5.4 Smoothing splines

Smoothing splines

Smoothing splines are similar to regression splines, but arise in a slightly different setting.

If there are too many knots, the fitted curve would be too rough. One way to overcome this problem is via shrinkage. This avoids the knot selection problem entirely by using a maximal set of knots. The complexity of the fit is controlled by regularisation.

In this problem we find a function f (with two continuous derivatives) that minimizes the **penalized residual sum of squares**

$$RSS(f,\lambda) = \sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda \int (f^{''}(t))^2 dt,$$

where λ is a fixed **smoothing parameter**. The first term measures closeness to the data and the second term penalizes curvature in the function. λ establishes a tradeoff between the two.

It can be shown that this problem has an **explicit, finite-dimensional, unique minimizer** which is a **natural cubic spline** with knots at the unique values of the x_i, $i = 1, \ldots, N$ and the solution can be expressed as

$$f(x) = \sum_{j=1}^{N} N_j(x)\theta_j,$$

where $N_j(x)$ are an N-dimensional set of basis functions for representing this family of natural splines.

The above criterion can be, therefore, rewritten as

$$RSS(\theta,\lambda) = \|\boldsymbol{y} - \boldsymbol{N}\theta\|^2 + \lambda\theta^\top \boldsymbol{\Omega}_N \theta,$$

where $\boldsymbol{N}_{ij} = N_j(x_i)$ and $\boldsymbol{\Omega}_{\boldsymbol{N}jk} = \int N_j^{''}(t)N_k^{''}(t)dt$. The solution is here

$$\hat{\theta} = (\boldsymbol{N}^\top \boldsymbol{N} + \lambda \boldsymbol{\Omega}_N)^{-1}\boldsymbol{N}^\top \boldsymbol{y},$$

a generalized ridge regression. The **fitted smoothing spline** is given by

$$\hat{f}(x) = \sum_{j=1}^{N} N_j(x)\hat{\theta}_j.$$

Degrees of freedom

Denote by $\hat{\boldsymbol{f}}$ the N-vector of fitted values $\hat{f}(x_i)$ at the training predictors x_i. Then

$$\hat{\boldsymbol{f}} = \boldsymbol{N}(\boldsymbol{N}^\top \boldsymbol{N} + \lambda \boldsymbol{\Omega}_N)^{-1}\boldsymbol{N}^\top \boldsymbol{y} = \boldsymbol{S}_\lambda \boldsymbol{y}.$$

Note that the fit is linear in $\boldsymbol{y}$, and the finite linear operator $\boldsymbol{S}_\lambda$ depends only on x_i and λ.

The effective degrees of freedom of a smoothing spline are

$$df_\lambda = \mathrm{trace}(\boldsymbol{S}_\lambda).$$

Since df_λ is monotone in λ for smoothing splines, we can invert the relationship and specify λ by **fixing** df_λ. **Fixing the degrees of freedom** is an intuitive way of finding λ, which is an alternative for cross-validation.

For example in `R` one can use $\texttt{smooth.spline}(\texttt{x}, \texttt{y}, \texttt{df} = 6)$ to specify the amount of smoothing. This encourages a more traditional mode of model selection, where we might try a couple of different values of `df`, and select one based on approximate F-tests, residual plots and other more subjective criteria. This is, in particular, useful when comparing many different smoothing methods. (Used in generalised additive models).

Cross-validation curve

The N-fold cross-validation curve is:

$$CV(\hat{f}\lambda) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}^{-i}\lambda(x_i))^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_\lambda(i,i)}\right)^2,$$

which can be computed for each value of λ, from the original fitted values and the diagonal elements $S_\lambda(i,i)$ of $\boldsymbol{S}_\lambda$. Recall that the notation $\hat{f}_\lambda^{-i}(x_i)$ indicates the fitted value for the smoothing spline evaluated at x_i, where the fit uses all of the training observations except for the ith observation

(x_i, y_i). This is an example of leave-one-out cross-validation.

