

Tema 4

Ajuste de Curvas

RESUMEN TEÓRICO

El problema de la aproximación consiste en encontrar un elemento de cierto subespacio S , de un espacio E , que se encuentra a la mínima distancia de un elemento dado $x \in E$. Si E es un espacio euclídeo y S es un subespacio de dimensión finita, sabemos que la solución viene dada por la proyección ortogonal de x sobre el subespacio S .

En este tema aplicaremos esta técnica para obtener la función, dentro de una cierta clase de funciones, que mejor se ajusta a una nube de puntos, en el sentido de minimizar el error cuadrático.

4.1. Ajuste de curvas.

Dado un conjunto de n puntos $\{(x_k, y_k)\}_{k=1}^n$, cuyas abscisas son todas distintas, es conocido que la recta de regresión ó recta óptima (en el sentido de los mínimos cuadrados) es la recta $y = ax + b$, donde el vector de los coeficientes $z = [a, b]$ es la única solución de las ecuaciones normales de Gauss $A^t Az = A^t c$

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

Esta recta, que puede que no pase por ninguno de los puntos anteriores, es la recta que hace mínimo el *error cuadrático*:

$$E^2 = \sum_{k=1}^n (ax_k + b - y_k)^2,$$

y en consecuencia es una línea de ajuste óptimo. Nótese que si la representación gráfica de los puntos $\{(x_k, y_k)\}_{k=1}^n$ sugiere que la relación entre x e y es lineal, construir el polinomio interpolador de dicha tabla de puntos al objeto de predecir el valor en otros puntos, sería una mala elección.

En ocasiones la matriz del sistema proveniente de las ecuaciones normales de Gauss $A^t A$, es una matriz que está mal condicionada. Para evitar los errores derivados de este problema podemos usar los

Comandos de Matlab:

- `p=polyfit(x,y,m)`

nos da los coeficientes, en sentido descendente, del polinomio p de grado $m < n$ que mejor aproxima, en el sentido de los mínimos cuadrados, al conjunto de puntos (x, y) donde $x = [x_1, x_2, \dots, x_n]$ e $y = [y_1, y_2, \dots, y_n]$. El polinomio obtenido puede evaluarse con el comando **polyval**.

A veces Matlab, al ejecutar el comando **polyfit**, recomienda normalizar las abscisas x_i , para que la media aritmética de todas ellas sea $\bar{x} = 0$ y la desviación típica $\sigma = 1$. Esto se consigue mediante la sintaxis

- `[p,s,mu]=polyfit(x,y,m)`

El polinomio p viene expresado en la nueva variable $z = \frac{x - \bar{x}}{\sigma}$. El segundo argumento de salida nos proporciona información sobre la matriz del sistema resultante y el tercer argumento es un vector con los valores de (\bar{x}, σ) , que son necesarios si queremos evaluar el polinomio p respecto de la variable inicial x .

En algunas ocasiones, la función propuesta como modelo no es lineal, pero podemos hallar un cambio de variables que linealice el problema y así poder aplicar el método de los mínimos cuadrados para encontrar dicha recta. Por ejemplo, si queremos ajustar una función exponencial del tipo $y = ce^{dx}$ a un conjunto de puntos $\{(x_k, y_k)\}_{k=1}^n$, tomando logaritmos en la expresión anterior obtenemos $\ln y = \ln c + dx$. Si hacemos el cambio de variables $X = x$, $Y = \ln y$ y denotamos $b = \ln c$ el problema consiste ahora en encontrar la recta $Y = dX + b$ que mejor aproxima, en el sentido de los mínimos cuadrados, a la nube de puntos $\{(x_k, \ln y_k)\}_{k=1}^n$. Una vez hallados los coeficientes d y b hay que deshacer el cambio para hallar el parámetro $c = e^b$.

En la referencia Mathews y Fink [2000,Cap 5] se pueden ver (figura 5.7) las gráficas de varios tipos de curvas que pueden usarse como modelo, dependiendo de los datos del problema, y una tabla (tabla 5.6) en la que se muestran algunos cambios de variable usados para linealizar.

