Machine Learning B

2022-2023

Home Assignment 3

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

The deadline for this assignment is **13 December 2022, 18:00**. 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 The VC-dimension (50 points)

1. Let $\mathcal{H}$ be a finite hypothesis set with $|\mathcal{H}| = M$ hypotheses. Bound the VC-dimension of $\mathcal{H}$.

2. Let $\mathcal{H}$ be a hypothesis space with 2 hypotheses (i.e., $|\mathcal{H}| = 2$). Prove that $d_{\text{VC}}(\text{H}) = 1$.

3. ***Optional question (0 points)*** Prove Lemma 3.15 in Yevgeny's lecture notes.

4. ***Optional question (0 points)*** Verify that Theorem 3.8, Theorem 3.13, and Lemma 3.15 together yield Theorem 3.16.

5. What should be the relation between $d_{\text{VC}}(\text{H})$ and n in the VC generalization bound in Theorem 3.16 in Yevgeny's lecture notes in order for the bound to be non-trivial? [A bound on the loss that is greater than or equal to 1 is trivial, because we know that the loss is always bounded by 1. You do not have to make an exact calculation, giving an order of magnitude is sufficient.]

6. In the case of a finite hypothesis space, $|\mathcal{H}| = M$, compare the generalization bound that you can obtain with Theorem 3.16 in Yevgeny's lecture notes with the generalization bound in Theorem 3.2 in Yevgeny's lecture notes. In what situations which of the two bounds is tighter?

7. How many samples do you need in order to ensure that the empirical loss of a linear classifier selected out of a set of linear classifiers in $\mathbb{R}^{10}$ does not underestimate the expected loss by more than 0.01 with 99% confidence?

   Clarifications: (1) you are allowed to use the bound on the VC-dimension of linear classifiers mentioned in Yevgeny's lecture notes; (2) solving the question analytically is a bit tricky, you are allowed to provide a numerical solution. In either case (numerical or analytical solution), please, explain clearly in your report what you did.

8. Let $\mathcal{H}_+$ be the class of "positive" circles in $\mathbb{R}^2$ (each $h \in \mathcal{H}_+$ is defined by the center of the circle $c \in \mathbb{R}^2$ and its radius $r \in \mathbb{R}$; all points inside the circle are labeled positively and outside negatively). Prove that $d_{VC}(\mathcal{H}_+) \geq 3$.

9. Let $\mathcal{H} = \mathcal{H}_+ \cup \mathcal{H}_-$ be the class of "positive" and "negative" circles in $\mathbb{R}^2$ (the "negative" circles are negative inside and positive outside). Prove that $d_{VC}(\mathcal{H}) \geq 4$.

10. ***Optional question (0 points)*** Prove the matching upper bounds $d_{VC}(\mathcal{H}_+) \leq 3$ and $d_{VC}(\mathcal{H}) \leq 4$. [Doing this formally is not easy, but will earn you extra honor.]

11. What is the VC-dimension of the hypothesis space $\mathcal{H}_d$ of binary decision trees of depth d?

12. What is the VC-dimension of the hypothesis space $\mathcal{H}$ of binary decision trees of unlimited depth?

13. You have a sample of 100,000 points and you have managed to find a linear separator that achieves $\hat{L}_{\mathtt{FAT}}(h, S) = 0.01$ with a margin of 0.1. Provide a bound on its expected loss that holds with probability of 99%. The input space is assumed to be within the unit ball and the hypothesis space is the space of linear separators.

14. **The fine details of the lower bound.** We have shown that if a hypothesis space $\mathcal{H}$ has an infinite VC-dimension, it is possible to construct a worst-case data distribution that will lead to overfitting, i.e., with probability at least $\frac{1}{8}$ it will be possible to find a hypothesis for which $L(h) \geq \hat{L}(h, S) + \frac{1}{8}$. But does it mean that hypothesis spaces with infinite VC-dimension are always deemed to overfit? Well, the answer is that it depends on the data distribution. If the data distribution is not the worst-case for $\mathcal{H}$, there may still be hope.

    Construct a data distribution $p(X, Y)$ and a hypothesis space $\mathcal{H}$ with infinite VC-dimension, such that for any sample S of more than 100 points with probability at least 0.95 we will have $L(h) \leq \hat{L}(h, S) + 0.01$ for all h in $\mathcal{H}$.

