Machine Learning B

2021-2022

Final Exam

Christian Igel **Yevgeny Seldin** **Sadegh Talebi**
Department of Computer Science
University of Copenhagen

You must submit your individual solution of the exam electronically via the **Digital Exam / Digital Eksamen** system. The deadline for submitting the exam is **Friday, 21 January 2022, at 16:00**. The exam must be solved **individually**. You are **not allowed** to work in groups or discuss the exam questions with other students. For fairness reasons any questions about the exam will be answered on Absalon. If your question may reveal the answer to other students, please, email it personally to the lecturers and we will either answer it on Absalon or tell you that we cannot answer your question.

WARNING: The goal of the exam is to evaluate your personal achievements in the course. We believe that take-home exams are most suitable for this evaluation, because they allow to test both theoretical and practical skills. However, our ability to give take-home exams strongly depends on your honesty. Therefore, any suspicion of cheating, in particular collaboration with other students, will be directly reported to the head of studies and prosecuted in the strictest possible way. It is also strictly prohibited to post the exam questions or parts thereof on the Internet or on discussion forums and to seek help on discussion forums. And you are not allowed to store your solutions in open access version control repositories or to post them on the Internet or on discussion forums. Be aware that if proven guilty you may be expelled from the university. Do not put yourself and your fellow students at risk.

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 complete source code in this PDF file. Do *not* include the task description or parts thereof in your report.
- Please, use the provided LaTeX template for typing your report. Hand-

written solutions will not be accepted.

- 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.
- Your code should be structured such that there is one main file 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.
- Your code should also include a README text file describing how to run your program.

1 Perceptron

In this assignment (inspired by Blum et al., 2020), we extend the perceptron algorithm as introduced and analyzed on slides 27–33 from the "Linear Classification" lecture.

Let $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$ be a non-trivial training set (i.e., containing patterns of both classes), $\boldsymbol{w}_0 = \mathbf{0}$, and let

$$R \leftarrow \max_{1 \leq i \leq N} \|\boldsymbol{x}_i\| \ .$$

Suppose that there exists $\boldsymbol{w}_{\text{opt}}$ with $\|\boldsymbol{w}_{\text{opt}}\| = 1$ such that the training patterns are separable by margin $\rho > 0$, that is,

$$y_i \langle \boldsymbol{w}_{\text{opt}}, \boldsymbol{x}_i \rangle \geq \rho > 0$$

for $1 \leq i \leq N$. Then Novikoff's theorem tells us that the number of updates k made by the online perceptron algorithm on S is at most

$$\left(\frac{R}{\rho}\right)^2 \ .$$

However, the standard perceptron algorithms typically does not find a separator of large margin. If we also want to find a separator of large margin, a natural alternative is to update the parameter vector whenever an example $(\boldsymbol{x}, y)$ does not meet the target margin, that is, for a target margin of 1 if $y\langle \boldsymbol{w}, \boldsymbol{x} \rangle < 1$. This leads to the following algorithm:

Algorithm 1 Margin Perceptron

```
Input: separable data $\{(\boldsymbol{x}_1, y_1), \dots\} \subseteq (\mathbb{R}^n \times \{-1, 1\})^N$
Output: hypothesis $h(\boldsymbol{x}) = \mathrm{sgn}(\langle \boldsymbol{w}_k, \boldsymbol{x} \rangle) = \mathrm{sgn}(f(\boldsymbol{x})$
$\boldsymbol{w}_0 \leftarrow \boldsymbol{0}; k \leftarrow 0$ repeat
    for $i = 1, \dots, N$ do
        if $y_i \langle \boldsymbol{w}_k, \boldsymbol{x}_i \rangle < 1$ then
            $\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k + y_i \boldsymbol{x}_i$   $k \leftarrow k + 1$
        end
    end
until no mistake made within for loop
```

Recall the hinge loss

$$L_{\text{hinge}}(y, f(x)) = [1 - yf(x)]_+ = \max(0, 1 - yf(x))$$

from the lecture on “Support Vector Machines” as well as stochastic gradient descent (SGD) from the lecture “Linear Classification” and the slide “Online vs batch learning iteration” from the lecture “Deep Learning Part I: Neural Networks”.