Hacemos notar que la aproximación así obtenida es posiblemente distinta de la solución que se obtendría al plantear el problema de mínimos cuadrados para la función no lineal del modelo inicial. En efecto, el método de los mínimos cuadrados para la función $y = ce^{dx}$ consiste en hallar el mínimo de la función

$$E(c, d) = \sum_{k=1}^n (ce^{dx_k} - y_k)^2.$$

Al imponer que las derivadas parciales de la función E se anulen

$$\frac{\partial E}{\partial c} = 0, \quad \frac{\partial E}{\partial d} = 0,$$

aparecen las correspondientes ecuaciones normales. El sistema de ecuaciones resultante, a diferencia del caso anterior donde linealizabamos la función dada, es un sistema de ecuaciones no lineales que tendríamos que resolver mediante los métodos estudiados en el tema segundo.

De manera similar, mediante el procedimiento de los mínimos cuadrados, se resuelve el problema de aproximar una nube de puntos, $\{(x_k, y_k)\}_{k=1}^n$, por un polinomio de grado $m > 1$ (*ajuste polinomial*). No obstante si los datos no muestran una naturaleza polinomial, puede ocurrir que la curva resultante presente grandes oscilaciones conforme aumenta el grado del polinomio (*oscilación polinomial*), por ello no deben utilizarse polinomios de grado alto.

El problema del ajuste polinomial es un caso particular del problema de las combinaciones lineales en mínimos cuadrados. Es decir, dadas m funciones linealmente independientes, $\{f_1(x), f_2(x), \dots, f_m(x)\}$, y n puntos $\{(x_k, y_k)\}_{k=1}^n$, hallar m coeficientes $\{a_j\}_{j=1}^m$ para que la función $f(x)$, combinación lineal de las dadas, $f(x) = a_1 f_1(x) + a_2 f_2(x) + \dots + a_m f_m(x)$ minimice el error cuadrático

$$E^2 = \sum_{k=1}^n (f(x_k) - y_k)^2.$$

Imponiendo que la función error $E(a_1, a_2, \dots, a_m)$ tenga un mínimo, se obtiene un sistema cuadrado de m ecuaciones con m incógnitas que nos proporciona la solución al problema planteado. También se obtiene la misma solución si imponemos que $f(x_k) = y_k$, para $k = 1, \dots, n$, obteniéndose en este caso un sistema de n ecuaciones y m incógnitas que, si resulta ser incompatible, mediante las correspondientes ecuaciones normales nos permite obtener los coeficientes de la combinación lineal.

El siguiente programa calcula los coeficientes de la combinación lineal de las funciones linealmente independientes definidas en el conjunto S , que mejor se ajusta, en el sentido de los mínimos cuadrados, a la nube de puntos $\{(x_k, y_k)\}_{k=1}^n$.

```
function [C,A,B]=ajustefuncional(X,Y,S,m)
%
%      Datos de entrada
% X es el vector de las abscisas de orden 1xn
% Y es el vector de las ordenadas de orden 1xn
% S es el fichero donde estan definidas las funciones que intervienen
% en la combinacion lineal, introducido como una cadena de caracteres
% m es el numero de funciones que hay en S
%
%      Datos de salida
% C es un vector mx1 con los coeficientes de la combinacion lineal
% A es la matriz de los coeficientes para las ecuaciones normales
% B es el vector de los terminos independientes para las ecuaciones normales
%
n=length(X);
% en la matriz F se almacenan los valores que toman las funciones fi del
% subespacio S en los nodos X(k)
for k=1:n
    F(k,:)=feval(S,X(k),m);
end
```

```

% resolucion de las ecuaciones normales
format long
A=F'*F;
B=F'*Y';
C=A\B;

```

4.2. Series de Fourier y Polinomios Trigonométricos

Un caso particular del ajuste de curvas, mediante una combinación lineal de funciones, es el llamado polinomio trigonométrico de grado m .

Consideremos el conjunto de $2m + 1$ funciones

$$S = \left\{ \frac{1}{2}, \cos(x), \cos(2x), \dots, \cos(mx), \operatorname{sen}(x), \operatorname{sen}(2x), \dots, \operatorname{sen}(mx) \right\}.$$

Definición (polinomio trigonométrico)

Un polinomio trigonométrico de grado m es una función polinómica $P(x)$ obtenida mediante combinación lineal de las funciones del conjunto S , es decir

$$P_m(x) = \frac{a_0}{2} + \sum_{j=1}^m a_j \cos(jx) + b_j \operatorname{sen}(jx).$$

Dada una función $f(x)$ periódica de período 2π y el conjunto de puntos $\{(x_k, y_k)\}_{k=1}^n$, con $n \geq 2m + 1$, donde las abscisas están equiespaciadas en el intervalo $[-\pi, \pi]$

$$x_k = -\pi + \frac{2\pi k}{n}, \quad y_k = f(x_k), \quad k = 1, \dots, n$$

nos planteamos el problema de encontrar el polinomio trigonométrico que mejor se ajusta a la nube de puntos dada.

