# Smoothing Splines

Modellazione Statistica

Vito Simone Lacatena 747810

## 1. Introduzione

Prima di discutere della struttura dell'algoritmo di smoothing splines, occorre introdurre dei concetti utili alla discussione del metodo, partendo dal definire il problema della regressione.

### 1.1 Il problema della regressione

La regressione è un approccio di apprendimento supervisionato in cui si ha l'obiettivo di prevedere il valore di una **variabile di risposta** $Y$, dato un insieme di **variabili predittive** $X_1, ..., X_p$. Questo viene fatto stimando una funzione ignota $f$ con una funzione $\hat{f}$. Questo obbiettivo può essere raggiunto attraverso:

- **metodi parametrici**: In cui si fanno assunzioni sulla forma della funzione $f$ (ad esempio l'assunzione di linearità) e successivamete si stimano i parametri;
- **metodi non parametrici**: non si fanno assunzioni su una particolare forma della funzione $f$ ma si cerca una stima che si avvicini il più possibile alle osservazioni senza risultare troppo irregolare, tale approccio necessita di un numero elevato di esempi per poter ottenere una buona stima.

### 1.2 Regressione Lineare

Il più semplice modello di regressione è il modello di **regressione lineare** in cui si assume una relazione lineare tra una singola variabile predittiva $X$ e la variabile di risposta, tale funzione lineare viene approssimata come segue: $Y$

$$Y = \beta_0 + \beta_1 X + \epsilon$$

I parametri $\beta_0$ e $\beta_1$ sono sconosciuti e vanno stimati utilizzando i dati, il termine $\epsilon$ è un termine di errore con media 0.

L'obiettivo è determinare le stime $\hat{\beta}_0$ e $\hat{\beta}_1$, dove $\hat{\beta}_0$ è l'intercetta e $\hat{\beta}_1$ è la pendenza della retta che dovrebbe passare il più vicino possibile alle osservazioni (Figura 1).

Per determinare il valore ottimale di questi parametri è possibile utilizzare diversi metodi, il più comune è il criterio dei **minimi quadrati**, ovvero scegliere le stime dei parametri $\hat{\beta}_0$ e $\hat{\beta}_1$ in modo da minimizzare la **somma dei residui al quadrato** definita come :

$$RSS = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{N} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

La regressione lineare standard ha dei *limiti in termini di capacità predittiva*, dato che necessita dell'assunzione di linearità, e se la vera funzione non risulta lineare i risultati ottenuti saranno molto distanti da quelli reali, poiché il presupposto di linearità è molto spesso un'approssimazione scarsa del problema reale.

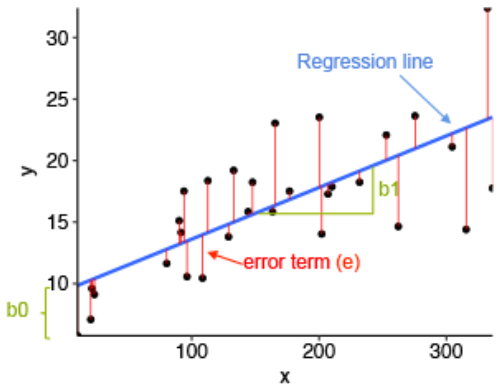



Figure 1: Regressione Lineare

## 1.3 Regressione Polinomiale

Il modello di regressione polinomiale estende il modello di regressione lineare attraverso l'aggiunta di ulteriori predittori ottenuti ciascuna delle variabili originali ad una potenza.

Formalmente, dato il modello lineare classico:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

esso viene esteso con la seguente forma polinomiale:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_i^2 + ... + \beta_d x_i^d + \epsilon_i$$

In questo modo è possibile ottenere curve non lineari. Non è tuttavia consigliabile utilizzare polinomi di grado troppo elevato poiché la curva diventerebbe troppo flessibile, nella maggior parte delle applicazioni è infatti suggeribile non andare oltre al grado 3 o 4.

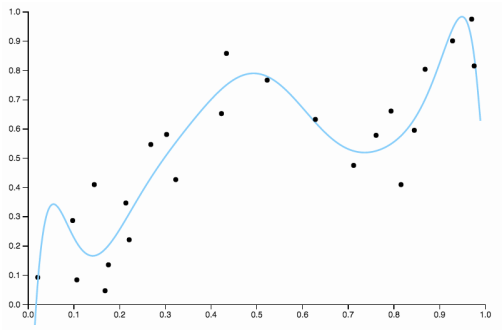


Figure 2: Regressione Polinomiale

## 1.4 Funzioni a gradino

In alcune applicazioni, anche la regressione polinomiale può essere considerata limitata, tale limite è dato dalla loro natura globale: *determinare dei coefficienti per ottenere una specifica forma funzionale in una regione può far sì che la funzione assuma una forma troppo irregolare in regioni distanti.*

Un modo per superare questo limite è quello di utilizzare delle **funzioni a gradino**.

Tali funzioni dividono l'intervallo dei valori in $K$ bins distinti creando un insieme di valori di soglia (detti anche **cutpoints**) $c_1, c_2, ..., c_k$ e per ogni bin si va ad adattare una funzione costante, avremo quindi una funzione costante a tratti:

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + ... + \beta_k C_k(x_i) + \epsilon_i$$

dove $C_j$ con $j = 1...k$ è una funzione che assume valore 1 se $c_{j-1} < x_i < c_j$ altrimenti 0.

## 1.5 Funzioni base

I modelli di regressione polinomiali e funzoni a scalini sono casi particolari di un framework generico di funzioni base, in cui si va ad adattare il seguente modello:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + ... + \beta_k b_k(x_i) + \epsilon_i$$

dove $b_j(.)$ sono funzioni base scelte in anticipo.

Questo significa che è possibile rivedere i casi precedenti come istanze di questo generico modello, ovvero:

- regressione polinomiale: $b_j(x_i) = x_i^j$ ;
- funzioni a scalini : $b_j(x_i) = C_j(x_i)$ ;

## 1.6 Regressione polinomiale a tratti e Spline

Esiste una classe di funzioni base che estende l' approccio che di regressione polinomiale e la regressione constante a tratti.

**Idea**: Invece di adattare un polinomio di grado elevato su l'intero intervallo dei valori della variabile predittrice $X$ ,si potrebbe pensare di utilizzare una funzione polinomiale a tratti, dividendo il dominio di X in intervalli contigui e rappresentando f con un polinomio di grado inferiore separato in ogni intervallo.

Tale approccio è chiamato **regressione polinomiale a tratti**.

**Esempio**: Se si considerano polinomi di terzo grado e si divide l'intervallo originale X sul punto $c$, avremmo due cubiche con differenti coefficienti:

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \varepsilon_i & \text{se } x_i < c \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \varepsilon_i & \text{se } x_i \geq c \end{cases}$$

In questo caso si adatteranno due diversi polinomi con i propri coefficienti, il punto $c$ dell'intervallo in cui si passa da un polinomio ad un altro è chiamato **nodo** ( o **knot** ).

**Osservazione**: con $K$ nodi si avranno $K + 1$ polinomi.

I **gradi di libertà** per questo esempio sono 8, per gradi di libertà si intende il numero di parametri liberi (come il numero dei coefficienti in una funzione polinomiale), nell'esempio precedente il numero di gradi di libertà è quattro per ogni polinomio (poiché di grado 3), quindi in tutto otto.

### 1.6.1 Vincoli di continuità

Per avere una curva che non sia troppo flessibile e non appaia discontinua nel passaggio da un intervallo all'altro è importante aggiungere tre vincoli: la **funzione deve essere continua** e la **derivata prima e seconda dei polinomi a tratti devono essere continue**, in questo modo non solo il polinomio a tratti sarà continuo ma sarà anche smussato (Figura 3).

Ad ogni vincolo il numero di gradi di libertà si riduce di 1, in questo modo i gradi libertà del polinomio di grado 3 visto prima si riducono a 5.

