Smoothing with penalized splines

A brief introduction and an illustrative application

Antonio Gasparri

Department of Social and Environmental Health Research
London School of Hygiene and Tropical Medicine (LSHTM)

Centre for Statistical Methodology – LSHTM
29 May 2015
Outline

1. The issue
2. Splines
3. A penalized approach
4. A comparison
5. An extension
6. Software
7. Some comments
Outline

1. The issue
2. Splines
3. A penalized approach
4. A comparison
5. An extension
6. Software
7. Some comments

Gasparrini A
Smoothing with penalized splines
Relationships between x and y

Scatterplot of x and y
Non-linearity

Scatterplot of x and y
Regression models

A relationship between a predictor $x$ and an outcome $y$ is usually estimated through regression models, controlling for potential confounders.

In the simple linear case:

$$y_i = \alpha + f(x_i) + \sum_{p=1}^{P} \gamma_p z_{ip}$$  \hspace{1cm} (1)

A number of alternative options are available for representing $f(x)$, describing the relationship as a smooth shape.
Smoothing methods

**Parametric**
Polynomials, fractional polynomials, regression splines

**In between**
Penalized splines

**Non-parametric**
Lowess, kernel, smoothing splines
Outline

1. The issue
2. Splines
3. A penalized approach
4. A comparison
5. An extension
6. Software
7. Some comments

Gasparrini A
Smoothing with penalized splines

LSHTM
Splines: basis representation

A spline is a numeric function composed by **piecewise-connected polynomial functions**

The advantage of using regression splines is that $f(x)$ can be represented in a **basis form**:

$$f(x_i; \beta) = \sum_{j=1}^{d} \beta_j b_j(x_i) = x^T \beta$$  \(2\)

where $b_j(x)$ are a series of $d$ (known) basis transformations of $x$
Regression splines

This type of splines allow the use of **standard estimation methods**, derived by minimizing the usual least square objective:

$$\sum_{i=1}^{N} \left( y - \alpha - f(x; \beta) + \sum_{p=1}^{P} \gamma_p z_p \right)^2 = \| y - \alpha - X\beta - Z\gamma \|^2$$  \(3\)

Several different transformations for \(b_j(x)\) (e.g. B-splines, natural splines), determining the mathematical properties
Graphical representation - I

Knots and splines
Graphical representation - II

Splines and coefficients

Splines

0 2 4 6 8 10
0.0 0.5 1.0

x

Splines

1.01, -0.49, 1.09, 4.45, 13.29, 4.26

Gasparrini A

Smoothing with penalized splines
Graphical representation - III

Sum of linear terms

Terms

0 5 10 15

-5 0 5 10 15

X

Smoothing with penalized splines
Graphical representation - IV

Estimated relationship

Gasparrini A
Smoothing with penalized splines
Graphical representation - V

Estimated and true

Estimated and true

Gasparrini A
Smoothing with penalized splines
Problems and limitations

In regression splines, the smoothness of the fitted curve is determined by:

- the degree of the spline
- the specific parameterization
- the number of knots
- the location of knots

No general selection method for number and position of knots
Number of knots

Degree of smoothness

Gasparrini A
Smoothing with penalized splines
Knots location

Best fitting models

5k at eq.-spaced val
8k at eq.-spaced per

Gasparrini A
Smoothing with penalized splines
Outline

1. The issue
2. Splines
3. A penalized approach
4. A comparison
5. An extension
6. Software
7. Some comments
A general framework of smoothing methods is offered by **generalized additive models** (GAMs)

GAMs extends traditional GLMs by allowing the linear predictor to depend linearly on unknown smooth functions. In the linear case:

\[ y_i = \alpha + f(x_i) + \sum_{p=1}^{P} f(z_{ip}) \]  

where \( f \) are traditionally represented by non-parametric terms such as smoothing splines of lowess
Penalty

The idea is to define a flexible function and control the smoothness through a **penalty term**, usually on the second derivative.

