

 UNIVERSIDAD DE OVIEDO DEPARTAMENTO DE MATEMÁTICAS	Asignatura	Cálculo Numérico	Página 1 de 7
	Tema	Aproximación y ajuste Mínimos cuadrados	
	Práctica	10	
	Autor	César Menéndez Fernández	

1 Mínimos cuadrados

Como se ha visto en teoría, el ajuste por mínimos cuadrados de un conjunto de valores

$\{(x_i, y_i)\}_{i=1}^m$ mediante una función $y(x) = \sum_{k=1}^N c_k \varphi_k(x)$ donde $\{\varphi_1(x), \varphi_2(x), \varphi_3(x), \dots, \varphi_N(x)\}$

forman una base de un espacio de funciones, conducen al sistema

$$\begin{pmatrix} \sum_{k=1}^m \varphi_1(x_k) \varphi_1(x_k) & \sum_{k=1}^m \varphi_1(x_k) \varphi_2(x_k) & \cdots & \sum_{k=1}^m \varphi_1(x_k) \varphi_N(x_k) \\ \sum_{k=1}^m \varphi_2(x_k) \varphi_1(x_k) & \sum_{k=1}^m \varphi_2(x_k) \varphi_2(x_k) & \cdots & \sum_{k=1}^m \varphi_2(x_k) \varphi_N(x_k) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=1}^m \varphi_N(x_k) \varphi_1(x_k) & \sum_{k=1}^m \varphi_N(x_k) \varphi_2(x_k) & \cdots & \sum_{k=1}^m \varphi_N(x_k) \varphi_N(x_k) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^m \varphi_1(x_k) y_k \\ \sum_{k=1}^m \varphi_2(x_k) y_k \\ \vdots \\ \sum_{k=1}^m \varphi_N(x_k) y_k \end{pmatrix} \Leftrightarrow \mathbf{A} \mathbf{c} = \mathbf{b}$$

Donde la matriz del sistema y el segundo miembro se pueden poner como

$$\mathbf{A} = \mathbf{X}^t \cdot \mathbf{X} \quad \mathbf{b} = \mathbf{X}^t \mathbf{Y} \quad \text{donde} \quad \mathbf{X} = \begin{pmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \varphi_3(x_1) & \cdots & \varphi_N(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \varphi_3(x_2) & \cdots & \varphi_N(x_2) \\ \varphi_1(x_3) & \varphi_2(x_3) & \varphi_3(x_3) & \cdots & \varphi_N(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_m) & \varphi_2(x_m) & \varphi_3(x_m) & \cdots & \varphi_N(x_m) \end{pmatrix} \quad \mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix}$$

1.1 Base genérica

Se puede aplicar directamente lo visto en teoría. Por ejemplo, ajustar un conjunto de datos a la

función: $y(x) = c_1 + c_2 x^2 + c_3 \sin(x) + c_4 \log(x)$ y representar los datos y la función.

Comenzamos planteando y resolviendo el sistema lineal para obtener los coeficientes

```

» x=[0.1600,0.9700,0.9600,0.4900,0.8000,0.1400,0.4200,0.9200,0.7900,0.9600];
» y=[0.5208,0.7583,0.7396,0.3414,0.5011,0.5368,0.3549,0.6693,0.4899,0.7396];
» f1=inline('ones(size(x))');
» f2=inline('x.^2');
» f3=inline('sin(x)');
» f4=inline('log10(x)');
» X=[f1(x);f2(x);f3(x);f4(x)]';
» c=(X'*X)\(X'*y')
c =
    0.9996
    1.4998
   -1.9993
    0.2497

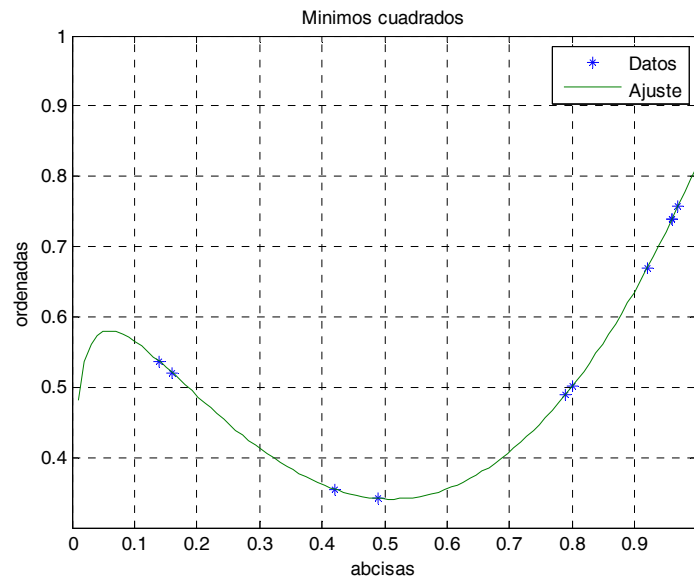
```