Genearalized Cross-Validation

Generalised Cross-validation (GCV) replaces S_{ii} above by $tr(\boldsymbol{S}_\lambda)/N$, which achieves greater numerical stability. We then have

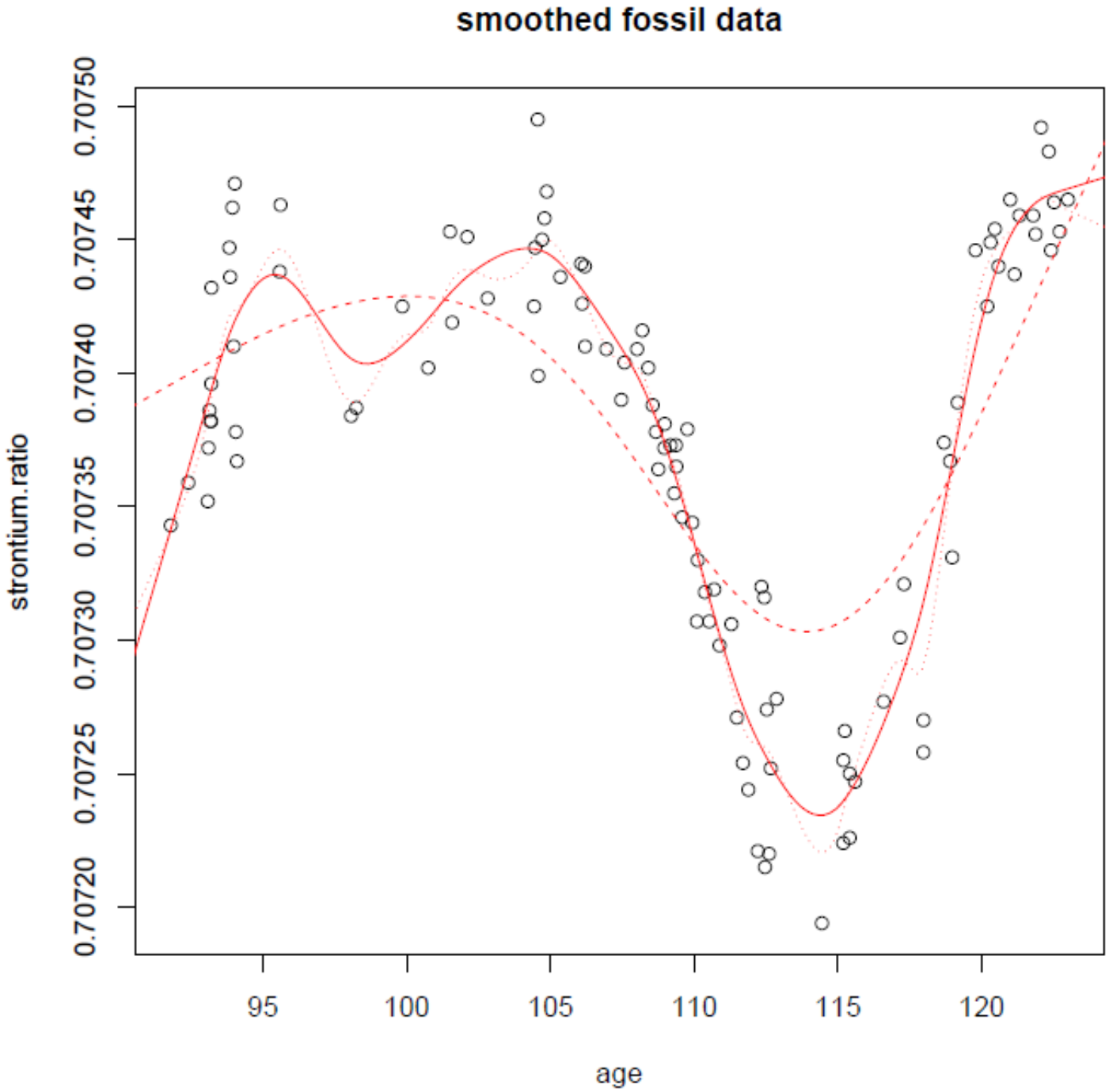
$$\mathrm{GCV} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{f}_\lambda^{-i}(x_i)}{1 - \boldsymbol{S}_\lambda(i,i)}\right)^2,$$

which we can rewrite as

$$\mathrm{GCV} = \frac{1}{N}\frac{\mathrm{RSS}}{(1 - \mathrm{tr}(\boldsymbol{S}_\lambda)/N)^2}.$$

