Basis Expansions and Regularization

by

AINDRILA GARAI

MSC STATISTICS, IIT KANPUR

aindrilag22@iitk.ac.in

Introduction:

The true function $f(X)$ is not always linear in X. So, we replace the inputs X with additional variables, which are transformations of X and then use linear models based on these derived input features.

Denote by $h_m(X): \mathbb{R}^p \mapsto \mathbb{R}$ the mth transformation of X, $m = 1, \ldots, M$. Then,

$$f(X) = \sum_{m=1}^{M} \beta_m \, h_m(X)$$

a linear basis expansion in X.

$h_m(x)$ can be (i) $X_m, m = 1, \cdots, p$ (original linear model) (ii) $h_m(X) = X_j^2$ or $h_m(X) = X_j X_k$ (polynomial terms to achieve higher-order Taylor expansions) (iii) $h_m(X) = \log(X_j), \sqrt{X_j}, \ldots$ (iv) $h_m(X) = I(L_m \leq X_k < U_m)$ (a piecewise constant contribution for X_k).

Dictionary:

Modeling signals and images produce a dictionary D consisting of a very large number $|D|$ of basis functions. Along with this a method for controlling the complexity of our model is required, using basis functions from the dictionary. 3 approaches are -

(i) **Restriction methods** where we decide before-hand to limit the class of functions $f(X) = \sum_{j=1}^{p} \sum_{m=1}^{M_j} \beta_{jm} h_{jm}(X_j)$

(ii) **Selection methods** which adaptively scan the dictionary and include only those basis functions h_m that contribute significantly to the fit of the model such as Boosting.

(iii) **Regularization methods** where we use the entire dictionary but restrict the coefficients such as Ridge regression.

Piecewise Polynomials and Splines:

Assume that X is one-dimensional. A piecewise polynomial function $f(X)$ is obtained by dividing the domain of X into contiguous intervals, and representing f by a separate

polynomial in each interval. We would typically prefer which is also piecewise linear but restricted to be continuous at the two knots.

- When the function is continuous and has continuous first and second derivatives at the knots is known as a cubic spline. More generally, an order-M spline with knots ϵ_j , $j = 1, \ldots, K$ is a piecewise-polynomial of order M and has continuous derivatives up to order $M - 2$. The general form for the truncated-power basis set would be

$$\begin{aligned} h_j(X) &= X^{j-1}, j = 1, \ldots, M \\ h_{M+\ell}(X) &= (X - \xi_\ell)_+^{M-1}, \ell = 1, \ldots, K \end{aligned}$$

These fixed-knot splines are also known as regression splines.

Natural Cubic Splines:

A natural cubic spline adds additional constraints, namely that the function is linear beyond the boundary knots. A natural cubic spline with K knots is represented by K basis functions.

$$N_1(X) = 1, \quad N_2(X) = X, \quad N_{k+2}(X) = d_k(X) - d_{K-1}(X)$$

where

$$d_k(X) = \frac{(X - \xi_k)_+^3 - (X - \xi_K)_+^3}{\xi_K - \xi_k}$$

Each of these basis functions can be seen to have zero second and third derivative for $X \geq \xi_K$.

Filtering and Feature Extraction:

The preprocessing need not be linear but can be a general (nonlinear) function of the form $x^* = g(x)$. The derived features x^* can be used as inputs into any (linear or nonlinear) learning procedure. For signal or image recognition a approach is to first transform the raw features via a wavelet transform and then use the features as inputs into a neural network.

Smoothing Splines:

A spline basis method that avoids the knot selection problem completely by using a maximal set of knots. The complexity of the fit is controlled by regularization. The fitted smoothing spline is given by

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

where the $N_j(x)$ are an N-dimensional set of basis functions for representing this family natural splines. - The degrees of freedom of a smoothing spline is $\mathrm{df}_\lambda = \mathrm{trace}(\mathbf{S}_\lambda)$ where $\mathbf{S}_\lambda$ is smoother matrix. - Bias will decrease and variance will increse if we increase the degrees of freedom.