Calculamos el error cuadrático

```
» norm(c(1)*f1(x)+c(2)*f2(x)+c(3)*f3(x)+c(4)*f4(x)-y)
ans =
    9.5466e-005
```

Y finalmente se generan los puntos de representación y la figura

```
» xx=linspace(0,1);
» yy=c(1)*f1(xx)+c(2)*f2(xx)+c(3)*f3(xx)+c(4)*f4(xx);
» plot(x,y,'*',xx,yy);grid on;
» xlabel('abscisas');ylabel('ordenadas');
» legend('Datos','Ajuste')
» title('Mínimos cuadrados')
```



Ejercicios recomendados:

1. Los siguientes datos recogen el censo asturiano en los años indicados¹

Año: {1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008}

Censo={1112186, 1115016, 1123301, 1125419, 1128372, 1093937, 1099296, 1112415, 1117762, 1117370, 1087885, 1081834, 1084314, 1076567, 1075329, 1073971, 1075381, 1073761, 1076635, 1076896, 1074862, 1080138}

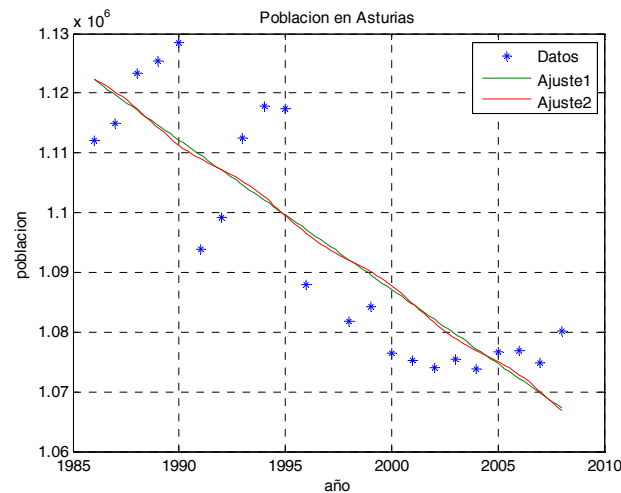
Ajustar los datos con los siguientes tipos de curvas y calcular su error cuadrático

$$y_1 = c_1 + \frac{c_2}{x} \quad y_2 = c_3 + c_4 x + c_5 \sin(x)$$

Solución:

$c_1 = -3.8921e+006$, $c_2 = 9.9585e+009$, $c_3 = 6.0705e+006$, $c_4 = -2.4917e+003$, $c_5 = 7.1850e+002$,
 $EC_1 = 4.6891e+004$, $EC_2 = 2.3342e+007$

¹ <http://www.ine.es/inebmenu/indice.htm#6d>



1.2 Base polinómica

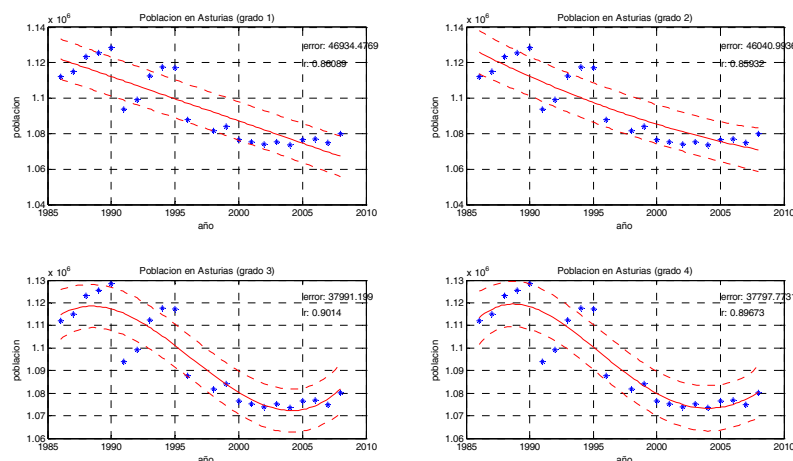
Cuando la base es polinómica, no es necesario realizar los pasos anteriores, ya que MATLAB dispone de la instrucción *polyfit* que realiza simultáneamente el cálculo de los coeficientes y el error. Para evaluarla en un punto determinado, se emplea *polyval*.

