04/03/2024

Programación Orientada a Objetos

11/03/2024

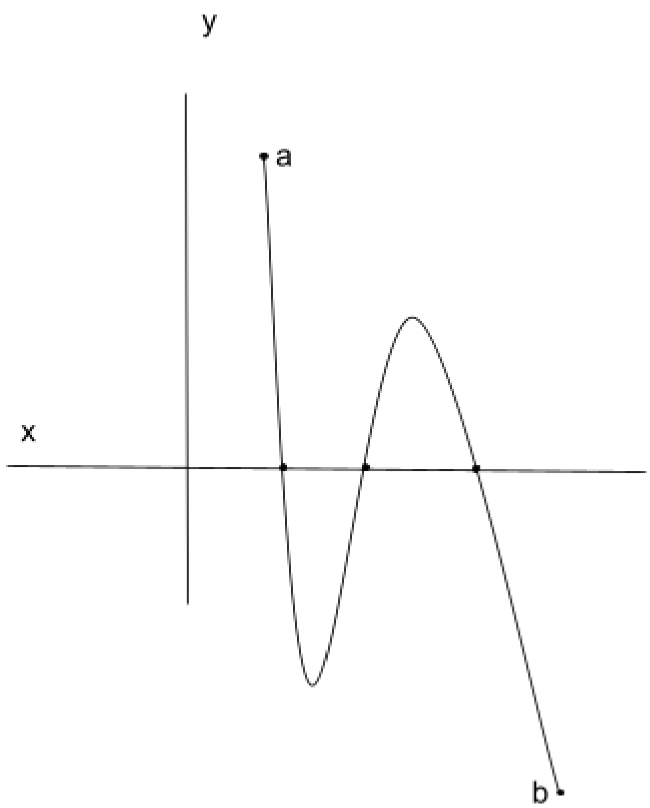
15/03/2024

En finanzas, usamos la raíz de un polinomio para sacar una TIR.

Sacar raíces de un polinomio

Bisección

- Necesito que f(x) sea continua.
- Usaremos el teorema de valor medio, por lo que, arbitrariamente, elegimos que a < b
- Suponemos que f(a)*f(b)<= 0, por lo que nos aseguramos que hay una cantidad de raíces >= 1.

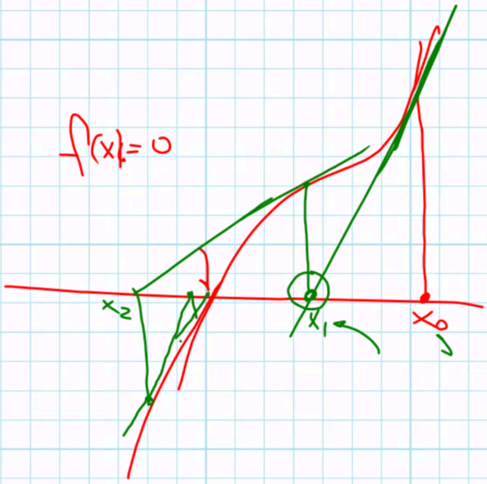


- Si la raíz no está en a y b, se divide la superficie de búsqueda a la mitad y se busca en esa sección.

$$c = (b-a)/2$$

Newton Raphson

Newton



$x_0 \quad f(x_0)$

$f'(x_0)$

$$f(x) \approx f(x_0) + f'(x_0)\,(x - x_0)$$

$$0 = f(x_{n+1}) = f(x_n) + f'(x_n)(x_{n+1} - x_n)$$

$$\boxed{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$f(x) = x^2 - n = 0$$

$$f'(x) = 2x$$

$$x_{n+1} = x_n - \frac{x_n^2 - n}{2x_n}$$

$$= \frac{2x_n^2 - x_n^2 - n}{2x_n}$$

$$= \frac{x_n^2 - n}{2x_n} = \frac{x_n}{2} - \frac{n}{2x_n}$$

$$\boxed{x_{n+1} = \frac{x_n - \frac{n}{x_n}}{2}}$$

Para el cálculo de TIR, la cual se calcula sobre valor futuro, se utiliza Newton Raphson