Best Subset Selection:

```
library (ISLR2)
sum (is.na(Hitters$Salary)) # to identify the missing observations

## [1] 59

library (leaps)
regfit.full <- regsubsets (Salary ~ ., Hitters)
summary (regfit.full)

## Subset selection object
## Call: regsubsets.formula(Salary ~ ., Hitters)
## 19 Variables  (and intercept)
##            Forced in Forced out
## AtBat          FALSE      FALSE
## Hits           FALSE      FALSE
## HmRun          FALSE      FALSE
## Runs           FALSE      FALSE
## RBI            FALSE      FALSE
## Walks          FALSE      FALSE
## Years          FALSE      FALSE
## CAtBat         FALSE      FALSE
## CHits          FALSE      FALSE
## CHmRun         FALSE      FALSE
## CRuns          FALSE      FALSE
## CRBI           FALSE      FALSE
## CWalks         FALSE      FALSE
## LeagueN        FALSE      FALSE
## DivisionW      FALSE      FALSE
## PutOuts        FALSE      FALSE
## Assists        FALSE      FALSE
## Errors         FALSE      FALSE
## NewLeagueN     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns C
RBI
## 1  ( 1 ) " "   " "  " "   " "  " " " " " "   " "   " "    " "   " "    " "   "
*"
## 2  ( 1 ) " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "   "
*"
## 3  ( 1 ) " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "   "
*"
## 4  ( 1 ) " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "   "
*"
## 5  ( 1 ) "*"   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "   "
*"
## 6  ( 1 ) "*"   "*"  " "   " "  " " " "*"   " "   " "    " "   " "    " "   "
*"
## 7  ( 1 ) " "   "*"  " "   " "  " " " "*"   " "   "*"    "*"   "*"    " "   "
```

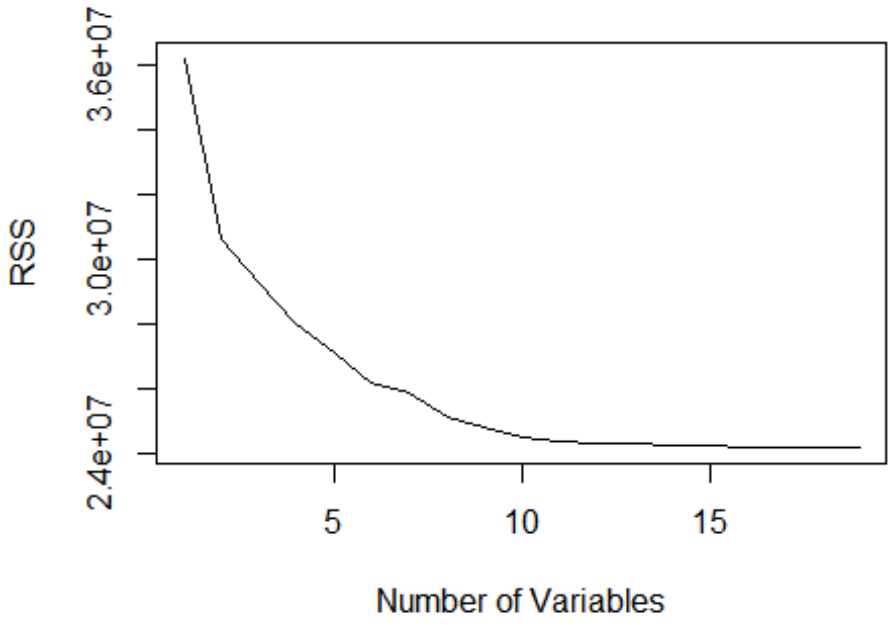
```
"
## 8  ( 1 ) "*"   "*"  " "   " "  " " "*"   " "   " "    " "   "*"    "*"   "
"
##          CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1  ( 1 ) " "    " "     " "       " "     " "     " "    " "
## 2  ( 1 ) " "    " "     " "       " "     " "     " "    " "
## 3  ( 1 ) " "    " "     " "       "*"     " "     " "    " "
## 4  ( 1 ) " "    " "     "*"       "*"     " "     " "    " "
## 5  ( 1 ) " "    " "     "*"       "*"     " "     " "    " "
## 6  ( 1 ) " "    " "     "*"       "*"     " "     " "    " "
## 7  ( 1 ) " "    " "     "*"       "*"     " "     " "    " "
## 8  ( 1 ) "*"    " "     "*"       "*"     " "     " "    " "
```

```
regfit.full <- regsubsets (Salary ~ ., data = Hitters , nvmax = 19)
reg.summary <- summary (regfit.full)
reg.summary
```