    Hint: this can be achieved with an extremely simple example.

2 Variable Stars (50 points)

In this part of the assignment, you should get familiar with support vector machines (SVMs).

SVM Software You need an SVM implementation, and you are welcome to use existing SVM software.[1] You are supposed to use Python in this course, still we make some comments on SVM implementations available in other languages, which may be useful in other contexts. We might get back to this question in future assignments, so it may pay off to write clean, reusable code.

[1]Carefully study what the SVM implementation you use is doing under the hood. Some implementations consider C/ℓ instead of C in the SVM objective function, where C denotes the regularization parameter. The SVM from the Matlab Bioinformatics Toolbox may by default use different regularization parameters depending on the class and the class frequency.

We recommend using LIBSVM Chang and Lin (2011), which can be downloaded from `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. Interfaces to LIBSVM for many programming languages exist, including Matlab and Python. The SVM implementation `sklearn.svm.SVC` (see `https://scikit-learn.org/stable/modules/svm.html`) of Scikit-learn Pedregosa et al. (2011) is also based on LIBSVM.[2] For this exercise, use Gaussian kernels of the form

$$k(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{z}\|^2) \ . \tag{1}$$

Here $\gamma > 0$ is a bandwidth parameter that has to be chosen in the model selection process. Note that instead of γ often the parameter $\sigma = \sqrt{1/(2\gamma)}$ is considered.

Variable Stars A variable star changes its intensity, as observed by a telescope, over time. This can be caused extrinsically, for example by other objects temporarily occluding it, but also intrinsically, when the star changes its physical properties over time. Figure 1 shows an example. The graph of the varying intensity as a function of time is called the light curve. Variable stars can be further divided into many classes depending on other physical properties. The task we are trying to solve is to predict the class of a variable star by its light curve. To achieve this, we train a classifier in a supervised setting using labeled data from the All Sky Automated Survey Catalog of Variable Stars (ACVS) (Pojmanski, 2000).

The data considered in the following is based on the study by Richards et al. (2012). Each sample encodes the astronomical properties of a variable star in a 61-dimensional feature vector. The features are listed in Table 2, for a detailed description of their meaning we refer to Dubath et al. (2011) and Richards et al. (2011). From the originally 25 different classes, we consider just two classes, namely "Beta Persei" and "Beta Lyrae", in order to evaluate binary SVMs.

2.1 Data understanding and preprocessing (8 points)

Read the data, for example like this:

```
y_train = np.loadtxt('y_train_binary.csv', delimiter=',')
y_test = np.loadtxt('y_test_binary.csv', delimiter=',')
X_train = np.loadtxt('X_train_binary.csv', delimiter=',')
X_test = np.loadtxt('X_test_binary.csv', delimiter=',')
```

[2]For R, package such as E1071 (recommended) and KERNLAB are available. If you are comfortable with C++, you are encouraged to use the SVM implementation within the SHARK machine learning library Igel et al. (2008). Get the latest snapshot from `http://www.shark-ml.org/`.

