# *Machine Learning B*

*2022-2023*

## Final Exam

**Fabian Gieseke** **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, 20 January 2023, at 12:00**. 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. (Please, be aware that by default your grades will be based on PDF reports only. The source code will be inspected when deemed necessary, including in cases of suspected collaboration.)
- 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 Gaussian Kernel (20 Points) [Christian]

Prove that the Gaussian kernel on $\mathbb{R}^d$ for positive integer $d$

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2) \tag{1}$$

for $\gamma > 0$ can be expressed as the inner product in an infinite-dimensional feature space.

To do so, first prove that if $k'$ is a kernel on $\mathcal{X}$, then for $\phi : \mathcal{X} \to \mathbb{R}$ it holds that

$$k''(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) k'(\mathbf{x}, \mathbf{x}') \phi(\mathbf{x}') \tag{2}$$

is also a kernel on $\mathcal{X}$.

Then use the following hints and facts for the main proof (not necessarily in that order):

I. As the polynomial kernel $k_D(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^D$ is a proper kernel for each positive integer $D$, there exist feature maps $\Phi_D$ such that

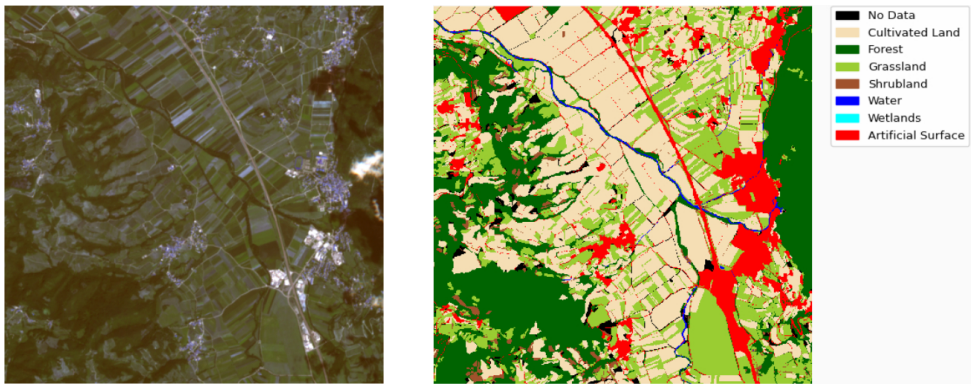
$$k_D(\mathbf{x}, \mathbf{x}') = \langle \Phi_D(\mathbf{x}), \Phi_D(\mathbf{x}') \rangle \ . \tag{3}$$

II. The real exponential function can be defined by the power series

$$\exp(x) := \sum_{i=0}^{\infty} \frac{x^i}{i!} \ . \tag{4}$$

III. The kernel defined in (1) can be split in the following way:

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2) = \exp(-\gamma \langle \mathbf{x}, \mathbf{x} \rangle) \exp(2\gamma \langle \mathbf{x}, \mathbf{x}' \rangle) \exp(-\gamma \langle \mathbf{x}', \mathbf{x}' \rangle) \tag{5}$$




(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.

# 2 Landcover Classification with XGBoost (20 Points) [Fabian]

In this exercise, you are supposed to apply XGBoost to the data set you already know from HA4, see again Figure 1. As input data, you are given two files, `train.npz` and `test.npz`, containing some training and some test instances.[1] Each instance is composed of images of size $13 \times 13$ pixels from 6 bands 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. 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. Split up the training data into a set used for the fitting process (90%) and a another validation set (10%) to monitor the performance during the fitting process.

2. Train a XGBoost classification model using `colsample_bytree=0.02`, `learning_rate=0.1`, `max_depth=8`, `reg_lambda=1`, and `n_estimators=100`.

[1] As for HA4, you can download both files as well as a Jupyter notebook from here `https://sid.erda.dk/cgi-sid/ls.py?share_id=c9SgMJSGik`. The files are also available via Absalon, see the announcement "Data (HW4 and HW5)" made on December 21st 2022.

For regression scenarios, we have used the square loss as objective. For this classification task, make use the `multi:softmax` loss/objective function for fitting the model.[2] Note that you have to "flatten" the image data to obtain vector vectors that the XGBoost library can process. In contrast to HA4, consider all the pixels per instance as input data (not only the central pixel).

3. As for the regression task considered in HA6, you can monitor/visualize the training process. For this task, make use of the multiclass loglosses evaluation metric and plot the training and validation loss vs. the boosting rounds.

