Elements of Machine Learning, WS 2022/2023
Jilles Vreeken and Aleksandar Bojchevski
Exercise Sheet #4: *Beyond Linear*

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

Solution.

1. A parametric method learns a function $\hat{f}(\cdot;\theta)$ of a fixed form described by a finite number of parameters $\theta_1, \ldots, \theta_k$, where k does not depend on the number of samples in X.
   In contrast, in a non-parametric method the number k of parameters used to describe $\hat{f}$ can increase with the number of samples in X. Furthermore, the shape of the function $\hat{f}$ will also depend on the data to which it is fit.
2. - OLS: Parametric, the number of parameters is simply $p+1$ when $X \in \mathbb{R}^{n\times p}$.
   - LASSO: As in OLS, the number of parameters is still $p+1$.
   - Polynomial regression: The degree of the polynomial is fixed at d, and for an X as above we need at most $\sum_{k=0}^{d}\binom{p}{k} \leq 1 + d\binom{p}{d}$ coefficients. The method is therefore parametric.

- Smoothing Splines: The smoothing spline solution for data $X, y \in \mathbb{R}^n$ is always a natural spline with n knots. Therefore, the number of parameters required increases with n and Smoothing Splines are therefore non-parametric.
- Local Regression: For every point x_0, a different parameter vector $\hat{\beta}$ is fit and used for prediction, so no finite set of parameters can describe the model. Local Regression is therefore in general non-parametric.
- Generalized Additive Models: It's complicated. When an explicit basis representation is given for the GAM, and each of the individual functions can be described by finitely many parameters, then this is parametric. An example would be linear regression as a special case of GAMs.
  When the back-fitting algorithm is used to fit the model, whether the resulting function is parametric or not depends on the smoothing operators S_j.

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.

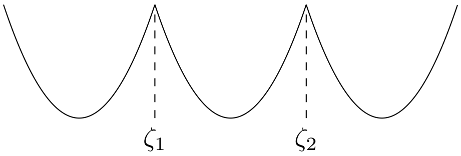

Figure 1: Three Parabolas side by side.

Solution.

1. f being twice differentiable at $x = \zeta$ means that the zeroth, first and second derivatives of both cases in the definition of f match at this point. The constraints are therefore

$$\begin{aligned} a_3\zeta^3 + a_2\zeta^2 + a_1\zeta + a_0 &= b_0 && f \\ 3a_3\zeta^2 + 2a_2\zeta + a_1 &= b_1 && f' \\ 6a_3\zeta + 2a_2 &= 2b_2 && f'' \end{aligned}$$

   so that given $a_0, \ldots, a_3$ the only free parameter is b_3. We therefore have only $K = 1$ additional degree of freedom in choosing our spline.

2. For the quadratic spline with one knot, we can write

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

   Then the zeroth and first-order constraints are

$$\begin{aligned} a_2\zeta^2 + a_1\zeta + a_0 &= b_0 \\ 2a_2\zeta + a_1 &= b_1 \end{aligned}$$

   so again, the only free parameter given $a_0, \dots, a_2$ would be b_2. Therefore we have $K + 3 = 4$ free parameters a_0, a_1, a_2, b_2.

3. The result that cubic splines have $K + 4$ free parameters and quadratic splines have $K + 3$ parameters is rather unintuitive. After all, for large K we have "almost" the same number of parameters—relative to how many parameters we have in total. That is, for each of the $K + 1$ intervals separated by the K knots we have on average

$$\begin{aligned} \frac{K+4}{K+1} &= 1 + \frac{3}{K+1} \quad \text{(cubic)} \\ \frac{K+3}{K+1} &= 1 + \frac{2}{K+1} \quad \text{(quadratic)} \end{aligned}$$

   free parameters, which for large K is not very much of a difference.
   And yet, in every single interval the function is a cubic polynomial, which we would expect to be much more expressive than a quadratic polynomial.

