Machine Learning B

2021-2022

Home Assignment 5

Fabian Gieseke **Yevgeny Seldin** **Sadegh Talebi**
Department of Computer Science
University of Copenhagen

The deadline for this assignment is **11 January 2022, 22:00**. 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.
- Do NOT zip the PDF file, since zipped files cannot be opened in speed grader. Zipped PDF submissions will not be graded.
- Your PDF report should be self-sufficient. 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 PAC-Bayesian Aggregation (75 points)

In this question you are asked to reproduce an experiment from Thiemann et al. (2017, Section 6, Figure 2) (the paper is an outcome of a master project). Figure 2 corresponds to "the second experiment" in Section 6 "Experimental Results". You are only required to reproduce the experiment for the first dataset, Ionosphere, which you can download from the UCI repository (Asuncion and Newman, 2007). You are allowed to use any programming language you like and any SVM solver you choose. Please, document carefully what you do and clearly annotate your graphs, including legend and axis labels.

Comments:

1. Assuming that you have followed the lectures and read "PAC-Bayesian Analysis" chapter in Yevgeny's lecture notes it should be sufficient to read only the "Experimental Results" section of the paper in order to reproduce the experiment, but you are of course welcome to read the full article.

2. Theorem 6 in the paper corresponds to Theorem 3.32 in Yevgeny's lecture notes, but uses a slightly tighter version of PAC-Bayes-kl inequality. Specifically, the $\ln(n+1)$ term is replaced by $\ln(2\sqrt{n})$, which leads to $\ln((n-r)+1)$ in Theorem 3.32 being replaced by $\ln(2\sqrt{n-r})$ in Theorem 6 in the paper. We do not mind which one you select to work with, but remember to document it in your report.

3. Ideally, you would repeat the experiment several times, say 10, and report the average + some form of deviation, e.g. standard deviation or quantiles, over the repetitions. We have committed a sin by not doing it in the paper. We encourage you to make a proper experiment, but in order to save time you are allowed to repeat the sin (we will not take points for that). Please, do not do that in real papers.

Hint: Direct computation of the update rule for ρ,

$$\rho(h) = \frac{\pi(h)e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}(h,S)}}{\sum_{h'}\pi(h')e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}(h',S)}},$$

is numerically unstable, since for large $n-r$ it leads to division of zero by zero. A way to fix the problem is to normalize by $e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}_{\min}}$, where $\hat{L}^{\mathrm{val}}_{\min} = \min_h \hat{L}^{\mathrm{val}}(h,S)$. This leads to

$$\rho(h) = \frac{\pi(h)e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}(h,S)}}{\sum_{h'}\pi(h')e^{-\lambda(n-r)\hat{L}^{\mathrm{val}}(h',S)}} = \frac{\pi(h)e^{-\lambda(n-r)\left(\hat{L}^{\mathrm{val}(h,S)}-\hat{L}^{\mathrm{val}}_{\min}\right)}}{\sum_{h'}\pi(h')e^{-\lambda(n-r)\left(\hat{L}^{\mathrm{val}}(h',S)-\hat{L}^{\mathrm{val}}_{\min}\right)}}.$$

Calculation of the latter expression for $\rho(h)$ does not lead to numerical instability problems.

Optional Add-on

1. Repeat the experiment with the tandem bound on the majority vote, Theorem 3.38 in Yevgeny's lecture notes, which corresponds to Masegosa et al. (2020, Theorem 9). You can find the details of optimization procedure for the bound in Masegosa et al. (2020, Appendix G). Tandem losses should be evaluated on overlaps of validation sets and n in the bounds should be replaced with the minimal overlap size for all pairs of hypotheses. Compare the results with the first order bound.

2 AdaBoost (10 points)

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. 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

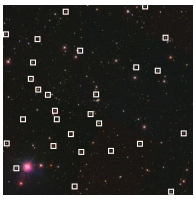
$$w_i^{(b+1)} = \frac{\exp(-y_i f_b(\mathbf{x}_i))}{\sum_{j=1}^n \exp(-y_j f_b(\mathbf{x}_j))},$$

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

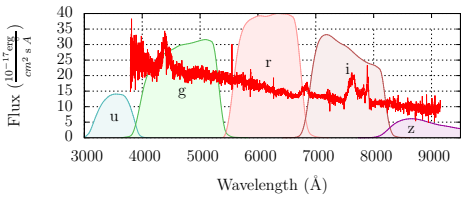
3 XGBoost (20 points)

An important problem in astrophysics is to estimate the distance of objects to our Earth. Since one cannot directly measure these distances, one resorts to the light emitted by the objects and that reaches our Earth. In particular, one considers the so-called *redshift* z of an object: the larger the redshift, the larger is the distance an object to our Earth. This redshift can be measured very accurately in case one has access to the detailed spectrum of the light for an object at hand. Unfortunately, obtaining such spectra is very time-consuming and are, hence, only available for a small subset of the detected objects, see Figure 1. Instead, one has access to image data, which is used to estimate the redshift of the detected objects.

