

Regresión

Lección 16

Dr. Pablo Alvarado Moya

CE3102 Análisis Numérico para Ingeniería
Área de Ingeniería en Computadores
Tecnológico de Costa Rica

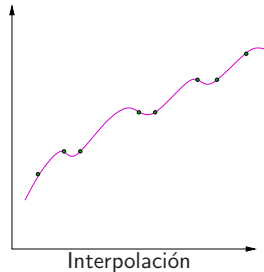
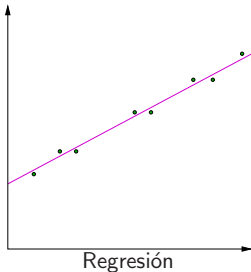
I Semestre 2018

Contenido

- 1 Introducción
- 2 Regresión Lineal
- 3 Mínimos cuadrados
 - Linealización
- 4 Regresión polinomial

Regresión

- **Interpolación** encuentra función que “pasa” por un número dado de puntos
- **Regresión** ajusta una función o modelo previamente conocido a un número de datos.
- Idea básica es minimizar distancia entre los puntos y el modelo.



Regresión Lineal

- Sea una línea recta con modelo

$$y_i = a_0 + a_1 x_i + e_i$$

con e_i el error entre el modelo y la i -ésima observación.

- Se ajustará dicho modelo al conjunto de observaciones (x_i, y_i)
 $i = 1 \dots n$
- Así, se minimizará el error por medio de **mínimos cuadrados**

$$S_r = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - (a_0 + a_1 x_i))^2$$

- Este criterio es derivable, lo que permite su minimización analítica.

Ajuste de línea por mínimos cuadrados

(1)

- Derivando el error con respecto a a_0 e igualando a cero:

$$\begin{aligned}\frac{\partial S_r}{\partial a_0} &= -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i) = 0 \\ \sum_{i=1}^n y_i - \sum_{i=1}^n a_0 - \sum_{i=1}^n a_1 x_i &= 0 \\ na_0 + \left(\sum_{i=1}^n x_i \right) a_1 &= \sum_{i=1}^n y_i\end{aligned}$$

Ajuste de línea por mínimos cuadrados

(2)

- Derivando el error con respecto a a_1 e igualando a cero:

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i) x_i = 0$$

$$\sum_{i=1}^n y_i x_i - \sum_{i=1}^n a_0 x_i - \sum_{i=1}^n a_1 x_i^2 = 0$$

$$\left(\sum_{i=1}^n x_i \right) a_0 + \left(\sum_{i=1}^n x_i^2 \right) a_1 = \sum_{i=1}^n x_i y_i$$

Ajuste de línea por mínimos cuadrados

(3)

- Las ecuaciones anteriores reciben el nombre de **ecuaciones normales**:

$$\begin{aligned} na_0 + \left(\sum_{i=1}^n x_i \right) a_1 &= \sum_{i=1}^n y_i \\ \left(\sum_{i=1}^n x_i \right) a_0 + \left(\sum_{i=1}^n x_i^2 \right) a_1 &= \sum_{i=1}^n x_i y_i \end{aligned}$$

o en forma matricial

$$\begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}$$

Ajuste de línea por mínimos cuadrados

(4)

- El sistema de dos ecuaciones con dos incógnitas se resuelve con:

$$a_1 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}$$
$$a_0 = \frac{1}{n} \left(\sum_{i=1}^n y_i - a_1 \sum_{i=1}^n x_i \right)$$

que minimiza la varianza del error.

Medición del error

(1)

- El error estándar de la estimación de línea de regresión es:

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}} \quad S_r = \sum_{i=1}^n e_i^2 \quad e_i = (y_i - (a_0 + a_1 x_i))$$

que cuantifica la dispersión de los datos alrededor de la línea de regresión.

Medición del error

(2)

- El coeficiente de determinación r^2 se define como

$$r^2 = \frac{S_t - S_r}{S_t}$$

donde S_t es la varianza los datos

$$S_t = \sum (y_i - \bar{y})^2 \qquad \bar{y} = \frac{1}{n} \sum y_i$$

y S_r la suma de cuadrados de residuos descrita anteriormente.

- $r = \sqrt{r^2}$ es el coeficiente de correlación
- r^2 cuantifica la mejora o reducción del error entre describir los datos por su media y utilizar la línea de regresión.