4. Finally, compute the induced accuracy on the test set and generate a confusion matrix to visualize the errors made by the model. What is the accuracy of the model on the test set?

*Deliverables:* Provide the extended notebook containing your code. In addition, provide the training/validation plot as well as the confusion matrix to your write-up.

# 3 PAC-Bayes-Unexpected-Bernstein (40 points) [Yevgeny]

**Background** The kl and PAC-Bayes-kl inequalities that we have studied in the course work well for binary random variables (the zero-one loss), but, even though they apply to any random variables bounded in the $[0, 1]$ interval, they are not necessarily a good choice if a random variable is non-binary and has a high probability mass inside the interval, because the kl inequality does not exploit small variance. For example, if you have a sample of Bernoulli random variables taking values $\{0, 1\}$ with probability half-half, and you have another sample of non-Bernoulli random variables from a distribution, which is concentrated on $\frac{1}{2}$ (i.e., the random variables always take the value $\frac{1}{2}$), the kl bound on the expectation will be the same in both cases, because it is only based on the empirical average $\hat{p}_n$, even though in the second case the random variables are much more concentrated than in the first.

Non-binary random variables occur, for example, if the loss of false positives and false negatives is asymmetric; in learning with abstention, where an algorithm is occasionally allowed to abstain from prediction and pay an abstention cost

[2] See the corresponding documentation available at `https://xgboost.readthedocs.io/en/stable/python/python_api.html?highlight=classifier#xgboost.XGBClassifier` and `https://xgboost.readthedocs.io/en/stable/parameter.html`.

$c \in (0, \frac{1}{2})$; in working with continuous loss functions, such as the square or the absolute loss (although the Unexpected Bernstein inequality you will derive in this question still requires that the loss is one-side bounded); and many other problems (Wu and Seldin, 2022).

In this question you will derive a concentration of measure inequality belonging to the family of Bernstein's inequalities, which exploit small variance to provide tighter concentration guarantees.

**Guidance** The question is built step-by-step, and if you fail in one of the steps you can still proceed to the next, because the outcomes of the intermediate steps are given. While it is possible to find alternative derivations of the inequality in the literature, you are asked to follow the steps.

1. Let $Z \leq 1$ be a random variable. Show that for any $\lambda \in [0, \frac{1}{2}]$:

$$\mathbb{E}\left[e^{-\lambda Z - \lambda^2 Z^2}\right] \leq e^{-\lambda \mathbb{E}[Z]}.$$

   Point out where you are using the assumption that $\lambda \in [0, \frac{1}{2}]$ and where you are using the assumption that $Z \leq 1$.

   Hint: the following two inequalities are helpful for the proof. For any $z \geq -\frac{1}{2}$ we have $z - z^2 \leq \ln(1+z)$ (Cesa-Bianchi et al., 2007, Lemma 1). And for any $z$, we have $1 + z \leq e^z$.

2. Prove that for $Z \leq 1$ and $\lambda \in [0, \frac{1}{2}]$,

$$\mathbb{E}\left[e^{\lambda(\mathbb{E}[Z] - Z) - \lambda^2 Z^2}\right] \leq 1.$$

3. Let $Z_1, \ldots, Z_n$ be independent random variables upper bounded by 1. Show that for any $\lambda \in [0, \frac{1}{2}]$

$$\mathbb{E}\left[e^{\lambda \sum_{i=1}^n (\mathbb{E}[Z_i] - Z_i) - \lambda^2 \sum_{i=1}^n Z_i^2}\right] \leq 1.$$

4. Let $Z_1, \ldots, Z_n$ be independent random variables upper bounded by 1. Show that for any $\lambda \in (0, \frac{1}{2}]$

$$\mathbb{P}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n Z_i\right] \geq \frac{1}{n}\sum_{i=1}^n Z_i + \frac{\lambda}{n}\sum_{i=1}^n Z_i^2 + \frac{\ln \frac{1}{\delta}}{\lambda n}\right) \leq \delta.$$

5. [Unexpected Bernstein inequality] Explanation: the right hand side of the inequality inside the probability above is minimized by $\lambda^*(Z_1, \ldots, Z_n) =$

$\min\left\{\frac{1}{2}, \sqrt{\frac{\ln\frac{1}{\delta}}{\sum_{i=1}^n Z_i^2}}\right\}$, but we cannot plug $\lambda^*(Z_1,\ldots,Z_n)$ into the bound, because it depends on the sample $Z_1,\ldots,Z_n$, and if you trace the proof back to Point 1, it assumes that $\lambda$ is independent of the sample. And, while the bound in Point 4 holds for any $\lambda$, it does not hold for all $\lambda$ simultaneously. What you will do instead is take a grid of $\lambda$ values and a union bound over the grid, and select $\lambda$ from the grid, which minimizes the bound.