- Este algoritmo falla cuando las derivadas se hacen 0

- Puede entrar en un loop infinito

18/03/2024

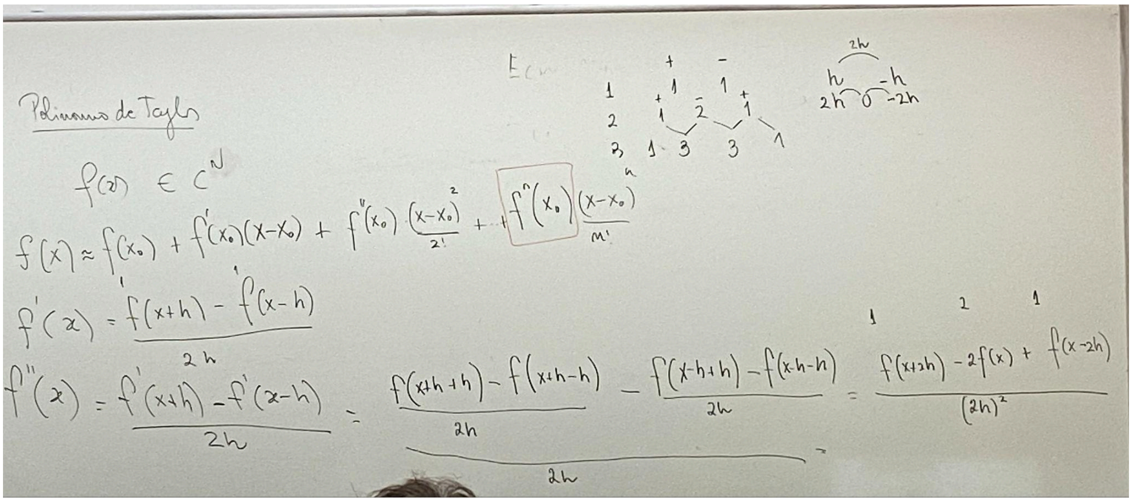
Polinomios

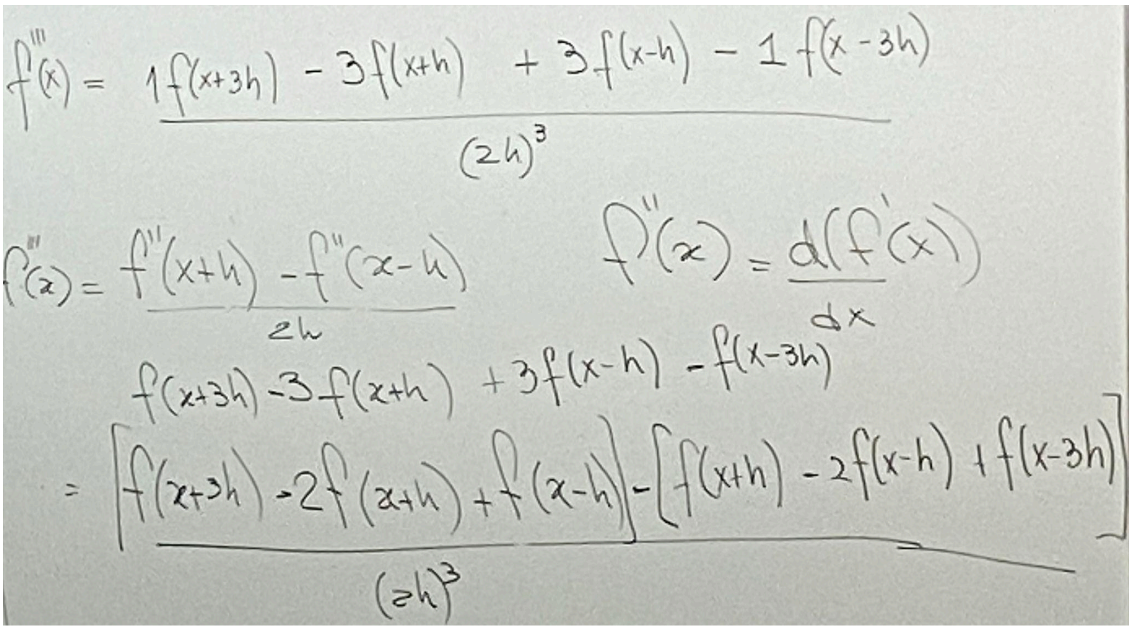
Su función es:

- Aproximar funciones → usamos polinomio de Taylor.
- Fiteo de datos, que es buscar polinomios que pasen por todos los puntos de datos → polinomios de Lagrange.
- Least squares, que busca cual es el mejor polinomio para un dado grado, y aproxima de mejor manera a todos los datos.

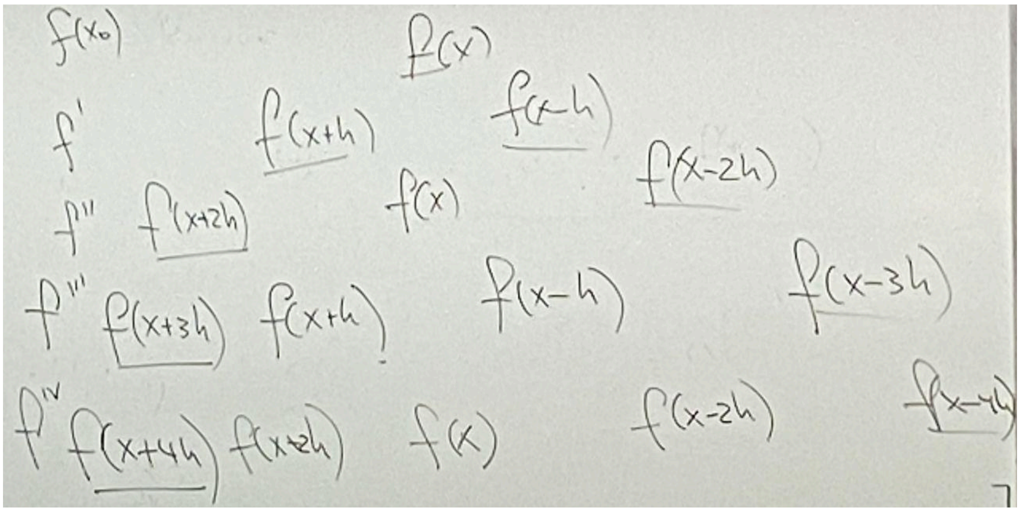
Polinomio de Taylor

Si no tenemos la derivada de f(x), podemos usar el método de la secante, que nos da el valor numérico en lugar de f'(x). Esto se llama derivada adelantada.





Para sacar el polinomio de Taylor, usamos el triángulo de Pascal.



