Elements of Machine Learning, WS 2022/2023
Jilles Vreeken and Aleksandar Bojchevski
Exercise Sheet #3: *Beyond Linear*

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY | UNIVERSITÄT DES SAARLANDES

Deadline: Thursday, December 22, 2022, 16:00

Before solving the exercises, read the instructions on the course website.

- For each theoretical problem, submit a single `pdf` file that contains your answer to the respective problem. This file may be a scan of your (legible) handwriting.
- For each practical problem, submit a single `zip` file that contains
  - the completed jupyter notebook (`.ipynb`) file,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows all your results.
- For the bonus question, submit a single `zip file` that contains
  - a `pdf` file that includes your answers to the theoretical part,
  - the completed jupyter notebook (`.ipynb`) file for the practical component,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows your results.
- Every team member has to submit a signed Code of Conduct.

Problem 1 (T, 4 Points). **Parametric? Non-parametric?** We often like to describe methods as "parametric" or "non-parametric". But what does this mean?

1. (1 Point) Describe, in your own words, what the difference between a parametric and a non-parametric method is.
2. (3 Points) For each of the following methods you have learned about in the lectures so far, decide if it is parametric or non-parametric and explain your decision:
   - Ordinary Least Squares
   - LASSO
   - Polynomial Regression
   - Smoothing Splines
   - Local Regression
   - Generalized Additive Models

Problem 2 (T, 5 Points). **Splines**
In the lectures, you have learned that the space of cubic splines with K knots has dimension $K + 4$. But how did we arrive at this number?

1. (2 Points) Assume that we have $K = 1$ knot, and let ζ be this knot. Further, let the spline be written as

$$f(x) = \begin{cases} a_3x^3 + a_2x^2 + a_1x + a_0, & x \leq \zeta \\ b_3(x-\zeta)^3 + b_2(x-\zeta)^2 + b_1(x-\zeta) + b_0, & x > \zeta. \end{cases}$$

   Let $a_0, \ldots, a_3$ be given. Show that b_0, b_1, b_2 are fully determined by the constraint imposed by f being twice differentiable at $x = \zeta$. What does this mean for the degrees of freedom of the model?
2. (1 Point) Write down a similar presentation for a *quadratic* spline with $K = 1$ knot at ζ. How many parameters does this model have under the requirement that the spline be differentiable *once*?

3. (1 Point) How large is the difference in free parameters between quadratic and cubic splines? Is this difference bigger than you would intuitively expect? Is it smaller?

4. (1 Point) Explain why cubic splines would be more suitable than quadratic splines for the data in Fig. 1. The $K = 2$ knots ζ_1, ζ_2 are located as shown.

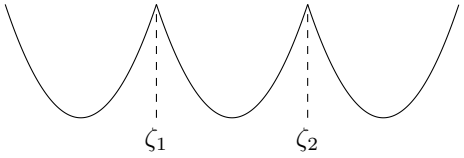

Figure 1: Three Parabolas side by side.

Problem 3 (T, 5 Points). **Generalized Additive Models**
In Generalized Additive Models (GAMs), we are interested in predicting our target variable $Y \in \mathbb{R}$ based on the variables $X_1, \ldots, X_p$ as follows:

$$g(Y) = \alpha + \sum_{j=1}^{p} f_j(X_j) \ ,$$

where we assume that $\mathbb{E}(f_j(X_j)) = 0$ for all j. For the rest of this exercise, we will assume $g = \text{id}$ to be the identity function and that the dimensionality of X is $p = 2$.

1. (2 Points) In the lecture we have seen the backfitting algorithm. Let the smoothing operators $\mathcal{S}_j = \mathcal{S}_\lambda$ for $\lambda \geq 0$ take the following form

$$\hat{\beta}_j = \arg\min_{\beta} \sum_i \left(y_i - \alpha - \sum_{k:k \neq j} \hat{f}_k(x_{ki}) - \beta x_{ji} \right)^2 + \lambda \beta^2$$
$$\hat{f}_j(X_j) = \hat{\beta}_j X_j \ .$$

   Write out the first iteration of the backfitting algorithm. That is, compute the parameters $\hat{\beta}_j$ for both $\hat{f}_1$ and $\hat{f}_2$ after the first iteration of the algorithm.

2. (1 Point) Without proof, will iterating this algorithm produce the same result as one of the methods you have learned about in class? Explain your reasoning.

3. (1 Point) Under which conditions will the results of the backfitting algorithm with this smoothing operator $\mathcal{S}_\lambda$ depend on the order of which $\hat{f}_j$ is updated first?

4. (1 Point) Write down the smoothing operator based on cubic smoothing splines.

Problem 4 (P, 6 Points). **Local Regression**
In this exercise, we will look at the effect of the choice of different kernels on the outputs of the models fit by local regression.
Note: You should use the `scikit-misc` package. If, for whatever reason, this does not work for you, an alternative implementation of local regression has been provided in the file `util.py`.

1. (1 Point) Load the data and plot it. How would you describe the relationship between the variables?

2. (2 Points) Implement the following four kernels

$$K_1(x, x') = 1$$
$$K_2(x, x'; \lambda) = \left(1 - \left|\frac{x - x'}{\lambda}\right|^3\right)^3$$
$$K_3(x, x'; \lambda) = \exp(-\lambda |x - x'|^2)$$
$$K_4(x, x'; L) = \begin{cases} 1, & \exists l \in \{1, \dots, L\} : \frac{l-1}{L} \leq x, x' < \frac{l}{L} \\ 0, & \text{otherwise} \end{cases}$$

3. (2 Points) Write a function which takes the data as well as a kernel as input, and use `skmisc.loess.loess` to fit the model at each data point. Use the parameters $\lambda = 1$ for K_2, K_3 and $L = 2$ for K_4.

4. (1 Point) Plot the predictions of each model and explain how these models differ. Which one looks best to you? Why?

Problem 5 (Bonus). **Ex Pluribus Unum**
In the lectures, you have seen several methods to learn non-linear models. However, all of them are based on fitting one complex model. Why don't we instead fit multiple simple models and combine them?
Let our data be $(x_i, y_i)_{i=1}^n$ where $x_i \in [-1, 1)$ and $y_i \in \mathbb{R}$ are both centered. As usual, our goal will be to predict y from x. In the following, we will explore how we could fit and combine multiple simple models.

1. Our first attempt will be to split the data x, y into multiple equally-sized subsets $S_k, k = 1, \dots, K$ at random and fit an OLS independently on each S_k. Write down the parameters $\hat{\beta}_0^{(k)}, \hat{\beta}^{(k)}$ for each dataset.

2. With the above models, given input point x_0, we predict $\hat{y}_0$ by averaging all the obtained models. Write down this model prediction.

3. What shape does the model that results from this procedure take? Explain why doing this is worse than fitting one OLS model over all the data if we assume that the data is indeed generated from the model $y = \beta_0^* + \beta_1^* x + \epsilon$. (Hint: Use the Gauss-Markov Theorem.)

4. Now, let us try something different. Split the data x, y into K different datasets $S_k, k = 1 \dots, K$ as follows: $S_k = \left\{(x_i, y_i) : -1 + \frac{2(k-1)}{K} \leq x_i < -1 + \frac{2k}{K}\right\}$ and fit an OLS independently on each S_k. Draw an illustrative example of the data split and the resulting model for $K = 3$.

5. Given a new input point $x_0 \in [-1, 1)$, how would you use this model to predict $\hat{y}_0$?

6. Explain the problem with this way of splitting the data into multiple intervals. How could we fix it?

7. Do you think this method would generalize well to higher-dimensional X? Why? Why not?