Proyecto 1

Análisis aplicado

Mario Humberto Becerra 124362
José Carlos Castro 127049
José Manuel Incera 125360
Rodrigo Andrés Morales Mendoza 124341

Profesor: Dr Zeferino Parada

30 de Septiembre de 2015

Introducción

El modelo logístico de crecimiento está determinado por la ecuación diferencial

$$\frac{dP}{dt} = rP(t)(1 - \frac{P(t)}{K}) \tag{1}$$

donde $P(t)$ es la población al tiempo t, r es la tasa de crecimiento y K es una constante con la cantidad máxima permitida de la población. A continuación usaremos indistitamente $P(t) = N(t)$ como notación para la poblacion en el tiempo t.

Para resolver esta ecuación diferencial, considérese el cambio de variable $X = 1/N$. Entonces si se deriva X se tiene

$$\dot{X} = \frac{dx}{dt} = \frac{dx}{dN}\frac{dN}{dt} = -N^{-2}\dot{N} = \frac{\dot{N}}{-N^2} \tag{2}$$

Si se despeja $\dot{N}$ de (2) entonces tenemos $\dot{N} = \frac{-\dot{X}}{X^2}$, de donde:

$$\begin{aligned}
-\dot{X}\frac{1}{X^2} &= r\frac{1}{X}(1 - \frac{1}{Xk}) \\
\Rightarrow \dot{X} &= -r\frac{X^2}{X}(\frac{XK-1}{XK}) \\
&= -r\frac{X^2}{KX^2}(XK-1) \\
&= \frac{-r}{K}(XK-1) \\
&= -rX + \frac{r}{K}
\end{aligned}$$

Primero resolvemos para X_h con lo que $\dot{X} = -rX \Rightarrow X = ce^{-rt}$

$$\begin{aligned}
\int \frac{dx}{x} &= \int -r dt \\
\ln x &= -rt + k \\
x_h &= ce^{-rt}
\end{aligned}$$

Segunda solución particular

$$\begin{aligned}
x(t) &= A_1 t + A_0 \\
\Rightarrow x^{'}(t) &= A_1 \\
\Rightarrow A_1 &= -r(A_1 t + A_0) + \frac{r}{k} \\
\therefore A_1 + rA_1 t + rA_0 + \frac{r}{k} = 0 &\iff A_1 = 0,\ A_0 = \frac{1}{k}
\end{aligned}$$

Solución general

$$\begin{aligned}
x(t) &= ce^{-rt} + \frac{1}{k};\ N = \frac{1}{x} \\
\therefore N(t) &= \frac{1}{ce^{-rt} + \frac{1}{k}}
\end{aligned}$$

finalmente, despejamos a c para obtener su valor

$$\begin{aligned}
(ce^{-rt} + \frac{1}{k})N(t) &= 1 \\
\Rightarrow c &= (\frac{1}{N(t)} - \frac{1}{k})e^{rt}
\end{aligned}$$

Por lo que junto con la condición inicial $N(t_0) = P(t_0) = P_0$ se tiene:

$$\begin{aligned}
c &= e^{r0}(\frac{1}{P_0} - \frac{1}{k}) \\
\Rightarrow c &= (\frac{1}{P_0} - \frac{1}{k}) \\
\therefore P(t) &= \frac{k}{1 + (\frac{k}{P_0} - 1)e^{-rt}}
\end{aligned}$$

Modelación de ventas de *iPad*

La ecuación (1) puede usarse para explicar la adaptación a nuevas tecnologías, donde $P(t)$ es la población que ha aceptado la innovación al tiempo t y donde r es la tasa de aceptación. Este tipo de modelos se usan porque tiene sentido pensar que en un principio la venta de un producto nuevo va a ser lenta, y a medida que pasa el tiempo más gente conoce y se familiariza con el producto, por lo que la pendiente de la venta aumenta; pero también asume que este crecimiento no va a seguir por siempre, sino que debe llegar a un límite, la K.

Supongamos que las ventas siguen un modelo logístico (1) y que se desean conocer los parámetros r, K y P_0 (suponemos que el primer valor puede tener errores).