Prove that the margin perceptron algorithm above is equivalent to running stochastic gradient descent on the class of linear decision functions $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ using hinge loss as the loss function and using a constant learning rate of 1.

2 Invariance properties of the polynomial kernel

If we use PCA for preprocessing, we rotate the input to our machine learning algorithm. How does rotation affect different methods? Consider supervised learning for classification. Let the input space be $\mathbb{R}^d$. Now consider the transformation of the input data by a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{d \times d}$, which only rotates the data. That is, instead of $\boldsymbol{x}$ the algorithm is provided $\boldsymbol{R}\boldsymbol{x}$. The transformation is applied to train and test data.

In mathematical terms, we say that $\boldsymbol{R}$ is a member of the d-dimensional *special orthogonal group* and write $\boldsymbol{R} \in \mathrm{SO}(d)$. The formal definition of a rotation matrix is that $\boldsymbol{R}$ is a rotation matrix if and only if $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$ and $\det \boldsymbol{R} = 1$.

Now consider a spport vector machine with polynomial kernel, that is, for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, positive integer d, and non-negative $c \in \mathbb{R}^+$

$$k(\boldsymbol{x}, \boldsymbol{y}) = ((\langle \boldsymbol{x}, \boldsymbol{y} \rangle) + c^d \quad .$$

Is the classification affected by the rotation transformation in the sense that the classification performance changes if trained and tested on the transformed data compared to the original data?

Deliverables: Provide a formal proof either showing that the classification performance will not change or may change. For the former, you may ignore numerical issues. For the latter, note that it is sufficient to give a single toy example where the classification changes. The toy examples must explain why rotation may influence classification outcomes, just sampling points to illustrate the effect is *not* sufficient.

3 Sparse classifiers

Consider linear classification with d real-valued features ($\mathcal{X} = \mathbb{R}^d$). As we have seen in the course, if d is relatively large in relation to n (where n is the number of training samples) a naive solution that finds a separating hyperplane without making any restrictions is prone to overfitting. One tool for preventing overfitting that we have studied in the course, the one used in Support Vector Machines, is to put a constraint on the norm of a vector $\mathbf{w}$ that defines a separating hyperplane. In this question we will analyze an alternative approach based on sparsity. A sparse classifier is a classifier based on a small subset of d features. It has the advantage of being "simple", since sparse solution is based on a small number of features and often easy to interpret. While finding the best subset of features is computationally expensive, greedy approaches that iteratively add most valuable features to the solution can be quite successful (of course, depending on the problem). The focus of this question is on generalization properties of sparse classifiers.

All answers should be accompanied by explanations, derivations, and/or relevant citations.

1. Let $\mathcal{H}$ be the space of all possible linear classifiers in $\mathbb{R}^d$ (including non-homogeneous). Write down a generalization bound for learning with $\mathcal{H}$.

2. Let $\mathcal{H}_i$ be the space of all possible one-dimensional linear classifiers (including non-homogeneous) that are only based on the i-th feature (just a single feature). Let $\mathcal{H}_{\texttt{sparse}} = \bigcup_{i=1}^{d} \mathcal{H}_i$ be the space of all possible linear classifiers based on only one out of the d features. Derive a generalization bound for learning with $\mathcal{H}_{\texttt{sparse}}$.

3. Let h^* be "the best" linear classifier in $\mathbb{R}^d$ (the one achieving the lowest empirical error) and let $h^*_{\texttt{sparse}}$ be "the best" sparse classifier based on a single feature. You have derived generalization bounds for $L(h^*)$ and

$L(h^*_{\texttt{sparse}})$ above. Now you should explain in which situations (if any) the generalization bound for $L(h^*)$ will be smaller than the generalization bound for $L(h^*_{\texttt{sparse}})$, and in which situations (if any) it will be the other way around.

4 Bayesian inference of discrete distributions

In this assignment, we do Bayesian inference of a particular discrete distribution.

Suppose an item can take on one of the three possible colors, red, green, blue. We index the three colors by $k \in \{1, 2, 3\}$. Let X denote the random variable taking values in $\{1, 2, 3\}$ and representing the color of the item. Let $q = [q_1, q_2, q_3]$ denote the probability distribution of colors, i.e., $\mathbb{P}(X = k) = q_k$.

The color distribution q is unknown, but we have access to n observations sampled i.i.d. from q.