Método de la secante

f'(x) = $\lim\limits_{h \to 0} \frac{f(x+h)-f(x-h)}{2h}$

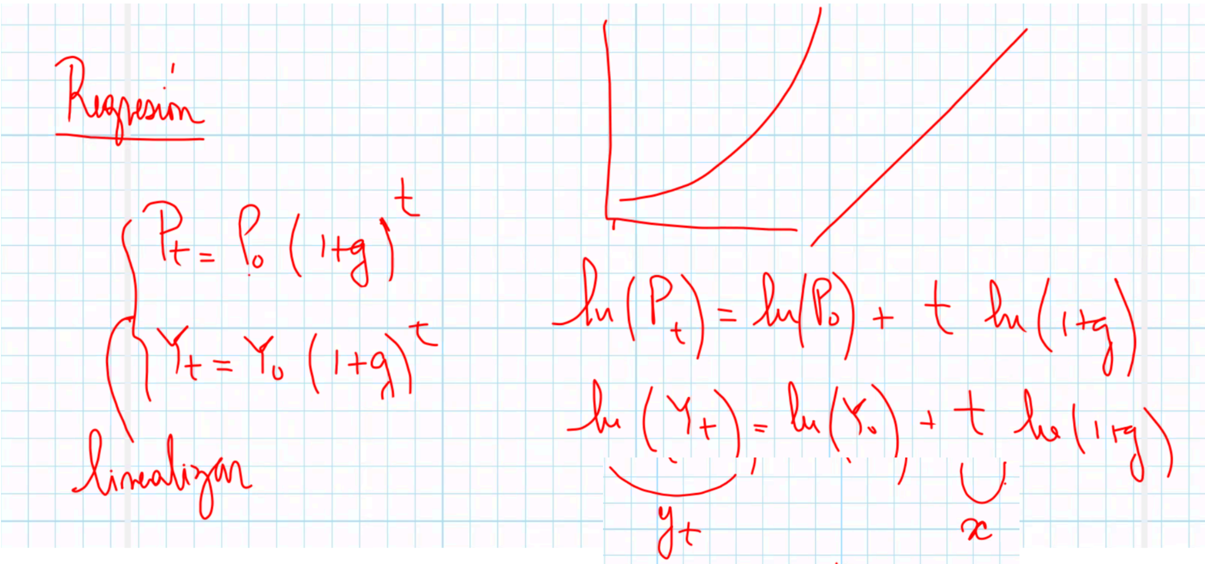
22/03/2024

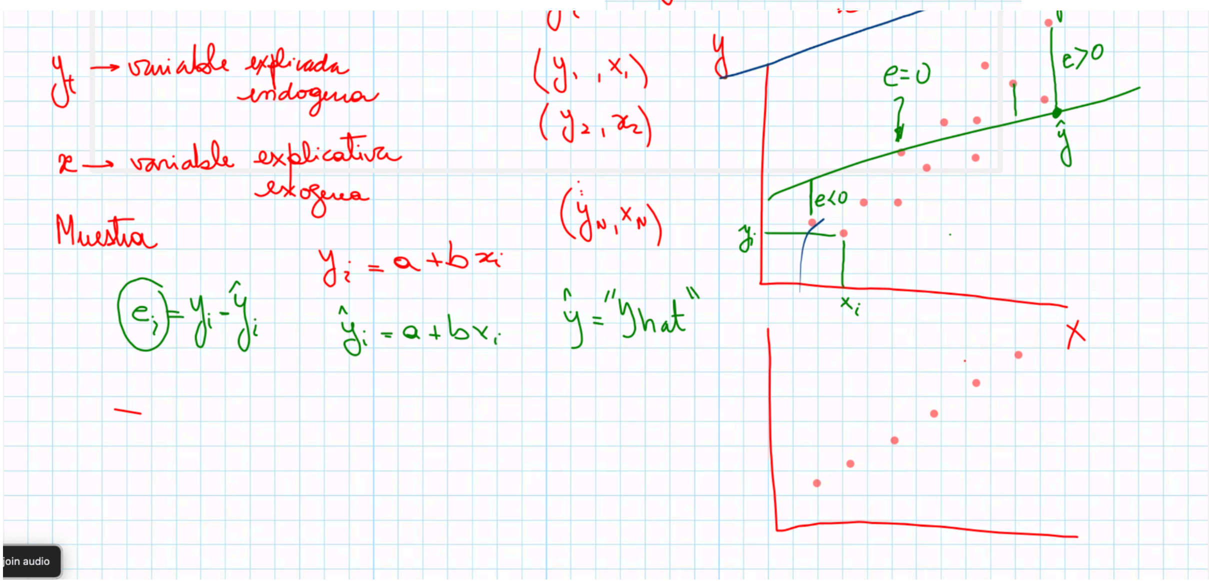
Regresiones lineales

Hay que buscar la recta que pasa más próxima a un conjunto de puntos.

El logaritmo sirve para medir la tasa de crecimiento de las cosas, suavizando curvas.

$\ln(P_t) = \ln(P_0) + t\ln(1+g) \Rightarrow \ln(Y_t) = \ln(T_0) + t\ln(1+g)$



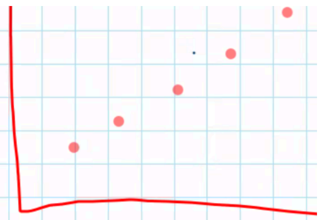


$$\frac{1}{N}\sum e_i = 0$$

$$\frac{1}{N}\sum_i \left(y_i - (a + b x_i)\right) = 0$$

$$\frac{1}{N}\sum_{i=1}^{N} y_i - \frac{1}{N}\sum_{i=1}^{N} a - b\,\frac{1}{N}\sum_{i=1}^{N} x_i = 0$$