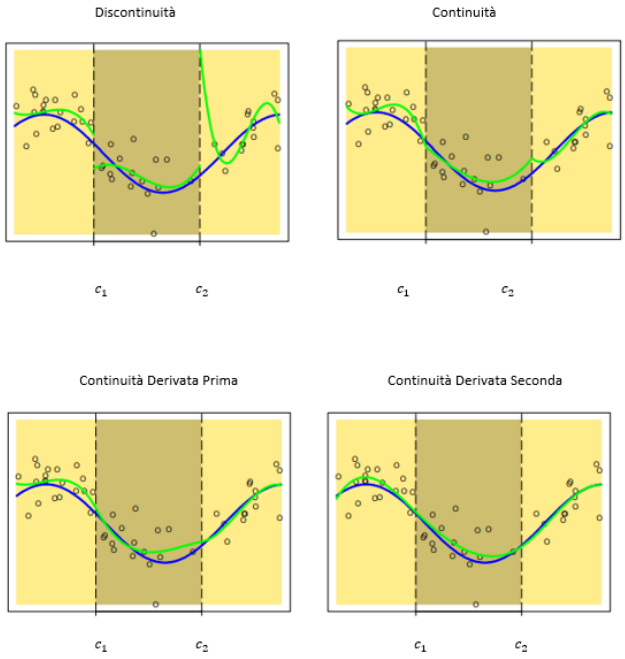



Figure 3: Differenze con diversi livelli di continuità

### 1.6.2 Spline

Una **spline di ordine** $d$ con nodi $c_j$, $j = 1, ..., K$ *è un polinomio a tratti di ordine* $d$ *avente derivate continue sino all' ordine* $d - 1$.

Le spline più utilizzate sono le **spline cubiche**, con $K$ nodi utilizzano $K + 4$ gradi di libertà e sono due volte differenziabile nell'intero intervallo X. Questo tipo di spline sono popolari perchè risulta difficile all'occhio umano individuare una discontinuità ai nodi.

### 1.6.3 Basi della Spline

Una regressione spline può essere rappresentata in termini di funzioni basi, come le **funzioni base a potenza troncate**.

**Caso semplice:** regressione spline lineare con grado $d = 1$ e $K = 1$ nodi

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \epsilon_i$$

dove $b_1(x) = x$ e successivamente si usano funzioni base troncate

$$h(x, c) = (x - c)_+ = \begin{cases} (x - c) & \text{se } x > c \\ 0 & \text{se } x \leq c \end{cases}$$

Se $x_i \leq c$ allora $(x_i - c)_+ = 0$ e $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$

Se $x_i > c$ allora $(x_i - c)_+ = (x_i - c)$ e $y_i = \beta_0 + \beta_1 x_i + \beta_2 (x_i - c) + \epsilon_i$

**Caso generale:** regressione spline di grado $d$ e $K$ nodi

$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_K b_K(x_i) + \beta_{K+1} b_{K+1}(x_i) + ... + \beta_{K+d} b_{K+d}(x_i) + \epsilon_i$

le funzioni base sono

$$x, x^2, ..., x^d, h(x, c_1), ..., h(x, c_K)$$

dove in questo caso

$h(x, c) = (x - c)_+^d = (x - c)^d$ se $x > c$ altrimenti 0.

#### 1.6.4 Splines cubiche naturali

Le Spline di grado superiore ad secondo mostrano un elevata variabilità agli estremi dell'intervallo.

Le **spline naturale** risolvono questo problema, e permettono di produrre stime più stabili agli estremi dell' intervallo, aggiungendo un ulteriore vincolo che impone che la funzione sia lineare agli estremi dei nodi, con l' aggiunta di questo vincolo si liberano in tutto 4 gradi di libertà.

#### 1.6.5 La scelta del numero e della posizione dei nodi

Un metodo obiettivo per determinare il numero e la posizione dei nodi consiste nell'utilizzare la cross-validation in cui si effettuano più iterazioni utilizzando una parte dei dati per adattare la spline con un determinato numero K di nodi e i dati non visti rimanenti sono utilizzati per fare la previsioni, si calcola una somma dei quadrati dei residui complessiva per avere una misura di valutazione. La procedura può essere effettuata più volte per diversi valori di K e scegliere il modello che ha fornito un risultato di RSS più piccolo.

#### 1.6.6 Vantaggi rispetto alla regressione polinomiale

Le spline di regressione permettono di ottenere risultati superiori rispetto a quelli della regressione polinomiale, perchè a differenza dei polinomi che per produrre una curva flessibile deve utilizzare un alto grado, le spline ottengono flessibilità aumentando il numero dei nodi ma mantenendo un grado basso, producendo in questo modo stime più stabili permettendo di posizionare più nodi(quindi più flessibilità) in regioni in cui la funzione da stimare tende a cambiare rapidamente e meno nodi in regioni in cui la funzioni è più stabile.

# 2. Smoothing Splines

## 2.1 Panoramica

Si consideri il problema di trovare la funzione $f(x)$ tale che minimizzi la **somma dei quadrati dei residui**

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

In assenza di alcun vincolo si potrebbe ottenere un RSS pari a zero scegliendo la funzione $f$ che interpoli perfettamente tutti i puntii $y_i$, tuttavia la curva di regressione risulterebbe troppo flessibile.

Occorre quindi e il modo di smussare la funzione, modificando la $RSS$ da minimizzare nella seguente forma:

$$RSS(f, \lambda) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(t)^2 dt$$

La funzione $f$ è definita **smoothing spline**.

L'approccio smoothing splines è una tecnica molto utilizzata di **regressione non parametrica**, dato che non si fa alcuna assunzione sulla forma di $f$ ma si cerca una stima di una funzione sconosciuta che si avvicini il più possibile alle osservazioni senza risultare né troppo approssimativa né, nel caso opposto, troppo irregolare.

La prima parte di $RSS$ definisce una misura di distanza dai dati mentre la seconda parte definisce una **termine penalità** che va a penalizzare la variabilità della funzione $f$, la quantità $\lambda \int f''(t)^2 dt$ è quindi una misura del cambiamento globale nella funzione $f'(t)$

- se $f$ è smussata $\int f'(t)$ sarà costante e $\int f''(t)$ avrà un valore molto piccolo;
- se $f$ è irregolare $\int f'(t)$ sarà molto variabile e $\int f''(t)$ avrà un valore molto grande;

Il parametro $\lambda$ è un **parametro di smoothing** che porterà $f$ ad essere smussata, il parametro assume valori compresi in $(0, \infty)$ :

- Con valore $\lambda = 0$ i suoi effetti sono nulli e quindi la funzione assumerà un comportamento molto irregolare;
- con $\lambda = \infty$ la funzione sarà una linea retta che passa il più vicino possibile ai dati.

Quindi grandi valori di $\lambda$ producono curve smussata, mentre piccoli valori di questo parametro producono curve più irregolari.

Occorre notare che all'aumentare del valore di $\lambda$ da 0 a $\infty$ il numero di gradi di libertà descresce da $n$ a 2, il parametro di regolarizzazione $\lambda$ controlla l'irregolarità della spline e quindi anche i gradi di libertà effettivi.

La funzione $f(x)$ che **minimizza la RSS** è una **spline cubica naturale** avente nodi ad ogni punto $x_1, ..., x_n$, ovvero è un polinomio cubico a tratti con nodi a valori distinti di X, derivate prime e seconde continue ad ogni nodo, ed inoltre è lineare nelle regioni al di fuori dei nodi estremi.

Il vettore n-dimensionale $\hat{f}_\lambda$ contenenti i valori adattati ai dati di training $x_1, ..., x_n$, nonchè soluzione di $RSS(f, \lambda)$ per un determinato valore di $\lambda$ può essere quindi scritto nella seguente forma :

$$\hat{f}_\lambda = S_\lambda y$$

Questo perché, riscrivendo la spline cubica naturale in termini delle sue funzioni base :

$$f(x) = \sum_{j=1}^{n} N_j(x)\theta_j$$

Dove $N_j$ denota la j-esima funzione base n-dimensionale della spline cubica naturale, e $\theta_j$ è un generico parametro.

