Machine Learning

2021-2022

Home Assignment 3

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

The deadline for this assignment is **14 December 2021, 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.
- IMPORTANT: 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. 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 Kernels (50 points)

The first question should improve the understanding of the geometry of the kernel-induced feature space. You can directly use the result to implement a kernel nearest-neighbor algorithm. The second question should make you more familiar with the basic definition of the important concept of positive definiteness. The third question is important to understand the real dimensionality of learning problems using a linear kernel – one reason why linear kernels are often treated differently in efficient implementations.

1.1 Distance in feature space

Given a kernel k on input space $\mathcal{X}$ defining RKHS $\mathcal{H}$. Let $\Phi : \mathcal{X} \to \mathcal{H}$ denote the corresponding feature map (think of $\Phi(x) = k(x, .)$). Let $x, z \in \mathcal{X}$. Show that the distance of $\Phi(x)$ and $\Phi(z)$ in $\mathcal{H}$ is given by

$$\|\Phi(x) - \Phi(z)\| = \sqrt{k(x,x) - 2k(x,z) + k(z,z)}$$

(if distance is measured by the canonical metric induced by k).

1.2 Sum of kernels

Let $k_1, k_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be positive-definite kernels.

Prove that $k(x,z) = a \cdot k_1(x,z) + b \cdot k_2(x,z)$ for $a, b \in \mathbb{R}^+$ is also positive-definite.

1.3 Rank of Gram matrix

Let the input space be $\mathcal{X} = \mathbb{R}^d$. Assume a linear kernel, $k(x,z) = x^{\mathrm{T}} z$ for $x, z \in \mathbb{R}^d$ (i.e., the feature map Φ is the identity) and m input patterns $x_1, \ldots, x_m \in \mathbb{R}^d$.

Prove a non-trivial upper bound on the rank of the Gram matrix from the m input patterns in terms of d and m.

2 Early stopping (30 points)

Early stopping is a widely used technique to avoid overfitting in models trained by iterative methods, such as gradient descent. In particular, it is used to avoid overfitting in training neural networks. In this question we analyze several ways of implementing early stopping. The technique sets aside a validation set S_{val}, which is used to monitor the improvement of the training process. Let $h_1, h_2, h_3, \ldots$

be a sequence of models obtained after $1, 2, 3, \ldots$ epochs of training a neural network or any other prediction model (you do not need to know any details about neural networks or their training procedure to answer the question). Let $\hat{L}(h_1, S_{\text{val}}),\ \hat{L}(h_2, S_{\text{val}}),\ \hat{L}(h_3, S_{\text{val}}), \ldots$ be the corresponding sequence of validation errors on the validation set S_{val}.

1. Let h_{t^*} be the neural network returned after training with early stopping. In which of the following cases is $\hat{L}(h_{t^*}, S_{\text{val}})$ an unbiased estimate of $L(h_{t^*})$ and in which cases is it not. Please, explain your answer.

   (a) Predefined stopping: the training procedure always stops after 100 epochs and always returns the last model $h_{t^*} = h_{100}$.

   (b) Non-adaptive stopping: the training procedure is executed for a fixed number of epochs T, and returns the model h_{t^*} with the lowest validation error observed during the training process, i.e., $t^* = \arg\min_{t \in \{1,\ldots,T\}} \hat{L}(h_t, S_{\text{val}})$.

   (c) Adaptive stopping: the training procedure stops when no improvement in $\hat{L}(h_t, S_{\text{val}})$ is observed for a significant number of epochs. It then returns the best model observed ever during training. (This procedure is proposed in Goodfellow et al. (2016, Algorithm 7.1) or `https://www.quora.com/How-does-one-employ-early-stopping-in-TensorFlow`, but again, you do not need to know the details of the training procedure.)