Teorema (serie de Fourier discreta de la función f)

Los coeficientes del polinomio trigonométrico $P_m(x)$ que mejor se ajustan, en el sentido de los mínimos cuadrados, al conjunto de puntos $\{(x_k, y_k)\}_{k=1}^n$ vienen dados por

$$a_j = \frac{2}{n} \sum_{k=1}^n f(x_k) \cos(jx_k), \quad j = 0, 1, \dots, m$$

y

$$b_j = \frac{2}{n} \sum_{k=1}^n f(x_k) \operatorname{sen}(jx_k), \quad j = 1, 2, \dots, m$$

Notas

(1) Si la función dada $f(x)$ es una función periódica de período l , entonces la nueva función

$$g(x) = f\left(\frac{lx}{2\pi}\right) \text{ es también periódica de período } 2\pi.$$

(2) Si la función $f(x)$ es impar ($f(-x) = -f(x)$ para todo x), entonces los coeficientes $a_j = 0$, para $j = 0, 1, \dots, m$. Si la función $f(x)$ es par ($f(-x) = f(x)$ para todo x), entonces los coeficientes $b_j = 0$, para $j = 1, 2, \dots, m$.

- (3) Si partimos de otro intervalo $[a, b]$ hay que hacer un cambio de variable lineal que nos lleve dicho intervalo al intervalo $[-\pi, \pi]$:

$$x \in [a, b] \rightarrow z(x) := \pi \left(\frac{2(x-a)}{b-a} - 1 \right) \in [-\pi, \pi],$$

entonces el nuevo conjunto de puntos es

$$\left\{ \left(z_k = z(x_k), f \left(a + \frac{(z_k + \pi)(b-a)}{2\pi} \right) \right) \right\}_{k=1}^n.$$

- (4) Si la función periódica $f(x)$, de período 2π , es continua a trozos en el intervalo $[-\pi, \pi]$ y hacemos tender $m \rightarrow \infty$ entonces aparece la *serie de Fourier* de la función $f(x)$

$$S(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} a_j \cos(jx) + b_j \operatorname{sen}(jx),$$

donde los coeficientes vienen dados por las fórmulas

$$a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(jx) dx, \quad j = 0, 1, \dots, \quad b_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \operatorname{sen}(jx) dx, \quad j = 1, 2, \dots$$

Referencias básicas: Burden y Faires [1985, Cap. 8]; Cordero, Hueso, Martínez y Torregrosa [2006, Cap 7]; Mathews y Fink [2000, Cap 5].

Ejercicios Resueltos Tema 4

PROBLEMA 4-1:

El número de habitantes censados en una determinada ciudad desde 1995 a 2004 se resume en la tabla siguiente:

Año	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004
Número	575873	576182	577001	574830	580940	579911	581920	582373	587278	571452

- (1) Hallar la recta de regresión y la matriz de las ecuaciones normales.
- (2) Ajustar dichos datos mediante un polinomio de grado tres, calculando el error cuadrático.
- (3) Obtener el polinomio anterior de grado tres con el comando *polyfit*, de dos maneras: a partir de los datos iniciales y normalizando las abscisas, comparándolos gráficamente con el obtenido en el apartado anterior. Estimar el número de habitantes en 2005.