Si può così riscrivere il criterio di minimizzazione nella seguente forma:

$$RSS(f, \lambda) = (y - N\theta)^T(y - N\theta) + \lambda\theta^T\Omega_n\theta$$

Con $N_{i,j} = N_j(x_i)$ e $\{\Omega_n\}_{j,k} = \int N_j''(t)N_k''(t)dt$

Il vettore di coefficenti $\hat{\theta} = (\hat{\theta_1}, ..., \hat{\theta_n})^T$ è soluzione del seguente problema di minimizzazione

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} (y - N\theta)^T(y - N\theta) + \lambda\theta^T\Omega_n\theta$$

che è un problema di *regressione ridge generalizzata*, avente come soluzione:

$$\hat{\theta} = (N^TN + \lambda\Omega_n)^{-1}N^Ty.$$

Dato ll vettore $\hat{f}_\lambda$ soluzione di RSS

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

può essere scritto come:

$$\hat{f}_\lambda = N(N^T N + \lambda\Omega_n)^{-1} N^T y$$

definendo $S_\lambda = N(N^T N + \lambda\Omega_n)^{-1} N^T$ si ha che:

$$\hat{f}_\lambda = S_\lambda y$$

La matrice di dimensione $n \times n$ $S_\lambda$ è nota come **matrice di smooothing**.

## 2.2 Proprietà della Matrice di Smoothing

La matrice di smoothing:

- é una matrice quadtrata $n \times n$ simmetrica di rango $n$
- è semi-definitia positiva ,ovvero $\forall x \neq 0 \; xS_\lambda x \geq 0$

Dato che $S_\lambda$ è simmetrica e semi definita positiva, essa ammette una **eigendecomposition**.

Assumendo che N sia invertibile (ovvero esiste una matrice $N^{-1}$ tale che il prodotto matriciale tra le due restituisce la matrice identità), la matrice di smoothing può essere riscritta nel seguente modo:

$S_\lambda = N(N^T N + \lambda\Omega_N)^{-1} N^T$

$= N(N^T N + \lambda(N^T N^{-T})\Omega_N(N^{-1}N))^{-1} N^T$

$= N[N^T(N + \lambda N^{-T})\Omega_N(N^{-1})N]^{-1} N^T$

$= NN^{-1}(I + \lambda N^{-T}\Omega_N N^{-1})^{-1} N^{-T} N^T$

$= (I + \lambda K)^{-1}$

Dove $K = N^{-T}\Omega_N N^{-1}$ è una **matrice di penalità**.

La decomposizione della matrice in termini dei suoi autovalori e autovettori è quindi la seguente:

$$S_\lambda = \sum_{k=1}^{N} \rho_k(\lambda) u_k u_k^T$$

Con

$$\rho_k(\lambda) = \frac{1}{1 + \lambda d_k}$$

Dove $d_k$ è il k-esimo autovalore di $K$.

La **somma degli elementi diagonali della matrice di smoothing** corrisponde al **numero dei gradi di libertà** della smoothing spline.

$$df_\lambda = trace(S_\lambda)$$

## 2.3 Selezione automatica dei parametri di smoothing

In generale i parametri di smoothing da stimare per la regressione spline sono:

- il grado delle spline;
- il numero e la posizione dei nodi.

Nel caso delle smoothing splines si avrà un solo parametro $\lambda$ da stimare, poichè:

- Il grado della spline è fissato a 3 e
- I nodi della spline corrispondono alle osservazioni distinte;

Il parametro $\lambda$ dato regola il gradi di libertà effettivi della smoothing spline, inoltre è possibile ricavare il valore di $\lambda$ dato un fissato numero di gradi di libertà, questo perchè dato che $df_\lambda = trace(S_\lambda)$ è monotona in $\lambda$ è possibile invertire la relazione.

Uno dei metodi per valutare la scelta del parametro $\lambda$ è la **Cross-Validation**, in particolare la **Leave-One-Out (LOOCV)** che consiste nell'adattare il modello su N-1 osservazioni e effettuare una predizione sull'unica osservazione non vista. Questa procedura viene eseguita per N volte.

L'**errore di Cross-Validation Leave One Out** è calcolato come segue:

$$RSS_{CV}(\lambda) = \sum_{i=1}^{N} \left( y_i - \hat{f}_\lambda^{(-i)}(x_i) \right)^2$$

Dove la notazione $\hat{f}_\lambda^{(-i)}$ indica la funzione smoothing spline adattata su tutti i dati di training tranne che su $x_i$.

Si può dimostrare che questo errore può essere calcolato in modo efficente utilizzando la seguente formula:

$$RSS_{CV}(\lambda) = \sum_{i=1}^{N} \left( y_i - \hat{f}_\lambda^{(-i)}(x_i) \right)^2 = \sum_{i=1}^{N} \left[ \frac{y_i - \hat{f}_\lambda(x_i)}{1 - \{S_\lambda\}_{ii}} \right]^2$$

In questa formula:

- il termine $\hat{f}_\lambda$ indica la funzione spline adattata a *tutte* le osservazioni ( anche $x_i$) e valutata in $x_i$
- $\{S_\lambda\}_{ii}$ l'elemento diagonale i-esimo della matrice $S_\lambda$.

Questa formula afferma che è possibile calcolare in modo rapido ciascuno degli **adattameti leave one out** utilizzando solo la funzione adattata a tutte le osservazioni.

La LOOCV può essere interpretata come una somma dei residui quadtrati *pesata* con $w_i = \frac{1}{(1-\{S_\lambda\}_{ii})^2}$, dato che $\{S_\lambda\}_{ii}$ è diverso per ogni osservazione, il criterio da un peso diverso ad ogni osservazione. Un miglioramento della LOOCV è la Generalized Cross-Validation (GCV) che permette di equilibrare il peso delle diverse osservazioni, questo criterio trova il valore di $\lambda$ che minimizza

$$GCV(\lambda) = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_\lambda(x_i))^2}{(1 - trace(S_\lambda)/n)^2}$$

## 2.4 Compromesso Bias-Varianza

Il compromesso bias-varianza è una proprietà specifica di tutti i modelli di apprendimento automatico (supervisionati), che impone un compromesso tra la "flessibilità" del modello e il comportamento su dati non visti.

La **varianza** si riferisce alla *variazione del valore della funzione stimata $\hat{f}$ al variare dei dati di training* che vengono utilizzati, se un metodo presenta una varianza elevata allora piccole variazioni nei dati di training possono portare a grandi variazioni in $\hat{f}$.

Il **bias** si riferisce alla *distorsione introdotta approssimando un problema complesso con un modello più semplice*.

Occorre definire un compromesso tra bias e varianza poiché al crescere dell'una, decrescerà l'altra.

La scelta del valore di $df_\lambda$ ha effetto sul bias e sulla varianza in quanto:

- valori **troppo bassi di** $df_\lambda$ provocano un **alto bias** e una **bassa varianza** e la funzione andrebbe in **underfitting** dato che il modello sarà troppo semplice;

- valori **troppo alti di** $df_\lambda$ provocano un **basso bias** ma un **alta varianza** e la funzione andrebbe in **overfitting** dato che il modello sarà troppo flessibile;

Dato che $\hat{f}_\lambda = S_\lambda y$, si ha che:

- $Cov(\hat{f}) = S_\lambda Cov(y) S_\lambda^T = S_\lambda S_\lambda^T$ (Gli elementi diagonali sono le varianze puntuali al punto $x_i$)
- $Bias(\hat{f}) = f - E[\hat{f}] = f - S_\lambda f$

dove $f$ è il vettore (sconosciuto) delle valutazioni della vera f nei punti di training X.

## 2.5 Caso Multidimensionale

Sino ad ora si è considerato il caso unidimensionale del metodo delle smoothing splines, ma tale approccio può essere generalizzato anche per un numero superiore di dimensioni. Data la coppia $y_i, x_i$ , con $y_i \in \mathbb{R}$ e $x_i \in \mathbb{R}^d$, e una funzione di regressione d-dimensionale $f(x)$, si consideri il seguente problema:

$$\min_f \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda J[f]$$

dove $J$ è una funzione di penalità appropriata per stabilizzare una funzione f in $\mathbb{R}^d$. Esempio in $\mathbb{R}^2$:

$$J[f] = \int\int_{\mathbb{R}^2} \left[ \left( \frac{\partial^2 f(x)}{\partial x_1^2} \right)^2 + 2 \left( \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} \right)^2 + \left( \frac{\partial^2 f(x)}{\partial x_2^2} \right)^2 \right] dx_1 dx_2.$$

Utilizzando per l'ottimizzazione questa penalità si ottiene una superficie bidimensionale smussata, nota come **spline a superficie sottile** (thin-plate spline). Condivide molte proprietà con la smoothing spline unidimensionale.