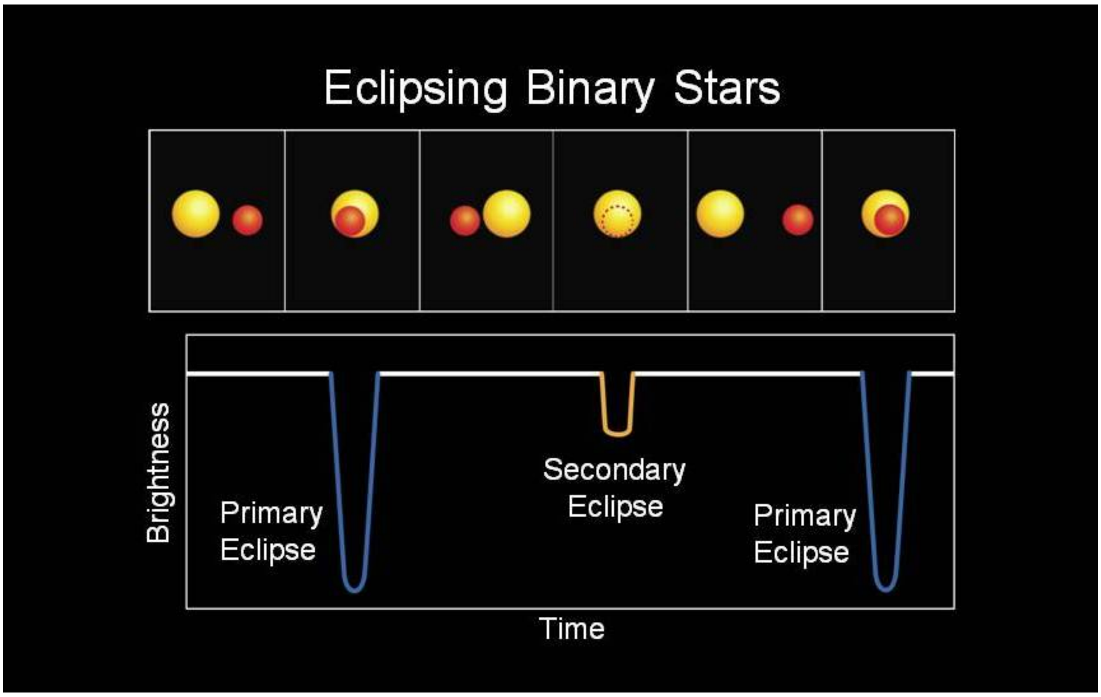

Figure 1: A variable star changes its intensity as observed from a telescope due to another smaller orbiting star. The image is taken from the NASA, `http://kepler.nasa.gov/news/nasakeplernews/index.cfm?FuseAction=ShowNews&NewsID=152`.

The are four files, the input patterns and their labels for training and testing, respectively. Report the class frequencies, that is, for each class report the number of data points belonging to that class divided by the total number of data points in the training data.

Then normalize the input data to zero mean and unit variance on the training data set. Consider the training data. Compute the mean and the variance of the input features (i.e., for each component of the input vector independently). Find the affine linear mapping f_{norm} that transforms the training data such that the mean and the variance of every feature in the transformed data are 0 and 1, respectively (verify by computing these values).

Use the function f_{norm} to also encode the test data. Compute the mean and the variance of every feature in the transformed test data.

The normalization is part of the model building process. Thus, you may only use the training data for determining f_{norm} (always remember that you are supposed to not know the test data). See also Example 5.3 on pages 174-175 in the course textbook Abu-Mostafa et al. (2012).

#	Feature name	#	Feature name
0	amplitude	31	freq3_harmonics_rel_phase_2
1	beyond1std	32	freq3_harmonics_rel_phase_3
2	flux_percentile_ratio_mid20	33	freq_amplitude_ratio_21
3	flux_percentile_ratio_mid35	34	freq_amplitude_ratio_31
4	flux_percentile_ratio_mid50	35	freq_frequency_ratio_21
5	flux_percentile_ratio_mid65	36	freq_frequency_ratio_31
6	flux_percentile_ratio_mid80	37	freq_signif
7	fold2P_slope_10percentile	38	freq_signif_ratio_21
8	fold2P_slope_90percentile	39	freq_signif_ratio_31
9	freq1_harmonics_amplitude_0	40	freq_varrat
10	freq1_harmonics_amplitude_1	41	freq_y_offset
11	freq1_harmonics_amplitude_2	42	linear_trend
12	freq1_harmonics_amplitude_3	43	max_slope
13	freq1_harmonics_freq_0	44	median_absolute_deviation
14	freq1_harmonics_rel_phase_1	45	median_buffer_range_percentage
15	freq1_harmonics_rel_phase_2	46	medperc90_2p_p
16	freq1_harmonics_rel_phase_3	47	p2p_scatter_2praw
17	freq2_harmonics_amplitude_0	48	p2p_scatter_over_mad
18	freq2_harmonics_amplitude_1	49	p2p_scatter_pfold_over_mad
19	freq2_harmonics_amplitude_2	50	p2p_ssqr_diff_over_var
20	freq2_harmonics_amplitude_3	51	percent_amplitude
21	freq2_harmonics_freq_0	52	percent_difference_flux_percentile
22	freq2_harmonics_rel_phase_1	53	QSO
23	freq2_harmonics_rel_phase_2	54	non_QSO
24	freq2_harmonics_rel_phase_3	55	scatter_res_raw
25	freq3_harmonics_amplitude_0	56	skew
26	freq3_harmonics_amplitude_1	57	small_kurtosis
27	freq3_harmonics_amplitude_2	58	std
28	freq3_harmonics_amplitude_3	59	stetson_j
29	freq3_harmonics_freq_0	60	stetson_k
30	freq3_harmonics_rel_phase_1		

Table 1: Different features are used to describe the light curve of a variable star.