Your task: Let $\Lambda = \{\lambda_1,\ldots,\lambda_k\}$ be a grid of $k$ values of $\lambda$, such that $\lambda_i \in (0, \frac{1}{2}]$ for all $i$. Prove that:

$$\mathbb{P}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n Z_i\right] \geq \frac{1}{n}\sum_{i=1}^n Z_i + \min_{\lambda\in\Lambda}\left(\frac{\lambda}{n}\sum_{i=1}^n Z_i^2 + \frac{\ln\frac{k}{\delta}}{\lambda n}\right)\right) \leq \delta.$$

We will call the above inequality an Unexpected Bernstein inequality.

6. [Empirical comparison of the kl and Unexpected Bernstein inequalities.] We compare the Unexpected Bernstein inequality with the kl inequality. Take a ternary random variable (a random variable taking three values) $Z \in \{0, \frac{1}{2}, 1\}$. Let $p_0 = \mathbb{P}(Z=0)$, $p_{\frac{1}{2}} = \mathbb{P}\left(Z=\frac{1}{2}\right)$, and $p_1 = \mathbb{P}(Z=1)$. Set $p_0 = p_1 = (1-p_{\frac{1}{2}})/2$, i.e., the probability of getting $Z=0$ or $Z=1$ is equal, and we have one parameter $p_{\frac{1}{2}}$, which controls the probability mass of the central value. We will compare the bounds as a function of $p_{\frac{1}{2}} \in [0,1]$. Let $p = \mathbb{E}[Z]$ (in the constructed example, for any value of $p_{\frac{1}{2}}$ we have $p = \frac{1}{2}$, because $p_0 = p_1$). For each value of $p_{\frac{1}{2}}$ in a grid covering the $[0,1]$ interval draw a random sample $Z_1,\ldots,Z_n$ from the distribution we have constructed and let $\hat{p}_n = \frac{1}{n}\sum_{i=1}^n Z_i$ and $\hat{v}_n = \frac{1}{n}\sum_{i=1}^n Z_i^2$. Generate a figure, where you plot the Unexpected Bernstein bound on $p - \hat{p}_n$ and the kl bound on $p - \hat{p}_n$ as a function of $p_{\frac{1}{2}}$ for $p_{\frac{1}{2}} \in [0,1]$. The Unexpected Bernstein bound on $p-\hat{p}_n$ is $\min_{\lambda\in\Lambda}\left(\lambda\hat{v}_n + \frac{\ln\frac{k}{\delta}}{\lambda n}\right)$, and the kl bound on $p-\hat{p}_n$ is $\mathrm{kl}^{-1^+}\left(\hat{p}_n, \sqrt{\frac{\ln\frac{n+1}{\delta}}{n}}\right) - \hat{p}_n$; put attention that in contrast to one of your home assignments, we subtract the value of $\hat{p}_n$ after inversion of kl to get a bound on the difference $p-\hat{p}_n$ rather than on $p$. Take the following values for the comparison: $n = 100$, $\delta = 0.05$, $|\Lambda| = k = \lceil \log_2(\sqrt{n/\ln(1/\delta)}/2)\rceil$, and $\Lambda = \{\frac{1}{2}, \frac{1}{2^2}, \ldots, \frac{1}{2^k}\}$. Briefly comment on the result of empirical evaluation.

Explanation: The kl inequality depends only on the empirical first moment of the sample, $\hat{p}_n = \frac{1}{n}\sum_{i=1}^n Z_i$ and, therefore, it is "blind" to the variance and cannot exploit it. The Unexpected Bernstein inequality depends on the empirical first and second moments of the sample, $\hat{p}_n$ and $\hat{v}_n = \frac{1}{n}\sum_{i=1}^n Z_i^2$. The second moment is directly linked to the variance, $\mathrm{Var}[Z] = \mathbb{E}[Z^2] -$

$\mathbb{E}[Z]^2$ and, therefore, the Unexpected Bernstein inequality is able to exploit small variance.