- Per $\lambda \to 0$ si ottiene come soluzione una funzione di interpolazione;
- Per $\lambda \to \infty$ la soluzione converge al piano dei minimi quadrati;

Per valori intermedi, la soluzione può essere rappresentata come un **espansione lineare delle funzioni base**. La soluzione ha la forma

$$f(x) = \beta_0 + \beta^T x + \sum_{j=1}^{T} a_j h_j(x) w$$

Dove $h_j = ||x - x_j||^2 log||x - x_j||$. I coefficienti si determinano sostituendo la $f(x)$ in

$$\min_f \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda J[f]$$

In generale si può rappresentare $f \in \mathbb{R}^d$ come una espansione di qualsiasi grande insieme di funzioni base, e controllare la complessità applicando un regolarizzatore come quello visto precedentemente.

Per esempio se si considera il caso di $X \in \mathbb{R}^2$ si ha :

- Una base di funzioni $h_{11}(X_1), ..., h_{1K_1}(X_1)$ per rappresentare la funzione di coordinate $X_1$
- Una base di funzioni $h_{21}(X_2), ..., h_{2K_2}(X_2)$ per rappresentare la funzione di coordinate $X_2$

Una funzione di coordinate $f(X_1, X_2)$ può essere rappresentata in termini del prodotto delle basi delle due funzioni su $X_1$ e $X_2$:

$$f(X_1, X_2) = \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} \theta_{ij} h_{1i}(X_1) h_{2j}(X_2)$$

Si può quindi costruire una base di una funzione spline multidimensionale dai prodotti di tutte le coppie di basi di funzioni di ogni smoothing-spline univariata.

L'aumentare della dimensione comporta una crescita esponenziale in termini di funzioni base e occorre quindi ridurre il numero di funzioni per ogni coordinata, uno dei modi più naturali utilizzata da una classe ristretta delle splines mutlidimensionali è quello di assumere che la funzione $f$ sia **additiva**, ovvero:

$$f(X) = \alpha + f_1(X_1) + ... + f_d(X_d)$$

In questo modo il termine di penalità può essere espresso come

$$J[f] = J(f_1 + f_2 + ... + f_d) = \sum_{j=1}^{d} \int f_j''(t_j)^2 dt_j$$

# 3. Implementazioni in R

R è un linguaggio e ambiente di programmazione nato negli anni 90 per usi statistici. R è oggi molto diffuso, grazie alla flessibilità della sintassi, alla sua natura modulare e la sempre più vasta comunità che ne supporta lo sviluppo. R rappresenta oggi un framework in grado di acquisire moli di dati dalle più disparate sorgenti, offrendo strumenti per una loro analisi immediata ed efficiente.

Sono presenti due principali implementazioni della smoothing spline in R, in due package distinti:

- la funzione **smooth.spline** nel package **stats**;
- la funzione **ss** nel package **npreg**;

## 3.1 Funzione smooth.spline

Gli argomenti in input alla funzione smooth.spline sono i seguenti:

| Argomento | Descrizione |
|---|---|
| x | Vettore di valori della variabile predittiva |
| y | Vettore di valori della variabile di risposta |
| w | Vettore di pesi della stessa dimensione di x (opzionale), di default è un vettore di 1 |
| df | Numero dei gradi di libertà desiderati. Più alto è il numero, più flessibile sarà la curva adattata, il valore é compreso essere compreso tra 1 e il numero di punti distinti in x |
| spar | Parametro di smoothing, se specificato il parametro $\lambda$ Può essere speè una funzione monotona di spar |
| lambda | Può essere specificato direttamente al posto di scar o df |
| cv | Booleano, se impostato a TRUE viene utilizzata la cross-validation di Tipo Leave-One-Out (LOOCV) altriementi se FALSE la Generalized Cross-Validation (GCV). Viene usato per il calcolo dei parametri di smoothing solo quando sia spar che df non sono specificati è anche se vengono speficiati è comunque usato per determinare cv.crit nel risultato. Impostando questo argoemento su NA si salta la valutazione |
| all.knots | Se TRUE utilizza tutii i punti distinti come nodi, se FALSE usa un sottoinsieme di questi punti |
| nknots | Può essere settato come il numero o un criterio che resituisce il numero di nodi ( funziona solo se all.nots = FALSE) |

| Argomento | Descrizione |
|---|---|
| keep.data | Booleano che specifica se i dati di input devono essere mantenuti nel risultato. Se TRUE (come per default), i valori adattati e i residui sono disponibili dal risultato |
| df.offset | Permette di aumentare i gradi di libertà di df.offset nel criterio GCV. |
| penality | Coefficiente della penalità per i gradi di libertà nel criterio GCV. |
| control.spar | Lista opzionale con i componenti nominati che controllano la ricerca della radice quando il parametro di smoothing spar è calcolato. |
| tol | Soglia di tolleranza per l'uguaglianza o l'unicità dei valori x. I valori sono suddivisi in intervalli di dimensione tol e i valori che cadono nello stesso intervallo sono considerati uguali. Deve essere strettamente positivo (e finito) |
| keep.stuff | Argomento booleano sperimentale che indica se il risultato deve tenere extra dai calcoli interni. Dovrebbe permettere di ricostruire la matrice X e altro. |

I componenti più importanti presenti nell'oggetto restituito dalla funzione smooth.spline sono i seguenti:

- **x** : i valori distinti di x;
- **y** : i valori adattati corrispondenti a x;
- **w**: i pesi utilizzati ai valori unici di x;
- **cv.crit**: punteggio della cross-validation effettiata a seconda di cv. Il punteggio CV è spesso chiamato "PRESS" , per 'PREdiction Sum of Squares';
- **pen.crit** : il criterio penalizzato, ovvero la somma (ponderata) dei quadrati residui (RSS);
- **df** : gradi di libertà utilizzati.;
- **lambda** : il valore di $\lambda$;
- **fit**: Lista di oggetti quali la sequenza dei nodi, il numero di coefficenti, e i loro valori, valore minimo e il range di valori di x;

## 3.2 Funzione ss

La funzione *ss* è ispirata alla funzione *smooth.spline* del package stats, in aggiunta a *smooth.splin*:

- Invece dell'argomento cv disponde di *method* permettendo di scegliere tra otti diversi metodi (GCV, OCV, GACV, ACV, REML, ML, AIC) per selezionare il parametro di smoothing;
- Permette di definire tre tipi di spline(linear, cubic,quintic);
- permette di definire un vincolo di periodicità
- permette di specificare i valori dei nodi

GLi argomenti x,y,w,df,spar,lambda,all.knots,nknots,keep.data,df.offset,penality,control.spar sono gli stessi ritrovati in smooth.spline. Di seguito verranno descritti gli argomenti presenti in *ss* e non presenti in *smooth.spline*.

| Argomento | Descrizione |
|---|---|
| method | Metodo per selezionare il parametro di smoothing. Ignorato se viene fornito spar o lambda. |
| m | Il valore predefinito è m = 2, che è una spline di smoothing cubica. Imposta m = 1 per uno spline lineare o m = 3 per uno spline quintico |
| periodic | Se TRUE, la funzione stimata f(x) è costretta ad essere periodica |
| knots | Vettore dei valori dei nodi per la spline. I valori dovrebbero essere singoli e all'interno dell'intervallo dei valori x (per evitare un warning). |

| Argomento | Descrizione |
|---|---|
| bernoulli | Se TRUE, vengono utilizzati polinomi di Bernoulli scalati per le funzioni di base e di penalizzazione. Se FALSE, produce la definizione "classica" di una spline di smoothing. |

# 4. Esempio: Graduate Admission

## 4.1 Dataset

Vediamo ora un esempio pratico utilizzando come dataset di riferimento **Graduate Admission** ( link: https://www.kaggle.com/mohansacharya/graduate-admissions). Il dataset è stato creato per la previsione delle ammissioni dei laureati, contiene diversi variabili considerati importanti durante l'applicazione per i programmi di master. Le feature incluse sono:

- **GRE Scores** (General Test Score)
- **TOEFL Scores** (Test Of English as a Foreign Language)
- **University Rating** ( su 5 )
- **Statement of Purpose (SOP)** e **Letter of Recommendation Strength (LOR)** ( su 5 )
- **Undergraduate GPA** (Grade Point Average) ( su 10 )
- **Research Experience** ( booleano )
- **Chance of Admit** (da 0 a 1)

Questo dataset ha lo scopo di aiutare gli studenti nella selezione delle università con i loro profili. Il risultato previsto dà loro un'idea delle loro possibilità per una particolare università.

