SB1b/SM2 Computational Statistics HT18

Lecturer: Geoff Nicholls

Lecture 7: B-Splines; Multivariate Regression

Notes and Problem sheets are available at

http://www.stats.ox.ac.uk/~nicholls/sm2-sb1b/(L1-7)

http://www.stats.ox.ac.uk/~caron/teaching/sb1b/(L8-13)

and via the MSc weblearn pages.
**B-splines** To construct solution, need basis for natural polynomial splines. Convenient is the so-called **B-spline** basis.

Splines: TL constant; TR linear; LL quadratic; LR cubic.
For knots $a = \xi_0 < \xi_1 < \xi_2 < \ldots, \xi_k \leq \xi_{k+1} = b$ in $(a,b)$, define new knots $\tau$ as (recall $M = 4$ for CS)

- $\tau_0 \leq \tau_1 \leq \ldots \leq \tau_M = \xi_0 = a$
- $\tau_{j+M} = \xi_j$
- $b = \xi_{k+1} = \tau_{k+M+1} \leq \tau_{k+M+2} \leq \ldots \leq \tau_{k+2M}$

Define recursively. For $m = 1$ and $i = 1, \ldots, k + 2M - 1$.

$$B_{i,1}(x) = \begin{cases} 1 & \tau_i \leq x < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

For $m \leq M$ and $i = 1, \ldots, k + 2M - m$.

$$B_{i,m}(x) = \frac{x - \tau_i}{\tau_{i+m-1} - \tau_i}B_{i,m-1}(x) + \frac{\tau_{i+m} - x}{\tau_{i+m} - \tau_{i+1}}B_{i+1,m-1}(x).$$

Support of $B_{i,m}(x)$ is $[\tau_i, \tau_{i+m}]$.

An implementation for $R$ is in package splines. [See R-code examples in L6spline.R]
Fitting Splines  Take $B_i(x) = B_i,M(x)$, with $M = 4$, we can write $m$ in the B-spline basis as

$$m(x) = \sum_{j=1}^{n+4} B_j(x) \beta_j$$

The objective function

$$\sum_{i=1}^{n} (y_i - m(x_i))^2 + \lambda \int (m''(x))^2 \, dx$$

reads now

$$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{n+4} B_j(x_i) \beta_j)^2 + \lambda \sum_{j,j'=1}^{n+4} \beta_j \beta_{j'} \int B_j''(x) B_{j'}''(x) \, dx$$
• $Y$ is the $n \times 1$ vector $Y = (y_1, \ldots, y_n)$.
• $B$ is the $n \times (n + 4)$-matrix with entries $B_{ij} = B_j(x_i)$ and
• $\Omega$ is as $(n + 4) \times (n + 4)$-matrix with entries

$$\Omega_{jk} = \int B''_j(x)B''_k(x) \, dx.$$ 

$$\hat{\beta} = \arg\min_\beta \quad (Y - B\beta)^T(Y - B\beta) + \lambda \beta^T\Omega\beta.$$ 

And $\hat{m}(x) = \sum_{j=1}^{n+4} \hat{\beta}_j B_j(x)$. The solution to

$$(Y - B\beta)^T(Y - B\beta) + \lambda \beta^T\Omega\beta.$$ 

is

$$\hat{\beta} = (B^TB + \lambda\Omega)^{-1}B^TY.$$
The estimated function values are thus

\[ \hat{Y} = \hat{m}(x) = SY, \]

The smoothing matrix \( S \) in \( \hat{Y} = SY \) is given by

\[ S = B(B^T B + \lambda \Omega)^{-1} B^T. \]

Smoothing splines are thus another example of linear smoothers.
Solution depends on the regularization parameter $\lambda$.

- Value $\lambda$ non-intuitive
- Can find $\lambda = \lambda(df)$ for a given degrees of freedom $df$.

Here $df = \text{trace}(S)$ set to 10, 50, 150 corresponds to $\lambda$ approximately $2e - 3$, $2e - 6$ and $2e - 20$ (!)
Choose $\lambda$ by LOO-CV or GCV [see R-code in L6spline.R]
Multivariate smoothing

Consider smoothing functions \( g(\cdot) : \mathbb{R}^p \mapsto \mathbb{R} \) with \( p > 1 \) now.

Can we just extend the methods and model functions \( \mathbb{R}^p \mapsto \mathbb{R} \) nonparametrically?