Sean $x = (r, K, P_0)^T$ los parámetros a determinar y (t_k, v_k) las ventas en el trimestre k-ésimo. Defina la función residual k-ésima

$$r_j(x) = P(t_j) - v_j,\ j = 1, 2, \ldots, 16. \tag{3}$$

Entonces el problema es minimizar la función

$$f(x) = \frac{1}{2} \sum_{j=1}^{16} r_j(x)^2. \tag{4}$$

En el presente trabajo se minimiza la función (4) usando el algoritmo de Levenberg–Marquardt, diseñado específicamente para problemas de mínimos cuadrados no lineales.

Agricultura

La producción de trigo durante 24 años consecutivos por hectárea de cultivo está determinada por los datos:

11.72, 13.38, 14.10, 13.87, 14.80, 15.68, 14.36, 16.30, 16.91, 18.16, 18.43, 18.70, 20.46 19.16, 20.02, 22.41, 21.21, 22.81, 23.97, 23.27, 23.80, 25.59, 24.93, 26.59.

Suponiendo que la producción sigue el modelo logístico, se determinan los parámetros que minimizan la función de mínimos cuadrados (4) utilizando el mismo método que para el problemas de las *iPad*.

Este modelo se utiliza por las mismas razones que se utiliza en la venta de *iPads*: la producción en un principio es lenta, después de un tiempo aumenta, pero tampoco puede aumentar por siempre porque los recursos son limitados.

Por otro lado, la producción depende de la fertilidad de la tierra, que entre otros factores se deriva de los nutrientes de la misma, entonces a medida que estos nutrientes se absorven la producción crece, sin embargo a medida que se acerca a la capacidad máxima de producción por unidad de area, la producción depende cada vez menos de la fertilidad de la tierra y es por ello que la tasa de crecimiento cae significativamente, similarmente que con la función logística.

Algoritmo de Levenberg–Marquardt

El algoritmo de Levenberg-Marquardt (LM) es un método que se utiliza para resolver problemas de mínimos cuadrados no lineales, como el del problema que se presenta en este trabajo. En general, en un problema de mínimos cuadrados, la función objetivo f es de la forma:

$$f(x) = \frac{1}{2} \sum_{j_1}^{m} r_j^2(x),$$

donde cada r_j es una función suave de $\mathbb{R}^n$ en $\mathbb{R}$, conocido como el residual. Se asume que $n \leq m$.

La forma que toma la función objetivo en problemas de mínimos cuadrados la hace más fácil de resolver que en problemas de optimización generales. Para probar esto, consideremos el vector de residuales $r : \mathbb{R}^n \to \mathbb{R}^m$ como:

$$r(x) = (r_1(x), \ldots, r_m(x))^T.$$

Utilizando esta notación, se puede reescribir f como $f(x) = \dfrac{1}{2} \|r(x)\|_2^2$. Las derivadas de $r(x)$ pueden ser expresadas en términos de la Jacobiana $J(x)$, definida como

$$J(x) = \left[\frac{\partial r_j}{\partial x_i}\right]_{j=1,\ldots,m \;\; i=1,\ldots,n} = \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix}$$

donde cada $\nabla r_j(x)$, $j = 1, \ldots, m$ es el gradiente de r_j.

El gradiente de f se puede expresar como

$$\nabla f(x) = \sum_{j=1}^{m} r_j(x) \nabla r_j(x) = J(x)^T r(x),$$

mientras que la Hessiana de f es

$$\nabla^2 f(x) = \sum_{j=1}^{m} \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) = J(x)^T J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x). \tag{5}$$

Nótese que la Hessiana se puede aproximar mediante el producto matricial $J(x)^T J(x)$ sin necesidad de evaluar segundas derivadas. Esta aproximación se puede hacer porque el término $J(x)^T J(x)$ es más importante que el segundo término de la suma de la Hessiana en (5), ya sea porque los residuales r_j son cerca a ser afines cerca de la solución (i.e., $\nabla^2 r_j(x)$ son pequeños) o porque los residuales son cercanos a cero (i.e., $r_j(x)$ son pequeños). El algoritmo LM explota esta propiedades estructurales de la Hessiana y lo hacen muy eficiente.