Utilizzando il Dataset è possibile effettuare un task di regressione utilizzando le variabili predittive fornite (punteggio GRE, punteggio TOEFL, rating universitario, ecc.) per determinare la relazione tra i punteggi ottenuti e la probabilità di ammissione di un nuovo candidato, valutando quanto quindi quanto questi test siano importanti per la probabilità di essere ammessi.

### Caricamento del dataset

```
data <- read.csv(file = './data/admission.csv',sep=',')


head(data,20)
```

```
##    ID GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research
## 1   1       337         118                 4 4.5 4.5 9.65        1
## 2   2       324         107                 4 4.0 4.5 8.87        1
## 3   3       316         104                 3 3.0 3.5 8.00        1
## 4   4       322         110                 3 3.5 2.5 8.67        1
## 5   5       314         103                 2 2.0 3.0 8.21        0
## 6   6       330         115                 5 4.5 3.0 9.34        1
## 7   7       321         109                 3 3.0 4.0 8.20        1
## 8   8       308         101                 2 3.0 4.0 7.90        0
## 9   9       302         102                 1 2.0 1.5 8.00        0
## 10 10       323         108                 3 3.5 3.0 8.60        0
## 11 11       325         106                 3 3.5 4.0 8.40        1
## 12 12       327         111                 4 4.0 4.5 9.00        1
## 13 13       328         112                 4 4.0 4.5 9.10        1
## 14 14       307         109                 3 4.0 3.0 8.00        1
## 15 15       311         104                 3 3.5 2.0 8.20        1
## 16 16       314         105                 3 3.5 2.5 8.30        0
```

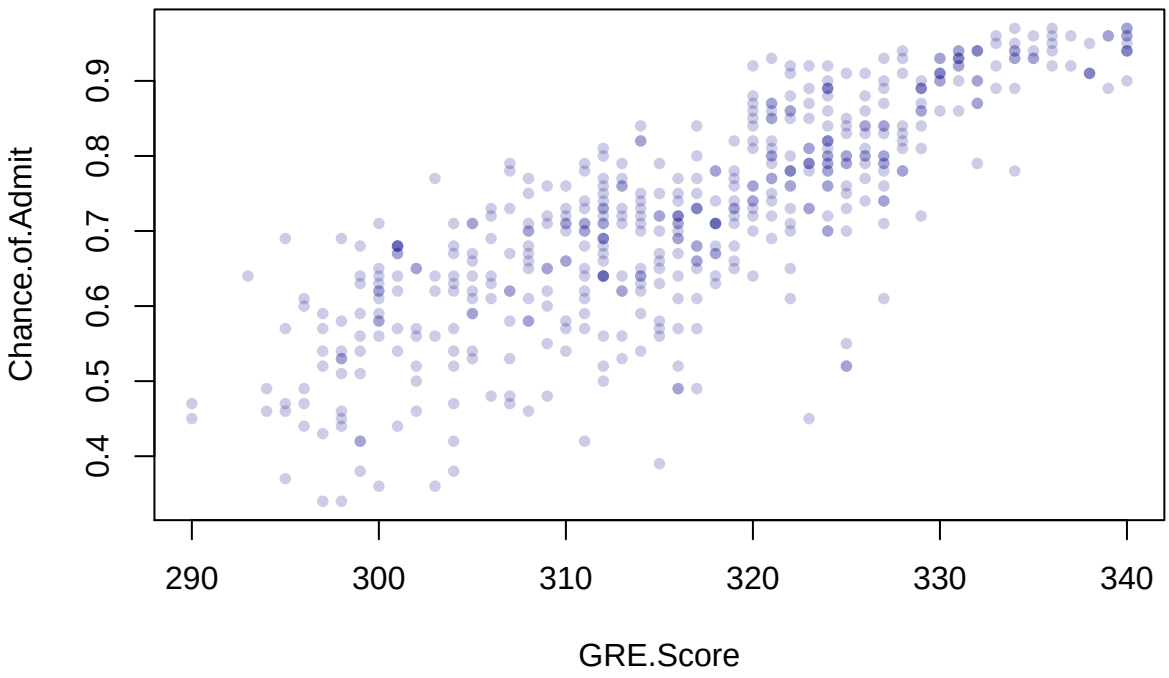
```
## 17 17       317         107                3 4.0 3.0 8.70        0
## 18 18       319         106                3 4.0 3.0 8.00        1
## 19 19       318         110                3 4.0 3.0 8.80        0
## 20 20       303         102                3 3.5 3.0 8.50        0
##    Chance.of.Admit
## 1             0.92
## 2             0.76
## 3             0.72
## 4             0.80
## 5             0.65
## 6             0.90
## 7             0.75
## 8             0.68
## 9             0.50
## 10            0.45
## 11            0.52
## 12            0.84
## 13            0.78
## 14            0.62
## 15            0.61
## 16            0.54
## 17            0.66
## 18            0.65
## 19            0.63
## 20            0.62
```

Di seguito verranno mostrate diverse applicazioni della smoothing spline utilizzando come variabile di risposta la variabile **Chance.of.Admit**, e un'unica variabile predittiva, scelta tra le variabili numeriche del dataset.

**Variabile Predittiva: GRE Score**

Considero come variabile predittive il punteggio GRE, plottando i valori di **Chance.of.Admit** per diversi valori della variabile **GRE Score**.

```
library(scales)
xlabel = 'GRE.Score'
ylabel = 'Chance.of.Admit'
x <- data$GRE.Score
y <- data$Chance.of.Admit
plot(x,y,xlab = xlabel,ylab = ylabel,type = "p",col=alpha('darkblue', 0.2),pch =16,cex=0.8)
```




### 4.2 Casi Estremi

Impostando il parametro **df** dei gradi di libertà con df = n (numero di osservazioni uniche ) e df = 2 si ottengono i casi estremi.

Si ricorda che al crescere del valore di $\lambda$ il numero di gradi di libertà decresce, la funzione *smooth.spline* determina automaticamente a quali valori di $\lambda$ corrispondono i gradi di libertà impostati e adatta la funzione in base ai parametri specificati.

```
n = length(unique(x))

spline_A <- smooth.spline(x,y,df = 2,all.knots = TRUE)
spline_B <- smooth.spline(x,y,df = n,all.knots = TRUE)

print(spline_A)
```

```
## Call:
## smooth.spline(x = x, y = y, df = 2, all.knots = TRUE)
##
## Smoothing Parameter  spar= 1.49996  lambda= 4366.614 (34 iterations)
## Equivalent Degrees of Freedom (Df): 2.000185
## Penalized Criterion (RSS): 0.4564486
## GCV: 0.006880627
```

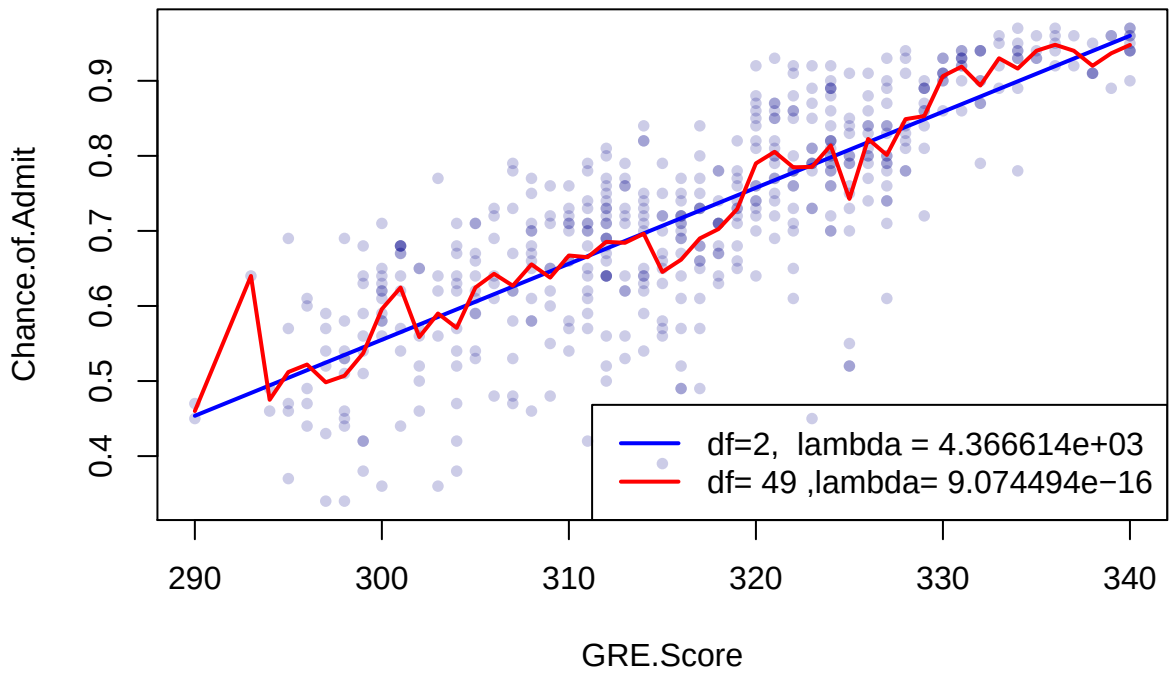
Speficando un numero di gradi di libertà pari a 2 si ottenuto un valore di $\lambda = 4366.614$ , la somma dei residui al quadrato *RSS* con questa funzione è di circa 0.456

```
print(spline_B)
```

```
## Call:
## smooth.spline(x = x, y = y, df = n, all.knots = TRUE)
##
## Smoothing Parameter  spar= -1.085928  lambda= 9.074494e-16 (27 iterations)
## Equivalent Degrees of Freedom (Df): 49
## Penalized Criterion (RSS): 2.808197e-20
## GCV: 0.007267405
```