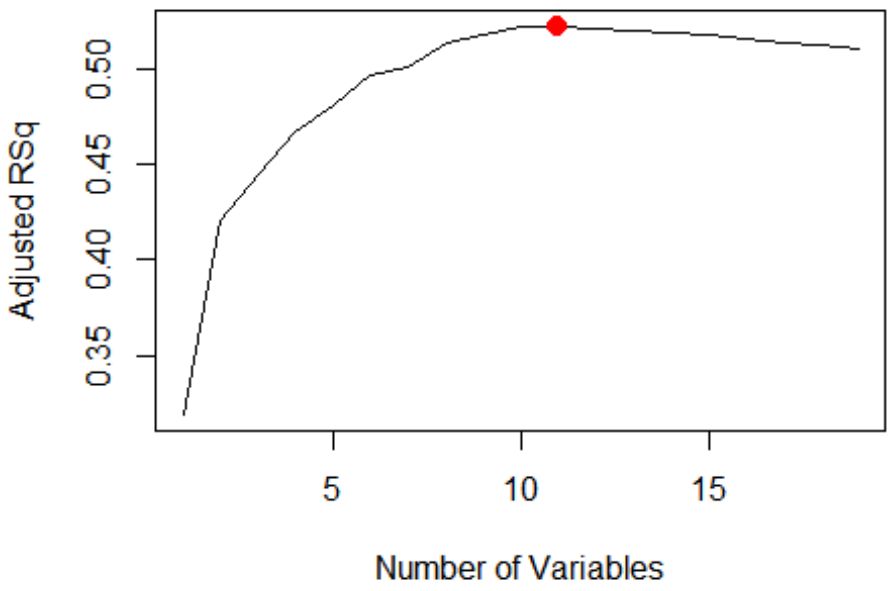
```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19)
## 19 Variables  (and intercept)
##            Forced in Forced out
## AtBat          FALSE      FALSE
## Hits           FALSE      FALSE
## HmRun          FALSE      FALSE
## Runs           FALSE      FALSE
## RBI            FALSE      FALSE
## Walks          FALSE      FALSE
## Years          FALSE      FALSE
## CAtBat         FALSE      FALSE
## CHits          FALSE      FALSE
## CHmRun         FALSE      FALSE
## CRuns          FALSE      FALSE
## CRBI           FALSE      FALSE
## CWalks         FALSE      FALSE
## LeagueN        FALSE      FALSE
## DivisionW      FALSE      FALSE
## PutOuts        FALSE      FALSE
## Assists        FALSE      FALSE
## Errors         FALSE      FALSE
## NewLeagueN     FALSE      FALSE
## 1 subsets of each size up to 19
## Selection Algorithm: exhaustive
##           AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns
CRBI
## 1  ( 1 )  " "   " "  " "   " "  " " " " " "   " "   " "    " "   " "    " "
"*"
## 2  ( 1 )  " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
"*"
## 3  ( 1 )  " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
"*"
```

```
## 4  ( 1 )  " "   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
"*"
## 5  ( 1 )  "*"   "*"  " "   " "  " " " " " "   " "   " "    " "   " "    " "
"*"
## 6  ( 1 )  "*"   "*"  " "   " "  " " " "*"   " "   " "    " "   " "    " "
"*"
## 7  ( 1 )  " "   "*"  " "   " "  " " " "*"   " "   "*"    "*"   "*"    " "
" "
## 8  ( 1 )  "*"   "*"  " "   " "  " " " "*"   " "   " "    " "   "*"    "*"
" "
## 9  ( 1 )  "*"   "*"  " "   " "  " " " "*"   " "   "*"    " "   " "    "*"
"*"
## 10  ( 1 ) "*"   "*"  " "   " "  " " " "*"   " "   "*"    " "   " "    "*"
"*"
## 11  ( 1 ) "*"   "*"  " "   " "  " " " "*"   " "   "*"    " "   " "    "*"
"*"
## 12  ( 1 ) "*"   "*"  " "   "*"  " " " "*"   " "   "*"    " "   " "    "*"
"*"
## 13  ( 1 ) "*"   "*"  " "   "*"  " " " "*"   " "   "*"    " "   " "    "*"
"*"
## 14  ( 1 ) "*"   "*"  "*"   "*"  " " " "*"   " "   "*"    " "   " "    "*"
"*"
## 15  ( 1 ) "*"   "*"  "*"   "*"  " " " "*"   " "   "*"    "*"   " "    "*"
"*"
## 16  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   " "   "*"    "*"   " "    "*"
"*"
## 17  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   " "   "*"    "*"   " "    "*"
"*"
## 18  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   "*"   "*"    "*"   " "    "*"
"*"
## 19  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   "*"   "*"    "*"   "*"    "*"
"*"
##           CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1  ( 1 )  " "    " "     " "       " "     " "     " "    " "
## 2  ( 1 )  " "    " "     " "       " "     " "     " "    " "
## 3  ( 1 )  " "    " "     " "       "*"     " "     " "    " "
## 4  ( 1 )  " "    " "     "*"       "*"     " "     " "    " "
## 5  ( 1 )  " "    " "     "*"       "*"     " "     " "    " "
## 6  ( 1 )  " "    " "     "*"       "*"     " "     " "    " "
## 7  ( 1 )  " "    " "     "*"       "*"     " "     " "    " "
## 8  ( 1 )  "*"    " "     "*"       "*"     " "     " "    " "
## 9  ( 1 )  "*"    " "     "*"       "*"     " "     " "    " "
## 10  ( 1 ) "*"    " "     "*"       "*"     "*"     " "    " "
## 11  ( 1 ) "*"    "*"     "*"       "*"     "*"     " "    " "
## 12  ( 1 ) "*"    "*"     "*"       "*"     "*"     " "    " "
## 13  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    " "
## 14  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    " "
## 15  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    " "
## 16  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    " "
## 17  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"    "*"
```