$$\bar{Y} - a - b\bar{X} = 0$$

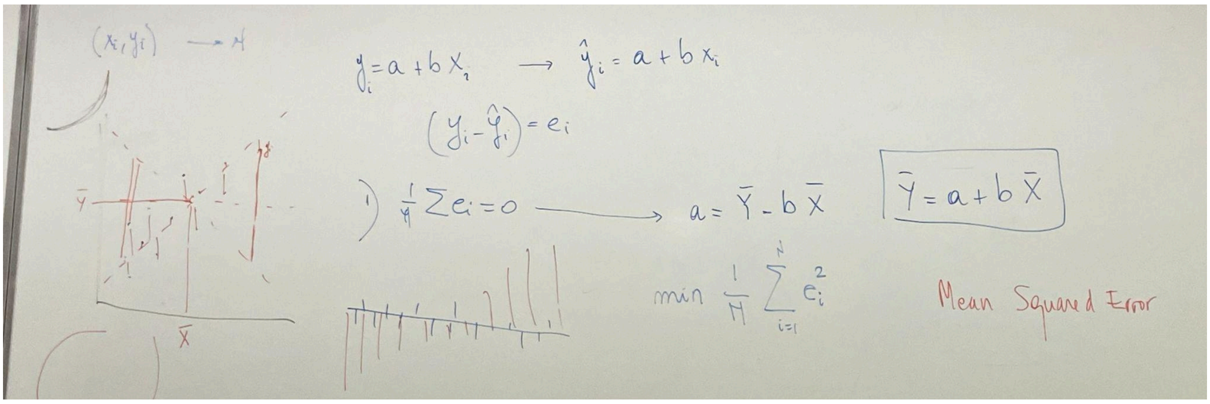


$\bar{X}$

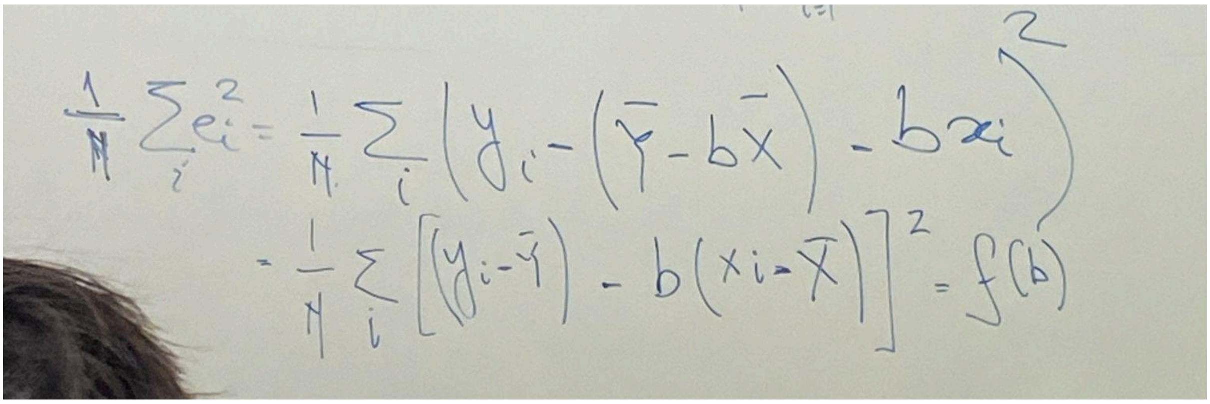
$$a = \bar{Y} - b\bar{X} \checkmark$$

$$\boxed{\bar{Y} = a + b\bar{X}}$$

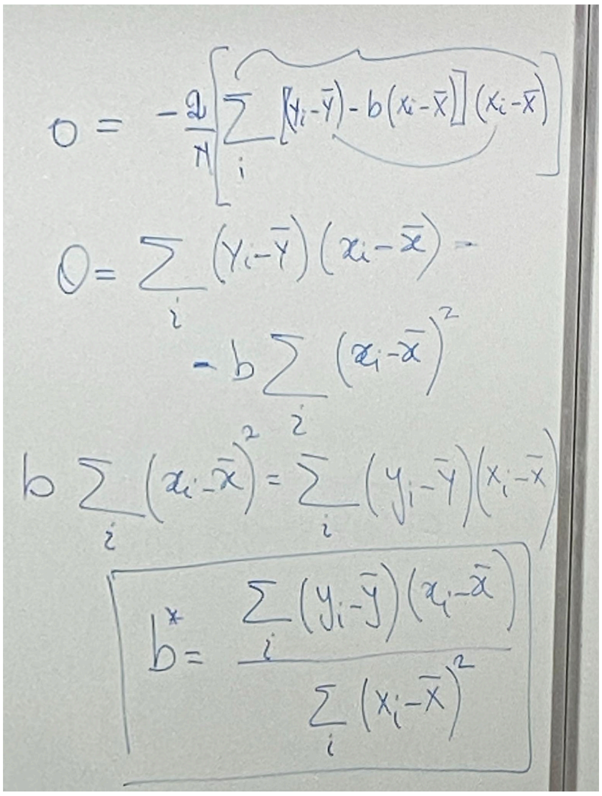
25/03/2024



Buscamos elevar el error al cuadrado, para poder buscar cual crece más y eliminarlo. Usamos una función llamada Mean Square Error.



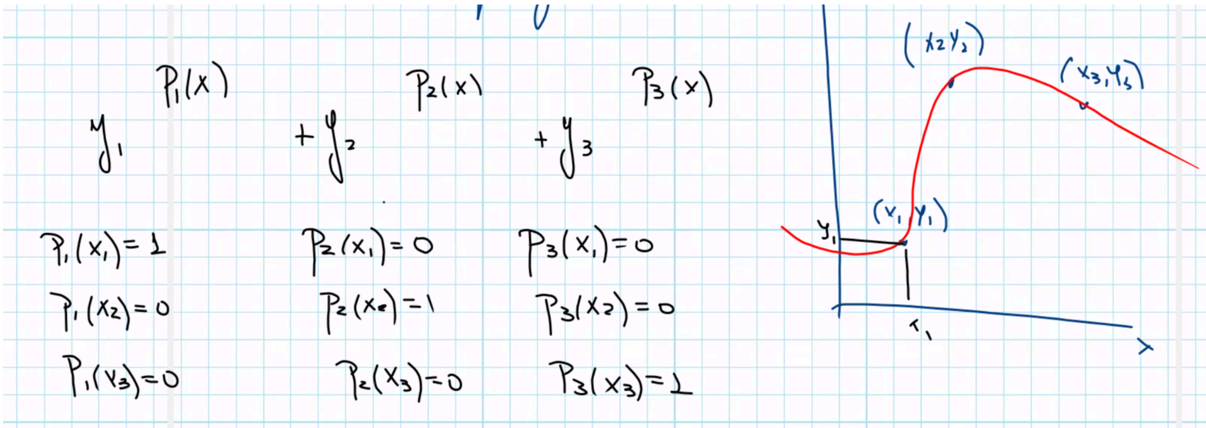
f’(b) = 0

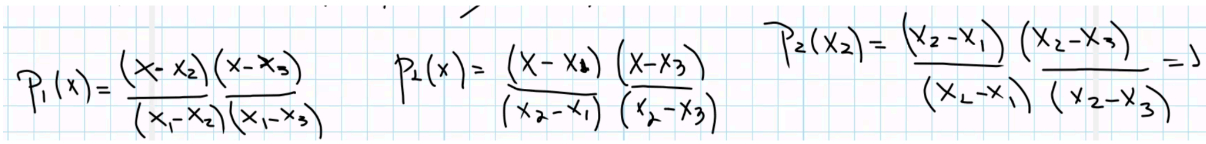


05/04/2024

Lagrange

Buscamos un polinomio que aproxima un conjunto de puntos

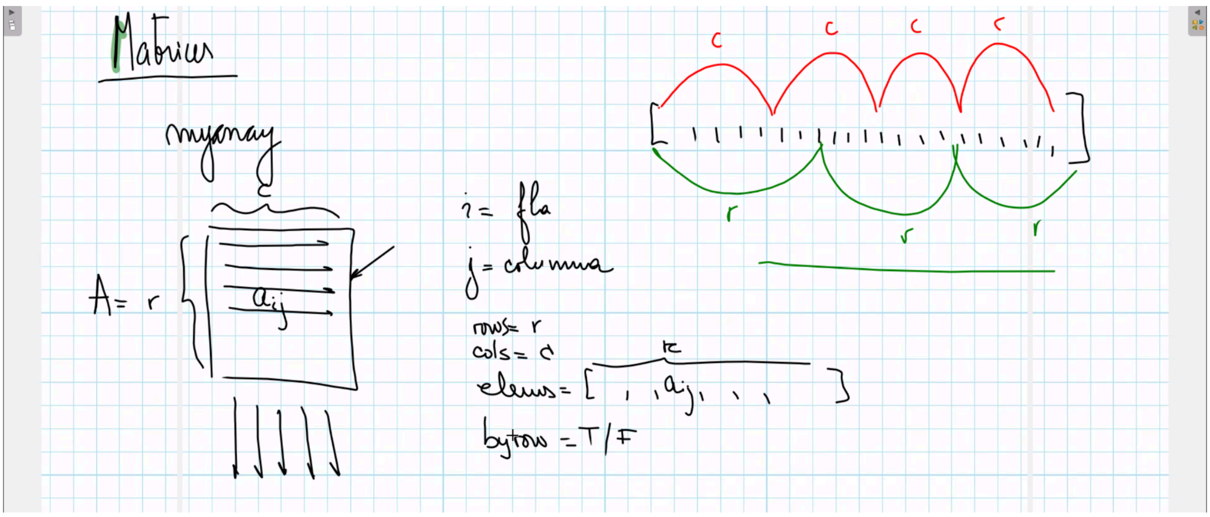




Matrices

Es un arreglo bidimensional. Tiene un parámetro **r** que es la cantidad de filas y un parámetro **c** que es la cantidad de columnas.