Example: Generalized Cross-Validation

```
fossil <- read.table("/course/data/fossil.dat",header=T)
plot(fossil,pch=1, main="smoothed fossil data")
ss1=smooth.spline(fossil$age,fossil$strontium.ratio)
ss2=smooth.spline(fossil$age,fossil$strontium.ratio,df = 4)
ss3=smooth.spline(fossil$age,fossil$strontium.ratio,df = 25)
xx=seq(90,130,length.out = 200)
lines(predict(ss1,xx), col=2)
lines(predict(ss2,xx), col=2,lty=2)
lines(predict(ss3,xx), col=2,lty=3)
```

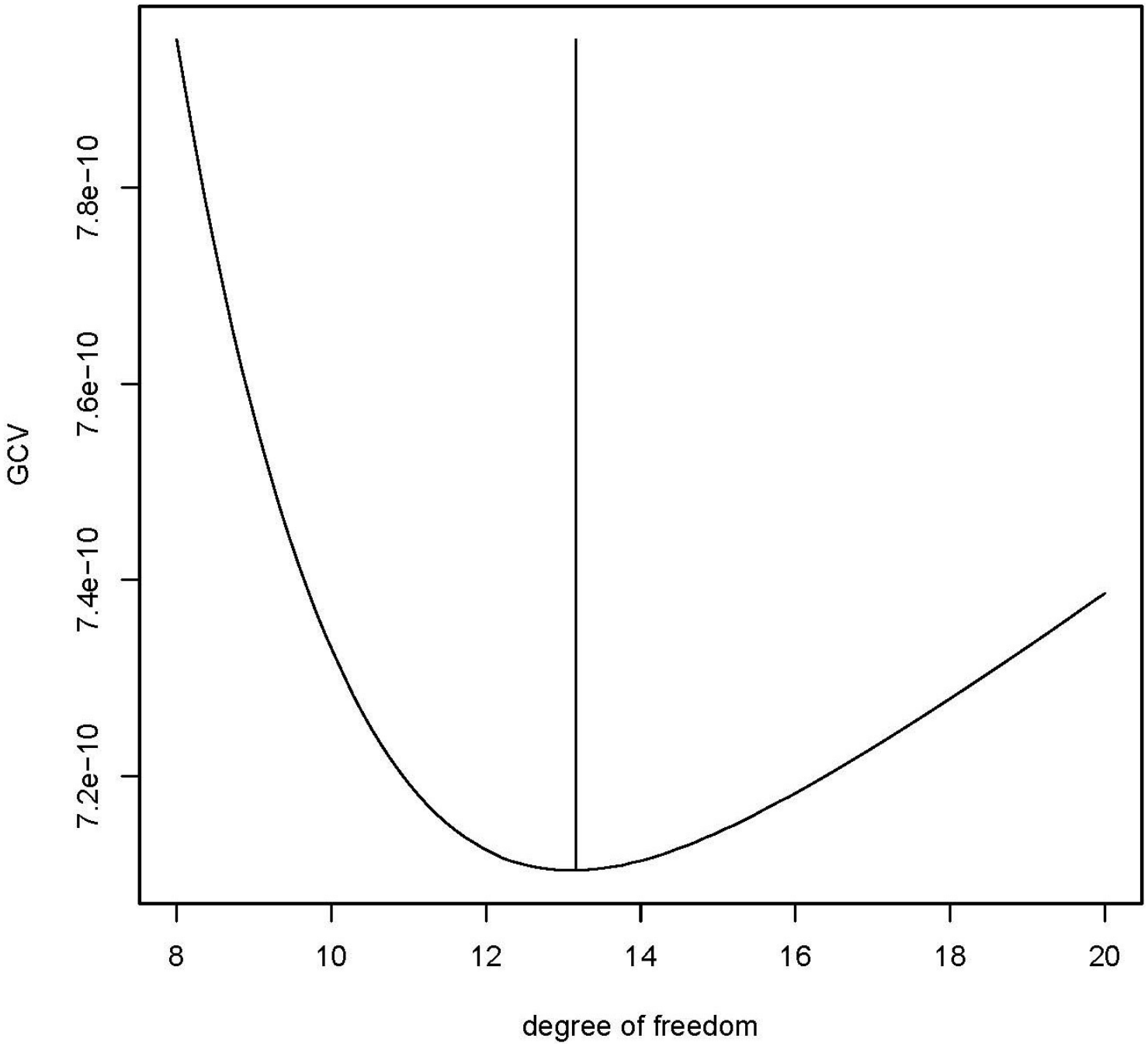


The following code shows the dependence of the GCV on the d.o.f.

```
fossil <- read.table("/course/data/fossil.dat",header=T)
num.df <- 101
df.grid<-seq(8,20,length=num.df)
GCV <- rep(NA,num.df)
n <- length(fossil$age)
for (i in 1:num.df)
{
 fit    <- smooth.spline(fossil$strontium.ratio~fossil$age,df=df.grid[i])
 RSS    <- sum((fossil$strontium.ratio-predict(fit,fossil$age)$y)^2)
 GCV[i] <- (1/n)*RSS/(1-df.grid[i]/n)^2
}
```

```
plot(df.grid,GCV,type="l",xlab="degree of freedom", ylab="GCV")

ind.min <- order(GCV)[1]
lines(rep(df.grid[ind.min],2),c(min(GCV),max(GCV)))
```



The same result can be obtained by simply setting **`cv = FALSE`** in **`smooth.spline`** function:

```
fossil <- read.table("/course/data/fossil.dat",header=T)
sp<-smooth.spline(fossil$age,fossil$strontium.ratio,cv=FALSE)
sp$df
```

```
[1] 13.10385
```

Activity: Smooth splines

Question *Submitted Feb 18th 2024 at 9:29:22 pm*

Let

$$y_i = f(x_i) + \varepsilon_i \text{ for } i = 1, \ldots, n$$