1. Show that the joint probability of n i.i.d. samples is:

$$p_{X_1,\ldots,X_n|Q}(x_1, \ldots, x_n|q) = \prod_{k=1}^{3} q_k^{N_k}$$

   where N_k is the number of occurrences of color k, i.e., $N_k = \sum_{i=1}^{n} \mathbb{I}\{x_i = k\}$.

2. Assume that q is treated as a random variable Q with a prior distribution f_Q or p_Q. Write down the posterior distribution. Indicate which variable is discrete and which is continuous. Please follow the same notation used in the lecture.

3. Consider a distribution with parameters $\alpha_1, \ldots, \alpha_3$, which we call the `MLB` Distribution, and which we denote by $\texttt{MLB}(\alpha_1, \alpha_2, \alpha_3)$. Its probability density function (pdf) is defined over the simplex probability distribution in $\mathbb{R}^3$. More specifically, let Y be a random variable taking values $y = [y_1, y_2, y_3] \in \mathbb{R}^3$, with y satisfying

$$y_1 + y_2 + y_3 = 1 \quad \text{and} \quad y_k \geq 0,\ \forall k.$$

   If Y is distributed according to $\texttt{MLB}(\alpha_1, \alpha_2, \alpha_3)$, its density is

$$f_Y(y; \alpha_1, \alpha_2, \alpha_3) = \frac{\Gamma(\sum_{k=1}^{3} \alpha_k)}{\prod_{k=1}^{3} \Gamma(\alpha_k)} \prod_{k=1}^{3} y_k^{\alpha_k - 1}$$

   where Γ denotes the Gamma function:

$$\Gamma(z) := \int_0^\infty x^{z-1} e^{-x} \mathrm{d}x\,.$$

Assume that prior for q is chosen to be $\texttt{MLB}(\alpha_1, \alpha_2, \alpha_3)$. Find the posterior distribution for q.

4. Argue that `MLB` is a conjugate prior distribution.

5 PAC-Bayes with distribution-dependent priors

In this question you will do a fun thing with PAC-Bayes. The "prior" distribution $\pi(h)$ in PAC-Bayesian analysis should be independent of the data S. But it can depend on the data generating distribution $p(x, y)$! (Side remark: recall that in the proof of the PAC-Bayes lemma we had to swap the expectations $\mathbb{E}_{h\sim\pi(h)}[\mathbb{E}_{S\sim p(x,y)^n}[e^{f(h,S)}]] = \mathbb{E}_{h\sim\pi(h)}[\mathbb{E}_{S\sim p(x,y)^n}[e^{f(h,S)}]]$. If π depends on S we cannot do the swap, but if π depends on the data generating distribution $p(x, y)$, we can do the swap and the rest of the analysis goes through.) Alas, the data generating distribution is unknown to us. But the only place where $\pi(h)$ appears in the PAC-Bayesian bounds is in the KL-divergence between the posterior and the prior, $\mathrm{KL}(\rho\|\pi)$. Therefore, as long as we are able to bound $\mathrm{KL}(\rho\|\pi)$, we can work with the PAC-Bayesian bounds, even though we have no direct access to the prior $\pi(h)$.

The ideal prior $\pi(h)$ would put all of its mass on the best performing prediction rule h, but for such prior we would not be able to bound $\mathrm{KL}(\rho\|\pi)$. Instead, we smooth the prior and define $\pi(h) = \frac{e^{-\beta L(h)}}{\sum_{h'\in\mathcal{H}} e^{-\beta L(h')}}$, assuming for simplicity that $\mathcal{H}$ is finite (we could do exactly the same for uncountable $\mathcal{H}$). This prior does what we would generally like to do, it puts higher mass on prediction rules with lower expected error $L(h)$ and lower mass on prediction rules with high expected error $L(h)$. This form of distribution is known as the Gibbs distribution. The parameter β, known as the inverse temperature parameter (the name comes from statistical physics), controls the spread of the prior. If we take $\beta = 0$, then $\pi(h)$ becomes a uniform distribution, and if we take $\beta \to \infty$, then it puts all the mass on the hypothesis/hypotheses with the minimal loss. If we take the posterior to have the same form of Gibbs distribution, then we can bound $\mathrm{KL}(\rho\|\pi)$, which is all we need.