Cada elemento de la matriz es a_{i*j} donde i = filas; j = columnas. Elems = [lista de los elementos de la matriz]. La cantidad de elementos es r*c. Además sumamos un parámetro byRow = bool, que sirve para saber si leemos por filas o por columnas.



Métodos de la clase

__init__

get_pos(self, j, k) = buscamos el elemento $a_{j\ k}$,y buscamos m y conseguimos su posición en la lista

get_cords(self,

switch(self) = cambiamos los elementos para pasar de leer en columnas a leer en filas o viceversa. Sin embargo, hay que reordenar los elementos de la lista, en lugar de cambiar el parámetro byrow. Si hago a.switch().switch() = me devuelve A.

trans(self) = busca la traspuesta de la matriz

12/04/2024

<u>Lineq</u>

Es una subclase de matrices para resolver ecuaciones lineales

$$N\begin{bmatrix} a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}=\begin{pmatrix}b_1\\ b_2\\ b_3\end{pmatrix}$$

$$\underset{3\times 3}{A}\cdot\underset{3\times 1}{x}=\underset{3\times 1}{b}$$

$$x = A^{-1}b$$

$$A^{-1}=\frac{1}{\det(A)}\begin{bmatrix} & & \\ & & \end{bmatrix}_{cof(A)^T}$$

1 pasada $\rightarrow a_{11}\neq 0$

$$\begin{bmatrix}\frac{1}{a_{11}} & & \\ -\frac{a_{21}}{a_{11}} & 1 & \\ -\frac{a_{31}}{a_{11}} & & 1\end{bmatrix}\begin{bmatrix}\boxed{a_{11}} & a_{12} & a_{13}\\ \boxed{a_{21}} & a_{22} & a_{23}\\ \boxed{a_{31}} & a_{32} & a_{33}\end{bmatrix}\quad\begin{bmatrix}1 & a'_{12} & a'_{13}\\ 0 & a'_{22} & a'_{23}\\ 0 & a'_{32} & a'_{33}\end{bmatrix}$$

$$L_1 \qquad A'$$

$$\begin{pmatrix}a_{11}\\ a_{21}\\ a_{31}\end{pmatrix}=a \rightarrow (-1)\,a \rightarrow \begin{pmatrix}1\\ -a_{21}\\ -a_{31}\end{pmatrix}\rightarrow \frac{1}{a_{11}}\begin{pmatrix}1\\ -a_{21}\\ -a_{31}\end{pmatrix}$$

a_{11}

a'_{22}

2da pasada

$$\begin{bmatrix}1 & & \\ & \boxed{1} & \\ & & 1\end{bmatrix}\begin{bmatrix}1 & \boxed{a'_{12}} & a'_{13}\\ 0 & \boxed{a'_{22}} & a'_{23}\\ 0 & \boxed{a'_{32}} & a'_{33}\end{bmatrix}=$$

$$(-1)\begin{bmatrix}a_{12}\\ a_{22}\\ a_{32}\end{bmatrix}\rightarrow\begin{bmatrix}-a_{12}\\ 1\\ -a_{32}\end{bmatrix}$$

$$\frac{1}{a_{22}}\begin{pmatrix}-a_{12}\\ 1\\ -a_{32}\end{pmatrix}$$

$$L_2 \qquad A''$$

$$\begin{bmatrix}1 & & -\frac{a''_{13}}{a''_{33}}\\ & 1 & -\frac{a_{23}}{a_{33}}\\ & & \frac{1}{a_{33}}\end{bmatrix}\begin{bmatrix}1 & 0 & a''_{13}\\ 0 & 1 & a''_{23}\\ 0 & 0 & a''_{33}\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$(L_3L_2L_1)=A^{-1} \qquad x = A^{-1}b$$

2^{da} pasada

$$\underbrace{\begin{bmatrix} 1 & -\frac{a'_{12}}{a'_{22}} & \\ & \frac{1}{a'_{22}} & \\ & -\frac{a'_{32}}{a'_{22}} & 1 \end{bmatrix}}_{L_2} \overbrace{\begin{bmatrix} 1 & a'_{12} & a'_{13} \\ 0 & 0 & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{bmatrix}}^{L_1 A}_{A'} = \begin{bmatrix} 1 & 0 & a''_{13} \\ 0 & 1 & a''_{23} \\ 0 & 0 & a''_{33} \end{bmatrix}$$

$a_{22} = 0$

A

$$A' = P_{32} A' \longrightarrow \begin{matrix} 1 & a'_{12} & a'_{13} \\ 0 & a'_{32} & a'_{33} \\ 0 & 0 & a'_{23} \end{matrix}$$

$L_2 \cdot P_{32}$

$$\left(L_3 \left(L_2 P_{32}\right)\left(L_1 A\right)\right)$$

$$P_{32} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

15/04/2024

Cuando el punto de pivote de la matriz es 0, hay que pre-multiplicarla por un swapeador de filas.

Transformaciones lineales

10/05/2024

Optimización multivariada

Vamos a trabajar con funciones f: $\mathbb{R}^N \rightarrow \mathbb{R}$
Vamos a pensar (por ahora) en problemas sin restricciones (como una restricción presupuestaria).
Para computar una optimización sin restricciones, la función debe ser C^1.
La condición que define un punto crítico es que el gradiente de f(x) = 0

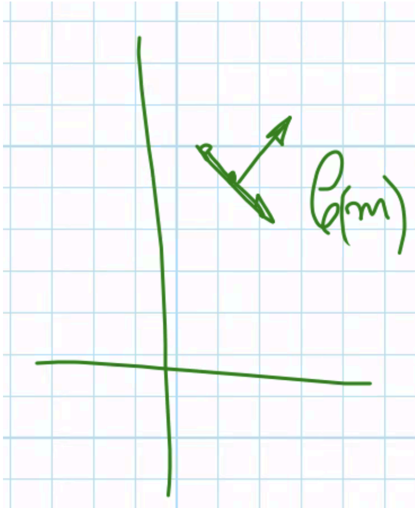
- Sin restricciones
- $\nabla f(x^*) = 0$

$$\begin{cases} \dfrac{f(x+h, y) - f'(x-h, y)}{2h_x} = \dfrac{\partial f}{\partial x} \\[2ex] \dfrac{f(x, y+h) - f(x, y-h)}{2h_y} = \dfrac{\partial f}{\partial y} \end{cases}$$

