Dr. Boqiang Huang

Mathematical Foundations of Data Analysis II — SS 2019

Problem Set 05

Published on 22.05.2019 To be collected on 28.05.2019

Problem 1: (8 points)
The cubic spline interpolation problem can be explained as follows:
(a) given K knot pairs $(k_i, s_i), i = 1, \ldots, K$, suppose that there is a piecewise function of the form

$$f(x) := \begin{cases} f_1(x) & \text{if } k_1 \le x \le k_2; \\ f_2(x) & \text{if } k_2 \le x \le k_3; \\ \cdots & \\ f_{K-1}(x) & \text{if } k_{K-1} \le x \le k_K, \end{cases} \tag{1}$$

where $f_i(x)$ is a third degree polynomial defined by

$$f_i(x) := a_i(x-k_i)^3 + b_i(x-k_i)^2 + c_i(x-k_i) + d_i,\ i = 1, \ldots, K-1, \tag{2}$$

and following properties are satisfied:

(I) the piecewise function $f(x)$ can interpolate all data points for $x \in [k_1, k_K]$;
(II) $f(x)$ is continuous on the interval $[k_1, k_K]$;
(III) $f'(x)$ is continuous on the interval $[k_1, k_K]$;
(IV) $f''(x)$ is continuous on the interval $[k_1, k_K]$.

Property (I) implies $K-1$ equations

$$f(k_i) = f_i(k_i) = s_i,\ i = 1, \ldots, K-1; \tag{3}$$

Property (II) implies $K-2$ equations

$$f_i(k_i) = f_{i-1}(k_i),\ i = 2, \ldots, K-1; \tag{4}$$

Property (III) implies $K-2$ equations

$$f_i'(k_i) = f_{i-1}'(k_i),\ i = 2, \ldots, K-1; \tag{5}$$

Property (IV) implies last $K-2$ equations

$$f_i''(k_i) = f_{i-1}''(k_i),\ i = 2, \ldots, K-1. \tag{6}$$

Let $\boldsymbol{x} := [b_1, \ldots, b_{K-1}]^T \in \mathbb{R}^{K-1}$ represent the unknown vector of all coefficient b_i in the equation (2), please derive the elements of matrix A and vector $\boldsymbol{b}$ satisfying

$$A\boldsymbol{x} = \boldsymbol{b} \tag{7}$$

based on equation (2) and above-mentioned properties.
(b) Unfortunately the dimention of matrix A in equation (7) is $(K-2) \times (K-1)$, which implies that there are infinite solutions of $\boldsymbol{x}$. In order to obtain an unique solution, we need more bounday conditions, e.g.

(i) Natural Spine: set the second derivative be euqal to zero at the boundary knots k_1 and k_K;
(ii) Parabolic Runout Spline: require the second derivatives at the knots k_1 and k_K should be equal to the ones at the nearest knots k_2 and k_{K-1}, respectively;
(iii) Cubic Runout Spline or "Not-A-Knot": set the third derivative of $f_1(x)$ and $f_2(x)$ at the knot k_2 should be equal, and the third derivative of f_{K-1} and f_{K-2} at the knot k_{K-1} should be equal.

Please update equation (7) based on the "Not-A-Knot" condition.

Problem 2: (8 points)
Consider a general nonlinear model:

$$y_i \sim N(f(x_i), \sigma_\epsilon^2),\ i = 1, \dots, n, \tag{8}$$

where $f(x)$ is a smooth, but otherwise unspecified, function of some univariate x, that needs to be estimated from observation pairs (x_i, y_i), $i = 1, \dots, n$, and σ_ϵ^2 denotes the variance of an additive zero-mean normal noise in the considered system. To capture/recognize/learn a complex non-linear structure of $f(x)$, we define K knots $k_1, \dots, k_K$ and extend a parametric polynomial model with the truncated polynomial basis functions, i.e.