Deliverables: number of training and test examples; code snippets for the normalization; mean and variance of features in the test data

2.2 Model selection using grid-search (21 points)

The performance of your SVM classifier depends on the choice of the regularization parameter C and the kernel parameters (here γ). Adapting these *hyperpa-*

rameters is referred to as SVM *model selection.*

Use grid-search to determine appropriate SVM hyperparameters γ and C. Look at all combinations of $C \in \{c_1, c_2, \ldots, c_7\}$ and $\gamma \in \{g_1, \ldots, g_7\}$, where you have to choose proper grid points for yourself. Consider values around $C = 10$ and $\gamma = 0.1$ of different orders of magnitude. It is common to vary the values on a logarithmic scale (e.g., $0.001, 0.01, 0.1, 1, 10, 100$). For each pair, estimate the performance of the SVM using *5-fold cross validation.*

Cross validation is an important technique in machine learning and it is very important that you become familiar with it. You can read about cross-validation on pages 145-149 in the course textbook Abu-Mostafa et al. (2012).

Pick the hyperparameter pair with the lowest average 0-1 loss (classification error) and train an SVM with these hyperparameters using the complete training dataset. Only the training data must be used in the model selection process.

Report the values for C and γ you found in the model selection process, the 0-1 loss on the training data, as well as the 0-1 loss on the test data.

Deliverables: Description of software used; a short description of how you proceeded (e.g., did the cross-validation); training and test errors as well as the best hyperparameter configuration

2.3 Inspecting the kernel expansion (21 points)

A support vector x_i is bounded if the corresponding coefficient in the kernel expansion (usually denoted by α_i) has an absolute value of C. If $0 < \alpha_i < C$ then the support vector is free. Free support vectors lie on the margin (i.e., have a functional margin of 1). All wrongly classified patterns as well as the patterns that are correctly classified but are inside the margin area (i.e., they are too close to the decision boundary and have a functional margin less than 1) end up as bounded support vectors.

Let us consider the effect of the regularization parameter C. How do you expect the number of bounded and free support vectors change if C is drastically decreased? Briefly justify your claim giving rigorous arguments. In addition, verify and illustrate your claim experimentally by counting the number of free and bounded support vectors in solutions for the problem in this exercise for different (smaller) values of C. Report the C values and the (different) numbers of free and bounded support vectors. Show the part of the code that computes the number of free and bounded support vectors, respectively, in the report (in addition to the attached full source code in an archive).

Optional: What happens with the number of support vectors for increasing C?

Deliverables: Answer and rigorous argumentation, results of empirical validation including description of how these results were computed

References

Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning from Data*. AMLbook, 2012.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions Intelligent Systems Technology*, 2(3):27:1–27:27, 2011.

P. Dubath, L. Rimoldini, M Süveges, J. Blomme, M López, L. M. Sarro, J. De Ridder, J. Cuypers, L. Guy, I. Lecoeur, et al. Random forest automated supervised classification of hipparcos periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, 414(3):2602–2617, 2011.

C. Igel, T. Glasmachers, and V. Heidrich-Meisner. Shark. *Journal of Machine Learning Research*, 9:993–996, 2008.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

G. Pojmanski. The all sky automated survey. Catalog of about 3800 variable stars. *Acta Astronomica*, 50:177–190, 2000.

J. W. Richards, D. L. Starr, N. R. Butler, J. S. Bloom, J. M. Brewer, A. Crellin-Quick, J. Higgins, R. Kennedy, and M. Rischard. On machine-learned classification of variable stars with sparse and noisy time-series data. *The Astrophysical Journal*, 733(1):10, 2011.

J. W. Richards, D. L. Starr, H. Brink, A. A. Miller, J. S. Bloom, N. R. Butler, J. B. James, J. P. Long, and J. Rice. Active learning to overcome sample selection bias: Application to photometric variable star classification. *The Astrophysical Journal*, 744(2):192, 2012.