# Series 5

**1.** The leave-one-out CV-score for the cubic smoothing spline and the least squares parametric estimator can be written in such a way that it depends only on the estimator $\hat{m}(\cdot)$ which is computed from the *full* dataset. To obtain the CV-score, it is therefore not necessary to calculate the leave-one-out estimators $\hat{m}_{n-1}^{(-i)}(\cdot)$. From the manuscript (Formula 4.5) we learn:

$$n^{-1}\sum_{i=1}^{n}\left(Y_i - \hat{m}_{n-1}^{(-i)}(X_i)\right)^2 = n^{-1}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{m}(X_i)}{1 - S_{ii}}\right)^2,$$

where $S$ is the *hat-matrix* of the linear estimator $\hat{m}(\cdot)$. In this exercise we are going to prove this formula step by step in the case of multiple-linear-regression $y_i = \mathbf{x_i^T}\beta + \epsilon_i$.

**a)** Show that for an invertible $p \times p$-matrix $A$ and two $p$-vectors $\mathbf{a}$ and $\mathbf{b}$ with $\mathbf{b^T}A^{-1}\mathbf{a} \neq 1$ the matrix $A - \mathbf{ab^T}$ is invertible too and that its inverse can be computed as follows:

$$(A - \mathbf{ab^T})^{-1} = A^{-1} + \frac{1}{1 - \mathbf{b^T}A^{-1}\mathbf{a}} \cdot A^{-1}\mathbf{ab^T}A^{-1}.$$

**b)** Show the following formula which describes the influence of omitting the $i^{\text{th}}$ observation for the multiple-linear-regression estimator:

$$\hat{\beta}^{(-\mathbf{i})} - \hat{\beta} = -\frac{y_i - \mathbf{x_i^T}\hat{\beta}}{1 - S_{ii}}(X^TX)^{-1}\mathbf{x_i}.$$

**Hints:** Let $A := X^TX = \sum_{i=1}^{n}\mathbf{x_i x_i^T}$, $\mathbf{c} := X^Ty = \sum_{i=1}^{n} y_i\mathbf{x_i}$.
Now you might start as follows: $\hat{\beta}^{(-\mathbf{i})} = (A - \mathbf{x_i x_i^T})^{-1}(\mathbf{c} - y_i\mathbf{x_i})$, then use **a)**.

**c)** From **b)** you can finally conclude the desired result:

$$y_i - \mathbf{x_i^T}\hat{\beta}^{(-\mathbf{i})} = \frac{1}{1 - S_{ii}}(y_i - \mathbf{x_i^T}\hat{\beta}).$$

**2.** Consider the `diabetes`-dataset from the lecture notes (Section 3.2) and the model

$$Y_i = m(X_i) + \epsilon_i,$$

where the response $Y$ is a log-concentration of a serum (in connection with diabetes) and the predictor variable $X$ is the age in months of children.

We want to know if a complicated nonparametric regression gives us valuable information and which one is the best. The generalization error of the following fits should be compared:

1. the kernel regression fit from `ksmooth`,
2. the local polynomial fit from `loess`,
3. a smoothing spline fit from `smooth.spline` with a fixed degree of freedom,
4. a smoothing spline fit from `smooth.spline` with the smoothing parameter selected automatically by cross-validation,
5. a constant "fit" by the overall mean of $Y_i$, simply ignoring the $X_i$-values.

**a)** Read in the dataset as follows:

```
> diabetes <-
    read.table("http://stat.ethz.ch/Teaching/Datasets/diabetes2.dat",
               header = TRUE)
> reg <- diabetes[, c("Age", "C.Peptide")]
> names(reg) <-     c("x",   "y")
```

Sort the observations along $x$, for easier dealing with the hat matrix:

```
> reg <- reg[sort.list(reg$x), ]
```

Plot the data; choose a reasonable bandwidth $h$ for a Nadaraya-Watson kernel estimator by eye. Perform a non-parametric regression on the dataset using the kernel estimator of `ksmooth` with the bandwidth $h$ of your choice.

Calculate the leave-one-out CV-score of your fit; the CV-score is an estimator for the generalization error. Because you will have to do that for the other regression methods too, it is recommended to write a utility function for calculating the CV-score. An R-skeleton for such a function is available on the course homepage, but you can of course also implement everything on your own from scratch.

Finally, calculate the degrees of freedom `df.nw` that corresponds to your fit. For that aim, calculate the hat matrix and its trace, cf. Formula (3.6) of the lecture notes and the above-mentioned code skeleton. We will use the same degrees of freedom as smoothing parameter for the other regression methods.

**b)** Perform a non-parametric regression on the dataset `diabetes` using the local polynomial fit from `loess`. Use the degrees of freedom `df.nw` from task a) as a smoothing parameter:

```
loess(..., enp.target = df.nw, surface = "direct")
```

The last argument, `surface = "direct"`, ensures that you can also predict values outside the range of $x$-values you used for estimation.

Calculate the CV-value for this estimator, too.

**c)** Perform a non-parametric regression on the dataset `diabetes` using a smoothing splines fit from `smooth.spline`; use the degrees of freedom `df.nw` from task a).

`smooth.spline` can calculate the CV-value internally; start by looking at that value:

```
> est.ss <- smooth.spline(..., cv = TRUE, df = df.nw)
> est.ss$cv.crit
```

Compare this internally calculated value with a value calculated “on your own”, i.e. similarly to the CV-values in tasks a) and b). When doing your own calculations, do not provide the smoothing parameter via the argument `df`, but use the parameter `spar` from the automatic calculation above:

```
smooth.spline(..., spar = est.ss$spar)
```

You may also take advantage of Formula (4.5) in the lecture notes to calculate the CV-value; cf. Exercise 1 of this series.

**d)** Perform a non-parametric regression on the dataset `diabetes` using a smoothing splines fit from `smooth.spline`, this time using automatically selected degrees of freedom. Report the CV-value that is calculated internally:

```
> smooth.spline(..., ..., cv = TRUE)$cv.crit
```

**e)** Perform a constant fit of the data, i.e. neglect the $x$-values and calculate the mean of the $y$-values. Calculate the corresponding CV-value also for that estimator.

**f)** The CV-scores calculated in tasks a) to e) are estimators of the generalization error. According to the values you calculated, which of the five estimators has the lowest generalization error?

The comparison of the CV-value of method no. 4 (smoothing splines with automatically selected degrees of freedom) with the others is not fair. Can you explain the problem?

**Preliminary discussion:** Friday, March 30.

**Deadline:** Friday, April 20.