denote a model in which the response variable y depends on a single predictor variables x. Assume that the x_i are distinct and ordered, so $x_1 < x_2 < ... < x_n$. Let a and b denote real numbers such that

$$a = x_0 < x_1 < x_2 < ... < x_n < x_{n+1} = b.$$

Let g denote a natural cubic spline on $[a, b]$. List 2 defining properties of natural cubic splines. Let $\mathcal{S}[a, b]$ denote the space of all functions h defined on $[a, b]$ which are continuous with two continuous derivatives and put

$$S(h) = \sum_{i=1}^{n} \left[y_i - h\left(x_i\right)\right]^2 + \lambda \int_a^b h''(x)^2 dx$$

with $\lambda > 0$. Let $z_1, ...z_n$ denote real numbers and assume that

$$g(x_i) = h(x_i) = z_i \text{ for } i = 1, ..., n$$

where g is a natural cubic spline on $[a, b]$ and $h \in \mathcal{S}[a, b]$. Show that

$$\int_a^b g''(x)^2 dx \leq \int_a^b h''(x)^2 dx.$$

Solution

Two defining properties of natural cubic spline: cubic polynomial between knots continuous and two continuous derivatives linear beyond the first and last knot points.

Let $\tilde{g} = h - g$.

Then $\tilde{g}(x_i) = 0, i = 1, ..., n$. Since g is a natural cubic spline (linear beyond x_1 and x_n), $g''(x) = 0$ on $[a, x_1]$ and $[x_n, b]$.

Now

$$\begin{aligned}\int_a^b g''(x)\tilde{g}''(x)dx &= [g''(x)\tilde{g}'(x)]_a^b - \int_a^b g'''(x)\tilde{g}'(x)dx \\ &= -\int_a^b g'''(x)\tilde{g}'(x)dx\end{aligned}$$

integrating by parts and since $g''(a) = g''(b) = 0$. Since $g(x)$ is a cubic spline, (hence a cubic polynomial on each interval (x_j, x_{j+1})) $g'''(x)$ is a constant on (x_j, x_{j+1}), $j = 0, ..., n$.

Write $g'''(x_j^+)$ for the value of $g'''(x)$ on the interval (x_j, x_{j+1}).

