

Assignment 5

以降の実験で使用するソースコードは、別に添付する `cubic_spline.py` に記載する。

1.

A)

今回使うスプライン基底を図 1 に示す。各基底は、ソースコード内”__init__”の 39～42 行目で獲得している

39 行目では、指定した k の個数に応じてノットの座標を獲得している。41 行目では、 $h_0(x) \sim h_3(x)$ のスプライン基底を”basis”に格納している。42 行目では、 $h_4(x) \sim h_{k+3}(x)$ のスプライン基底を”basis”に格納している。

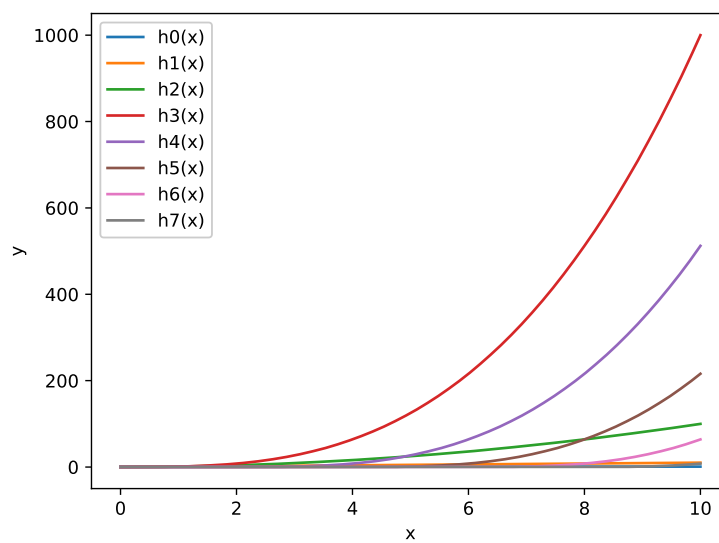


図 1 使用するスプライン基底

B)

獲得されたスプライン関数を図 2 に示す。スプライン関数の係数はソースコード内”calc_beta”関数内の 47, 48 行目で計算した。なお、引数”data”は入力を、引数”ans”は出力を代入している。47 行目では、入力データにスプライン基底を適用し、 \mathbf{X} を計算している。48 行目では、上で計算した \mathbf{X} を利用して、 $\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ を計算している。

また、スプライン関数の出力は”predict”関数内の 52, 53 行目で計算した。52 行目では \mathbf{X} を計算し、53 行目で \mathbf{X} に係数 $\boldsymbol{\beta}$ を掛けて出力を計算している。

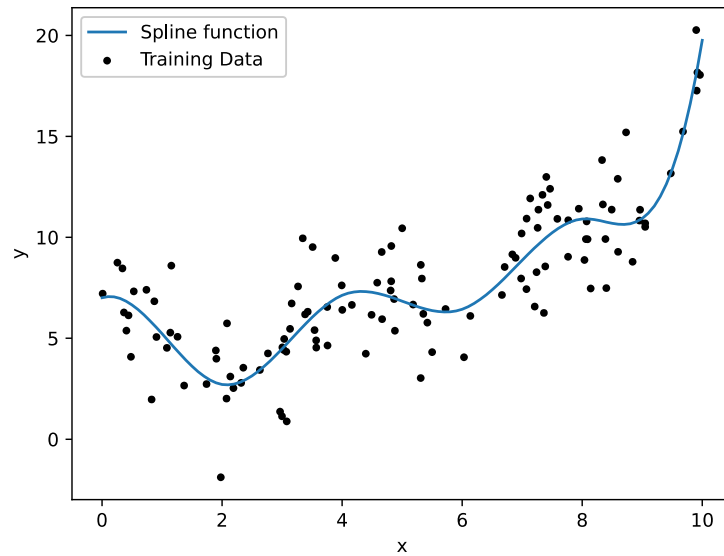


図 3 獲得されたスプライン関数

C)

magic fomula によって計算された CV_{Loo} は 3.727 となった。 CV_{Loo} の計算はソースコード内”calc_magic_predict”の 59～62 行目で行った。 59 行目では、入力データにスプライン基底を適用し、 \mathbf{X} を計算している。 60 行目では、上で計算した \mathbf{X} を利用して $\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ を計算している。 61 行目は、”predict”関数でスプライン関数の出力を計算している。 62 行目では、計算した \mathbf{H} およびスプライン関数の出力”out”を利用して magic fomula の値を計算している。

D)

magic fomula を使わずに計算された CV_{Loo} は 3.727 となった。 CV_{Loo} の計算はソースコード内 66 行目”calc_loss”関数および 86 行目で行った。 なお関数 calc_loss(spl, data, ans, idx)は、”idx”番目の入力を使用せずに基底”basis”と係数”beta”を定義したスプライン関数”spl”利用して、”idx”番目の入力”data”に対するスプライン関数の 2 乗誤差を出力する関数である。 86 行目では、各 $i(i = 1, 2, \dots, 120)$ を使用せずに作成したスプライン関数に対して、 i 番目の入力に対する出力の誤差を計算し、その平均値を計算している。

2.

各 $K(K = 1, 2, \dots, 15)$ について、計算された CV_{Loo} の値を表 1 に示す。 なお CV_{Loo} の値は、magic fomula を使用して計算した。 各 K に対する CV_{Loo} は 73 行目の”calc_magic_loo”関数および 90 行目で、最適なノット数は 91 行目で求めている。 ここでは、ノットの数”k”が与えられたときの、magic fomula による CV_{Loo} の計算を関数 calc_magic_loo(data, ans, k) として定義している。 89 行目では、ノットの個数を決定し、90 行目で各 K について、 CV_{Loo}

を計算している．そして，91 行目で最適な K を獲得している．

表 1 各 K に対する CV_{Loo} の値

ノット数	CV_{Loo}	ノット数	CV_{Loo}
1	4.375	9	4.392
2	4.366	10	4.649
3	4.411	11	4.688
4	3.727	12	5.269
5	3.959	13	5.767
6	3.789	14	6.121
7	3.844	15	5.927
8	4.152	-	-

3.

多項式回帰では，自由度を増やしたモデルでも，自由度を増やす前のモデルを表現することができた．すなわち，適当な係数を使用することにより，自由度が高いモデルで，自由度の低いモデルが表現できた．そのため，自由度を増やすと必ずエラーは減少する．

一方，スプライン回帰では，ノットの数が 1 以上のとき，ノットの数が増加すると，増加後のモデルでは増加前のモデルが表現できない．具体的には，ノットの個数が 1 つ増えると，増加前とノットの位置が一致しなくなるため，どのような係数であっても増加前のモデルと等しいモデルにはならない．ゆえに，スプライン回帰では自由度を増やしてもエラーが減少するとは限らない．