Ejemplo:

- Realizar el ajuste de los datos del censo asturiano utilizando polinomios de primer, segundo, tercer y cuarto grado, y representar las funciones así como el margen del ajuste del 50%.

```

vary=var(y);N=length(x);
for i=1:4;
    [p,S,mu] = polyfit(x,y,i);[yy,delta] = polyval(p,xx,S,mu);
    subplot(2,2,i);plot(x,y,'*',xx,yy,'r',xx,yy-delta,'r:',xx,yy+delta,'r:');
    grid on;xlabel('año');ylabel('poblacion');
    title(['Poblacion en Asturias (grado ',num2str(i),')']);
    text(0.8,0.9,['error: ',num2str(S.normr)],'Units','Normalized');
    r2=1-S.normr.^2/vary/(N-i);
    text(0.8,0.8,['r: ',num2str(sqrt(r2))],'Units','Normalized');
end
  
```



1.3 Ajuste no lineal

El ajuste por mínimos cuadrados exige que la función de ajuste se pueda poner como combinación lineal de funciones de base, por lo que cuando esto no sucede es necesario, cuando sea posible, linealizar la función. A continuación se muestran como ejemplos los ya vistos en teoría:

$$y = ax^b \rightarrow \ln(y) = \ln(ax^b) = \ln(a) + b\ln(x) \Rightarrow \hat{y} = A + b\hat{x}$$

```
» p = polyfit(log(x),log(y),1)
» b=p(1),a=exp(p(2))
» yy = a*xx.^b; % dos formas de evaluar la función de ajuste
» yy = exp(polyval(p,log(xx)));
```

$$y = ae^{bx} \rightarrow \ln(y) = \ln(ae^{bx}) = \ln(a) + bx \Rightarrow \hat{y} = A + bx$$

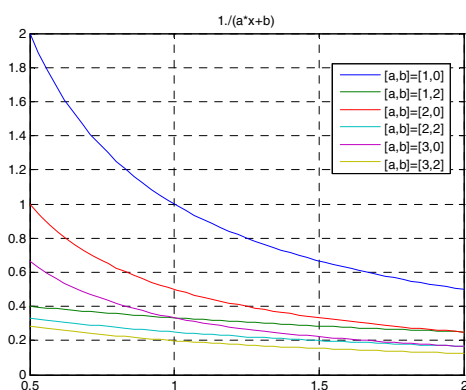
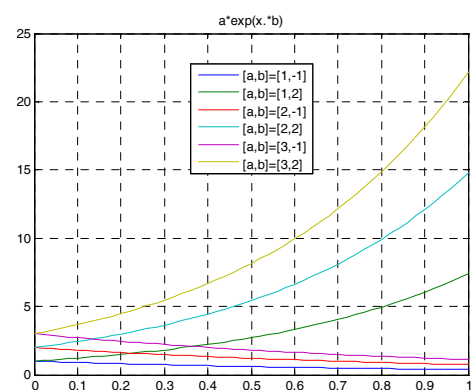
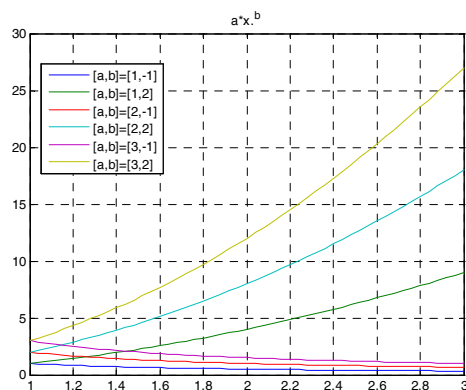
```
» p = polyfit(x,log(y),1)
» b=p(1),a=exp(p(2))
» yy = a*exp(xx*b); % dos formas de evaluar la función de ajuste
» yy = exp(polyval(p,xx));
```

$$y = \frac{x}{ax+b} \rightarrow \frac{1}{y} = \frac{ax+b}{x} = a + b\frac{1}{x} \Rightarrow \hat{y} = a + b\hat{x}$$

```
» p = polyfit(1./x,1./y,1)
» yy = xx./(p(2)*xx+p(1)); % dos formas de evaluar la función de ajuste
» yy = 1./polyval(p,1./xx);
```

Ejercicios recomendados:

3. Representar las funciones de ajuste para diferentes valores de a y b.



4. El fichero IG_pr10D1.txt contiene los valores de la edad y su correspondiente esperanza de vida. Leer el fichero y ajustar mediante la función no lineal más adecuada de las vistas, calculando para cada una su error cuadrático

1.4 Ajuste multimodal

Se puede extender la teoría vista al ajuste de funciones de varias variables, siempre que sea posible expresar la función de ajuste de forma lineal en los parámetros.