2. Derive a high-probability bound (a bound that holds with probability at least $1 - \delta$) on $L(h_{t^*})$ in terms of $\hat{L}(h_{t^*}, S_{\text{val}})$, δ, and the size n of the validation set S_{val} for the three cases above. In the second case the bound may additionally depend on the total number of epochs T, while in the third case the bound may additionally depend on the index t^* of the epoch providing the optimal model. Please, solve the last case using the series $\sum_{t=1}^{\infty} \frac{1}{i(i+1)} = 1$.[1]

3. The adaptive approach suggests stopping when "no improvement in $\hat{L}(h_t, S_{\text{val}})$ is observed for a significant number of epochs". A natural way of redefining the stopping criterion once we have the generalization bound is to stop when "no improvement in the generalization bound is observed for a significant number of epochs". The adaptive approach does not limit the number of epochs in advance, but what is the maximal number of epochs $T_{\max}$, after which it makes no sense to continue training according to the bound you derived in Point 2? Express $T_{\max}$ in terms of the number of

[1] We have $\sum_{i=1}^{\infty} \frac{1}{i(i+1)} = \sum_{i=1}^{\infty} \left(\frac{1}{i} - \frac{1}{i+1}\right) = 1$.

validation samples n. It is sufficient to provide an order of magnitude of $T_{\max}$ in terms of n, you do not have to calculate the explicit constants.

4. How would your answer to the previous point change if you use the series $\sum_{i=1}^{\infty} \frac{1}{2^i} = 1$ for deriving the bound? (You should get that with these series you can run significantly less epochs in the adaptive approach compared to the series used in Point 2. Thus, unlike in the case of decision trees from the previous question, here the choice of the series has a significant impact.)

5. In this question we compare the adaptive procedure with non-adaptive. Assume that the two procedures use the same initialization, so that the corresponding models at epoch t are identical, and assume that the adaptive procedure has reached $T_{\max}$ (but t^* may be smaller than $T_{\max}$). Show that the generalization bound for adaptive stopping in Point 2 is never much worse than the generalization bound for non-adaptive stopping, but in some cases the adaptive bound can be significantly lower.

   Guidance: To simplify the analysis, throughout the question we assume that the confidence parameter $\delta \leq \frac{1}{2}$. For $T \geq 1$ it gives $\delta \leq \frac{1}{2} \leq \frac{T}{T+1}$.

   (a) First, assume that $T \leq T_{\max}$. Let t^* be the index of the epoch selected by the adaptive procedure and T^* be the index of the epoch selected by the non-adaptive procedure. Since the adaptive procedure has selected t^* we know that the adaptive bound for epoch t^* is lower than the adaptive bound for epoch T^*. We also know that $T^* \leq T$, where T is the number of epochs in the non-adaptive approach. Use this information and do some bounding to show that for any confidence parameter $\delta \leq \frac{1}{2}$, the adaptive bound can be at most a multiplicative factor of $\sqrt{2}$ larger than the non-adaptive bound.

   (b) Now consider the case $T > T_{\max}$. Show that in this case the adaptive bound is trivially 1 and the non-adaptive bound is at least $\frac{1}{\sqrt{2}}$. So in this case the adaptive bound also cannot exceed the non-adaptive bound by more than a multiplicative factor of $\sqrt{2}$.

   (c) You have shown that under the assumption that $\delta \leq \frac{1}{2}$ the adaptive bound never exceeds the non-adaptive bound by more than a multiplicative factor of $\sqrt{2}$. Now explain in which situations the adaptive bound can be significantly smaller than the non-adaptive bound. You should have two cases. In both cases you should have $T < T_{\max}$ and $\delta \leq \frac{1}{2}$.

   Conclusion: depending on the data, the generalization bound for adaptive stopping can be significantly smaller than the generalization bound for non-adaptive stopping and at the same time it is guaranteed that it is never worse by more than a multiplicative factor of $\sqrt{2}$.

3 Learning by discretization (20 points)

We want to learn an arbitrary binary function on a unit square by discretizing the square into a uniform grid with d^2 cells. The hypothesis space is the space of all possible uniform grids with d^2 cells for $d \in \{1, 2, 3, \dots\}$, where each cell gets a binary label.

We have a sample S of size n to learn the function. Let $\mathcal{H}$ be the hypothesis set of uniform grids and let $f(h)$ denote the number of cells in the hypothesis h. (If we take $d(h) = \sqrt{f(h)}$, then $d(h) \in \{1, 2, 3, \dots\}$.)

1. Derive a generalization bound for learning with $\mathcal{H}$.
2. Explain how to use the bound to select a prediction rule $h \in \mathcal{H}$.
3. What is the maximal number of cells as a function of n, for which your bound is non-vacuous? (It is sufficient to give an order of magnitude, you do not need to make the precise calculation.)
4. Explain how the density of the grid affects the bound. Which terms in the bound increase as the density of the grid increases and which terms in the bound decrease as the density of the grid increases?

References

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.