(a) Observatory



(b) Image Data



(c) Spectrum (Quasar)

Figure 1: The Apache Point Observatory shown in Figure (a) gathers both images and spectra. The image data are given in terms of grayscale images that are based on five filters covering different wavelength ranges, called the `u`, `g`, `r`, `i`, and `z` bands. In Figure (b) an RGB image is shown that is based on such images of a particular region. For a small subset of detected objects (white squares), detailed follow-up observations in terms of spectra are available, see Figure (c). Image sources for Figures (a) and (b): `https://www.sdss.org`

Note that you do not need to understand the physical details given above to address this exercise (they are just provided for the sake of completeness). All you have to know is that this task can be formalized as regression problem and that you have access to a dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \subset \mathbb{R}^{10} \times \mathbb{R}$, which is stored in the file `quasars.csv`. Each line contains, for each object (row), a target value $y_i \in \mathbb{R}$ (last column) and a vector $\mathbf{x}_i \in \mathbb{R}^{10}$ of ten attributes (first 10 columns).

Your task is to build XGBoost regression models that can be used to predict the target (redshift) for new, unseen objects.[1] The corresponding library is available at `https://xgboost.readthedocs.io/en/stable/`.

1. Start by loading the dataset (`quasars.csv`; available on Absalon) and split it up into a training set containing about 80% and a test set containing about 20% of the instances.

2. Fit a XGBoost regression model (square loss) using the following parameter assignments (stick to the default assignments for the remaining parameters): `colsample_bytree=0.5`, `learning_rate=0.1`, `max_depth=4`, `reg_lambda=1`, `n_estimators=500`. Train the model on about 90% of the training data and make use of an hold-out validation set containing about 10% of the training set to monitor the training process by plotting both the training and the validation RMSE (y-axis) vs. the boosting iterations (x-axis).

   Finally, make predictions for the test set via the fitted model and compute the induced RMSE and the R2 score on the test set.

[1] The dataset contains data of so-called "quasars", which belong to the most distant objects that can be observed from our Earth!

3. Next, conduct a grid search to find good parameter assignments for your XGBoost model. Include the parameters used above as a starting point and extend the grid. Select the parameter combination with the lowest 3-fold cross validation RMSE (on the training set).

   Refit your model on all the training instances using the best parameter configuration. What is induced RMSE/R2 score on the test set? Can you beat a simple nearest neighbors regression model ($k = 5$ neighbors) that is fitted on the training set (note that beating this baseline is not a requirement for getting all the points :-))?

Hints: You can find many XGBoost tutorials on the internet. Have a look at some of them! Also, check out the parameters `eval_metric` *and* `eval_set` *of the fit function for monitoring the training process. For the grid search, note that you can combine XGBoost with Scikit-Learn; the whole grid-search can be conducted in a few lines of code.*

4 Illustration of Bernstein's Inequality for Bernoulli (45 points)

Consider a sequence of i.i.d. random variables, $X_1, \ldots, X_n$, sampled from a Bernoulli with mean μ, and let $\hat{\mu}_n$ be corresponding the empirical estimator, i.e., $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. We are interested in evaluating (or bounding) $\mathbb{P}(\hat{\mu}_n \geq \alpha)$ for some α.

4.a Let $n = 20$, $\mu = 0.5$, and $A = \{0.5, 0.55, 0.6, \ldots, 0.95, 1\}$. Make 1,000,000 repetitions of the experiment.

(i) Plot the empirical frequency of observing $\hat{\mu}_n \geq \alpha$ for $\alpha \in A$.

(ii) In the same figure, plot Hoeffding's bound[2] on $\mathbb{P}(\hat{\mu}_n \geq \alpha)$ for $\alpha \in A$. Write down explicitly the bound you are evaluating. (Note: Since $\mu \leq 1$, whenever the bound exceeds one, replace it with the trivial bound of one.)

(iii) Repeat (ii) for Bernstein's inequality.

(vi) Compare the curves and discuss.

