

Material de estudio

OCW 2018: Utilizando Mathematica como apoyo al cálculo algebraico en los grados de Ingeniería

Tema 5. Teoría de la aproximación

Equipo docente del curso

*Martín Yagüe, Luis
Unzueta Inchaurre, Aitziber
Arrospide Zabala, Eneko
García Ramiro, María Begoña
Soto Merino, Juan Carlos
Alonso González, Erik*

TEMA 5. TEORÍA DE LA APROXIMACIÓN

Introducción

En este tema se abordan los siguientes conceptos:

- producto escalar en los espacios vectoriales $\mathbb{R}^n$, $\mathbb{P}_n(x)$ y $M_{m \times n}(\mathbb{R})$
- expresión matricial del producto escalar: matriz de Gram
- ortogonalidad: bases ortogonales, proyección ortogonal, proyección de un vector sobre otro vector y proyección de un vector en un subespacio vectorial
- aproximación discreta y aproximación continua
- resolución aproximada de sistemas incompatibles en el sentido de mínimos cuadrados

Producto escalar

Producto escalar usual en $\mathbb{R}^n$

El producto escalar usual de dos vectores $\vec{u}(u_1, u_2, \dots, u_n), \vec{v}(v_1, v_2, \dots, v_n) \in (\mathbb{R}^n, \bullet)$ se define como:

$$\vec{u} \bullet \vec{v} = \sum_{i=1}^n u_i \cdot v_i = u_1 \cdot v_1 + u_2 \cdot v_2 + \dots + u_n \cdot v_n \in \mathbb{R}$$

Se utiliza la función :

- **Dot**. Calcula el producto escalar usual (*dot product*) de los dos vectores introducidos como argumentos. Puede usarse el operador `.`, obteniéndose el mismo resultado.

Ejemplo. Producto escalar usual de los vectores: $\vec{u}(0, -2, 1), \vec{v}(1, -1, 2) \in (\mathbb{R}^3, \bullet)$

`{u = {0, -2, 1}, v = {1, -1, 2}}; (* definición de los vectores *)`

`Dot[u, v] == u.v (* comprobación de la igualdad de resultados usando la función y el operador *)`

`True`

`u.v (* cálculo del producto escalar usual *)`

`4`

`v.u (* se verifica la propiedad conmutativa *)`

`4`

Otros productos escalares en $\mathbb{R}^n$

La expresión analítica del producto escalar se introduce como una función de dos variables.

El producto escalar de dos vectores se realiza ejecutando la función definida previamente indicando como argumentos los vectores cuyo producto se calcula.

Ejemplo. Se considera el espacio vectorial euclídeo $(\mathbb{R}^2, \langle, \rangle)$ con el siguiente producto escalar:

$$\langle \vec{u}, \vec{v} \rangle = \langle (x_1, y_1), (x_2, y_2) \rangle = x_1 \cdot x_2 - x_2 \cdot y_1 - x_1 \cdot y_2 + 2 y_1 \cdot y_2$$

- definición del producto escalar como una función de dos variables u y v

```
pe[u_, v_] := u[[1]] * v[[1]] - v[[1]] * u[[2]] - u[[1]] * v[[2]] + 2 u[[2]] * v[[2]]
```

- la notación $u[[i]]$ hace referencia a la coordenada i del vector denotado como u
- la asignación es diferida por lo que no produce salida
- producto escalar de dos vectores genéricos

```
{u = {x1, y1}, v = {x2, y2}}; (* definición de los vectores *)
```

```
pe[u, v] (* obtención del producto escalar *)
```

```
x1 x2 - x2 y1 - x1 y2 + 2 y1 y2
```

- el resultado puede usarse para comprobar que la función pe está bien definida
- caso particular, producto escalar de los vectores: $\vec{u}(1, 0), \vec{v}(3, -1) \in \mathbb{R}^2$

```
{u = {1, 0}, v = {3, -1}}; (* definición de los vectores *)
```

```
pe[u, v] (* obtención del producto escalar *)
```

```
4
```

- caso particular, producto escalar de los vectores: $\vec{w}(1, 1), \vec{r}(-6, 2) \in \mathbb{R}^2$

```
{w = {1, 1}, r = {-6, 2}}; (* definición de los vectores *)
```

```
pe[w, r] (* obtención del producto escalar *)
```

```
2
```

Producto escalar usual en $M_{m \times n}(\mathbb{R})$

Para calcular el producto escalar usual de dos matrices, $A, B \in (M_{m \times n}(\mathbb{R}), \bullet)$:

- se escriben los elementos de cada matriz como una lista (**Flatten**)
- se trabaja con esas listas de elementos como vectores de $\mathbb{R}^{m \times n}$

Ejemplo. Se considera el espacio vectorial euclídeo $(M_{2 \times 3}(\mathbb{R}), \bullet)$ con el producto escalar usual. Calcular $A \bullet B$, siendo:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & -1 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

- definición de las matrices cuyo producto escalar se quiere calcular

```
{a = {{1, 1, 1}, {0, -1, 2}}, b = {{1, 0, 1}, {0, 1, 1}}};
```

```
{MatrixForm[a], MatrixForm[b]}
```

```
{ $\begin{pmatrix} 1 & 1 & 1 \\ 0 & -1 & 2 \end{pmatrix}$ ,  $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ }
```

- transformar las matrices 2×3 en vectores de $2 \times 3 = 6$ coordenadas

```
{va = Flatten[a], vb = Flatten[b]}
```

```
{{1, 1, 1, 0, -1, 2}, {1, 0, 1, 0, 1, 1}}
```

- cálculo del producto escalar usual

```
va.vb
```

```
3
```

También, cabe otra posibilidad de realizar el producto escalar usual dado que se puede definir como:

$$A \bullet B = \text{tr}(B^t \cdot A)$$

En este caso:

- se define el producto escalar como una función de dos variables
- se definen las matrices en la forma habitual como listas de listas
- el producto escalar de las dos matrices se realiza ejecutando la función definida previamente indicando como argumentos las matrices cuyo producto se calcula

Ejemplo. Se considera el espacio vectorial euclídeo $(\mathbb{M}_{2 \times 3}(\mathbb{R}), \bullet)$ con el producto escalar usual.