SOLUCIÓN PROBLEMA 4-1:

Apartado (1)

Podemos utilizar el programa de ajustefuncional anterior para calcular los coeficientes de la recta y la matriz de las ecuaciones normales. En el fichero *S* metemos una base del subespacio vectorial de los polinomios de grado ≤ 1 , es decir $\{f_1(x) = 1, f_2(x) = x\}$.

```
function Y=S(X,m)
%
%      Datos de entrada
% X es un vector
% m es el numero de funciones que hay en el subespacio S
%
%      Datos de salida
% Y es un vector con el valor que toma cada una de las funciones del
% subespacio S en X
%
for j=1:m
    Y(j)=X.^(j-1);
end
```

Introducimos los vectores X, Y desde la ventana de comandos

```
>> X=1995:2004;
>> Y=[575873, 576182, 577001, 574830, 580940, 579911, 581920, 582373, 587278, 571452];
e invocamos al programa principal
>> [C,M]=ajustefuncional(X,Y,'S',2)
obteniéndose como respuesta
```

$$C \approx \begin{pmatrix} -451075,563636364 \\ 515,054545455 \end{pmatrix}.$$

Por tanto la recta pedida es

$$y = 515,054545455x - 451075,563636364$$

y la matriz del sistema

$$M = \begin{pmatrix} 10 & 19995 \\ 19995 & 39980085 \end{pmatrix}.$$

Dado que el vector C de los coeficientes está ordenado en sentido creciente de las potencias de x , necesitamos previamente invertirlo para poder evaluarlo posteriormente mediante el comando **polyval**, esto se consigue con el comando **flipud** por ser un vector columna. Si se trata de un vector fila el comando es **fliplr**.

```
>> P=flipud(C);
>> YR=polyval(P,X);
Finalmente representamos la recta de regresión junto a los datos de la tabla
>> plot(X,Y,'b*',X,YR,'r');
```

La gráfica resultante la podemos ver en la figura 4.1.

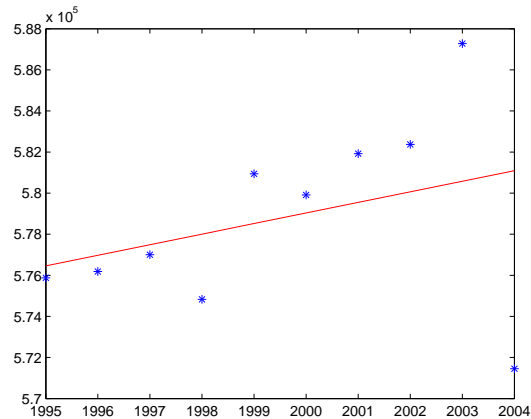


Figura 4.1: Gráfica de la recta de regresión (color rojo) y de los puntos de la tabla (color azul) en el intervalo [1995,2004].

Apartado (2)

En este caso el fichero S , con la base del subespacio vectorial de los polinomios de grado ≤ 3 , tendría como funciones $\{f_1(x) = 1, f_2(x) = x, f_3(x) = x^2, f_4(x) = x^3\}$.

Ahora al objeto de calcular el error cuadrático completamos el programa principal.

```
function [C,E]=ajustefuncionalerror(X,Y,S,m)
%
%      Datos de entrada
% X es el vector de las abscisas de orden 1xn
% Y es el vector de las ordenadas de orden 1xn
% S es el fichero donde están definidas las funciones que intervienen
% en la combinacion lineal, introducido como una cadena de caracteres
% m es el numero de funciones que hay en S
%
%      Datos de salida
% C es un vector mx1 con los coeficientes de la combinacion lineal
% E es el error cuadratico
%
n=length(X);
% en la matriz F se almacenan los valores que toman las funciones fi del
% subespacio S en los nodos X(k)
for k=1:n
    F(k,:)=feval(S,X(k),m);
end
% resolucioin de las ecuaciones normales
format long
A=F'*F;
B=F'*Y';
C=A\B;
```

```

% calculo del error cuadratico
s=0;
for j=1:n
    YE(j)=feval(S,X(j),m)*C;
    s=s+(YE(j)-Y(j)).^2;
end
E=sqrt(s);

```

A continuación llamamos al programa principal

```
>> [C,E]=ajustefuncionalerror(X,Y,'S',4)
```

Matlab nos avisa de que el sistema está mal condicionado, es casi singular, obteniéndose como respuesta

$$C \approx \begin{pmatrix} 5939208292,36221 \\ -9382130,84092 \\ 4928,01456 \\ -0,86080 \end{pmatrix},$$

$$E = 1,169115197437662e + 004.$$

El polinomio es

$$P3(x) = -0,86080x^3 + 4928,01456x^2 - 9382130,84092x + 5939208292,36221$$

Para representarlo gráficamente

```

>> P3=flipud(C);
>> X3=1995:0.1:2004;
>> YP3=polyval(P3,X3);
>> plot(X,Y,'b*',X,YR,'r',X3,YP3,'g');

```

La gráfica resultante la podemos ver en la figura 4.2.