Curse of Dimensionality

The usefulness of ‘local’ fitting is lost if \( p \to \infty \) and \( n \) constant. Say our data points \( (x, y), y \in \mathbb{R} \),

\[
x = (x^{(1)}, x^{(2)}, \ldots, x^{(p)}) \in [0, 1]^p
\]

are scattered uniformly in a unit cube.
A cube of side-length $0 < \ell < 1$ occupies a fraction $\ell^p$ of the volume $[0, 1]^p$. To capture say 5% of the points we need $\ell^p = 0.05$, or $\ell = 0.05^{1/p}$,

<table>
<thead>
<tr>
<th>Dimension $p$</th>
<th>side length $\ell$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>2</td>
<td>0.22</td>
</tr>
<tr>
<td>5</td>
<td>0.54</td>
</tr>
<tr>
<td>10</td>
<td>0.74</td>
</tr>
<tr>
<td>1000</td>
<td>0.997</td>
</tr>
</tbody>
</table>

So ‘local’ is essentially the whole space.
Additive Models Can ‘avoid’ the curse of dimensionality by imposing an additive univariate structure. In an additive model, $m : \mathbb{R}^p \mapsto \mathbb{R}$ has the form

$$m_{add}(x) = \mu + \sum_{j=1}^{p} m_j(x^{(j)}), \quad m \in \mathbb{R},$$

where $m_j(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ is the same sort of scalar nonparametric smoother we fit before and (for identifiability),

$$\sum_{i=1}^{n} m_j(x_i^{(j)}) = 0 \quad \text{for each } j = 1, \ldots, p.$$

In fact $\mu$ may depend on other covariates $z = (z^{(1)}, \ldots, z^{(q)})$ in a simple parametric way, in which case

$$m_{add}(z, x) = z^T \theta + \sum_{j=1}^{p} m_j(x^{(j)}), \quad m \in \mathbb{R}$$

with the variables in $z$ distinct from those in $x$ for identifiability.
There are two settings for this sort of fitting. In the context of
the course so far, we might assume $Y = m(x) + \epsilon$ with $E(\epsilon) = 0$ and choose $m(x)_{\text{add}}$ to minimise the MSSE $\sum_i E((m(x_i) - \hat{m}(x_i))^2)$, using, for example, LOO-CV. We would choose among models using the CV or GCV score.

However there is also a GLM version of this setup, with a linear predictor ($s(x)$ applies a non-parametric smoother to variable $x$)

$$\eta_i = z^T \theta + s(x_i^{(1)}) + s(x_i^{(2)}) + \ldots + s(x_p^{(1)})$$

link function $\eta_i = g(E(Y_i))$ and stochastic part $Y_i \sim F(\cdot; z_i, x_i)$. For example in a Poisson GAM with a log-link, $E(Y_i) = \exp(\eta_i)$ and $Y_i \sim \Pi(e^{\eta_i})$. In this setting the RSS is replaced by the deviance. Because we can now maximise a likelihood, we have many of the usual elements of testing, such as the AIC and Chi-Square tests, for model selection.
Backfitting Data $x^{(j)}_i$ are given for $1 \leq i \leq n$ and $1 \leq j \leq p$. Let $x^{(j)} = (x^{(j)}_1, \ldots, x^{(j)}_n)$ and $x^i = (x^{(1)}_i, \ldots, x^{(p)}_i)$ be the column and row vectors of the design matrix $X$.

Suppose we have a smoothing spline with a penalty term. We seek $m(x)$ and $\mu$ to minimise

$$L(\mu, m) = \sum_i (Y_i - \mu - \sum_{j=1}^p m_j(x^{(j)}_i))^2 + \sum_j \lambda_j \int (m''_j(x))^2 \, dx$$

with

$$m_j(x) = \sum_{k=1}^{n+4} B_{j,k}(x) \beta_{j,k}$$

and $\beta_j = (\beta_{j,1}, \ldots, \beta_{j,n+4})$ the spline parameters for the $j$th smoother.

Let $B_j$ be the $n \times (n + 4)$-matrix with entries $B_{jik} = B_{j,k}(x^{(j)}_i)$, so that $m_j(x^{(j)}) = B_j \beta_j$. 
Let $\Omega^{(j)}$ be the matrix with entries $\Omega^{(j)}_{ik} = \int B''_{j,i}(x)B''_{j,k}(x) \, dx$. We can write

$$L(\mu, m) = |Y - \mu 1_n - \sum_{j=1}^{p} B_j \beta_j|^2 + \sum_{j} \lambda_j \beta_j^T \Omega_j \beta_j.$$ 

Now suppose we know $\hat{\mu}$ and $\hat{m}_k$ for all $k \neq j$. We can split off the unknown term,

$$L(\hat{\mu}, \hat{m}) = |Y - \hat{\mu} 1_n - \sum_{i \neq j} B_i \hat{\beta}_i - B_j \beta_j|^2 + \lambda_j \beta_j^T \Omega_j \beta_j + ...$$

$$= |\tilde{Y} - B_j \beta_j|^2 + \lambda_j \beta_j^T \Omega_j \beta_j + \text{const wrt } \beta_j$$

We choose $\hat{m}_j(x^{(j)}) = B_j \hat{\beta}_j$ to minimise this, and this only reduces our loss function $L(\mu, m)$. We cycle through all $j = 1, \ldots, p$ repeating this until we get no improvement.
The backfitting algorithm