Calcular $A \bullet B = \text{tr}(B^t \cdot A)$, siendo:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & -1 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

- definición de las matrices cuyo producto escalar se quiere calcular

`{a = {{1, 1, 1}}, {0, -1, 2}}, b = {{1, 0, 1}}, {0, 1, 1}};`

- definición del producto escalar como una función de dos variables a y b

`pe[a_, b_] := Tr[Transpose[b].a]`

- cálculo del producto escalar usual

`pe[a, b]`

3

Productos escalares en $\mathbb{P}_n(x)$

Cuando el producto escalar está definido en función de los coeficientes de dos polinomios, $p(x)$ y $q(x)$, entonces:

- se escriben los coeficientes de cada polinomio como una lista
- se trabaja con esas listas de coeficientes como vectores de $\mathbb{R}^{n+1}$

Ejemplo. Se considera el espacio euclídeo $(\mathbb{P}_2(x), \bullet)$ con el producto escalar usual:

$$\forall p(x) = a_1 x^2 + b_1 x + c_1, q(x) = a_2 x^2 + b_2 x + c_2 : \langle p(x), q(x) \rangle = a_1 \cdot a_2 + b_1 \cdot b_2 + c_1 \cdot c_2$$

$$\text{Calcular: } (2x^2 - 1) \bullet (-2x^2 - 3x + 1)$$

- definición de los polinomios cuyo producto escalar se quiere calcular

`{polp[x_] = 2 x^2 - 1, polq[x_] = -2 x^2 - 3 x + 1};`

- representación vectorial de los polinomios con respecto a la base canónica

`{p = CoefficientList[polp[x], x, 3], q = CoefficientList[polq[x], x, 3]}`

`{{-1, 0, 2}}, {1, -3, -2}}`

- cálculo del producto escalar usual

`p.q`

-5

Si el producto escalar está definido en función de los distintos valores que toman dos polinomios entonces:

- se define el producto escalar como una función de dos variables
- se definen los polinomios como funciones de una variable

- el producto escalar de los dos polinomios se realiza ejecutando la función definida previamente indicando como argumentos los polinomios cuyo producto se calcula

Ejemplo. Se considera el espacio euclídeo $(\mathbb{P}_4(x), \langle, \rangle)$ con el producto escalar siguiente:

$$\forall p(x), q(x) \in \mathbb{P}_4(x) : \langle p(x), q(x) \rangle = p(-2) \cdot q(-2) + p(-1) \cdot q(-1) + p(0) \cdot q(0) + p(1) \cdot q(1) + p(2) \cdot q(2)$$

$$\text{Calcular: } \langle 2x^3 - 1, x^4 - 2x^2 - 3x + 1 \rangle$$

- definición del producto escalar como una función de dos variables p y q

```
pe[p_, q_] := p[-2] * q[-2] + p[-1] * q[-1] + p[0] * q[0] + p[1] * q[1] + p[2] * q[2]
```

- la notación $p[[\#i]]$ hace referencia al valor del polinomio denotado como p para el valor i de la variable
- la asignación es diferida por lo que no produce salida
- definición de los polinomios cuyo producto escalar se quiere calcular

```
Clear[p, q]
```

```
{p[x_] = 2 x^3 - 1, q[x_] = x^4 - 2 x^2 - 3 x + 1}
```

```
{-1 + 2 x^3, 1 - 3 x - 2 x^2 + x^4}
```

- cálculo del producto escalar

```
pe[p, q]
```

```
-223
```

Producto escalar usual en $C[a, b]$

Se considera el espacio vectorial euclídeo, de dimensión no finita, de las funciones continuas en el intervalo cerrado $[a, b]$, $(C[a, b], \bullet)$, donde $\bullet$ denota el producto escalar usual que, en este caso, se define como:

$$\forall f(x), g(x) \in C[a, b] : f(x) \bullet g(x) = \int_a^b f(x) \cdot g(x) dx$$

En la definición del producto escalar se utiliza la función:

- **Integrate.** Calcula la integral definida de la función $f(x)$ indicada como argumento en el intervalo $[x_{min}, x_{max}]$.

Ejemplo. Se considera el espacio euclídeo $(C[0, \pi], \bullet)$ con el producto escalar usual:

$$\forall f(x), g(x) \in C[0, \pi] : f(x) \bullet g(x) = \int_0^\pi f(x) \cdot g(x) dx$$

$$\text{Calcular: } x^2 \bullet \sin(x)$$

- definición del producto escalar como una función de dos variables f y g

```
pe[f_, g_] := Integrate[f * g, {x, 0, Pi}]
```

- definición de las funciones cuyo producto escalar se quiere calcular

```
{f[x_] = x^2, g[x_] = Sin[x]}
```

```
{x^2, Sin[x]}
```

- cálculo del producto escalar

```
pe[f[x], g[x]]
```

```
-4 + \pi^2
```

- interpretación geométrica del resultado: área del dominio sombreado en la gráfica

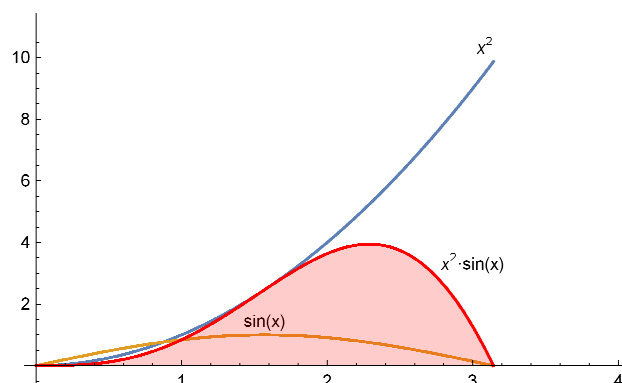
```

ar = Plot[f[x] * g[x], {x, 0, Pi}, PlotStyle -> {Red}, Filling -> Bottom, PlotLabels -> Placed[{"x^2 . sin(x)", {3, 3.4}}];

fg = Plot[{f[x], g[x]}, {x, 0, Pi}, PlotLabels -> Placed[{"x^2", "sin(x)", Above}];

Show[fg, ar]

