Machine Learning II
Week 2 Lecture – 17th October 2022
Non-linear regression methods 2

Contents

This week's short lecture follows on directly from last week.

- Spline smoothing

Last Week

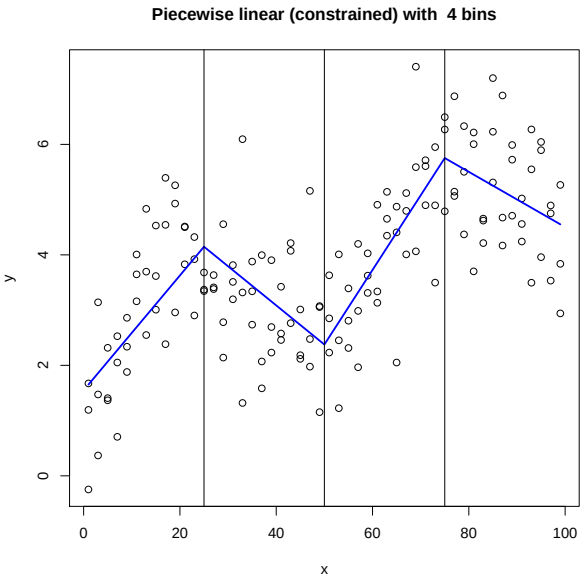
We looked at fitting:

- piecewise constant functions
- piecewise linear functions
- constrained piecewise linear functions (continuous) and
- constrained piecewise cubic functions

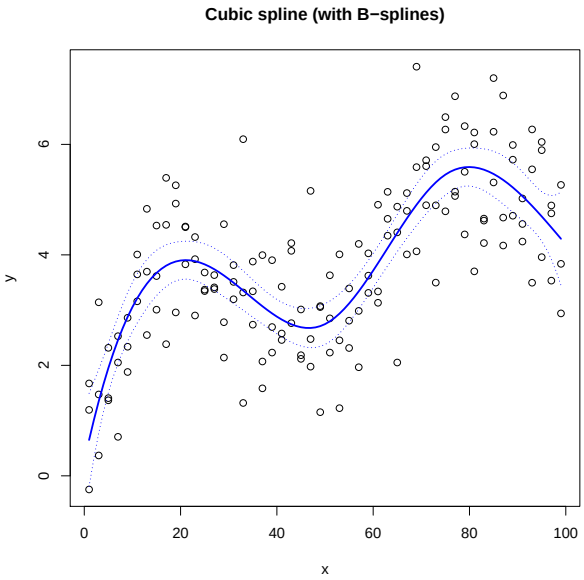
All of these methods involved adding together basis functions, which are usually zero outside of a specific interval.

The coefficients for each basis function is estimated by minimising the MSE ore RSS.

A constrained piecewise cubic function is called a cubic spline.



first derivative is 0



1st derivative is not zero, but 2nd derivative is not zero

All of the non linear regression methods considered last week require the analyst to specify how many knot points there and their locations.

A standard choice for the knot point locations is equally spaced knots, but the number still has to be chosen. (Hyperparameter)

We now move to a method which avoids this choice. There is still a hyperparameter to be chosen, but this is usually easier to chose than the knot points.

The method this week is called **spline smoothing**, but is different from the method last week, which we will call **spline fitting**!

Spline Smoothing

We return to the penalised least squares concept you learnt in ridge regression last semester:

minimise: squared error of fitted model + λ penalty term

If we have a completely free choice of a non-linear function, we could choose a function f which fits the data well. I.e. minimise squared residuals

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

A problem with this is that $f(x)$ is that this will almost certainly over fit the data. It won't *smooth* the data at all.

Another description of the over fitting is that f is too "*wiggly*".

The wiggliness of a function is measured using the second derivative of f. f'' measures of how quickly the *slope* of f changes.