Linealización de relaciones no lineales

- Método anterior asume dependencia lineal entre las variables independientes x_i y el valor de las funciones $f(x_i)$
- Si relación lineal no describe correctamente la relación, hay dos opciones:
 - 1 Regresión con otros modelos (p.ej. regresión polinomial)
 - 2 Transformar los datos para aplicar regresión lineal

Linealización de exponencial

- Si los datos siguen la tendencia exponencial

$$y = \alpha e^{\beta x}$$

se aplica el logaritmo natural a ambos lados para obtener

$$\ln y = \ln \alpha + \beta x$$

- La gráfica de $\ln y$ contra x es una línea recta de pendiente β y corte de ordenadas en $\ln \alpha$.
- Ejemplos: crecimiento poblacional, decaimiento radiactivo

Linealización de ecuación de potencias

- Si los datos siguen la tendencia de potencias

$$y = \alpha x^{\beta}$$

se aplica el logaritmo natural a ambos lados para obtener

$$\ln y = \ln \alpha + \beta \ln x$$

- La gráfica de $\ln y$ contra $\ln x$ es una línea recta de pendiente β y corte de ordenadas en $\ln \alpha$.

Linealización de ecuación de razón de crecimiento

- Si los datos siguen la tendencia de razón de crecimiento

$$y = \alpha \frac{x}{\beta + x}$$

se invierte a ambos lados

$$\frac{1}{y} = \frac{\beta}{\alpha} \frac{1}{x} + \frac{1}{\alpha}$$

- La gráfica de $1/y$ contra $1/x$ es una línea recta de pendiente β/α y corte de ordenadas en $1/\alpha$.

Suposiciones sobre regresión lineal

- Los valores x no son aleatorios y se conocen sin error
- El error en la medición de y tiene distribución normal y es independiente de x .
- En aplicaciones prácticas hay otros aspectos a considerar en la regresión lineal, relacionados con los *outliers* (valores atípicos)

Regresión curvilínea

- Anteriormente se ajustaron datos no lineales por medio de su transformación
- Otra alternativa es ajustar polinomios de grado conocido a los datos
- Procedimiento de mínimos cuadrados se extiende a polinomios de grado superior a 1.

Ajuste cuadrático

(1)

- Asíumase que se va a ajustar un polinomio de segundo grado

$$y = a_0 + a_1x + a_2x^2 + e$$

- El error es

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2)^2$$

- ¿Cuáles son las derivadas respecto a a_0 , a_1 y a_2 ?

Ajuste cuadrático

(2)

- Derivando con respecto a cada coeficiente

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^n x_i (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum_{i=1}^n x_i^2 (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

Planteamiento de sistema de ecuaciones

- Igualando las derivadas a cero y reagrupando se obtiene

$$\begin{aligned} na_0 + (\sum x_i) a_1 + (\sum x_i^2) a_2 &= \sum y_i \\ (\sum x_i) a_0 + (\sum x_i^2) a_1 + (\sum x_i^3) a_2 &= \sum x_i y_i \\ (\sum x_i^2) a_0 + (\sum x_i^3) a_1 + (\sum x_i^4) a_2 &= \sum x_i^2 y_i \end{aligned}$$

que es un sistema de tres ecuaciones con tres incógnitas de la forma $\mathbf{Ax} = \mathbf{b}$.

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

- La matriz **A** se denomina **matriz normal**

Generalización

- Para hacer la regresión a un polinomio de orden m se planteará un sistema de $m + 1$ incógnitas con $m + 1$ ecuaciones.
- La sistema de ecuaciones normales tiene la forma

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^m \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{m+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots & \sum x_i^{m+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum x_i^m & \sum x_i^{m+1} & \sum x_i^{m+2} & \cdots & \sum x_i^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \\ \vdots \\ \sum x_i^m y_i \end{bmatrix}$$

- Note que la matriz es simétrica, por lo que se puede utilizar la descomposición de Cholesky para solucionar el sistema.
- El sistema es **mal** condicionado, por diferencias de valores debido a potencias, particularmente si $m \gg 1$

Cálculo del error

- En el caso de polinomios de orden m y n datos, la desviación estándar es

$$s_{y/x} = \sqrt{\frac{S_r}{n - (m + 1)}}$$

- El coeficiente de determinación r^2 del mismo modo que en el caso de regresión lineal:

$$r^2 = \frac{S_t - S_r}{S_t}$$

donde S_t es la varianza los datos

$$S_t = \sum (y_i - \bar{y})^2 \qquad \bar{y} = \frac{1}{n} \sum y_i$$

y S_r la suma de cuadrados de residuos descrita anteriormente.