```



Expresión matricial del producto escalar

Matrices de Gram

Una matriz de Gram es aquella matriz simétrica que contiene el producto escalar considerado entre todos los vectores de una base cualquiera, $B = (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_n)$, de un espacio vectorial euclídeo $(\mathbb{E}_n(x), \langle, \rangle)$:

$$G_B = \begin{pmatrix} \langle \vec{u}_1, \vec{u}_1 \rangle & \langle \vec{u}_1, \vec{u}_2 \rangle & \dots & \langle \vec{u}_1, \vec{u}_n \rangle \\ \langle \vec{u}_2, \vec{u}_1 \rangle & \langle \vec{u}_2, \vec{u}_2 \rangle & \dots & \langle \vec{u}_2, \vec{u}_n \rangle \\ \dots & \dots & \dots & \dots \\ \langle \vec{u}_n, \vec{u}_1 \rangle & \langle \vec{u}_n, \vec{u}_2 \rangle & \dots & \langle \vec{u}_n, \vec{u}_n \rangle \end{pmatrix}$$

Entonces, $\forall \vec{u}(u_1, u_2, \dots, u_n), \vec{v}(v_1, v_2, \dots, v_n) \in (\mathbb{E}_n, \langle, \rangle): \langle \vec{u}, \vec{v} \rangle = \vec{u} \cdot G_B \cdot \vec{v}^T$

Si se considera el producto escalar usual y la base canónica, $B_C = (\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n)$, de un espacio vectorial euclídeo $(\mathbb{E}_n(x), \cdot)$:

$$G_{B_C} = \begin{pmatrix} \vec{e}_1 \cdot \vec{e}_1 & \vec{e}_1 \cdot \vec{e}_2 & \dots & \vec{e}_1 \cdot \vec{e}_n \\ \vec{e}_2 \cdot \vec{e}_1 & \vec{e}_2 \cdot \vec{e}_2 & \dots & \vec{e}_2 \cdot \vec{e}_n \\ \dots & \dots & \dots & \dots \\ \vec{e}_n \cdot \vec{e}_1 & \vec{e}_n \cdot \vec{e}_2 & \dots & \vec{e}_n \cdot \vec{e}_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} = \mathbb{I}_n$$

Con *Mathematica* se procede de la siguiente manera para generar una de estas matrices:

- se define la base B considerada como una lista de listas
- se define el producto escalar como una función de dos variables salvo que sea el producto escalar usual en $\mathbb{R}^n$
- la matriz de Gram se introduce con la función **Table** indicando como primer argumento (*expr*) el producto escalar de los vectores de la base y utilizando dos índices (uno para el número de filas y otro para el número de columnas)

Ejemplo. Se considera el espacio vectorial euclídeo $(\mathbb{R}^2, \langle, \rangle)$ con el siguiente producto escalar:

$$\langle \vec{u}, \vec{v} \rangle = \langle (x_1, y_1), (x_2, y_2) \rangle = x_1 \cdot x_2 + 2 y_1 \cdot y_2$$

Hallar la matriz de Gram referida a la base canónica. Repetir el ejercicio con una base diferente.

- definición del producto escalar como una función de dos variables u y v

```
pe[u_, v_] := u[[1]] * v[[1]] + 2 u[[2]] * v[[2]]
```

- comprobación realizando el producto escalar de dos vectores genéricos

```
{u = {x1, y1}, v = {x2, y2}};
```

```
pe[u, v]
```

```
x1 x2 + 2 y1 y2
```

- definición de la base usual

```
b = {{1, 0}, {0, 1}};
```

- obtención de la matriz de Gram

```
gb = Table[pe[b[[i]], b[[j]]], {i, Length[b]}, {j, Length[b]}; MatrixForm[gb]
```

```

$$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

```

- cálculo del producto escalar de los vectores $\vec{u}(1, 0), \vec{v}(3, -1) \in \mathbb{R}^2$

```
{u = {1, 0}, v = {3, -1}};
```

- con la función definida

```
pe[u, v]
```

```
3
```

- con la matriz de Gram

```
u.gb.v
```

```
3
```

- si se considera, ahora, una base diferente del espacio vectorial $\mathbb{R}^2$

```
b1 = {{1, 2}, {2, 1}};
```

```
gb1 = Table[pe[b1[[i]], b1[[j]]], {i, Length[b1]}, {j, Length[b1]}; MatrixForm[gb1]
```

```

$$\begin{pmatrix} 9 & 6 \\ 6 & 6 \end{pmatrix}$$

```

Ortogonalidad

Norma de un vector

Se define como: $\|\vec{u}\| = \sqrt{\langle \vec{u}, \vec{u} \rangle} \quad \forall \vec{u}(u_1, u_2, \dots, u_n) \in (\mathbb{E}_n, \langle, \rangle)$.

Se denomina **unitario** al vector cuya norma es la unidad con el producto escalar considerado.

Mathematica habilita la función **Norm** para el cálculo de la norma de un vector cuando se considera el producto escalar usual; en otro caso, debe definirse el producto escalar correspondiente como una función de dos variables y simular la fórmula.

Ejemplo. Calcular la norma del vector $\vec{v}(0, -1) \in \mathbb{R}^2$ considerando, en primer lugar, el producto escalar usual y, a continuación, el siguiente producto escalar $\forall \vec{u}, \vec{v} \in (\mathbb{R}^2, \langle, \rangle)$:

$$\langle \vec{u}, \vec{v} \rangle = \langle (x_1, y_1), (x_2, y_2) \rangle = x_1 \cdot x_2 + 2 y_1 \cdot y_2$$

```
v = {0, -1}; (* definición del vector *)
```

- usando el producto escalar usual

Norm[v] (* vector unitario *)

1

- usando el otro producto escalar dado

pe[u_, v_] := u[[1]] * v[[1]] + 2 u[[2]] * v[[2]]

Sqrt[pe[v, v]] (* vector no unitario *)

$\sqrt{2}$

Ángulo entre dos vectores

Se consideran los vectores : $\vec{u}(u_1, u_2, \dots, u_n), \vec{v}(v_1, v_2, \dots, v_n) \in (\mathbb{E}_n, \langle, \rangle)$.

El producto escalar entre dos vectores se define, también, como: $\langle \vec{u}, \vec{v} \rangle = \|\vec{u}\| \cdot \|\vec{v}\| \cdot \cos(\theta)$

De ahí, se define el ángulo entre los vectores como: $\theta = \arccos \frac{\langle \vec{u}, \vec{v} \rangle}{\|\vec{u}\| \cdot \|\vec{v}\|}$ ($0 \leq \theta \leq \pi$).

