Assignment 7

Statistical Machine Learning

Moritz Haas / Luca Rendsburg / Prof. Ulrike von Luxburg

Summer term 2022 — due **June 13th at 12:00**
Note that you have one additional week for this assignment because of the holidays!

Exercise 1 (Kernelization, 2 points)
Can logistic regression be kernelized? If so, derive the corresponding minimisation objective of the training loss using the representer theorem.

Exercise 2 (Kernel SVM, 1+1+2+1+1=6 points)
Download the dataset `train.csv` that you can find on our website.

(a) Demonstrate that a kernel SVM with a Gaussian kernel is powerful enough to interpolate the training data (achieve zero training error). What is the range of hyperparameters γ and C for which interpolation happens?

Hint: Look at a grid of hyperparameters.

(b) Assuming that the kernel matrix is of full rank, can kernel SVM achieve zero training error on any dataset? Explain.

(c) Now use cross-validation to obtain reasonable values for all hyperparameters. What is your training error with respect to the 0–1-loss? Download `test.csv` from our website. What is your test error with respect to the 0–1-loss?

(d) Use the provided code to plot the decision boundary of your classifiers together with the datapoints. What do you find?

(e) Play around with different kernels (Gaussian, linear, polynomial of degree 2 and 3, ...) and hyperparameters. Use the provided code to have a look at the decision boundaries. Which influence do the hyperparameters have?

Hint: For this exercise, use `sklearn.svm` and GridSearchCV.

Exercise 3 (Interpolation and Generalization, 3+1+2+2=8 points)
The first part of this exercise explores the relation between interpolation and generalization at the example of random feature regression. As a preprocessing step, the lecture discussed deliberately chosen basis functions $x \mapsto (\Phi_1(x), \dots, \Phi_l(x))$. It turns out that it can also be useful to choose these basis functions *randomly*, which is referred to as random features. This is based on the observation that some kernels can be approximated by basis functions drawn from a corresponding distribution. Specifically, some kernels $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ have a corresponding distribution P on $\mathbb{R}^d$ and activation function $\sigma\colon \mathbb{R} \to \mathbb{R}$ such that

$$k(x,x') = \mathbb{E}_{w\sim P}\left[\sigma(\langle w,x\rangle)\sigma(\langle w,x'\rangle)\right] \approx \frac{1}{l}\sum_{i=1}^{l}\sigma(\langle w_i,x\rangle)\sigma(\langle w_i,x'\rangle) = \langle \Phi_W(x), \Phi_W(x')\rangle\,,$$

where $\Phi_W(x) = \frac{1}{\sqrt{l}}(\sigma(\langle w_1,x\rangle),\dots,\sigma(\langle w_l,x\rangle))^T \in \mathbb{R}^l$ and $W = (w_1,\dots,w_l) \in \mathbb{R}^{d\times l}$ consists of independent draws $w_i \sim P$.

In this exercise, we are considering the first order arc-cosine kernel

$$k(x,x') = \frac{1}{\pi}(u(\pi - \arccos(u)) + \sqrt{1-u^2}), \quad \text{where } u = \frac{\langle x,x'\rangle}{\|x\|\|x'\|}\,.$$

The corresponding random features are given by the standard normal distribution $P = \mathcal{N}(0, I_d)$ and ReLU activation function $\sigma(t) = \max\{0, t\}$. We want to compare kernel regression for data $X = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} \in \mathbb{R}^{n\times d}, Y \in \mathbb{R}^n$ with linear regression on random features $\Phi_{\text{rf}} = \begin{pmatrix} \Phi_W(x_1)^T \\ \vdots \\ \Phi_W(x_n)^T \end{pmatrix} \in \mathbb{R}^{n\times l}$. In both cases, we intentionally choose no regularization.

(a) Complete the implementation of `kernel_regression` and `random_feature_regression`, which output the corresponding prediction functions. The latter function additionally outputs the normalized norm of the regression parameter $\|\alpha\|/\sqrt{d}$, where $f_{\text{rf}}(x) = \alpha^T \Phi_W(x)$.

(b) For the given train and test data, train multiple random feature predictors for different choices of l. Plot the corresponding train error, test error, and regression parameter norm against the overparameterization ration l/n. Also train the kernel predictor and include the corresponding train and test error as horizontal lines.
Hint: first choose the overparameterization ratio $\gamma = l/n$ and then define $l = \lceil n\gamma \rceil$.