(vii) For $\alpha = 1$ and $\alpha = 0.95$, calculate the exact probability $\mathbb{P}(\hat{\mu}_n \geq \alpha)$ (without plotting).

[2]Namely, the right hand side of Hoeffding's inequality.

4.b Repeat Part (a) with $n = 20$, $\mu = 0.1$, and $A = \{0.1, 0.15, \ldots, 1\}$.

4.c Compare the curves derived in (a) and (b).

4.d Now, we would like to compare confidence intervals for μ derived using Hoeffding's, Bernstein's, and empirical Bernstein's inequalities. For example, given $\delta \in (0, 1)$ and $X_1, \ldots, X_n$, recall that the confidence interval at the confidence level $1 - \delta$ defined by Hoeffding's inequality is

$$\mathsf{CI} = \left[\hat{\mu}_n - \sqrt{\frac{1}{2n}\log\left(\frac{2}{\delta}\right)},\ \hat{\mu}_n + \sqrt{\frac{1}{2n}\log\left(\frac{2}{\delta}\right)}\right]$$

In other words, by Hoeffding's inequality, the random interval CI traps the true mean μ with probability at least δ: $\mathbb{P}(\mu \in \mathsf{CI}) \geq 1 - \delta$. Note that as before, since $\mu \in [0, 1]$, whenever the upper bound of CI exceeds one (respectively, its lower bound falls below 0), we can replace it with the trivial bound of one (respectively, 0). One can define similar confidence intervals using Bernstein's and empirical Bernstein's inequalities.

Let $\mu = 0.15$, $\delta = 0.01$, $n = 25$, and make 1,000 repetitions of the experiment.

(i) Each repetition yields a confidence interval using Hoeffding's inequality. Report the mean and the variance of length of these confidence intervals.

(ii) Repeat (i) for confidnece intervals derived using Bernstein's inequality. Clearly write down the expression you are evaluating.

(iii) Repeat (i) for confidence intervals derived using empirical Bernstein's inequality. Clearly write down the expression you are evaluating.

(iv) Compare the results in (i)-(iii).

Some Optional Questions

5 [Optional question] Regularization by the relative entropy and the Gibbs distribution

In this question we will show that regularization by relative entropy leads to solutions in a form of the Gibbs distribution. Let's assume that we have a finite hypothesis class $\mathcal{H}$ of size m and we want to minimize

$$\mathcal{F}(\rho) = \alpha \mathbb{E}_\rho \left[\hat{L}(h, S)\right] + \mathrm{KL}(\rho \| \pi) = \alpha \sum_{h=1}^{m} \rho(h) \hat{L}(h, S) + \sum_{h=1}^{m} \rho(h) \ln \frac{\rho(h)}{\pi(h)}$$

with respect to the distribution ρ. This objective is closely related to the objective of PAC-Bayes-λ inequality when λ is fixed and this sort of minimization problem appears in many other places in machine learning. Let's slightly simplify and formalize the problem. Let $\rho = (\rho_1, \ldots, \rho_m)$ be the posterior distribution, $\pi = (\pi_1, \ldots, \pi_m)$ the prior distribution, and $L = (L_1, \ldots, L_m)$ the vector of losses. You should solve

$$\begin{aligned} \min_{\rho_1, \ldots, \rho_m} \quad & \alpha \sum_{h=1}^{m} \rho_h L_h + \sum_{h=1}^{m} \rho_h \ln \frac{\rho_h}{\pi_h} \\ s.t. \quad & \sum_{h=1}^{m} \rho_h = 1 \\ & \forall h : \rho_h \geq 0 \end{aligned} \tag{1}$$

and show that the solution is $\rho_h = \frac{\pi_h e^{-\alpha L_h}}{\sum_{h'=1}^{m} \pi_{h'} e^{-\alpha L_{h'}}}$. Distribution of this form is known as the Gibbs distribution.

Guidelines:

1. Instead of solving minimization problem (1), solve the following minimization problem

$$\begin{aligned} \min_{\rho_1, \ldots, \rho_m} \quad & \alpha \sum_{h=1}^{m} \rho_h L_h + \sum_{h=1}^{m} \rho_h \ln \frac{\rho_h}{\pi_h} \\ s.t. \quad & \sum_{h=1}^{m} \rho_h = 1, \end{aligned} \tag{2}$$

   i.e., drop the last constraint in (1).

2. Use the method of Lagrange multipliers to show that the solution of the above problem has a form of $\rho_h = \pi_h e^{-\alpha L_h + \texttt{something}}$, where `something` is something involving the Lagrange multiplier.