Speficando un numero di gradi di libertà pari al numero di osservazioni distinte, si ottenuto un valore di $\lambda$ vicino allo zero, la somma dei residui al quadrato $RSS$ con questa funzione è anch'essa vicino allo zero in quanto si sarà ottenuta una curva che passa perfettamente per i punti di training ma sarà troppo flessibile.

```
plot(x,y,xlab = xlabel,ylab = ylabel,type = "p",col=alpha('darkblue', 0.2),pch =16,cex=0.8)
lines(spline_A,lwd=2,col="blue")
lines(spline_B,lwd=2,col="red")
legend("bottomright", c(paste("df=2, ",'lambda',"=",format(spline_A$lambda,scientific = TRUE)),paste("d
```




Dal grafico si può notare infatti che con un numero di gradi di libertà di 2 il parametro labmda assume un valore molto alto, la curva risultante sarà quindi una retta, al contrario con un numero di gradi di libertà pari al numero di osservazioni il parametro $\lambda$ avrà un valore vicino a 0 e la curva risultante sarà molto irregolare .

### 4.3 Adattare la smoothing spline con la cross-validation

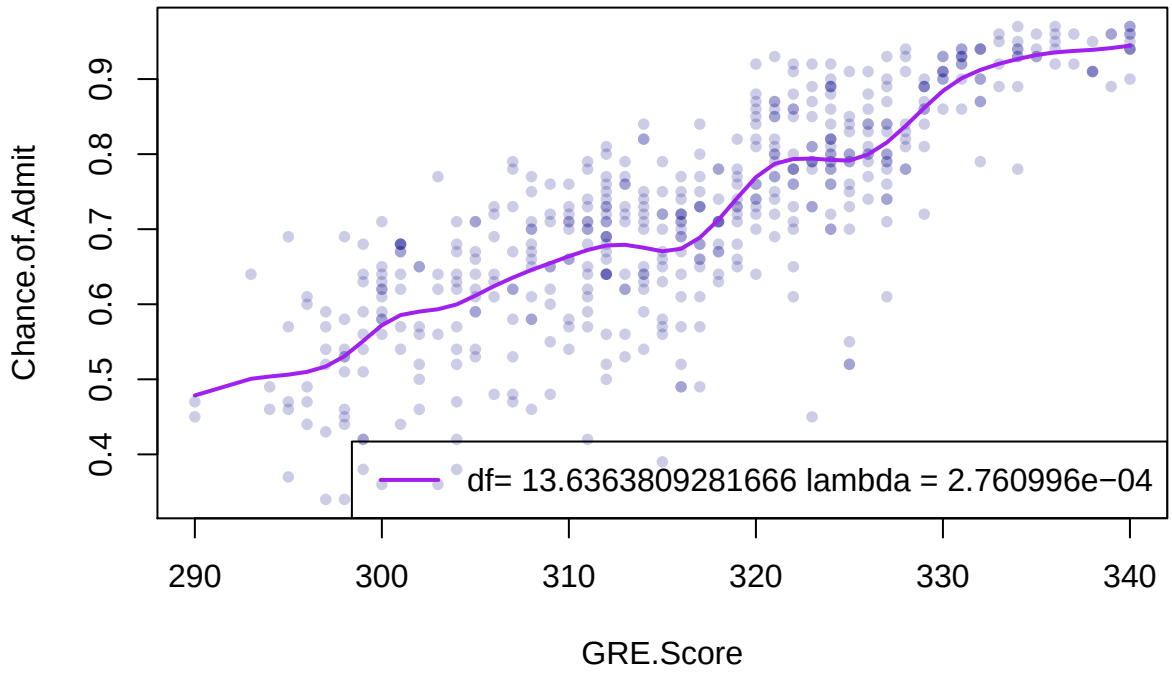
Vediamo di adattare la smoothing spline stimando i parametri di $\lambda$ e di *df* in modo da ottenere un risultato ottimale.

Richiamando la funzione *smooth.spline* non specificando nessuno dei parametri quali lambda, df e spar, la funzione effettuerà automaticamente una cross validation per stimare i parametri di smoothing.

```
GRE_fit <- smooth.spline(x, y,all.knots = TRUE)
GRE_fit
```

```
## Call:
## smooth.spline(x = x, y = y, all.knots = TRUE)
##
## Smoothing Parameter  spar= 0.5034762  lambda= 0.0002760996 (10 iterations)
## Equivalent Degrees of Freedom (Df): 13.63638
## Penalized Criterion (RSS): 0.191392
## GCV: 0.006653546
```

```
plot(x,y,xlab = xlabel,ylab = ylabel,type = "p",col=alpha('darkblue', 0.2),pch =16,cex=0.8)
lines(GRE_fit,lwd=2,col="purple")
legend("bottomright",paste("df=",GRE_fit$df ,'lambda',"=",format(GRE_fit$lambda,scientific = TRUE)),col=
```



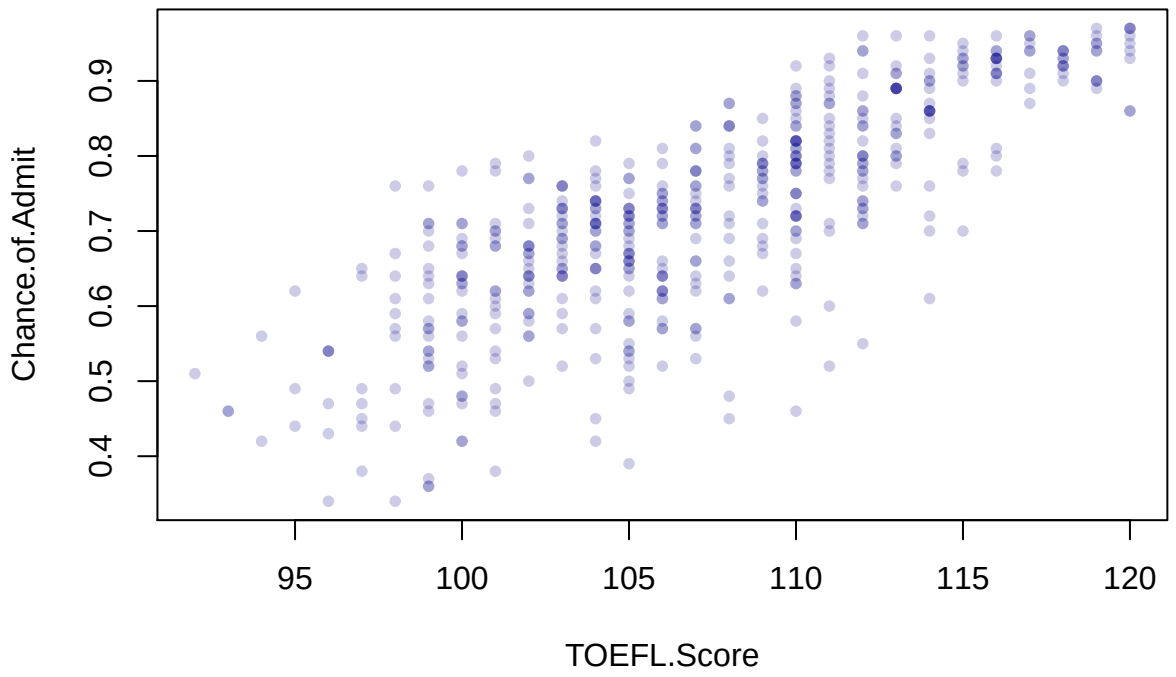

I parametri di smoothing ottenuti sono quelli che minimizzano l'errore di cross-validation. La smoothing spline adattata è la curva visualizzata nel grafico che minimizza la RSS ed assume una forma smussata.