Then, noting that $g'''(x_0^+) = g'''(x_n^+) = 0$,

$$\begin{aligned}\int_a^b g'''(x)\tilde{g}'(x)dx &= \sum_{j=1}^{n-1} g'''(x_j^+)\int_{x_j}^{x_{j+1}} \tilde{g}'(x)dx \\ &= \sum_{j=1}^{n-1} g'''(x_j^+)(\tilde{g}(x_{j+1}) - \tilde{g}(x_j))\end{aligned}$$

which is 0 since $\tilde{g}(x_j) = 0, j = 1, ..., n$. So $\int_a^b g''(x)\tilde{g}''(x) = 0$.

which is 0 since $\tilde{g}(x_j) = 0$, $j = 1, ..., n$. So $\int_a^b g''(x)\tilde{g}''(x) = 0$.

Now, $h = g + \tilde{g}$, so

$$\begin{aligned}\int_a^b h''(x)^2 dx &= \int_a^b (g''(x) + \tilde{g}''(x))^2 dx \\ &= \int_a^b g''(x)^2 dx + \int_a^b \tilde{g}''(x)^2 dx + 2\int_a^b g''(x)\tilde{g}''(x) dx \\ &= \int_a^b g''(x)^2 dx + \int_a^b \tilde{g}''(x)^2 dx \\ &\geq \int_a^b g''(x)^2 dx.\end{aligned}$$

Equality occurs only if $\tilde{g}(x) = 0$ (since $\tilde{g}(x_i) = 0$ for $i = 1, ..., n$).

This exercise shows that of all functions $h \in S(a, b)$, $g(x)$ is the one that is smoothest in the sense of minimizing

$$J(f) = \int_a^b f''(x)^2 dx.$$

Activity in R: Smoothing spline

Recall the **Wage** data set in the library **ISLR**. Visualise **age** vs **wage** using a scatterplot. Fit a smoothing spline using the `smooth.spline()` function in R. Compare two methods of fitting the smoothing spline: one of them is by providing $\texttt{df} = 16$ and the second one by using cross-validation $\texttt{cv} = \texttt{TRUE}$. Add the visualisation of both fitted splines onto the scatterplot. Explain how the fit is obtained in both cases.

Additional Activity

Question 1 *Submitted Feb 18th 2024 at 9:33:24 pm*

Which of the following statements are true about smoothing splines:

in the smoothing spline optimisation problem we look for a funciton f that minimises the penalised residual sum of squares:

$$RSS(f,\lambda) = \sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda \int (f^{'}(t))^2 dt$$

where λ is a fixed smoothing parameter;

- [x] the first term in the smoothing spline optimisation problem measures closeness to the data and the second term penalises curvature in the function f;

- [ ] the smoothing spline optimisation problem has an explicit, finite dimensional, unique minimiser which is a polynomial of degree 4.

Question 2 *Submitted Feb 18th 2024 at 9:33:30 pm*

The effective degrees of freedom of a smoothing spline are:

- [ ] $df_\lambda = \text{trace}(\boldsymbol{S}\lambda)/2$, where $\boldsymbol{S}_\lambda$ is given by the relation $\hat{\boldsymbol{f}} = \boldsymbol{S}_\lambda \boldsymbol{y}$;

- [x] $df_\lambda = \text{trace}(\boldsymbol{S}_\lambda)$, where $\boldsymbol{S}_\lambda$ is given by the relation $\hat{\boldsymbol{f}} = \boldsymbol{S}_\lambda \boldsymbol{y}$;

- [ ] $df_\lambda = \text{trace}(\boldsymbol{S}_\lambda)$, where $\boldsymbol{S}_\lambda$ is given by the relation $\hat{\boldsymbol{f}} = \boldsymbol{N}(\boldsymbol{N}^\top \boldsymbol{N} + \lambda \boldsymbol{S}_\lambda)^{-1}\boldsymbol{N}^\top \boldsymbol{y}$.

Question 3 *Submitted Feb 18th 2024 at 9:33:36 pm*

Generalised Cross-Validation replaces S_{ii} in the formula for the cross validation curve by $\text{trace}(\boldsymbol{S}_\lambda)$.

- [ ] Yes

- [x] No