The objective in (3) is modified to:

\[
\sum_{i=1}^{N} \left( y - \alpha - f(x; \beta) + \sum_{p=1}^{P} f(z_{ip}) \right)^2 + \lambda \int [f''(x)]^2 dx
\]

with \( \lambda \) as **smoothing parameter**.
Penalized splines

However, traditional GAMs are limited by complex and computationally-heavy estimation methods

Penalized splines offer an flexible and efficient version of GAM, based on low-rank basis transformations

The objective in (5) can be re-written in matrix terms as:

$$||y - \alpha - X\beta - Z\gamma||^2 + \lambda \beta^T S \beta$$

(6)

where $S$ is a penalty matrix
Smoothers

Alternative smoothers available, differing by parameterization and penalty:

- Thin-plate splines
- Cubic splines
- P-splines
- Random-effects
- Markov random fields
- Soap film smooths
- ...

Gasparrini A
Smoothing with penalized splines
Graphical representation - I

Increasing knots and splines
Graphical representation - II

Splines and coefficients

Splines

<table>
<thead>
<tr>
<th>0.79</th>
<th>0.39</th>
<th>0.11</th>
<th>0.26</th>
<th>1.07</th>
<th>1.70</th>
<th>3.25</th>
<th>4.52</th>
<th>7.62</th>
<th>9.25</th>
<th>9.43</th>
<th>7.42</th>
<th>4.82</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.20</td>
<td>0.11</td>
<td>0.54</td>
<td>2.20</td>
<td>5.88</td>
<td>10.07</td>
<td>2.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

x

Smoothing with penalized splines
Graphical representation - III

Sum of linear terms

Gasparrini A
Smoothing with penalized splines
Estimation

Estimation concerns coefficients of penalized and unpenalized terms ($\alpha$, $\beta$, $\gamma$) and smoothing parameters ($\lambda$s).

For the former, a penalized iteratively reweighted least squares (P-IRLS) scheme is used.

Estimation of $\lambda$s is integrated through either outer iteration or performance iteration, using GCV, UBRE/AIC or REML.
Advantages

Relatively *low-rank basis* and *simplified penalties*

Completely *parametric form*

Number and location of *knots* not critical

Automatic *smoothing selection*

Efficient *computational methods*

Well-grounded *theoretical framework*
Outline

1. The issue
2. Splines
3. A penalized approach
4. A comparison
5. An extension
6. Software
7. Some comments
Comparison of smoothing methods

- Fractional polynomials
- Regression splines
- Penalized splines
Simulations

Comparing alternative methods:

- Fractional polynomials
- Regression splines
- Penalized splines

Different shapes:

- Linear
- Decay
- Peak
- Complex
Simulated shapes

- **Linear**
- **Decay**
- **Peak**
- **Complex**
Simulation results - I

Fractional polynomials

- Linear
- Decay
- Peak
- Complex
Simulation results - II

Regression splines

Linear

Decay

Peak

Complex

Gasparrini A

Smoothing with penalized splines
Simulation results - III

Penalized splines

- Linear
- Decay
- Peak
- Complex

Smoothing with penalized splines
### Simulation results - IV

**Statistics**

<table>
<thead>
<tr>
<th>Fun</th>
<th>Model</th>
<th>(e)df</th>
<th>Bias</th>
<th>Cov</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>GLM</td>
<td>2.21</td>
<td>0.01</td>
<td>0.90</td>
<td>0.82</td>
</tr>
<tr>
<td>Linear</td>
<td>GAM</td>
<td>1.28</td>
<td>0.00</td>
<td>0.95</td>
<td>0.66</td>
</tr>
<tr>
<td>Decay</td>
<td>GLM</td>
<td>3.45</td>
<td>0.05</td>
<td>0.87</td>
<td>0.89</td>
</tr>
<tr>
<td>Decay</td>
<td>GAM</td>
<td>2.61</td>
<td>0.13</td>
<td>0.94</td>
<td>0.76</td>
</tr>
<tr>
<td>Peak</td>
<td>GLM</td>
<td>4.82</td>
<td>0.05</td>
<td>0.89</td>
<td>0.93</td>
</tr>
<tr>
<td>Peak</td>
<td>GAM</td>
<td>4.06</td>
<td>0.19</td>
<td>0.94</td>
<td>0.84</td>
</tr>
<tr>
<td>Complex</td>
<td>GLM</td>
<td>6.65</td>
<td>0.11</td>
<td>0.87</td>
<td>1.02</td>
</tr>
<tr>
<td>Complex</td>
<td>GAM</td>
<td>5.87</td>
<td>0.31</td>
<td>0.91</td>
<td>0.94</td>
</tr>
</tbody>
</table>
Distributed lag non-linear models