Dos vectores no nulos $\vec{u}, \vec{v} \in (\mathbb{E}_n(x), \langle, \rangle)$ se denominan **ortogonales** si $\langle \vec{u}, \vec{v} \rangle = 0$; se denota como $\vec{u} \perp \vec{v}$.

Mathematica habilita la función **VectorAngle** para el cálculo del ángulo entre dos vectores cuando se considera el producto escalar usual; en otro caso, debe definirse el producto escalar correspondiente como una función de dos variables y aplicar la fórmula.

Ejemplo. Calcular el ángulo entre los vectores $\vec{u}(1, -1), \vec{v}(2, 2) \in \mathbb{R}^2$ considerando, en primer lugar, el producto escalar usual y, a continuación, el siguiente producto escalar $\forall \vec{u}, \vec{v} \in (\mathbb{R}^2, \langle, \rangle)$:

$$\langle \vec{u}, \vec{v} \rangle = \langle (x_1, y_1), (x_2, y_2) \rangle = x_1 \cdot x_2 + 2 y_1 \cdot y_2$$

{u = {1, -1}, v = {2, 2}}; (* definición de los vectores *)

- usando el producto escalar usual

u.v (* vectores ortogonales *)

0

VectorAngle[u, v]

$\frac{\pi}{2}$

- usando el otro producto escalar dado

pe[u_, v_] := u[[1]] * v[[1]] + 2 u[[2]] * v[[2]]

pe[u, v] (* vectores no ortogonales *)

-2

ang = ArcCos[pe[u, v] / (pe[u, u] * pe[v, v])]

$\text{ArcCos}\left[-\frac{1}{18}\right]$

N[ang, 6]

1.62638

Bases ortonormales

Sea $B = (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_n)$ una base de un espacio vectorial euclídeo $(\mathbb{E}_n, \langle, \rangle)$.

La base B es **ortonormal** si:

- es ortogonal, es decir: $\langle \vec{u}_i, \vec{u}_j \rangle = 0 \quad \forall i \neq j$
- sus n vectores son unitarios, esto es: $\|\vec{u}_i\| = 1 \quad \forall i = 1, 2, \dots, n$

A partir de una base cualquiera, B , para obtener una base ortonormal:

- se aplica el método de ortogonalización de Gram-Schmidt para obtener una base ortogonal, B_O
- los vectores de B_O se dividen por su norma con lo que la base ortogonal se hace, además, unitaria y, en consecuencia, ortonormal

Este proceso, se hace con una sola función en *Mathematica*:

- **Orthogonalize**. Obtiene una base ortonormal a partir de una base definida previamente (b) usando el producto escalar que se incluye como argumento y que también ha sido definido (pe). **Nota**. Si sólo se incluye la base como argumento, por defecto, se considera el producto escalar usual.

Ejemplo. A partir de la base canónica de $\mathbb{R}^2$, obtener una base ortonormal del espacio considerando, en primer lugar, el producto escalar usual y, a continuación, el siguiente producto escalar $\forall \vec{u}, \vec{v} \in (\mathbb{R}^2, \langle, \rangle)$:

$$\langle \vec{u}, \vec{v} \rangle = \langle (x_1, y_1), (x_2, y_2) \rangle = x_1 \cdot x_2 + 2 y_1 \cdot y_2$$

`bc = {{1, 0}, {0, 1}}; (* definición de la base *)`

- usando el producto escalar usual

`Orthogonalize[bc] (* la base es ortonormal *)`

`{{1, 0}, {0, 1}}`

- usando el otro producto escalar dado

`pe[u_, v_] := u[[1]] * v[[1]] + 2 u[[2]] * v[[2]]`

`Orthogonalize[bc, pe]`

`{{1, 0}, {0,  $\frac{1}{\sqrt{2}}$ }}`

Ejemplo. A partir de la base $B = \{(1, 1), (-1, 1)\}$ de $\mathbb{R}^2$, obtener una base ortonormal del espacio considerando, en primer lugar, el producto escalar usual y, a continuación, el siguiente producto escalar $\forall \vec{u}, \vec{v} \in (\mathbb{R}^2, \langle, \rangle)$:

$$\langle \vec{u}, \vec{v} \rangle = \langle (x_1, y_1), (x_2, y_2) \rangle = x_1 \cdot x_2 + 2 y_1 \cdot y_2$$

`b = {{1, 1}, {-1, 1}}; (* definición de la base *)`

- usando el producto escalar usual

`Orthogonalize[b]`

`{{ $\frac{1}{\sqrt{2}}$ ,  $\frac{1}{\sqrt{2}}$ }, {- $\frac{1}{\sqrt{2}}$ ,  $\frac{1}{\sqrt{2}}$ }}`

- usando el otro producto escalar dado

`pe[u_, v_] := u[[1]] * v[[1]] + 2 u[[2]] * v[[2]]`

Orthogonalize[b, pe]

$$\left\{ \left\{ \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right\}, \left\{ -\sqrt{\frac{2}{3}}, \frac{1}{\sqrt{6}} \right\} \right\}$$

Proyección ortogonal

Una proyección ortogonal es aquella cuyas rectas proyectantes auxiliares son perpendiculares a la recta de proyección.

Sean dos vectores $\vec{u}, \vec{v} \in (\mathbb{E}_n, \langle, \rangle)$ tal que $\vec{v} \neq \vec{0}$. La proyección ortogonal del vector $\vec{u}$ sobre el $\vec{v}$ viene dada por la expresión:

$$\text{proy}_{\vec{v}} \vec{u} = \frac{\langle \vec{u}, \vec{v} \rangle}{\langle \vec{v}, \vec{v} \rangle} \cdot \vec{v} \in \mathcal{L}(\{\vec{v}\})$$

En *Mathematica* se usa la función:

- **Projection**. Calcula la proyección ortogonal del vector indicado como primer argumento (u) sobre el vector señalado como segundo argumento (v) con el producto escalar que aparece como tercer argumento (f) y que ha sido previamente definido. **Nota**. Si no se especifica el producto escalar se considera el usual por defecto.