(c) Discuss the plot under several aspects:

(i) How does the complexity of the hypothesis class change with the overparameterization ratio γ? Argue in terms of number of constraints and number of variables for the random feature regression problem $\Phi_{\text{rf}}\alpha \overset{!}{=} Y$.

(ii) Discuss train and test error of random feature regression in the underparameterized regime $\gamma < 1$.

(iii) What happens at the interpolation threshold $\gamma = 1$?

(iv) Discuss train and test error of random feature regression in the overparameterized regime $\gamma > 1$. What is surprising about the behavior of the test error? Do these regressors have an additional inductive bias which might explain this behavior?

The last part of this exercise gives a low-dimensional visualization of the phenomenon that interpolators can achieve good test error. To this end, we consider a different regressor: for training data $X \in \mathbb{R}^{n\times d}, Y \in \mathbb{R}^n$, the Nadaraya-Watson estimator simply predicts a weighted average of the m nearest neighbors

$$f(x) = \frac{\sum_{i=1}^m k(x, X_i)Y_i}{\sum_{i=1}^m k(x, X_i)} ,$$

where the training points $X_1, \ldots, X_m$ are the m nearest neighbors of x with respect to kernel k.

(d) Complete the implementation of `nadaraya_watson_regression` and plot the prediction function along with the training data for the singular kernel $k(x, x') = 1/\|x - x'\|$ with $m = 10$.
Hint: since kernels encode similarities, the points x and X_i are close if $k(x, X_i)$ is large.

Exercise 4 (Pentecoast Prediction Competition, Optional, 3x10, 7x5, otherwise 3 Bonus Points) This exercise is a machine learning competition! It is a bonus exercise, you do not have to participate. However, participation is highly encouraged. The idea is that you experience the process of machine learning first-hand by yourself. The best submissions can receive up to 10 bonus points. All students who take part in this competition receive 3 bonus points. The machine learning problem and competition details are as follows:

A group of biologists has laboriously collected a gene expression dataset. Unfortunately, half of the expressions of gene `GXYLT1` are lost. The biologists are heartbroken. You offer some relief by predicting the lost gene expressions. The biologists tell you that

$$\min\left\{1, \frac{|y - \hat{y}|}{50 + y}\right\} \tag{1}$$

is a meaningful measure of loss for this application (where y is the true gene expression level, and you predict $\hat{y}$).

- Download the gene expression dataset `geneexp.csv` from our website. This dataset contains 900 measurements of 1000 different genes. A gene expression is a non-negative number. The lost expressions of gene `GXYLT1` are indicated by the value -1.

- Predict the missing gene expressions, replace the value -1 with your predictions and submit the modified `geneexp.csv`, together with a short description of your solution and all code to reproduce your results to `sml2022competition@gmail.com`. The three best submissions (according to the test error with loss function (1) on our held out data with the true gene expressions) receive 10 bonus points, the next seven best submissions 5 bonus points and all other meaningful submissions 3 bonus points. The deadline of the competition is the deadline of this assignment. Have fun!

- You can use all machine learning methods and programming tools that you know about.

- Here are some things that you might want to do and think about:

  - Perform some exploratory data analysis. What does the distribution of features and outcome look like? Perhaps you should pre-process this dataset before you apply machine learning.
  - You might want to separate the data into a training and test set, or use cross-validation.
  - You should not spend too much time theorizing what might be the best machine learning method for this problem - get your hands dirty and try something! This will give you a first impression of what can be achieved. After you have this baseline: How can you improve?
  - How do you want to deal with the unusual loss function? At first, you might simply use another loss, like mean-squared error, and then see how it relates to (1). In a more advanced step, think how the loss can be incorporated more directly.

  Hint: You can assume that the gene expressions are missing at random. You can also use our forum on Ilias in order to discuss this problem - but not to exchange solutions.

Exercise 5 (Feedback Poll: Week 7, 0 bonus points but 1 gratitude)
Every week there will be a feedback survey on Ilias about the lecture, tutorials and exercise sheets. If you participate, it helps us in understanding what we can improve. Future polls will be anonymous again to ensure open feedback. The poll asks about the lectures that cover the topics treated in the respective sheet.