Algoritmo en versión ingenua

```
// int n      : número de datos
// vector<T> x: datos x
// vector<T> y: datos y
// matrix<T> a: matriz normal aumentada
T sum();
for (int i=0;i<=orden;++i) {
    for (int j=0;j<=i;++j) {
        int k=i+j;
        sum = 0;
        for (l=0;l<n;++l) {
            sum+=pow(x[l], k);
        }
        a[i][j] = a[j][i] = sum;
    }
    sum=T(0);
    for (int l=0;l<n;++l) {
        sum+= y[l] * pow(x[l], i);
    }
    a[i][orden+1] = sum;
}
```

Nuevo algoritmo

Nótese que las anti-diagonales comparten el mismo valor $\sum x_i^n$ con $n = 0 \dots 2m$, que pueden precalcularse.

```
// int orden : orden del polinomio
// vector<T> x: datos x
// vector<T> y: datos y
// matrix<T> a: matriz normal aumentada
int n=x.size(); // cuántos datos
T sum();
vector<T> acc(2*orden,0);
for (int l=0;l<n;++l) { // para cada dato
    tmp=1;
    for (int o=1;o<=2*orden;++o) { // para cada exponente
        tmp*=x[l];
        acc[o]+=tmp;
    }
}
acc[0]=n;
a.resize(orden+1,orden+1);
for (int i=0;i<=orden;++i) {
    a[i][i]=acc[i+i];
    for (int j=i+1;j<=orden;++j) {
        a[j][i]=a[i][j]=acc[i+j];
    }
}
```

Regresión lineal múltiple

(1)

- Conceptos anteriores se generalizan para ajustar los parámetros de un **plano** parametrizado con a_0 , a_1 y a_2 , con variables independientes x_1 , x_2

$$y = a_0 + a_1x_1 + a_2x_2 + e$$

- En este caso el error a minimizar es

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1x_{1i} - a_2x_{2i})^2$$

Regresión lineal múltiple

(2)

- Derivando e igualando a cero las derivadas se obtiene el sistema

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \end{bmatrix}$$

- Concepto es fácilmente extendible a hiperplanos en espacios m -dimensionales:

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \cdots a_mx_m + e$$

Algoritmo de regresión m -dimensional

```
// entrada: matrix<T> x: con primera fila y columna nulas
//                               (indices 0).
//                               Cada fila es un vector de entrada
//                               vector<T> y: valores y para cada vector de entrada
// salida:  matrix<T> a: matriz de sistema normal
int n=x.rows();           // total de datos disponibles
int dim = x.columns();    // dimensión de cada dato
T sum(0);
for (int i=1;i<=dim+1;++i) {
    for (int j=1;j<=i;++j) {
        for (int l=1;l<=n;++l) {
            sum += x[i-1][l]*x[j-1][l];
        }
        a[i][j] = sum;
        a[j][i] = sum;
    }
    sum = T(0);
    for (l=1;l<=n;++l) {
        sum += y[l]*x[i-1][l];
    }
    a[i][dim+2] = sum;
}
```

Linealización en múltiples dimensiones

Del mismo modo que se linealizaron relaciones para usar regresión lineal, para relaciones de la forma:

$$y = a_0 x_1^{a_1} x_2^{a_2} \cdots x_m^{a_m}$$

se aplican logaritmos a ambos lados para obtener

$$\ln y = \ln a_0 + a_1 \ln x_1 + a_2 \ln x_2 + \cdots + a_m \ln x_m$$

que se resuelve entonces como problema de regresión lineal múltiple

Resumen

- 1 Introducción
- 2 Regresión Lineal
- 3 Mínimos cuadrados
 - Linealización
- 4 Regresión polinomial

Este documento ha sido elaborado con software libre incluyendo \LaTeX , Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, L^T_EX- \LaTeX , GNU-Make y Subversion en GNU/Linux



Este trabajo se encuentra bajo una Licencia Creative Commons Atribución-NoComercial-LicenciarIgual 3.0 Unported. Para ver una copia de esta Licencia, visite <http://creativecommons.org/licenses/by-nc-sa/3.0/> o envíe una carta a Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

© 2005-2018 Pablo Alvarado-Moya Área de Ingeniería en Computadores Instituto Tecnológico de Costa Rica