Ejemplo. Calcular $\text{proy}_{\vec{v}} \vec{u}$, siendo $\vec{u} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)$, $\vec{v} = (1, 0) \in (\mathbb{R}^2, \bullet)$ donde $\bullet$ denota el producto escalar usual.

$$\{v = \{1, 0\}, u = \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\}\}; (* \text{definición de los vectores} *)$$

- proyección ortogonal

p1 = Projection[u, v]

$$\left\{ \frac{1}{\sqrt{2}}, 0 \right\}$$

- representación gráfica

(* vector que se proyecta *)

```
vu = {Arrowheads[Medium], Arrow[{0, 0}, u]};
vectoru = Graphics[{Black, vu}];
```

(* vector sobre el que se proyecta *)

```
vv = {Arrowheads[Medium], Arrow[{0, 0}, v]};
vectorv = Graphics[{Blue, vv}];
```

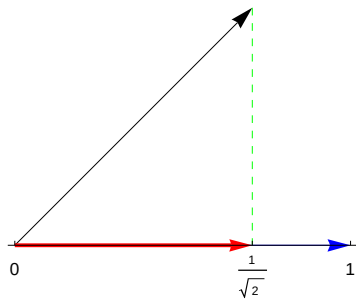
(* vector de proyección *)

```
vp = {Arrowheads[Medium], Arrow[{0, 0}, p1]};
proy = Graphics[{Red, Thick, vp}];
```

(* segmento de la recta de proyección *)

```
l = Line[{u, p1}];
linproy = Graphics[{Green, Dashed, l}];
```

Show[vectoru, vectorv, proy, linproy, Axes → {True, False}, Ticks → {{0, $\frac{1}{\sqrt{2}}$, 1}}]



Aproximación lineal en espacios vectoriales euclídeos

Mejor aproximación de un vector en un subespacio vectorial

Es la proyección ortogonal del vector que se aproxima sobre el subespacio vectorial considerado.

Se define como la suma de las proyecciones ortogonales del vector que se aproxima sobre los vectores de una base ortogonal del subespacio vectorial considerado.

Sea un vector $\vec{v} \in (\mathbb{E}_n, \langle, \rangle)$. Se considera una base ortogonal $B_S = (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_p)$ de un subespacio vectorial $S \subset \mathbb{E}_n$. Entonces:

$$\text{proy}_S \vec{v} = \sum_{i=1}^p \text{proy}_{\vec{u}_i} \vec{v} \quad \text{donde} \quad \text{proy}_{\vec{u}_i} \vec{v} = \frac{\langle \vec{v}, \vec{u}_i \rangle}{\langle \vec{u}_i, \vec{u}_i \rangle} \cdot \vec{u}_i \quad \forall i = 1, 2, \dots, p$$

El proceso se realiza combinando dos funciones:

- **Projection.** Anteriormente analizada.
- **Sum.** Realiza el sumatorio de una expresión que depende de un índice i que varía desde 1 hasta el valor denotado como $i_{\max}$ con incrementos de una unidad.

Ejemplo. Sea el vector $\vec{v} = (2, 5, 3) \in (\mathbb{R}^3, \cdot)$ donde $\cdot$ denota el producto escalar usual. Calcular la mejor aproximación de $\vec{v} \notin S$ en el subespacio $S = \langle \vec{e}_1 = (2, 0, 0), \vec{e}_2 = (0, 5, 0) \rangle$.

- vector que se aproxima

$v = \{2, 5, 3\};$

- base ortogonal de S

$b = \{e1 = \{2, 0, 0\}, e2 = \{0, 5, 0\}\};$ (* la familia generadora dada es libre y ortogonal *)

$e1.e2$ (* comprobación de ortogonalidad *)

0

- mejor aproximación de $\vec{v}$ en S

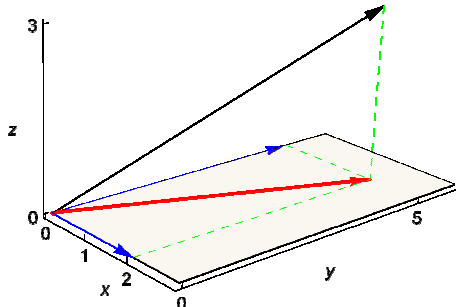
$\text{aprox} = \text{Sum}[\text{Projection}[v, b[[i]]], \{i, \text{Length}[b]\}]$

$\{2, 5, 0\}$

- representación gráfica

$o = \{0, 0, 0\};$

```
Graphics3D[{{Black, Arrowheads[Medium], Arrow[{0, v}]}, {Blue, Arrowheads[Medium], Arrow[{0, e1}]},
{Blue, Arrowheads[Medium], Arrow[{0, e2}]}, {Red, Thick, Arrowheads[Medium], Arrow[{0, {2, 5, 0}]}},
{Green, Dashed, Line[{v, aprox}]}, {Green, Dashed, Line[{e1, aprox}]}, {Green, Dashed, Line[{e2, aprox}]},
{LightGray, Opacity[.1], Polygon[{{0, 0, 0}, {0, 6, 0}, {3, 6, 0}, {3, 0, 0}]}], Axes -> True,
AxesOrigin -> Automatic, AxesLabel -> {x, y, z}, Boxed -> False, Ticks -> {{0, 1, 2}, {0, 5}, {0, 3}}]
```



Ejemplo. Calcular la mejor aproximación del vector $\vec{v}$ en el subespacio $S = \langle \vec{e}_1 = (1,1,0), \vec{e}_2 = (0,1,0) \rangle$, siendo $\vec{v} = (2, 5, 3) \in (\mathbb{R}^3, \langle, \rangle)$ y considerando el siguiente producto escalar $\forall \vec{u}, \vec{v} \in (\mathbb{R}^3, \langle, \rangle)$:

$$\langle \vec{u}, \vec{v} \rangle = \langle (x_1, y_1, z_1), (x_2, y_2, z_2) \rangle = x_1 x_2 + y_1 y_2 + 2 z_1 z_2 - x_1 z_2 - z_1 x_2$$

■ vector que se aproxima

$v = \{2, 5, 3\};$

■ producto escalar considerado

```
pe[u_, v_] := u[[1]] * v[[1]] + u[[2]] * v[[2]] + 2 u[[3]] * v[[3]] - u[[1]] * v[[3]] - u[[3]] * v[[1]]
```

■ base ortogonal de S

$b = \{e1 = \{1, 1, 0\}, e2 = \{0, 1, 0\}\};$ (* la familia generadora dada es libre pero NO ortogonal *)

```
pe[e1, e2]
```

1

```
b2 = Orthogonalize[b, pe]
```

$$\left\{ \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0 \right\}, \left\{ -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0 \right\} \right\}$$

■ mejor aproximación de $\vec{v}$ en S

```
aprox2 = Sum[Projection[v, b2[[i]]], {i, Length[b]}] // Simplify
```

```
{2, 5, 0}
```

Aproximación de una función continua por un polinomio

Una función continua en un intervalo $[a, b]$ puede aproximarse expresándola como combinación lineal de una familia de polinomios de un espacio vectorial $P_n(x)$.