1. Set $\hat{m}_j \equiv 0$ for all $j = 1, \ldots, p$. Set $\hat{\mu} \leftarrow n^{-1} \sum_{i=1}^{n} Y_i$.
2. Cycle through the indices $j = 1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots$
   (a) Set
   $$\tilde{Y} = Y - \hat{\mu} 1_n - \sum_{i \neq j} \hat{m}_i(x^{(i)}).$$
   (b) Compute a smoother $\hat{m}_j(x^{(j)}) = B_j \hat{\beta}_j$ for $\tilde{Y}$,
   $$\hat{\beta}_j = \arg\min_{\beta_j} |\tilde{Y} - B_j \beta_j|^2 + \lambda_j \beta_j^T \Omega_j \beta_j$$
   and normalize $\hat{m}_j(x^{(j)}) \leftarrow \hat{m}_j(x^{(j)}) - n^{-1} \sum_{i=1}^{n} \hat{m}_j(x^{(j)}_i)$.
   (c) Update $\hat{\mu} \leftarrow n^{-1} \sum_{i=1}^{n} (Y_i - \sum_k \hat{m}_k(x^{(k)}_i))$.
   (d) Stop when the targeted loss ceases to improve (it gets smaller at each iteration but the gain becomes negligible).
3. Return $\hat{\mu}$ and $\hat{m}_j(x^{(j)}) = B_j\hat{\beta}_j$, the smoothers for each covariate and
\[ \hat{m}_{\text{add}} \leftarrow \hat{\mu}1_n + \sum_{j=1}^{p} \hat{m}_j(x^{(j)}), \]
the $n \times 1$ vector of fitted values $\hat{Y} = \hat{m}_{\text{add}}$. 
Example: Simple 2D function

\[ Y_i = 1 + (x_i^{(1)})^2 + (x_i^{(2)})^2 + \epsilon_i, \quad \epsilon_i \sim N(0, (0.2)^2) \]

observed on a 20 × 20 grid in \([-1, 1]^2\).

\[
\begin{align*}
n &= 20; \quad N = n^2; \quad x1 = \text{seq}(-1, 1, \text{length.out}=n); \quad x2 = x1 \\
X &= \text{expand.grid}(x1, x2) \\
np &= 1 + X[,1]^2 - X[,2]^2 \quad \# \text{the truth!} \\
s &= 0.2; \quad y = np + s * \text{rnorm}(N) \quad \# \text{the data}
\end{align*}
\]

Use backfitting with splines to fit

\[ y = \mu + s(x12[, 1]) + s(x12[, 2]) + \epsilon / \]

See L7.R.
Example: Daily ozone - levels in LA basin with 9 predictors.

<table>
<thead>
<tr>
<th>O3</th>
<th>5300</th>
<th>5800</th>
<th>20</th>
<th>60</th>
<th>0</th>
<th>3000</th>
<th>0</th>
<th>200</th>
<th>0</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>vdht</td>
<td>0</td>
<td>10</td>
<td>20</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wind</td>
<td>0</td>
<td>30</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>humidity</td>
<td>0</td>
<td>3000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>temp</td>
<td>0</td>
<td>100</td>
<td>-50</td>
<td>100</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ibht</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dgpg</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>10</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ibtp</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>30</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vsty</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>100</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>day</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The R function `gam()` (Generalized Additive Models) uses penalized regression splines, a variation of smoothing splines. The degrees of freedom for each variable are determined by generalized cross-validation.
> addmod <- gam(O3 ~ s(vdht)+s(wind)+s(humidity)+s(temp)+s(ibht)+
   s(dgpg)+s(ibtp)+s(vsty)+s(day), data=ozone)
> summary(addmod)
Family: gaussian
Link function: identity
...
Approximate significance of smooth terms:

<table>
<thead>
<tr>
<th></th>
<th>edf</th>
<th>Ref.df</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(vdht)</td>
<td>1.000</td>
<td>1.000</td>
<td>8.712</td>
<td>0.003402 **</td>
</tr>
<tr>
<td>s(wind)</td>
<td>1.000</td>
<td>1.000</td>
<td>4.271</td>
<td>0.039595 *</td>
</tr>
<tr>
<td>s(humidity)</td>
<td>3.631</td>
<td>4.517</td>
<td>2.763</td>
<td>0.018556 *</td>
</tr>
<tr>
<td>s(temp)</td>
<td>4.361</td>
<td>5.396</td>
<td>4.864</td>
<td>0.000182 ***</td>
</tr>
<tr>
<td>s(ibht)</td>
<td>3.043</td>
<td>3.725</td>
<td>1.292</td>
<td>0.356960</td>
</tr>
<tr>
<td>s(dgpg)</td>
<td>3.230</td>
<td>4.108</td>
<td>10.603</td>
<td>3.46e-08 ***</td>
</tr>
<tr>
<td>s(ibtp)</td>
<td>1.939</td>
<td>2.504</td>
<td>1.808</td>
<td>0.197988</td>
</tr>
<tr>
<td>s(vsty)</td>
<td>2.232</td>
<td>2.782</td>
<td>5.822</td>
<td>0.000890 ***</td>
</tr>
<tr>
<td>s(day)</td>
<td>4.021</td>
<td>5.055</td>
<td>15.817</td>
<td>3.58e-14 ***</td>
</tr>
</tbody>
</table>
...
R-sq.(adj) = 0.797    Deviance explained = 81.2%
GCV = 14.137    Scale est. = 13.046    n = 330
We can go on to do model selection etc much as we would for a NLM or GLM. See L7.R for the rest of this example.