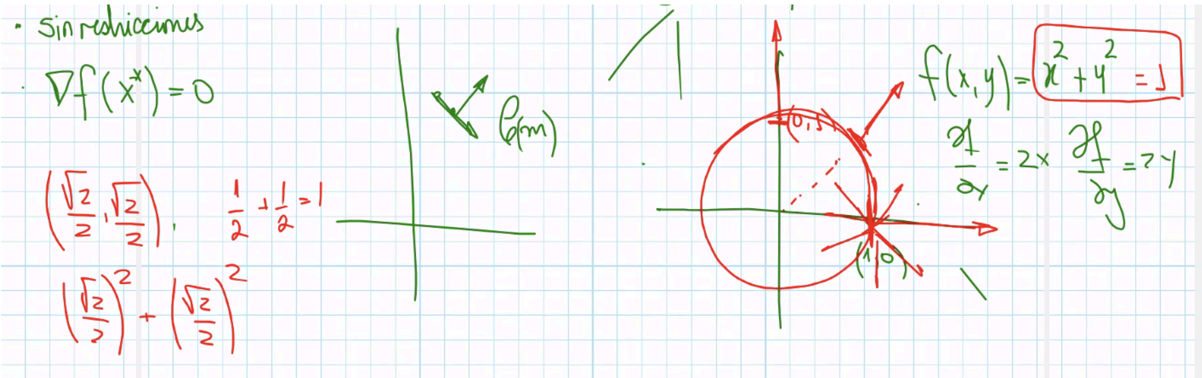
Si tengo una f(x,y), resuelvo las derivadas parciales y meto en Gauss Saidel, puedo luego:

$$\begin{cases} \dfrac{\partial f}{\partial x}(x, y) = 0 \\[2ex] \dfrac{\partial f}{\partial y}(x, y) = 0 \end{cases}$$

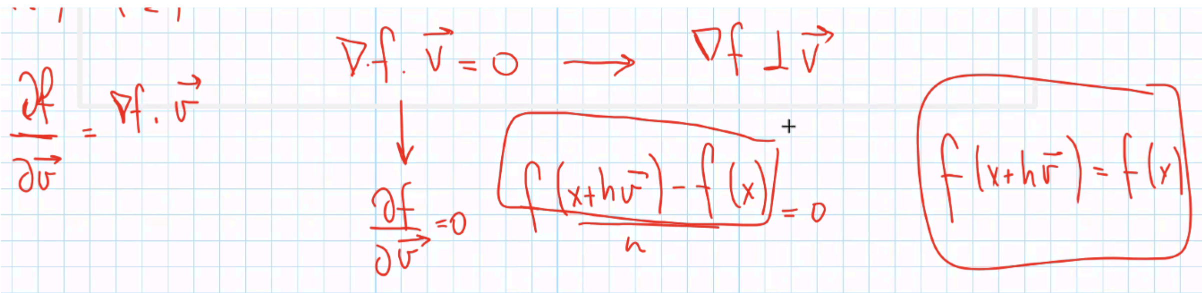
- El gradiente es un vector que indica la dirección de máximo crecimiento de la función
- En la dirección contraria, indica la dirección de máximo decrecimiento
- El gradiente es perpendicular a la curva de nivel o superficie de nivel que pasa por ese punto



Esta es una forma de calcular las curvas de nivel, que es para lo que usamos gradientes



Una curva de nivel cualquiera puede calcularse a partir de la derivada direccional df/dvector = gradiente f * vector ⇒ gradiente f es perpendicular a vector.



Matrices antisimétricas/rotación

Los valores de una matriz antisimétrica tiene valor 0.

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

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -y \\ x \end{pmatrix}$$

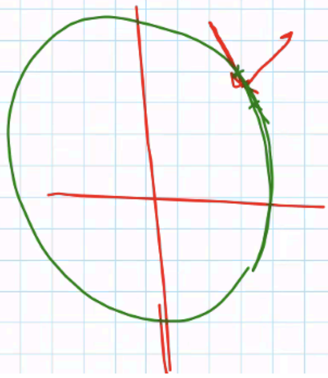
(x_0, y_0) —

$\nabla f(x_0, y_0)$

$\vec{v}_1 = P \, \nabla f(x_0, y_0)$

$\vec{v}_2 = P \, \nabla f(x_1, y_1)$

$(x_1, y_1) = (x_0, y_0) + \delta \vec{v}_1$



13/05/2024

Regresiones múltiples

En regresiones lineales, buscamos minimizar el error cuadrático medio, que es la diferencia entre el valor observado y el valor que el modelo le atribuye a esa y. ⇒ $e_i = y_i - (a + bx_i)$
Lo que buscamos minimizar es el Mean Squared Error (MSE)

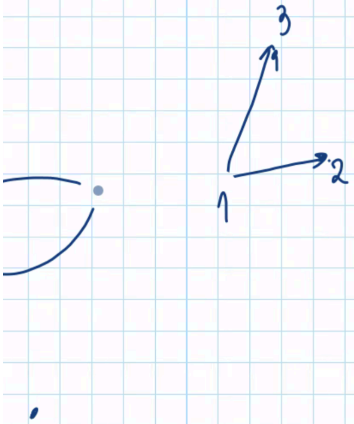
17/05/2024

<u>Polytopes</u>

Tengo una función en $\mathbb{R}^3$
Dados 3 puntos en (x,y), busco qué valores toman en z en la función
Dados los 3 puntos en z, los llamo x1, x2 y x3 y los ordeno de mayor a menor.

Dado x1, busco su perpendicular y lo llamo x1’. Si x1’ es menor a x1, reemplazo a x1 por x1’, vuelvo a ordenar la lista y vuelvo a hacer lo mismo

Si al volver a hacer el caso, x1’ es mayor que x1, no lo sustituyo y hago un shrink. A x1 * ½ y vuelvo a correr el algoritmo.
Corre hasta que el shrink se hace muy chico y/o x1 - x3 similar a 0
La distancia entre puntos la medimos con la distancia euclídea.
Recordar que en r3, los puntos x1,2 y 3 son vectores en $\mathbb{R}^2$
Almacenar los puntos para poder graficarlos.
Debería terminar en un sumidero de gradiente.
Es una versión mejorada de bisección.
Tener en cuenta que si el determinante de los 3 puntos es 0, son multiplos y no puedo usas esos puntos



$$\max\{f_i\} - \min\{f_i\}$$

$$(\overbrace{x_2 - x_1}^{a_1}, \overbrace{y_2 - y_1}^{b_1})$$
$$(\underbrace{x_3 - x_1}, y_3 - y_1)$$

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}$$

20/05/2024

Descomposición ortogonal para shrinking

La idea es que todo vector en $\mathbb{R}^2$ puedo escribirlo con dos versores que son vectores unitarios.
Los versores son:

- xsombrero (i sombrero tambien se lo llama) = (1, 0)
- ysombrero (j sombrero tambien se lo llama) = (0, 1)

Dado v = (2, 3)
2xsombrero + 3 ysombrero
Entonces tengo:

- xsombrero = 2*(1, 0) = (2, 0)
- ysombrero = 3*(0, 1) = (0, 3)

Si sumo ambos vectores, obtengo v

xsombrero e ysombrero entre sí, son perpendiculares y ortogonales, lo que quiere decir que su producto interno es cero. Lo bueno es que puedo calcular los coeficientes a y b usando productos internos.