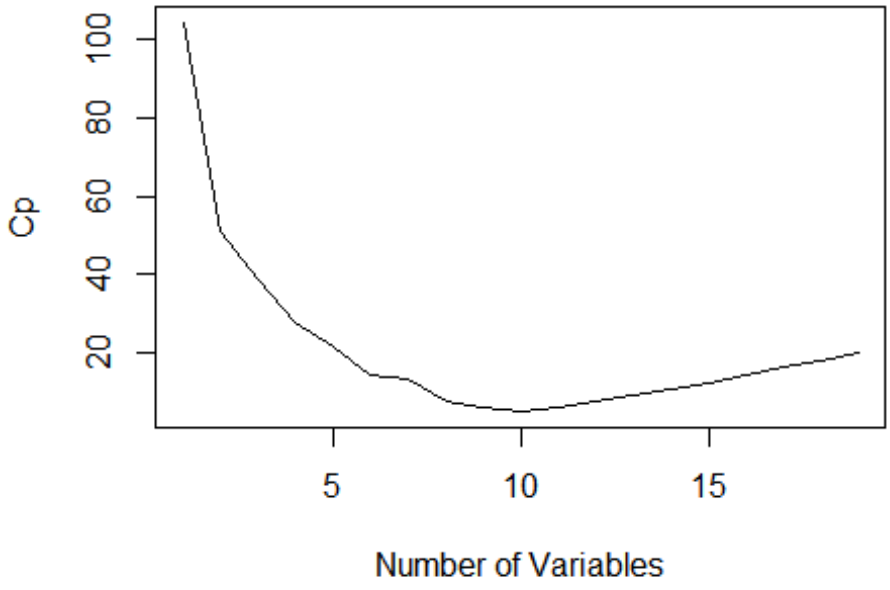
```
## 18  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"     "*"
## 19  ( 1 ) "*"    "*"     "*"       "*"     "*"     "*"     "*"

plot (reg.summary$rss , xlab = " Number of Variables ", ylab = " RSS ", type
= "l")
```



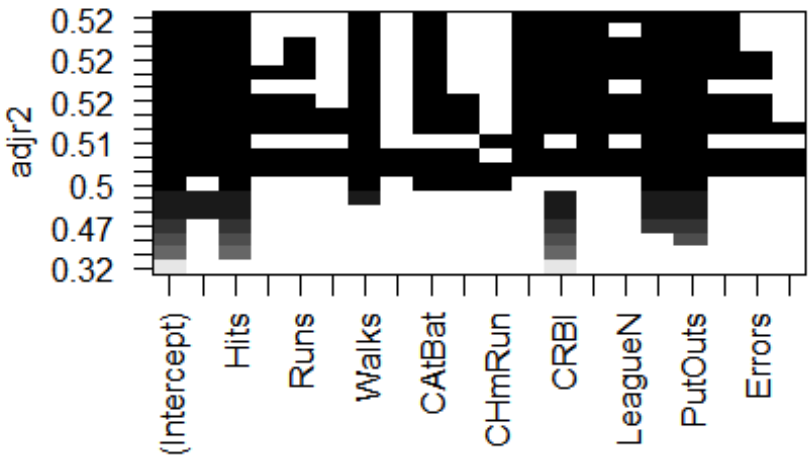

```
plot (reg.summary$adjr2 , xlab = " Number of Variables ", ylab = " Adjusted R
Sq ", type = "l")
points (11, reg.summary$adjr2[11], col = " red ", cex = 2, pch = 20)
```




```
plot (reg.summary$cp, xlab = " Number of Variables ", ylab = "Cp", type = "l"
)
```



```
plot (regfit.full , scale = "adjr2") # You can use r2, adjr2, Cp, bic
```




```
coef (regfit.full , 6)
```

```
##  (Intercept)        AtBat         Hits        Walks         CRBI    Divisi
onW
##   91.5117981   -1.8685892    7.6043976    3.6976468    0.6430169 -122.9515
338
##      PutOuts
##    0.2643076
```