Se considera:

■ el espacio vectorial euclídeo, de dimensión no finita, de las funciones continuas en $[a, b]$:

$$(C[a, b], \bullet)$$

■ el producto escalar usual de ese espacio:

$$\forall f(x), g(x) \in C[a, b]: f(x) \bullet g(x) = \int_a^b f(x) \cdot g(x) dx$$

- el subespacio vectorial de los polinomios en x de grado menor o igual que r :

$$\mathbb{P}_r(x) \subset C[a, b]$$

El problema queda resuelto calculando la mejor aproximación de la función dada en el subespacio $\mathbb{P}_r(x)$.

Ejemplo. Se considera el espacio vectorial euclídeo de las funciones continuas en el intervalo $[0,1]$ con el producto escalar usual, $(C[0,1], \bullet)$:

$$\forall f(x), g(x) \in C[0, 1]: f(x) \bullet g(x) = \int_0^1 f(x) \cdot g(x) dx$$

Aproximar la función $f(x) = \sinh(x)$ en el intervalo $[0,1]$ mediante un polinomio de grado 2.

- función que se aproxima

```
f[x_] := Sinh[x];
```

- definición del producto escalar usual del espacio vectorial euclídeo $C[0,1]$

```
pe[f_, g_] := Integrate[f*g, {x, 0, 1}]
```

- base usual del subespacio vectorial $S = \mathbb{P}_2(x) \subset C[0, 1]$

```
{f1[x_] := 1, f2[x_] := x, f3[x_] := x^2};
```

```
base = {f1[x], f2[x], f3[x]};
```

- base ortogonal del subespacio vectorial $S = \mathbb{P}_2(x) \subset C[0, 1]$

```
bort = Orthogonalize[base, pe] // Simplify
```

```
{1,  $\sqrt{3}(-1 + 2x)$ ,  $\sqrt{5}(1 - 6x + 6x^2)$ }
```

- mejor aproximación de la función $f(x) = \sinh(x)$ en S

```
vp[x_] = Sum[Projection[f[x], bort[[i]], pe], {i, Length[bort]}] // Simplify
```

$$-1 + \frac{5(19 - 26e + 7e^2)(1 - 6x + 6x^2)}{2e} + \cosh[1] + \frac{3(-1 + 2x)(2 + e - e \cosh[1])}{e}$$

```
N[vp[x]] // Expand
```

$$0.00925135 + 0.890837x + 0.265233x^2$$

- polinomio calculado para la aproximación

$$p(x) = 0.00925135 + 0.890837x + 0.265233x^2$$

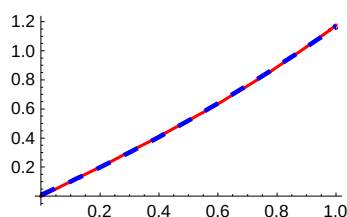
- representación gráfica de la función y el polinomio de aproximación

- en el intervalo $[0,1]$

```
grafv = Plot[f[x], {x, 0, 1}, PlotStyle -> {Red, Thickness[0.01]}];
```

```
grafvp = Plot[vp[x], {x, 0, 1}, PlotStyle -> {Blue, Dashing[0.05], Thickness[0.015]}];
```

```
Show[grafv, grafvp]
```

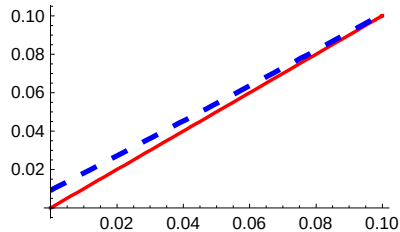


- en el intervalo $[0,0.1]$ (entorno del punto $x=0$)

```

gv = Plot[f[x], {x, 0, 0.1}, PlotStyle -> {Red, Thickness[0.01]};
gvp = Plot[vp[x], {x, 0, 0.1}, PlotStyle -> {Blue, Dashing[0.05], Thickness[0.015]};
Show[gv, gvp]

```

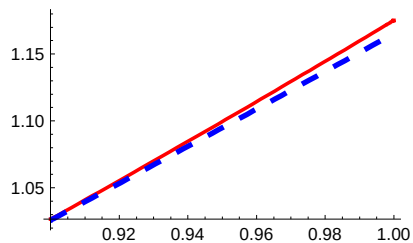


- en el intervalo $[0.9,1]$ (entorno del punto $x=1$)

```

gv2 = Plot[f[x], {x, 0.9, 1}, PlotStyle -> {Red, Thickness[0.01]};
gvp2 = Plot[vp[x], {x, 0.9, 1}, PlotStyle -> {Blue, Dashing[0.05], Thickness[0.015]};
Show[gv2, gvp2]

```

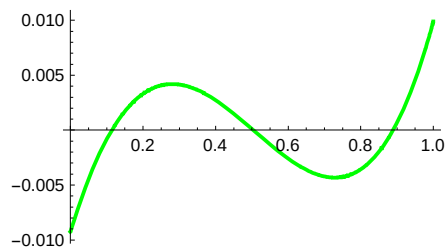


- error cometido en la aproximación

```
e[x_] := f[x] - vp[x]
```

- representación gráfica

```
graferror = Plot[e[x], {x, 0, 1}, PlotStyle -> {Green, Thickness[0.01]}]
```



- norma (cuantificación del error)

```

ne = Sqrt[pe[e[x], e[x]]] // N
0.00360283

```

$|error| = 0.00360283$

Solución aproximada de un sistema incompatible de ecuaciones lineales

El conocimiento y control de diferentes sistemas en el ámbito de la ingeniería implica realizar mediciones adecuadas. Las medidas realizadas permiten realizar cálculos y estimaciones útiles para la toma de decisiones.