4. While the displayed data is piecewise quadratic, a quadratic spline is not suitable for modeling it. The issue is the following: a quadratic spline can't have an inflection point in its interior. That is, once the left-most parabola is oriented, the orientation of the second parabola is fixed as shown in the Fig. 2 on the left. Therefore, when fitting a quadratic spline to the data, the best we can do (in terms of orientation, not scale), is to direct the first and third parabolas correctly, and hope the second isn't too bad.

   In contrast, with cubic splines we do not have this issue, as seen in Fig 2 on the right. Since cubic splines can have an inflection point on the inside, they can turn around and capture more of the parabola in this way.

   Of course, at the end of the day, when constraints (e.g. natural splines) or regularization (e.g. smoothing splines) are included, the difference between the two models becomes smaller.

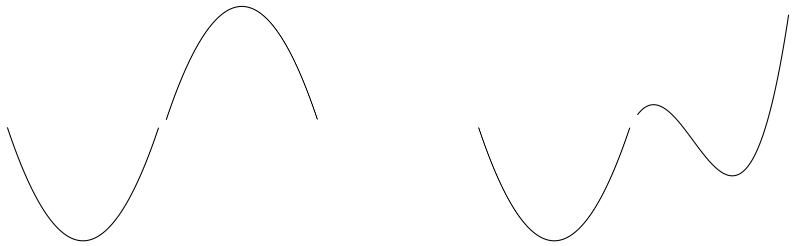

Figure 2: A representation of why quadratic splines do not do well at capturing three parabolas with the same direction, while cubic splines have a better ability to capture them. Left: The first parabola completely determines the orientation of the second parabola. Right: With well-chosen parameters, the left-most part going in the wrong direction can be made very small, while the correctly-oriented part can be made to capture the second parabola arbitrarily well.

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.

Solution.

1. We know that in the first step, $\hat{f}_2 = 0$ at the start, so that we can write

$$\begin{aligned}\hat{\beta}_1 &= \arg\min_{\beta} \sum_i (y_i - \alpha - \beta x_{i1})^2 + \lambda\beta^2 \\ &= \frac{x_{:,1}^\top y}{x_{:,1}^\top x_{:,1} + \lambda} \ ,\end{aligned}$$

   where $x_{:,1}$ denotes the vector containing all observations of the first feature. Consequently, the residual is $y' = y - \hat{\beta}_1 x_{:,1}$. plugging this into the update for $\hat{\beta}_2$ we obtain

$$\begin{aligned}\hat{\beta}_2 &= \frac{x_{:,2}^\top y'}{x_{:,2}^\top x_{:,2} + \lambda} \\ &= \frac{x_{:,2}^\top y}{x_{:,2}^\top x_{:,2} + \lambda} - \hat{\beta}_1 \frac{x_{:,2}^\top x_{:,1}}{x_{:,2}^\top x_{:,2} + \lambda} \ ,\end{aligned}$$

   which is an adjusted version of the estimate $\frac{x_{:,2}^\top y}{x_{:,2}^\top x_{:,2} + \lambda}$ obtained by "correcting for" the correlation between the variables X_1 and X_2.

2. Given the form of the smoothing function and the shape of the update for $\hat{\beta}_1$, it is easy to suspect that the result is the same as that of ridge regression. And this is in fact correct, but the proof of convergence is much too involved to include here.

3. We have seen on the previous exercise sheet that for OLS, i.e., $\lambda = 0$, the solution-space of $\hat{\beta}_1, \hat{\beta}_2$ contains more than a single value when $x_{:,1}, x_{:,2}$ are correlated with each other. In this case, the order in which $\hat{\beta}_1$ and $\hat{\beta}_2$ are updated matter and will result in different solutions.

4. For the smoothing operator based on cubic smoothing splines, we can simply use the definition from the lectures, as applied to the residuals after taking into account all other splines fit thus far:

$$\hat{f}_j(x) = \arg\min_{f_j} \sum_i \left(y_i - \alpha - \sum_{k:k\neq j} \hat{f}_k(x_{:,ki}) - f_j(x_{:,ji}) \right)^2 + \lambda_j \int f_j''(u)^2 du \, .$$

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

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

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 $S_k, k = 1, \ldots, K$ 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?