$$y(\mathbf{x}) = \sum_{i=1}^n a_i \varphi_i(x_i) + \sum_{i=1}^n \sum_{j=1}^n b_{ij} \varphi_{ij}(x_i, x_j) + \dots$$

Ejercicios recomendados:

5. El fichero IG_pr10D2.txt contiene los valores de la edad y su correspondiente esperanza de vida para varios años. Leer el fichero y ajustar mediante un ajuste multimodal de la forma $y(\mathbf{x}) = a_0 + a_1 x_1 + a_2 x_2 + a_{11} x_1^2 + a_{12} x_1 x_2 + a_{22} x_2^2$, calculando su error cuadrático

2 Anexo: Instrucciones necesarias

2.1 Ajuste polinómico

`p = polyfit(x,y,n)`

`[p,S] = polyfit(x,y,n)`

`[p,S,mu] = polyfit(x,y,n)`

`p = polyfit(x,y,n)` calcula los coeficientes del polinomio $p(x)$ de grado n que ajusta los valores (x_i, y_i) por mínimos cuadrados. El resultado es un vector fila con los n coeficientes del polinomio en orden de potencias decrecientes.

`[p,S] = polyfit(x,y,n)` calcula los coeficientes del polinomio y la estructura S que al usar con `polyval` permite obtener una estimación del error de predicción. La estructura S contiene los campos R , df y $normr$ que almacenan la matriz que realiza la factorización QR de la matriz de Vandermonde X , los grados de libertad y la norma del residuo respectivamente. Si los el error de los datos es ruido blanco, `polyval` produce límites de error que deben contener al menos el 50% de las predicciones.

`[p,S,mu] = polyfit(x,y,n)` normaliza previamente los datos, almacenando en μ los valores de la media y la desviación típica. Esta transformación mejora los resultados tanto del polinomio de interpolación como de su evaluación. En este caso es necesario transformar de igual modo los valores antes de evaluarlos, es decir, mediante

$$\bar{x}_k = \frac{x_k - \mu_1}{\mu_2}$$

`y = polyval(p,x)`

`y = polyval(p,x,[],mu)`

`[y,ey] = polyval(p,x,S)`

`[y,ey] = polyval(p,x,S,mu)`

`y = polyval(p,x)` evalúa el polinomio $p(x)$ en x .

`y = polyval(p,x,[],mu)` uses $\bar{x}_k = \frac{x_k - \mu_1}{\mu_2}$ in place of x .

`[y,ey] = polyval(p,x,S)` and `[y,ey] = polyval(p,x,S,mu)` utilize la estructura S para generar una banda $y \pm ey$ que contiene al menos el 50% de las predicciones si los errores son ruido blanco

2.2 Aproximación en general

Función	Toolbox	Notas
<code>lscurvefit</code>	Optimization	Resuelve por mínimos cuadrados el ajuste no lineal de una función
<code>lsqlin</code>	Optimization	Resuelve por mínimos cuadrados un problema de optimización lineal con restricciones
<code>lsqnonlin</code>	Optimization	Resuelve un sistema no lineal mediante minimización por mínimos cuadrados del funcional asociado
<code>lsqnonneg</code>	Optimization	Resuelve por mínimos cuadrados un problema lineal con restricciones no negativas
<code>tnlinfir</code>	Statistic	Regresión no lineal

2.3 Lectura de ficheros

Función	Datos	Delimitadores	Notas
<u>csvread</u>	Numéricos	comas	Usado fundamentalmente con hojas de cálculo
<u>dlmread</u>	Numéricos	Cualquier carácter	Flexible y fácil de usar
<u>fclose</u>	No		Cierre de un fichero
<u>fopen</u>	No		Apertura de un fichero para su posterior lectura
<u>fread</u>	Numéricos		Sólo válido con ficheros binarios
<u>fscanf</u>	Alfabéticos y numéricos, pero se devuelve en una única variable	Cualquier carácter	Flexible pero exige una programación compleja
<u>load</u>	Numéricos	Espacio, tabulador, comma o punto y coma	Fácil de usar y admite comentarios de Matlab
<u>textscan</u>	Alfabéticos y numéricos	Cualquier carácter	Flexible, potente y fácil de usar. Puede haber cabeceras y comentarios

También se pueden importar datos desde el entorno de Matlab, mediante:

- la selección de *Import Data* del menú *File*
- ejecutando `uiimport -file`, y pulsando **Enter**
- ejecutando `uiimport`, pulsando **Enter**, and seleccionado **File** en el diálogo **Import Data**.