## 4.3 Altre variabili del dataset

### Variabile Predittiva: TOEFL Score

Vediamo che rislutati si ottengono utilizzando come variabile predittiva il TOEFL Score per predire la probabilità di ammissione.

```
xlabel <- 'TOEFL.Score'
ylabel <- 'Chance.of.Admit'
x <- data$TOEFL.Score
y <- data$Chance.of.Admit
plot(x,y,xlab = xlabel,ylab = ylabel,type = "p",col=alpha('darkblue', 0.2),pch =16,cex=0.8)
```

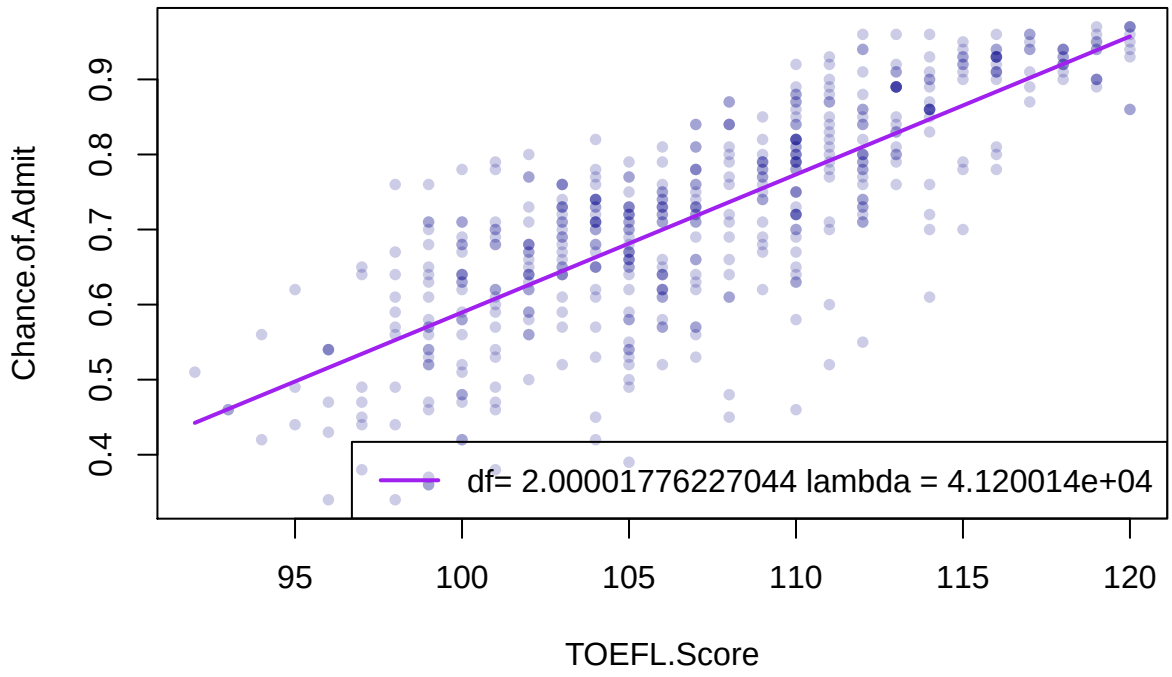



```
TOEFL_fit <- smooth.spline(x, y,all.knots = TRUE)
print(TOEFL_fit)
```

```
## Call:
## smooth.spline(x = x, y = y, all.knots = TRUE)
##
## Smoothing Parameter  spar= 1.499891  lambda= 41200.14 (28 iterations)
## Equivalent Degrees of Freedom (Df): 2.000018
## Penalized Criterion (RSS): 0.1849276
## GCV: 0.007462693
```

```
plot(x,y,xlab = xlabel,ylab = ylabel,type = "p",col=alpha('darkblue', 0.2),pch =16,cex=0.8)
lines(TOEFL_fit,lwd=2,col="purple")
```

```
legend("bottomright",paste("df=",TOEFL_fit$df ,'lambda',"=",format(TOEFL_fit$lambda,scientific = TRUE))
```




In questo caso la smoothing spline risulta lineare.

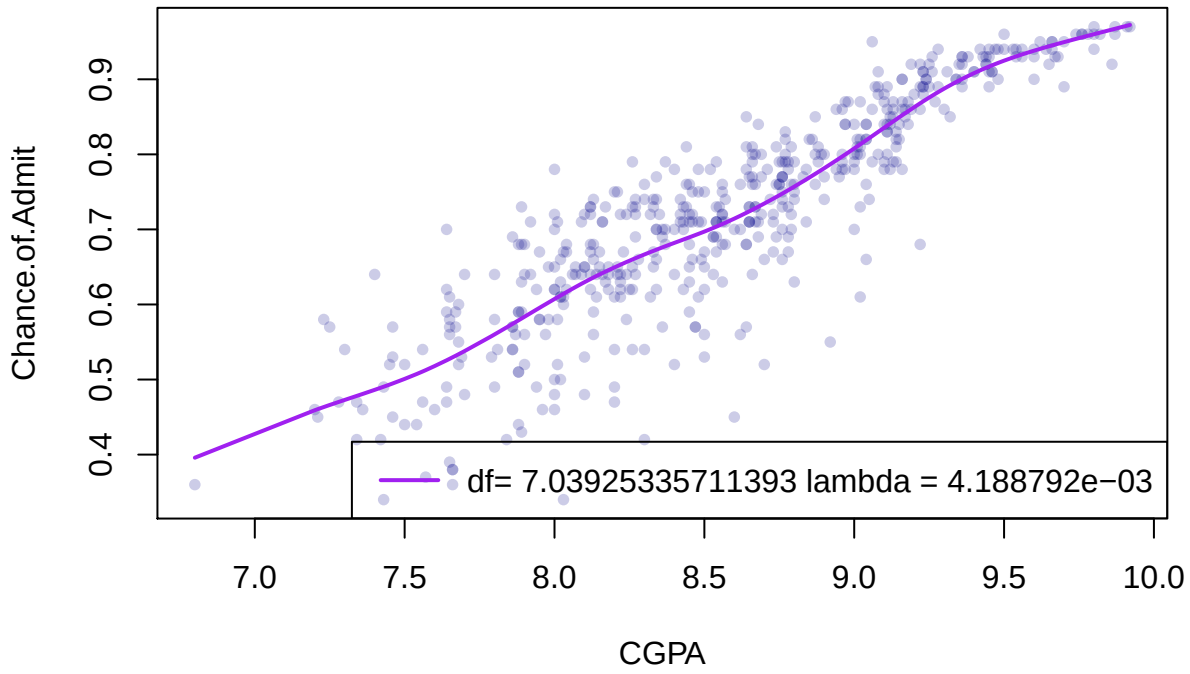
**Variabile Predittiva: CGPA**

```
plot(x= data$CGPA,
     y= data$Chance.of.Admit,
     xlab = 'CGPA',
     ylab = 'Chance.of.Admit',
     type = "p",
     col=alpha('darkblue', 0.2),
     pch =16,cex=0.8)

fit <- smooth.spline(x=data$CGPA,
                     y= data$Chance.of.Admit,
                     all.knots = TRUE)
print(fit)
```

```
## Call:
## smooth.spline(x = data$CGPA, y = data$Chance.of.Admit, all.knots = TRUE)
##
## Smoothing Parameter  spar= 1.045079  lambda= 0.004188792 (12 iterations)
## Equivalent Degrees of Freedom (Df): 7.039253
## Penalized Criterion (RSS): 0.7811333
## GCV: 0.004370443
```

```
lines(fit,lwd=2,col="purple")
legend("bottomright",paste("df=",fit$df ,'lambda',"=",format(fit$lambda,scientific = TRUE)),col="purple
```