En el ejemplo anterior:

- v*xsombrero = (2, 3) * (1, 0) = 2
- v*ysombrero = (2, 3) * (0, 1) = 3

Cualquier par de vectores, perpendiculares entre sí y de longitud uno, constituyen una base en $\mathbb{R}^2$

Si quiero que la línea 45grados sea la base, es el vector (1, 1).
Como debe tener norma uno, (1/(2^½), 1/(2^½)) ⇒ ||(1/(2^½))^½ , (1/(2^½))^½ || ⇒
||1/2, 1/2 || ⇒ (½ + ½)^½ = 1

Entonces para que la línea sea 45 grados, el vector es (1/(2^½), 1/(2^½))
La forma de hallar el perpendicular, es multiplicando por la matriz (0, -1, 1, 0), que da como resultado:
(-1/(2^½), 1/(2^½))

v = a*versor1 + b*versor2
v lo tengo, versor1 lo tengo. Me falta a, b y versor2

a = <v, versor1>
b ⇒ <v, versor1>*versor1 =b*versor2

Ejemplo

Dados los puntos: $1 = (1, 1)$; $2 = (1, 0)$; $3 = (0,1)$

$versor1 = x3 - x2 = (0, 1) - (1, 0) = (-1, 1)$ ⇒ lo normalizo para sacar el versor: ⇒ $(-1/(2^{\frac{1}{2}}), 1/(2^{\frac{1}{2}}))$
$v = x3 - x1 = (0, 1) - (1, 1) = (-1, 0)$

$v = a*versor1 + b*versor2$
$a = \langle v, versor1\rangle$ ⇒ $1/(2^{\frac{1}{2}})$

$b = v - a*versor1$ ⇒ $b = (-\frac{1}{2}, -\frac{1}{2})$

24/05/2024

Optimización con restricciones

Buscábamos un par x,y que minimiza la función f(x,y)

Ahora lo que buscamos es movernos por una curva de nivel de g(x,y)

$$\left.\begin{array}{l}\min_{(x,y)} f(x,y) \\ \text{s.a } g(x,y)=0\end{array}\right\} \text{restricciones con igualdad}$$

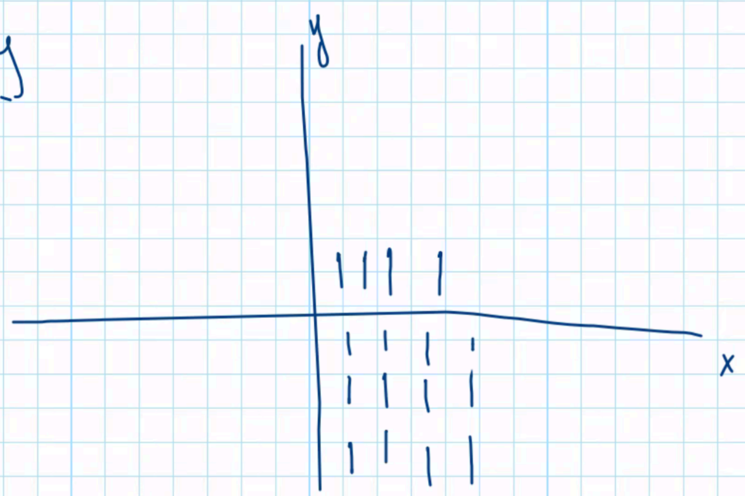
$$g(x,y) = x^2+y^2-1=0$$

En este caso, la función describe una circunferencia de radio 1.

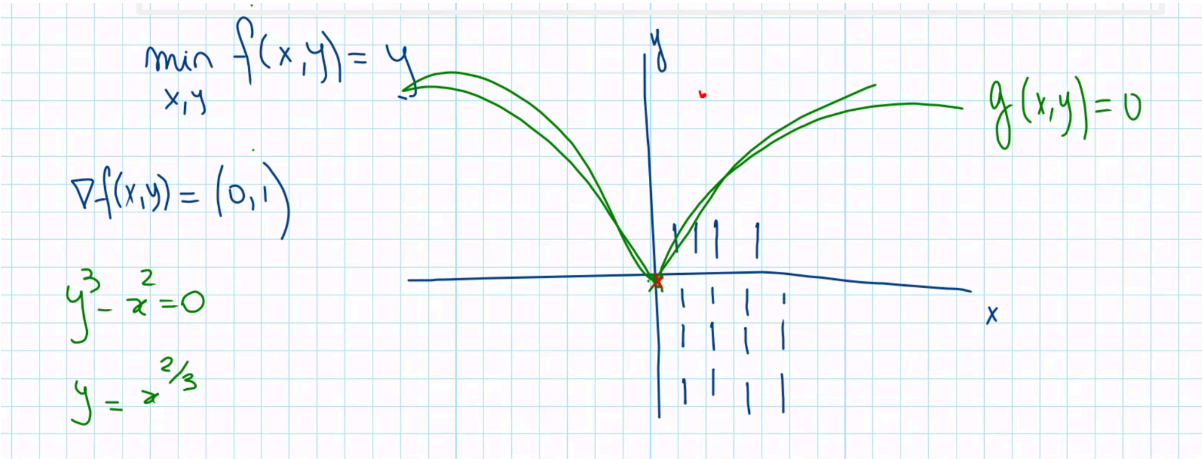
$g(x,y) = x^2 + y^2 - 1 = 0$ (función implícita)

$$\min_{x,y} f(x,y) = y$$

$$\nabla f(x,y) = (0,1)$$



En este caso, siempre son flechas que apuntan hacia arriba. En este caso, no se puede minimizar. Por ende, debo restringir la función



g(x,y) la restringe y el mínimo está en (0,0)

La forma de verlo es sí gráfico las curvas de nivel o el campo gradiente, y le superpongo la restricción

Los multiplicadores de lagrange nos lleva a la tangencia entre las curvas de nivel de f y la restricción g.