An overall measure of the wiggliness of our data is the integral of $f''(x)$ with x over the range of the predictor values.

$$\int_{x_1}^{x_n} f''(x)^2 dx$$

To balance out the data fitting with smooth ness of the function we can use the penalised sum of squares:

$$J(f, \lambda) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_{x_1}^{x_n} f''(x)^2 dx. \quad (1)$$

The aim is to minimise $J(f, \lambda)$ using an appropriate choice of λ.

Note that f can be any possible function as long as it is twice differentiable.

If $\lambda = \infty$ then any wiggliness will send $J(f, \lambda)$ to infinity, so any the function that minimises J has $f''(x) = 0$.

The resulting function has constant f', so f is linear.
When $\lambda = \infty$ we revert to linear regression.

If $\lambda = 0$ then there is no penalty for wiggliness.

If there are no duplicated x-values, the f that minimises $J(f, 0)$ will interpolate the data.

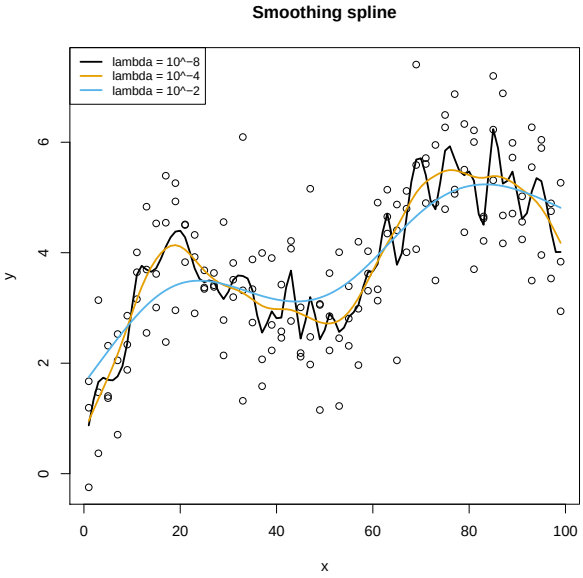
If there are duplicated x-values:

- Obtain the unique x-values. We could call them $u_1, u_2, \ldots u_m$.
- Calculate the *mean* of the y-values at each unique x-value. We could call them $v_1, v_2, \ldots v_m$.
- The f that minimises $J(f, 0)$ will interpolate the unique values: $(u_1, v_1), (u_2, v_2), \ldots (u_m, v_m)$.

For ease, we will call both cases the interpolating function.

There is a theorem which gives a very surprising result:
for a specific λ the function $f(x)$ which minimises Equation 1, the penalised least squares formula, will be a natural cubic spline.

The knot points for this cubic spline are the unique x-values $v_1, v_2, \ldots v_m$. The predictor function is a smoothed version of the interpolating function.



Effective degrees of freedom

Let N be the number of unique values of $x_1, \ldots, x_n$.

The interpolating function ($\lambda = 0$) depends on m pairs of x and y values (u and v).

A linear regression line ($\lambda = \infty$) can be defined using two pairs of x and y values.

The effective degrees of freedom is a value between 2 and m which corresponds to the level of smoothing, the value of λ.

In R you can specify either λ or the effective degrees of freedom. The latter is usually on a more intuitive scale.

Algorithm and cross validation

You do not need to learn the algorithm which fits the spline smoothing, but details are given in Hastie, Tibshirani & Friedman.

The algorithm involves fixing λ and solving a set of linear equations which is an algorithm order n (i.e. fast).

For ridge regression you learnt that cross validation is a method of choosing a good value of λ. A problem with calculating the leave one out cross validation (LOOCV) score in general is that is computationally expensive.

With the spline smoothing algorithm, the LOOCV score can be calculated directly from the numerical solution.

Effectively we get the LOOCV score for free!

Today's Workshop:

- The Worksheet has been updated to include spline smoothing
- Work through the relevant Lab in James et al.
- Apply what you have learnt to a data set which is a difficult problem in regression/smoothing methods.

Next week: we will look at a similar method to spline smoothing called localised regression (loess) and Generalised Additive Models (GAMS).