Cualquier medida se ve afectada por diferentes circunstancias que dan lugar a errores. Para paliar, en la medida de lo posible, los errores producidos se realiza un número de mediciones muy superior al que resultaría necesario. Se obtienen así, sistemas sobredimensionados, con más ecuaciones que incógnitas. Generalmente, esto conduce a casos de incompatibilidad.

Es posible obtener una solución aproximada de un sistema incompatible de ecuaciones lineales. Para ello, se considera:

- un sistema S con m ecuaciones lineales y n incógnitas ($m > n$) expresado en forma vectorial:

$$S \equiv A_{m \times n} \cdot \vec{X}_{n \times 1} = \vec{b}_{m \times 1} \Rightarrow S \equiv x_1 \cdot \vec{a}_1 + x_2 \cdot \vec{a}_2 + \dots + x_n \cdot \vec{a}_n = \vec{b}$$

- si el sistema S es incompatible: $\vec{b} \notin \mathcal{L}(\{\vec{a}_1, \vec{a}_2, \dots, \vec{a}_n\}) = W$

El método de aproximación (denominado de los mínimos cuadrados) consiste en resolver el sistema compatible:

$$S' \equiv x_1 \cdot \vec{a}_1 + x_2 \cdot \vec{a}_2 + \dots + x_n \cdot \vec{a}_n = \vec{b}' = \text{proy}_W \vec{b} \in W$$

La solución del sistema compatible anterior es la solución ó aproximación de mínimos cuadrados del sistema incompatible. Se denomina así porque el error cometido ($\vec{\varepsilon}$) en la aproximación es mínimo para la norma euclídea asociada al producto escalar usual considerado; es decir, se minimiza el error cometido cuantificado como: $\|\vec{\varepsilon}\| = \|\vec{b} - \vec{b}'\|$.

La solución del sistema compatible anterior es la solución ó aproximación de mínimos cuadrados del sistema incompatible. Se denomina así porque el error cometido ($\vec{\varepsilon}$) en la aproximación es mínimo para la norma euclídea asociada al producto escalar usual considerado; es decir, se minimiza el error cometido cuantificado como: $\|\vec{\varepsilon}\| = \|\vec{b} - \vec{b}'\|$.

Se considera, ahora, la expresión matricial del sistema: $S \equiv A_{m \times n} \cdot X_{n \times 1} = B_{m \times 1}$. Si $r(A) < n$, la familia $\{\vec{a}_1, \vec{a}_2, \dots, \vec{a}_n\}$ es libre. En este caso, la solución de mínimos cuadrados es única (ya que el sistema resultante es compatible determinado) y puede expresarse matricialmente de la siguiente manera:

$$X^* = (A^T \cdot A)^{-1} \cdot A^T \cdot B$$

La matriz $(A^T \cdot A)^{-1} \cdot A^T$ se denomina pseudoinversa de A .

Ejemplo. Aproximar con el método de los mínimos cuadrados la solución del sistema:

$$\begin{cases} x + y = 0 \\ x - y = 0 \\ y = 1 \end{cases}$$

- Expresión vectorial del sistema: $\vec{a}_1, \vec{a}_2, \vec{X}$ y $\vec{b}$

$\{\{a1 = \{1, 1, 0\}, a2 = \{-1, 1, 1\}\}, X = \{x, y\}, b = \{0, 0, 1\}\};$

- Sistema incompatible

`A = Transpose[{a1, a2}];`

`S = A.X == b;`

`Solve[S, X]`

`{}`

- Base ortogonal del subespacio de proyección $W = \mathcal{L}(\{\vec{a}_1, \vec{a}_2\})$

`a1.a2 (* el sistem generador es libre y ortogonal *)`

`0`

`bo = {a1, a2}; (* base ortogonal *)`

- Mejor aproximación del vector $\vec{b}$ en W

```
pb = Sum[Projection[b, bo[[i]]], {i, Length[bo]}]
```

$$\left\{-\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right\}$$

- Solución aproximada

```
SC = A.X == pb;
```

```
Solve[SC, X]
```

$$\left\{\left\{x \rightarrow 0, y \rightarrow \frac{1}{3}\right\}\right\}$$

- Error cometido con la aproximación

```
error = b - pb;
```

```
N[Norm[error]]
```

```
0.816497
```

- Solución aproximada usando la pseudoinversa

```
MatrixRank[A]
```

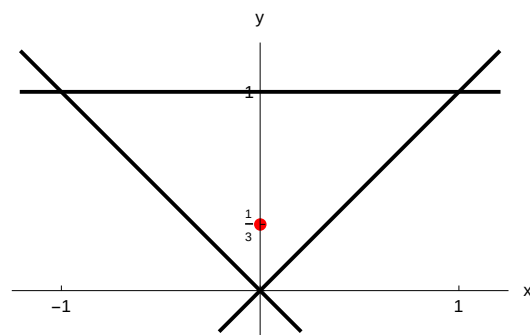
```
2
```

```
p = PseudoInverse[A].b
```

$$\left\{0, \frac{1}{3}\right\}$$

- Interpretación geométrica

```
Graphics[{Black, Thick, Line[{{-1.2, 1}, {1.2, 1}}], Line[{{1.2, 1.2}, {-0.2, -0.2}}],  
Line[{{-1.2, 1.2}, {0.2, -0.2}}], Red, PointSize[0.025], Point[p],  
Axes -> True, Ticks -> {{-1, 0, 1}, {0, 1/3, 1}}, AxesLabel -> {"x", "y"}}
```



Regresión lineal con mínimos cuadrados

La regresión con mínimos cuadrados es un método de extrapolación para hallar la línea que mejor se ajusta a una nube de puntos.

Dependiendo del tipo de función utilizada puede hablarse de regresión lineal, polinomial, exponencial, logarítmica, etc.

Se dispone de dos funciones:

- **Fit**. Encuentra un ajuste por mínimos cuadrados para la nube de puntos (primer argumento, *data*) como combinación lineal de las funciones (segundo argumento, *funcs*) y de las variables (tercer argumento, *vars*) indicadas.

- **LinearModelFit**. Construye el modelo de ajuste lineal permitiendo y permite analizar el error cometido.

Ejemplo. Se considera la nube de puntos: $A(1, 4)$, $B(-2, 5)$, $C(3, -1)$, $D(4, 1)$.