Desarollo

En esta clase hemos estudiado dos familias de métodos para resolver problemas de optimización sin restricciones. El ingrediente que las distingue es el mecanismo que utilizan para garantizar convergencia global. Específicamente nos referimos a los métodos basados en búsquedas lineales y a los que utilizan regiones de confianza.

En general nos basamos en un método de región de confianza para atacar este problema.

El método inicial que nos permitió comenzar con el desarollo fue el método visto en clase, de doblez. A partir del cual calculamos las direcciones de Cauchy y Newton las cuales nos permiten obtener el vector p de salida con la solución aproximada a la región de confianza.

Obteniendo la solución, pudimos pasar al desarollo de nuestro problema principal, que fue, como se describió en la sección anterior, resolver el problema de mínimos cuadrados no lineales. Para esto utilizamos la función Gauss Newton que describiremos a continuación.

Para empezar calculamos la Jacobiana y el vector de residuales, evaluados en el punto incial dados, al igual que el gradiente y el valor de la función en dicho punto.

```
Jx = jacob(fname, x);
rx = feval(fname, x);
grad = Jx’ * rx;
fx = (rx’ * rx)/2;
```

Para iniciar, calculamos la aproximación de la Hessiana, sin necesidad de calcular segundas derivadas, con ayuda del producto matricial de las Jacobianas. Con esta matriz calculamos sus eigenvalores y nos aseguramos que todos sean positivos agregandolos a la Hessiana en valor absoluto.

```
B = Jx’ * Jx;
b = eig(B);
    if (sum( b < 0 ) > 0)
        B = B + (1 + abs(min(b))) * eye(n);
    end
```

El siguiente paso consiste en obtener la dirección de descenso con la ayuda de la Hessiana calculada posteriormente. Para esto utilizamos la función de doblez que comentamos anteriormente. Al obtener la dirección con éste método, verificamos que se cumplan las condiciones Armijo, en caso de que no se cumplan se tiene que actualizar nuestra delta inicial y reiniciar el método hasta que la delta coincida con su cota menor o que se cumplan las condiciones de Armijo

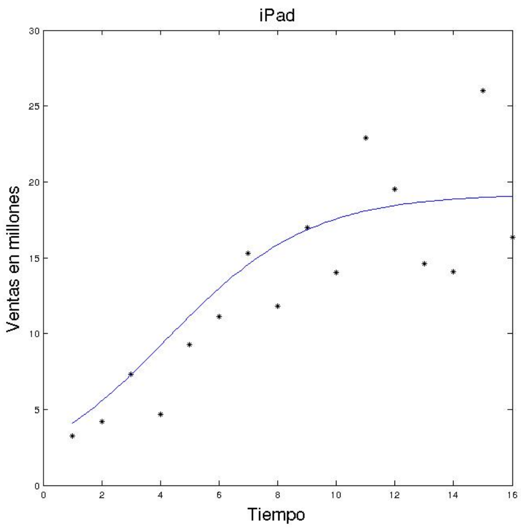
```
p_temp = Doblez(B, grad, delta);
r_x_p = feval(fname, x + p_temp);
c = (rx' * rx) + c1 * grad' * p_temp;
if( r_x_p' * r_x_p  <= c )
    p = p_temp;
    flag = 1;
else
    delta = max(delta/2, delta_lo);
        if(delta == delta_lo)
            p = p_temp;
            flag = 1;
        end
end
```

Finalmente, se actualizan los datos y se corre de nuevo el algoritmo para otra iteración.