Now you will do the analysis step-by-step. Note that since we provide the results for all the intermediate steps, you can proceed to the next step, even if you did not manage to prove the preceding step. In particular, the very first step is not hard, but it requires you to "click" on the right idea of what to do with $\frac{1}{Z_\pi}$. If you do not "click" on it quickly, we recommend that you proceed to the following points and get back to it at the end.

1. Let $\rho(h) = \frac{1}{Z_\rho} e^{-F(h,S)}$ and $\pi(h) = \frac{1}{Z_\pi} e^{-G(h)}$, where $Z_\rho = \sum_{h' \in \mathcal{H}} e^{-F(h',S)}$ and $Z_\pi = \sum_{h' \in \mathcal{H}} e^{-G(h')}$ are the normalization factors and F and G are arbitrary non-negative functions. Prove that:

   $$\mathrm{KL}(\rho \| \pi) \leq \mathbb{E}_{h \sim \rho(h)} \left[G(h) - F(h,S) \right] - \mathbb{E}_{h \sim \pi(h)} \left[G(h) - F(h,S) \right].$$

   Some hints:

   - In one step of the proof you need to use the fact that by definition of $\pi(h)$ we have that $\frac{1}{Z_\pi} = \frac{\pi(h)}{e^{-G(h)}}$. Note that this holds for any $h \in \mathcal{H}$. You will need to use this fact in order to get $\frac{1}{Z_\pi}$ into a summation over $h \in \mathcal{H}$.
   - In another step of the proof you will need to use Jensen's inequality.
   - You are allowed to assume that $\mathcal{H}$ is finite and replace all expectations with their definitions, as summations over $\mathcal{H}$.

2. Let $\mathbb{E}_\rho$ be a shorthand for $\mathbb{E}_{h \sim \rho(h)}$. Let $\beta > 0$ and $\lambda > 0$, and define $\rho(h) = \frac{1}{Z_\rho} e^{-\beta\left(1 - \frac{\lambda}{2}\right)\hat{L}(h,S)}$ and $\pi(h) = \frac{1}{Z_\pi} e^{-\beta L(h)}$. Use the result above and PAC-Bayes-λ inequality to show that with probability at least $1 - \delta$, for all $\lambda \in (0, 2)$ simultaneously

   $$\frac{1}{\beta} \mathrm{KL}(\rho \| \pi) \leq \frac{\lambda}{2} \mathbb{E}_\rho \left[L(h) \right] + \frac{\lambda}{2} \mathbb{E}_\rho \left[\hat{L}(h,S) \right] + \frac{\mathrm{KL}(\rho \| \pi) + \ln \frac{n+1}{\delta}}{\lambda n} + \frac{\ln \frac{n+1}{\delta}}{\lambda n}.$$

   Hint: You will need both PAC-Bayes-λ upper and PAC-Bayes-λ lower bound. You are allowed to use Theorem 3.31 in Yevgeny's lecture notes.

3. Continue the derivation and prove that for $\beta > 0$, with probability at least $1 - \delta$, for all $\lambda \in (0, 2)$ simultaneously

   $$\frac{1}{\beta} \mathrm{KL}(\rho \| \pi) \leq \frac{2\,\mathrm{KL}(\rho \| \pi)}{n\lambda(2 - \lambda)} + \frac{\lambda(4 - \lambda)\mathbb{E}_\rho \left[\hat{L}(h,S) \right]}{2(2 - \lambda)} + \frac{(4 - \lambda) \ln \frac{2(n+1)}{\delta}}{n\lambda(2 - \lambda)}.$$

4. Prove that for $\beta \in \left(0, \frac{n}{2}\right)$, with probability at least $1 - \delta$, for all $\lambda \in \left(1 - \sqrt{1 - \frac{2\beta}{n}}, 1 + \sqrt{1 - \frac{2\beta}{n}}\right)$ simultaneously

   $$\mathrm{KL}(\rho \| \pi) \leq \frac{n\lambda\beta(4 - \lambda)}{n\lambda(2 - \lambda) - 2\beta} \left(\frac{\lambda \mathbb{E}_\rho \left[\hat{L}(h,S) \right]}{2} + \frac{\ln \frac{n+1}{\delta}}{n\lambda} \right).$$

   Please, comment on where you are using the restrictions on the range of β and λ.