3. Show that $\rho_h \geq 0$ for all h. (This is trivial. But it gives us that the solutions of (1) and (2) are identical.)

4. Finally, $e^{\texttt{something}}$ should be such that the constraint $\sum_{h=1}^{m} \rho_h = 1$ is satisfied. So you can easily get the solution. You even do not have to compute the Lagrange multiplier explicitly.

6 [Optional question] Majority Vote

In this question we illustrate a few properties of the majority vote. Let MV denote a uniformly weighted majority vote.

1. Design an example of $\mathcal{H}$ and decision space $\mathbf{X}$, where $L(\text{MV}) = 0$ and $L(h) \geq \frac{1}{3}$ for all h. (Hint: three hypotheses and $|\mathbf{X}| = 3$ is sufficient.)

2. Design an example of $\mathcal{H}$ and $\mathbf{X}$, where $L(\text{MV}) > L(h)$ for all h.

   (a) Optional: design an example, where $L(\text{MV}) \xrightarrow[|\mathcal{H}|\to\infty]{} 2 \max_h L(h)$.

3. Let $\mathcal{H}$ be a hypothesis space, such that $|\mathcal{H}| = M$ and all $h \in \mathcal{H}$ have the same expected error, $L(h) = \frac{1}{2} - \varepsilon$ for $\varepsilon > 0$, and that the hypotheses in $\mathcal{H}$ make independent errors. Prove that $L(\text{MV}) \xrightarrow[|\mathcal{H}|\to\infty]{} 0$. (In words: derive a bound for $L(\text{MV})$ and show that as M grows the bound converges to zero, even though $L(h)$ can be almost as bad as $1/2$. In fact, $L(\text{MV})$ converges to zero exponentially fast with the growth of M.)

Bottom line: *If* the errors are independent *and* $L(h) < \frac{1}{2}$ for all $h \in \mathcal{H}$, the majority vote improves over individual classifiers. However, if $L(h) > \frac{1}{2}$ for some h the errors may get amplified, and if there is correlation it may play in either direction, depending on whether $L(h)$ is above or below $\frac{1}{2}$ and whether it is correlation or anti-correlation.

7 [Optional question] PAC-Bayes vs. Occam

The change of measure inequality that is at the basis of PAC-Bayesian analysis can be seen as a replacement of the union bound, which is at the basis of Occam's razor. In this question we compare the tightness of the two approaches.

1. Prove the following theorem.

   Theorem 1. Let S be an i.i.d. sample of n points, let ℓ be the zero-one loss, let $\mathcal{H}$ be countable, and let $\pi(h)$ be such that it is independent of the sample S and satisfies $\sum_{h\in\mathcal{H}} \pi(h) \leq 1$. Let $\delta \in (0,1)$. Then with probability greater than $1-\delta$, for all distributions ρ over $\mathcal{H}$ simultaneously:

   $$\mathrm{kl}\left(\mathbb{E}_\rho\left[\hat{L}(h,S)\right] \middle\| \mathbb{E}_\rho\left[L(h)\right]\right) \leq \frac{\mathbb{E}_\rho\left[\ln \frac{1}{\pi(h)}\right] + \ln \frac{n+1}{\delta}}{n}. \tag{3}$$

   You can use Occam's razor bound based on kl inequality that you have derived in one of the earlier questions to prove Theorem 1.

2. Recall that by PAC-Bayes-kl inequality, under the conditions of Theorem 1 we have that with probability greater than $1-\delta$, for all distributions ρ over $\mathcal{H}$ simultaneously:

   $$\mathrm{kl}\left(\mathbb{E}_\rho\left[\hat{L}(h,S)\right] \middle\| \mathbb{E}_\rho\left[L(h)\right]\right) \leq \frac{\mathrm{KL}(\rho\|\pi) + \ln \frac{n+1}{\delta}}{n}. \tag{4}$$

   Show that the PAC-Bayes-kl inequality in equation (4) is always at least as tight as the Occam's razor bound with kl in equation (3). Hint: the entropy of a discrete distribution is always non-negative, $\mathrm{H}(\rho) \geq 0$.

References

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Andrés R. Masegosa, Stephan S. Lorenzen, Christian Igel, and Yevgeny Seldin. Second order PAC-Bayesian bounds for the weighted majority vote. Technical report, `https://arxiv.org/abs/2007.13532`, 2020.

Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin. A strongly quasiconvex PAC-Bayesian bound. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, 2017.