Hallar y representar gráficamente la función que mejor se adapta a dicha nube de puntos realizando una regresión lineal y otra cuadrática con la función **Fit**.

- Definición de la nube de puntos

```
puntos = {{1, 4}, {-2, 5}, {3, -1}, {4, 1}};
```

- Regresión lineal

```
recta = Fit[puntos, {1, x}, x]
```

$3.57143 - 0.880952 x$

- Regresión cuadrática

```
parábola = Fit[puntos, {1, x, x^2}, x]
```

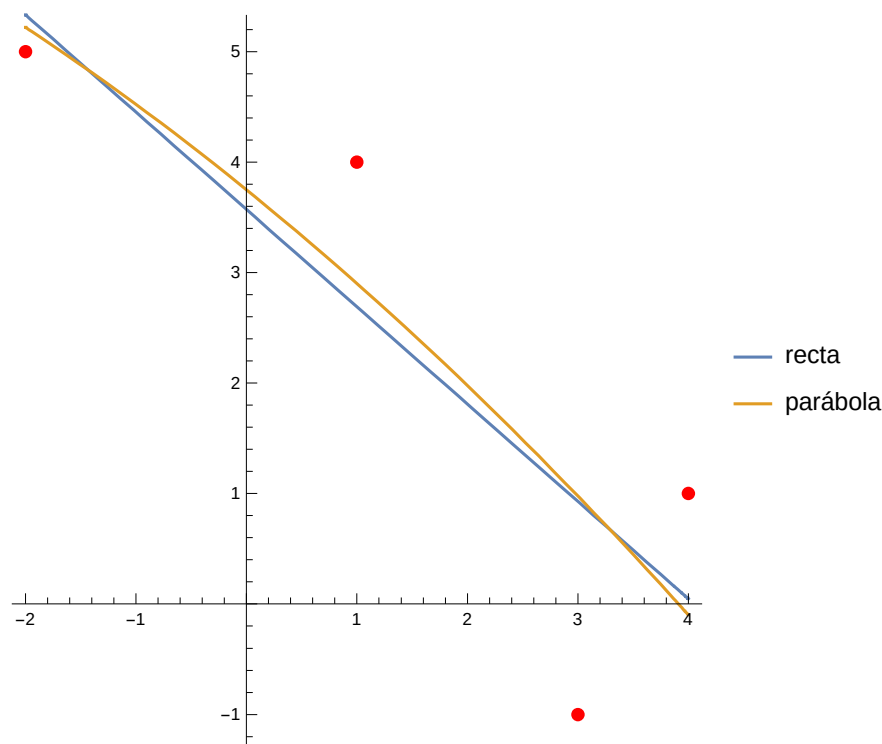
$3.75 - 0.810606 x - 0.0378788 x^2$

- Representación gráfica

```
nube = ListPlot[puntos, PlotStyle -> {Red, PointSize[Large]}];
```

```
líneas = Plot[{recta, parábola}, {x, -2, 4}, AspectRatio -> Automatic, PlotLegends -> "Expressions"];
```

```
Show[nube, líneas, AspectRatio -> Automatic]
```



Ejemplo. Se considera la nube de puntos: $A(1, 4)$, $B(-2, 5)$, $C(3, -1)$, $D(4, 1)$.

Hallar y representar gráficamente la función que mejor se adapta a dicha nube de puntos realizando una regresión lineal y otra cuadrática con la función **LinearModelFit**.

- Definición de la nube de puntos

```
puntos = {{1, 4}, {-2, 5}, {3, -1}, {4, 1}};
```

- Regresión lineal

■ Modelo

```
lm = LinearModelFit[puntos, {1, x}, x]
```

```
FittedModel[ 3.57143 - 0.880952 x ]
```

■ Recta de ajuste

```
rec = Normal[lm]
```

```
3.57143 - 0.880952 x
```

■ Evaluación en un punto

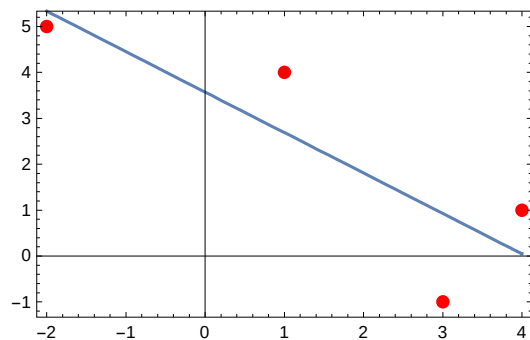
```
lm[4]
```

```
0.047619
```

■ Representación gráfica

```
nube = ListPlot[puntos, PlotStyle -> {Red, PointSize[Large]}];
```

```
Show[nube, Plot[lm[x], {x, -2, 4}], Frame -> True]
```



■ Obtención del error

```
error = lm["FitResiduals"] (* vector de error *)
```

```
{1.30952, -0.333333, -1.92857, 0.952381}
```

```
Norm[error] (* cuantificación del error *)
```

```
2.54015
```

■ Regresión cuadrática

■ Modelo

```
lm = LinearModelFit[puntos, {1, x, x^2}, x]
```

```
FittedModel[ 3.75 - 0.810606 x - 0.0378788 x^2 ]
```

■ Parábola de ajuste

```
par = Normal[lm]
```

```
3.75 - 0.810606 x - 0.0378788 x^2
```

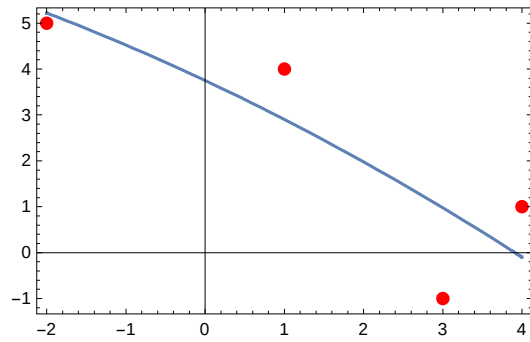
■ Evaluación en un punto

```
lm[4]
```

```
-0.0984848
```

■ Representación gráfica

```
Show[nube, Plot[lm[x], {x, -2, 4}], Frame -> True]
```



■ Obtención del error

```
error = lm["FitResiduals"] (* vector de error *)
```

```
{1.09848, -0.219697, -1.97727, 1.09848}
```

```
Norm[error] (* cuantificación del error *)
```

```
2.52413
```