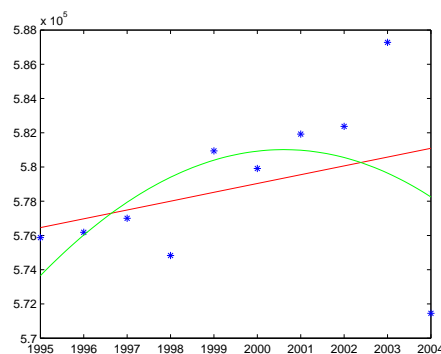


Figura 4.2: Gráfica de la recta de regresión (color rojo), del polinomio de grado 3 (color verde) y de los puntos de la tabla (color azul) en el intervalo [1995,2004].

Apartado (3)

Utilizando el comando *polyfit* se obtiene

```
>> p3=polyfit(X,Y,3);
```

Matlab nos avisa de que deberíamos normalizar las abscisas, obteniéndose el polinomio

$$p3(x) = -143,06296x^3 + 857927,73922x^2 - 1714953127,53494x + 1142699587571,780$$

Finalmente normalizamos las abscisas

```
>> [p3n,s,mu]=polyfit(X,Y,3);
```

obteniéndose el polinomio

$$p3n(x) = -3970,49173 \left(\frac{x - \bar{x}}{\sigma} \right)^3 - 2158,26389 \left(\frac{x - \bar{x}}{\sigma} \right)^2 + 7904,97276 \left(\frac{x - \bar{x}}{\sigma} \right) + 580718,43750$$

donde $\bar{x} = 1999,5$ y $\sigma = 3,02765035410$.

La representación gráfica del polinomio de grado tres calculado en el apartado anterior, y la de los polinomios, también de grado tres, obtenidos en este apartado la podemos ver en la figura 4.3.

```
>> yp3=polyval(p3,X3);
```

```
>> X3n=(X3-mu(1))./mu(2);
```

```
>> yp3n=polyval(p3n,X3n);
```

```
>> plot(X,Y,'b*',X3,YP3,'g',X3,yp3,'r+',X3,yp3n,'ko');
```

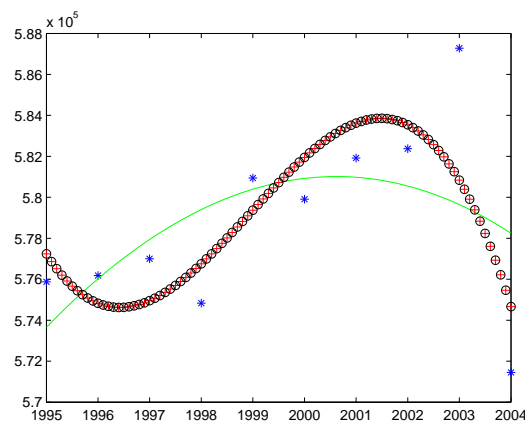


Figura 4.3: Gráfica de los puntos de la tabla (color azul), del polinomio de grado tres obtenido mediante las ecuaciones normales (color verde), del obtenido mediante el comando *polyfit* sin normalizar las abscisas (color rojo) y normalizando las abscisas (color negro) en el intervalo [1995,2004].

Como se observa no se distinguen los dos polinomios obtenidos mediante el comando *polyfit*, mientras que el polinomio calculado en el apartado (2) es claramente diferente debido a que el coeficiente de grado tres es muy pequeño en relación a los restantes coeficientes, cometándose errores de cancelación que hacen que se pierda el término de grado tres y su gráfica resulte la de una parábola en dicho intervalo.

Por último estimamos el número de habitantes en el año 2005, de seguir esa tendencia, usando el polinomio con las abscisas normalizadas.

```
>> xn=(2005-mu(1))/mu(2);  
>> polyval(p3n,xn)
```

obteniéndose $N \approx 564154$ habitantes, que coincide con el obtenido si utilizamos el polinomio sin normalizar las abscisas. Si utilizamos el polinomio del apartado anterior (2) se obtendría una peor previsión, $N \approx 576354$ habitantes.