Statistical tools to model **non-linear** and **lagged** dependencies

Defined by a **cross-basis** function of lagged exposures:

\[ s(x_{t-\ell_0}, \ldots, x_{t-\ell}) = \sum_{\ell=\ell_0}^{L} f \cdot w(x_{t-\ell}, \ell) \] (7)

The function is composed of an **exposure response** function \( f(x) \) and a **lag-response** function \( w(\ell) \)
An example

Gasparrini A

Smoothing with penalized splines
Tensor product basis

Parameterized by a **special tensor product**:

\[ s(x_{t-\ell_0}, \ldots, x_{t-\ell}) = (1^T_{L-\ell_0+1} A_t) \eta = w_t^T \eta \quad (8) \]

with

\[ A_t = (1^T_{\ell} \otimes R_t) \odot (C \otimes 1^T_x) \quad (9) \]

where \( R_t \) and \( C \) are basis matrices for \( x \) and \( \ell \), respectively.
Penalized DLNMs

**Question**: what about a penalized version of DLNMs?

Modify objective in (6) to:

$$\| y - \alpha - W\eta - Z\gamma \|^2 + \eta^T \left( \lambda_x \left( \mathbf{1}_{v_\ell} \otimes \mathbf{S}_x \right) + \lambda_\ell \left( \mathbf{S}_\ell \otimes \mathbf{1}_{v_x} \right) \right) \eta \tag{10}$$

with $\lambda_x$, $\lambda_\ell$ and $\mathbf{S}_x$, $\mathbf{S}_\ell$ as **smoothing parameters** and **penalty matrices** for each dimension.
Simulated surfaces

Scenario 1

Scenario 2

Scenario 3

Gasparrini A
Smoothing with penalized splines
Simulation results

Scenario 1

GAMps

Scenario 2

GAMps

Scenario 3

GAMps

Gasparrini A
Smoothing with penalized splines
An application

Exposure–lag–response

Lag–response for 29C

Overall cumulative exposure–response

Smoothing with penalized splines

Gasparrini A
The R package \texttt{mgcv}

Collection of functions implementing GAMs with penalized splines

Written by Simon Wood, extensively documented

Example of code:

\begin{verbatim}
litrary(mgcv)
model <- gam(y \sim s(x,bs="ps") + z, data, family=gaussian, method="REML")
\end{verbatim}

The function \texttt{s} determines the spline transformations and penalties
The R package *dlnm*

Collection of functions implementing DLNMs

Example of code:

```r
library(dlnm)
cb <- crossbasis(x, lag=c(10),
                 argvar=list(fun="bs", degree=2, knots=5, cen=0),
                 arglag=list(fun="ns", knots=c(3,6), int=F))
model <- glm(y ~ s(x, bs="ps") + z, data, family=gaussian)
pred <- crosspred(cb, model)
plot(pred, "3d", xlab="x", ylab="Lag", zlab="Effect")
```
Embedding \texttt{dlnm} and \texttt{mgcv}

Example of code:

```r
library(dlnm); library(mgcv)
cb <- crossbasis(x, lag=c(10), argvar=list(fun="ps"),
                 arglag=list(fun="ps"))
pen <- cbPen(cb)
model <- gam(y ~ cb + z, data, family=gaussian,
             method="REML", parapen=list(cb=list(pen)))
pred <- crosspred(cb, model)
plot(pred,"3d",xlab="x",ylab="Lag",zlab="Effect")
```
Outline

1. The issue
2. Splines
3. A penalized approach
4. A comparison
5. An extension
6. Software
7. Some comments

Gasparrini A
Smoothing with penalized splines
Some comments

Penalized splines combine the \textit{flexibility} of non-parametric methods with \textit{stability and simplicity} of parametric smoothers.

Based on \textit{theoretically-grounded} and \textit{computationally-efficient} estimators.

\textbf{Well implemented} in the package \texttt{mgcv} in R.

Research \textit{still ongoing}.