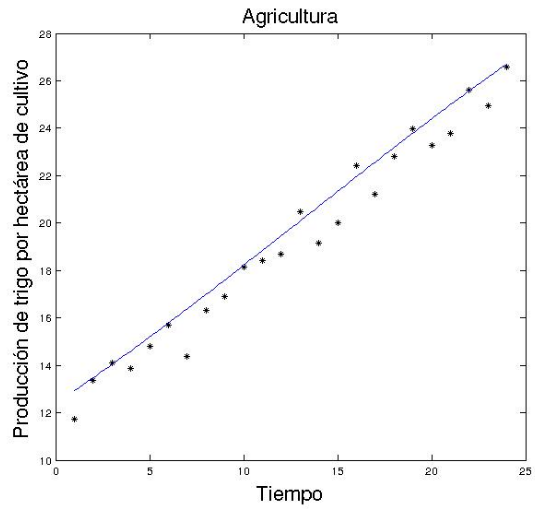
```
mc = 0.5 * p' * B * p + grad' * p + fx;
f_x_p = feval(fname, x + p);
rho = (f_x_p' * f_x_p - fx) / (mc - fx);
if (rho < 1.25 & rho > 0.75)
    delta = min(2 * delta, delta_hi);
end

% Actualizar variables
x = x + p;    rx = feval(fname, x);    Jx = jacob(fname, x);    grad = Jx' * rx;
fx = (rx' * rx) / 2;
iter = iter+1;
```

Para terminar y mostrar los resultados graficamos nuestro método aplicado tanto a los datos del problema de Ipads como al caso de Trigo.

```
clear;
global VECTOR_DE_DATOS;
VECTOR_DE_DATOS = [3.27, 4.19, 7.33, 4.69, 9.25, 11.12, 15.3, 11.80........];
n = length(VECTOR_DE_DATOS);
x0 = [0.01 500 3]';
t = (1:n)';
tic
[X, iter] = GaussNewton('residuales',  1.e-06, 10000, x0);
toc

disp('La solución final para el problema de iPads es:')
disp(strcat('r=', num2str(X(1))));
disp(strcat('K=', num2str(X(2))));
disp(strcat('P0=', num2str(X(3))));

time = linspace(1,n,100);
fit = func_log(X(1),X(2),X(3),time);

% Plots
figure(12)
```

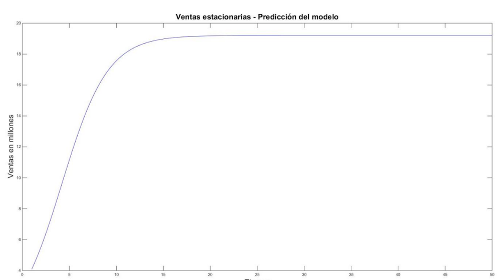


(a) *iPad*

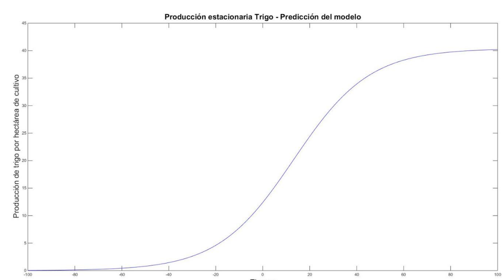


(b) Trigo

Figura 1: Datos y modelo ajustado para los datos de las ventas de *iPad* y de trigo



(a) *iPad*



(b) Trigo

Figura 2: Comportamiento en el largo plazo según la predicción del modelo en cada caso

```
plot(t,VECTOR_DE_DATOS,'*k')
title('iPad','FontSize',18)
xlabel('Tiempo','FontSize',18)
ylabel('Ventas en millones','FontSize',18)
hold on;
plot(time,fit,'-b')
hold on
hold off;
```

Resultados

Venta de *iPads*

La solución final para el problema de las *iPads* es el vector

$$(r, K, P_0)^T = (0.408, 19.21, 2.92)^T$$

La gráfica del modelo se puede ver en la figura 1.

Producción de trigo

La solución final para el problema de la producción de trigo es el vector

$$(r, K, P_0)^T = (0.061, 40.39, 12.39)^T$$

La gráfica del modelo se puede ver en la figura 1.

La figura 2 muestra el comportamiento predicho por el modelo en el largo plazo para ambas: venta de iPad y producción de trigo, cuyo punto estacionario es el K, que resulta sendamente en 19.21 y 40.39.

Referencias

[1] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization, second edition.* World Scientific, 2006.