7. Let $S$ be an i.i.d. sample, $h$ a prediction rule, and $\ell(y', y)$ a loss function upper bounded by 1. Define $\hat{V}(h, S) = \frac{1}{n}\sum_{i=1}^{n} \ell(h(X_i), Y_i)^2$ and $L(h)$ and $\hat{L}(h, S)$ as usual. Show that for any $\lambda \in [0, \frac{1}{2}]$ we have

$$\mathbb{E}\left[e^{n\left(\lambda\left(L(h)-\hat{L}(h,S)\right)-\lambda^2\hat{V}(h,S)\right)}\right] \leq 1.$$

8. Let $S$ and $\ell$ be as before. Let $\mathcal{H}$ be a set of prediction rules, let $\pi$ be a distribution on $\mathcal{H}$ that is independent of $S$. Show that for any $\lambda \in (0, \frac{1}{2}]$

$$\mathbb{P}\left(\exists \rho : \mathbb{E}_\rho[L(h)] \geq \mathbb{E}_\rho\left[\hat{L}(h,S)\right] + \lambda \mathbb{E}_\rho\left[\hat{V}(h,S)\right] + \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{1}{\delta}}{n\lambda}\right) \leq \delta,$$

where $\rho$ denotes a distribution on $\mathcal{H}$.

Hint: Take $f(h, S) = n\left(\lambda\left(L(h) - \hat{L}(h, S)\right) - \lambda^2\hat{V}(h, S)\right)$ and use PAC-Bayes bounding procedure and the result from the previous point.

Side remark: Note that the "optimal" value of $\lambda$ (the one that minimizes the right hand side of the inequality inside the probability) depends on the data, and that the bound does not hold for all $\lambda$ simultaneously. In the next step we resolve this issue by taking a grid of $\lambda$ values and using the best value in the grid.

9. [PAC-Bayes-Unexpected-Bernstein Inequality.] Let $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ be a grid of $k$ values of $\lambda$, such that $\lambda_i \in (0, \frac{1}{2}]$ for all $i$. Let $S$, $\ell$, and $\pi$ be as before. Prove that:

$$\mathbb{P}\left(\exists \rho : \mathbb{E}_\rho[L(h)] \geq \mathbb{E}_\rho\left[\hat{L}(h,S)\right] + \min_{\lambda\in\Lambda}\left(\lambda \mathbb{E}_\rho\left[\hat{V}(h,S)\right] + \frac{\mathrm{KL}(\rho\|\pi) + \ln\frac{k}{\delta}}{n\lambda}\right)\right) \leq \delta.$$

# 4 Dimensionality Reduction (20 points) [Sadegh]

In this question, we would like to visualize a dataset by producing a two-dimensional map of it using t-SNE (Van der Maaten and Hinton, 2008) and PCA. You can use some available implementation for PCA and t-SNE as long as you report the libraries/packages used together with the choice of parameters. Provide code snippets in the report, besides uploading your code.

Consider the dataset `dataset-vis-exam2022.txt`.

(i) Plot the dataset using a (three-dimensional) scatterplot. To generate nice scatterplots, you may use one color for the first half of the dataset (i.e., they have the same label) and another color for the other half. However, the algorithms are oblivious to such information.

(ii) Produce two-dimensional maps of the dataset using PCA on both *normalized* and *unnormalized data* and plot them.

(iii) Now we wish to map the dataset into the two-dimensional space using t-SNE. Apply t-SNE on the dataset, where the initial map points are chosen *randomly* (e.g., according to (Van der Maaten and Hinton, 2008); see the lecture slides). As for perplexity, try 4 choices chosen from $[30, 300]$ that are reasonably different but otherwise arbitrary. (For example, $\mathsf{Perplexity} = 32, 80, 180$, and 280.) Then plot the 4 produced representations.

(iv) Repeat (iii) for the same perplexities chosen but let the initial map points be the map points obtained using PCA on *normalized data*. Then plot the 4 produced representations.

(v) Compare the scattered plots in (i)-(iv) qualitatively. This should include: (a) a comparison between maps generated by PCA and t-SNE; (b) discussion on the impact of perplexity on t-SNE output; and (c) discussion on the impact of the initialization method on t-SNE output.

# References

Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66, 2007.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Yi-Shan Wu and Yevgeny Seldin. Split-kl and PAC-Bayes-split-kl inequalities for ternary random variables. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.