5. Prove that for $\beta \in \left(0, \frac{n}{2}\right)$, with probability at least $1-\delta$, for all $\lambda \in \left(1-\sqrt{1-\frac{2\beta}{n}}, 1+\sqrt{1-\frac{2\beta}{n}}\right)$ simultaneously

$$\mathbb{E}_\rho\left[L(h)\right] \leq \frac{\mathbb{E}_\rho\left[\hat{L}(h,S)\right]}{1-\frac{\lambda}{2}}\left(1+\frac{\beta\lambda(4-\lambda)}{2\left(n\lambda(2-\lambda)-2\beta\right)}\right)+\frac{\ln\frac{n+1}{\delta}}{n\lambda\left(1-\frac{\lambda}{2}\right)}\left(1+\frac{\beta(4-\lambda)}{n\lambda(2-\lambda)-2\beta}\right).$$

We conclude with a brief discussion to put your result into context (no action items for you). The result that you obtained is the following theorem.

Theorem 1. Let $\beta \in \left(0, \frac{n}{2}\right)$. Let $\rho(h) = \frac{1}{Z_\rho}e^{-\beta\left(1-\frac{\lambda}{2}\right)\hat{L}(h,S)}$ and $\pi(h) = \frac{1}{Z_\pi}e^{-\beta L(h)}$. Then with probability at least $1-\delta$ for all $\lambda \in \left(1-\sqrt{1-\frac{2\beta}{n}}, 1+\sqrt{1-\frac{2\beta}{n}}\right)$ simultaneously

$$\mathrm{KL}(\rho\|\pi) \leq \frac{n\lambda\beta(4-\lambda)}{n\lambda(2-\lambda)-2\beta}\left(\frac{\lambda\mathbb{E}_\rho\left[\hat{L}(h,S)\right]}{2}+\frac{\ln\frac{n+1}{\delta}}{n\lambda}\right)$$

and

$$\mathbb{E}_\rho\left[L(h)\right] \leq \frac{\mathbb{E}_\rho\left[\hat{L}(h,S)\right]}{1-\frac{\lambda}{2}}\left(1+\frac{\beta\lambda(4-\lambda)}{2\left(n\lambda(2-\lambda)-2\beta\right)}\right)+\frac{\ln\frac{n+1}{\delta}}{n\lambda\left(1-\frac{\lambda}{2}\right)}\left(1+\frac{\beta(4-\lambda)}{n\lambda(2-\lambda)-2\beta}\right).$$

We achieved the target: we defined $\pi(h)$ in terms of the data generating distribution and we managed to bound $\mathrm{KL}(\rho\|\pi)$ and $\mathbb{E}_\rho[L(h)]$ with no explicit access to $\pi(h)$. Note that since the bound holds for all λ simultaneously, it is possible to minimize it with respect to λ to obtain "the optimal" (with respect to the bound) posterior distribution ρ. (Side remark: we have discussed in the lecture notes that the optimal value of λ is always at most 1, so we could reduce the range of λ to $\lambda \in \left(1-\sqrt{1-\frac{2\beta}{n}}, 1\right]$.) Also note that the bound exhibits "fast convergence rates": when $\mathbb{E}_\rho\left[\hat{L}(h,S)\right]$ is close to zero it decreases at the rate of $1/n$. The bound does not allow selection of β, because then π would depend on S. Selection of β would require a union bound or some other correction.

6 Confidence intervals for Bernoulli

We are given two sequences of $S = (X_t)_{1\leq t\leq n}$ and $S' = (X'_t)_{1\leq t\leq n'}$. S is drawn i.i.d. from a Bernoulli with mean μ, whereas S' is drawn i.i.d. from a Bernoulli with mean μ' — See datasets `S.csv` and `Sprime.csv`.

1. Compute 0.95-CI (two-sided) for μ using Hoeffding's inequality, empirical Bernstein's inequality, and the kl-inequality. Document your calculations. (Final answers without explanation will earn zero point.)

2. Compare the three CIs.

3. Can we say with probability at least 0.99 that $\mu > \mu'$? Argue why. You can use any concentration inequality you like, but specify the one you use.

References

Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. Cambridge University Press, 2020.