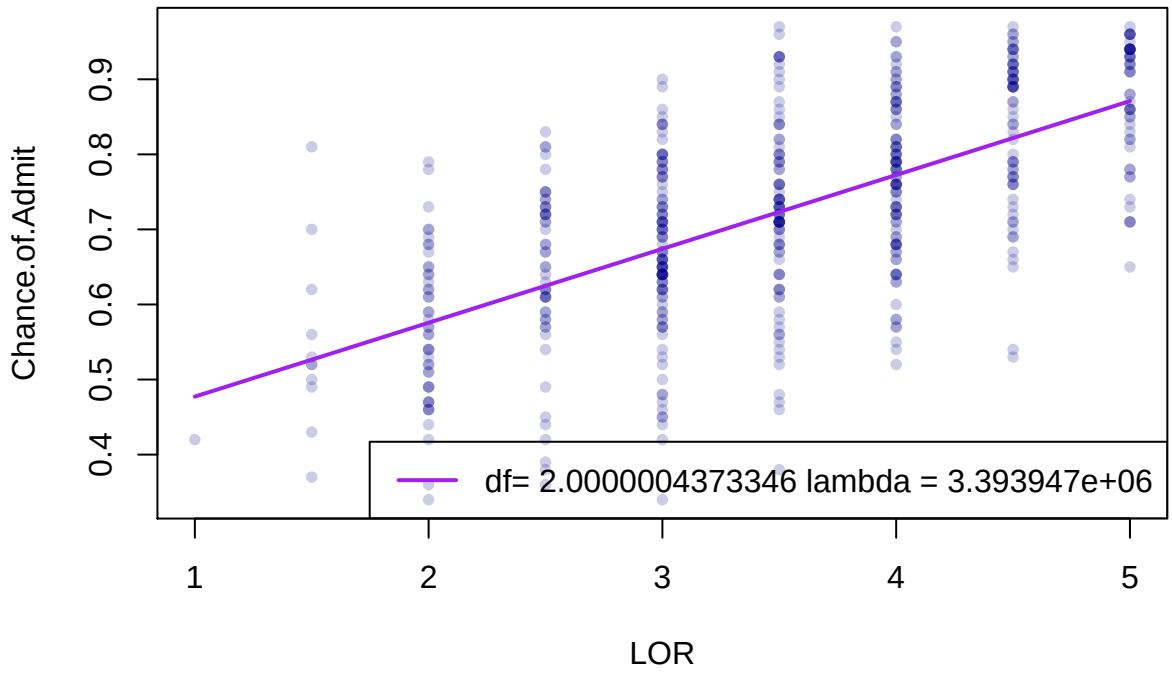

**Variabile Predittiva: LOR**

```
plot(x= data$LOR,
     y= data$Chance.of.Admit,
     xlab = 'LOR',
     ylab = 'Chance.of.Admit',
     type = "p",
     col=alpha('darkblue', 0.2),
     pch =16,cex=0.8)

fit <- smooth.spline(x=data$LOR,
                     y= data$Chance.of.Admit,
                     all.knots = TRUE)
print(fit)
```

```
## Call:
## smooth.spline(x = data$LOR, y = data$Chance.of.Admit, all.knots = TRUE)
##
## Smoothing Parameter  spar= 1.469608  lambda= 3393947 (27 iterations)
## Equivalent Degrees of Freedom (Df): 2
## Penalized Criterion (RSS): 0.04061662
## GCV: 0.01169389
```

```
lines(fit,lwd=2,col="purple")
legend("bottomright",paste("df=",fit$df ,'lambda',"=",format(fit$lambda,scientific = TRUE)),col="purple"
```



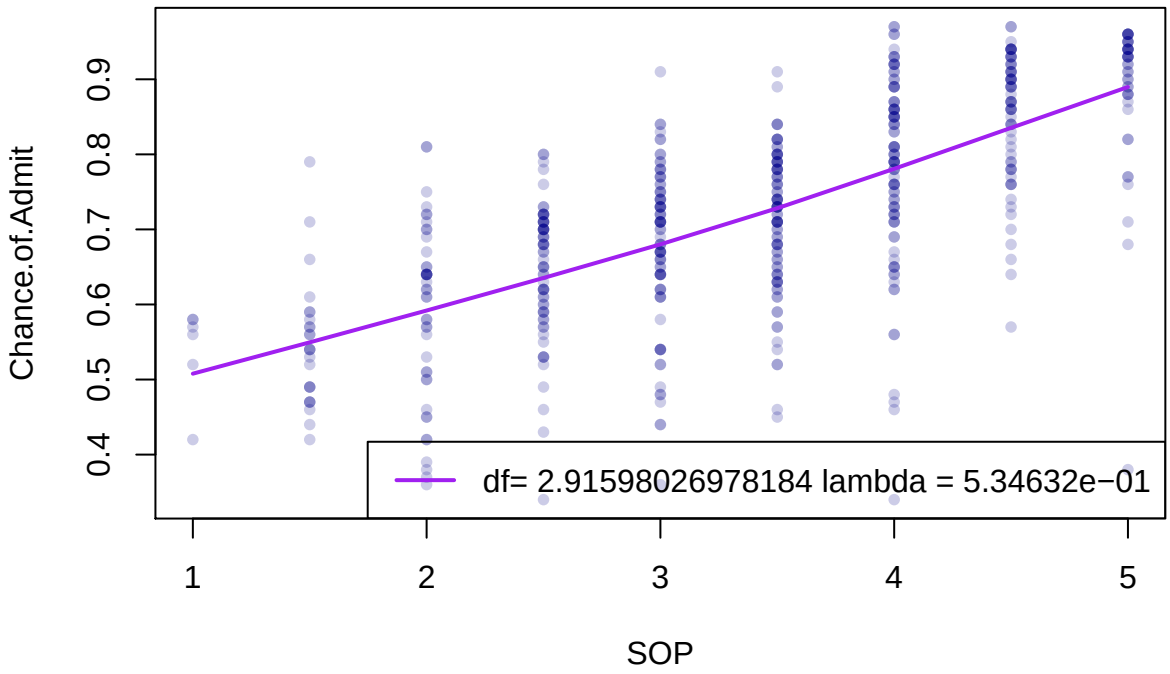

**Variabile Predittiva: SOP**

```
plot(x= data$SOP,
     y= data$Chance.of.Admit,
     xlab = 'SOP',
     ylab = 'Chance.of.Admit',
     type = "p",
     col=alpha('darkblue', 0.2),
     pch =16,cex=0.8)

fit <- smooth.spline(x=data$SOP,
                     y= data$Chance.of.Admit,
                     all.knots = TRUE)
print(fit)
```

```
## Call:
## smooth.spline(x = data$SOP, y = data$Chance.of.Admit, all.knots = TRUE)
##
## Smoothing Parameter  spar= 0.5281232  lambda= 0.534632 (11 iterations)
## Equivalent Degrees of Freedom (Df): 2.91598
## Penalized Criterion (RSS): 0.05065631
## GCV: 0.01062274
```

```
lines(fit,lwd=2,col="purple")
legend("bottomright",paste("df=",fit$df ,'lambda',"=",format(fit$lambda,scientific = TRUE)),col="purple"
```




## 5. Conclusioni

In conclusione le smoothing spline sono un tipo di spline ottenute dalla minimizzazione della somma dei quadrati dei residui soggetto ad una penalità di smoothing, viene utilizzato il parametro $\lambda$ per regolarizzare la funzione in modo che assumi una forma né troppo irregolare né troppo approssimativa in base ai dati.

Il metodo smoothing spline appartiene appartenente alla famiglia di metodi non parametrici e consente di ottenere risultati superiori rispetto all'utilizzo di approcci più limitati come la regressione lineare o polinomiale.

E stato mostrato un esempio di applicazione in R sul dataset Graduation Admission, con i seguenti risultati:

| X | df | $\lambda$ | RSS |
|---|---|---|---|
| GRE | 13.63638 | 0.0002760996 | 0.191392 |
| TOEFL | 2 | 41200.14 | 0.1849276 |
| CGPA | 7 | 0.004188792 | 0.7811333 |
| LOR | 2 | 3393947 | 0.04061662 |
| SOP | 3 | 0.534632 | 0.05065631 |