$$f(x) := \beta_0 + \beta_1 x + \cdots + \beta_p x^p + \sum_{j=1}^{K} w_j\, (x - k_j)_+^p, \tag{9}$$

where $(x - k_j)_+ := \max\{0, (x - k_j)\}$.
(a) let matrix $X_{n\times(p+1)} := [\mathbf{1}, \boldsymbol{x}, \dots, \boldsymbol{x}^p], \mathbf{1} = [1, \dots, 1]^T, \boldsymbol{x} = [x_1, \dots, x_n]^T, \boldsymbol{x}^p = [x_1^p, \dots, x_n^p]^T$, matrix $Z_{n\times K} := [(\boldsymbol{x} - k_1)_+^p, \dots, (\boldsymbol{x} - k_K)_+^p], \beta = [\beta_0, \dots, \beta_p]^T$, and $\boldsymbol{w} = [w_1, \dots, w_K]^T$, show that (8) can be represented as

$$\boldsymbol{y} \sim N(X\beta + Z\boldsymbol{w}, \sigma_\epsilon^2 I_n), \tag{10}$$

where $\boldsymbol{y} = [y_1, \dots, y_n]^T$. And the parametric model (10) can be easily estimated with ordinary least squares

$$\hat{\boldsymbol{y}} = C\left(C^T C\right)C^T \boldsymbol{y}, \tag{11}$$

with $C = [X, Z]$.
(b) recall the shrinkage method called ridge regression in Chapter 5, try to represent the corresponding optimization problem together with its analytic solution (Tip: define a new vector $\theta := [\beta^T, \boldsymbol{w}^T]^T$).
(c) the nonlinear model in (9) could be modified as

$$f(x) := \beta_0 + \beta_1 x + \sum_{j=1}^{K} w_j\, \varphi(|x - k_j|), \tag{12}$$

where $\varphi(r) := \varphi(|\cdot|)$ represents the radial basis function (RBF) defined in different versions:

$$\begin{aligned}
\text{Linear: } \varphi(r) &= r, &(13a)\\
\text{Cubic: } \varphi(r) &= r^3, &(13b)\\
\text{Thin plate: } \varphi(r) &= r^2 \ln(r+1), &(13c)\\
\text{Multiquadrics: } \varphi(r) &= \sqrt{1 + \frac{r^2}{\sigma^2}}, &(13d)\\
\text{Gaussian: } \varphi(r) &= \exp\left(-\frac{r^2}{2\sigma^2}\right). &(13e)
\end{aligned}$$

Taking Gaussian RBF as an example, please derive the corresponding optimization problem similar to the ridge regression, and present its analytic solution.

Problem 3: (4 points)
Given a training data set consists of five records, in which the target attribute is "Acceptable". The decision is made based on other three attributes, "furniture", "number of rooms", and "new kitchen".

House	Furniture	Rooms	New Kitchen	Acceptable
1	No	3	Yes	Yes
2	Yes	3	No	No
3	No	4	No	Yes
4	No	3	No	No
5	Yes	4	No	Yes

(a) compute the entropy of the target attribute $H(X = \text{"Acceptable"})$;
(b) define the Information Gain as

$$G(X, Y) := H(X) - H(X|Y), \tag{14}$$

e.g.

$$G(X = \text{"Acceptable"}, Y = \text{"Furniture"}) := H(X = \text{"Acceptable"}) - H(X = \text{"Acceptable"}|Y = \text{"Furniture"}). \tag{15}$$

Then, compute the information gains for other conditioning attributes $G(X =$ "Acceptable", $Y =$ "Furniture"), $G(X =$ "Acceptable", $Y =$ "Rooms"), $G(X =$ "Acceptable", $Y =$ "New Kitchen").
(c) set the attribute corresponding to the largest gain $G(X =$ "Acceptable", $Y =$ "·") as the root node of the decision tree. For each branch of the root node, determine the best sub-node corresponding to the largest gain.
For example, if the root node is "Furniture" one has to calculate and compare the gains
$G(X =$ "Acceptable", "Furniture" = "Yes"$|Y =$ "Rooms") and
$G(X =$ "Acceptable", "Furniture" = "Yes"$|Y =$ "New Kitchen")
for the branch "Furniture" = "Yes".
Implement such "best node selection" iteratively and finally construct a decision tree based on the given training data set.
This decision tree learning method is the classic "Iterative Dichotomiser 3 (ID3)" algorithm.