1) Lo primero a resolver es encontrar un x_0 sobre g(x,y) = 0 para encontrar la restricción
2) Lo segundo es navegar sobre la restricción g(x,y) = 0 hasta encontrar el mínimo local restringido de f(x,y)

Resolver 1):

- ¿Cómo encuentro un punto sobre la restricción g(x,y)?
  Fijo x, busco y tal que g(x,y) = 0
- Puedo fijar si g(x,y) > 0 o g(x,y) < 0. Como quiero ir hacia el 0, si es mayor a 0, busco el gradiente y voy en contra de él. Si quiero subir porque es menor a 0, voy a favor del gradiente

27/05/2024

$g(x,y) = 0$
$f(x,y) = 0$

Si quiero buscar la res del sistema de ecuaciones, elevo ambas funciones al cuadrado y las sumo, y minimizo la nueva función

$g^2(x,y) + f^2(x,y) = h(x,y) \Rightarrow$ **mínimos cuadrados**

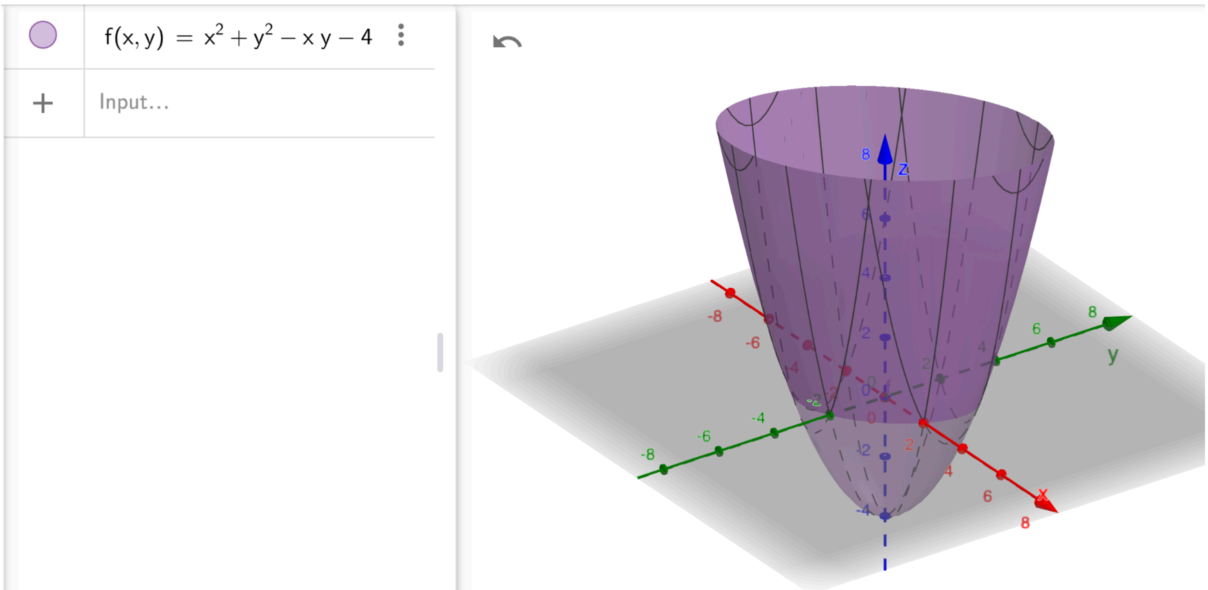
Mínimos cuadrados trata de buscar la recta que mejor aproxima todos los puntos

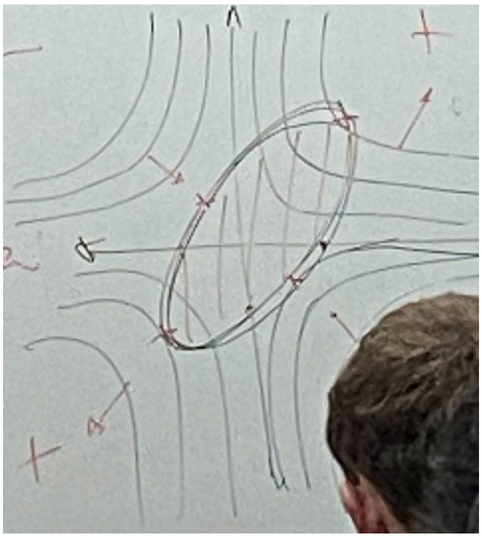
Optimización con restricciones de desigualdad

min $f(x,y)$ con $g(x,y) \geq 0$

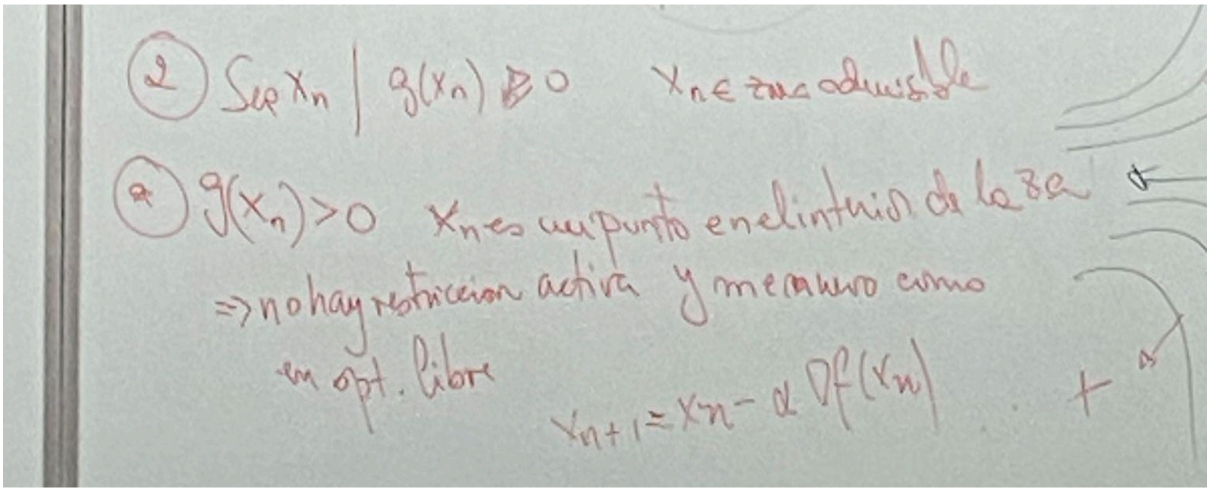
$\min_{x,y}$ $x*y$
Sujeto a: $-(x^2 + y^2 - xy - 4) \geq 0$

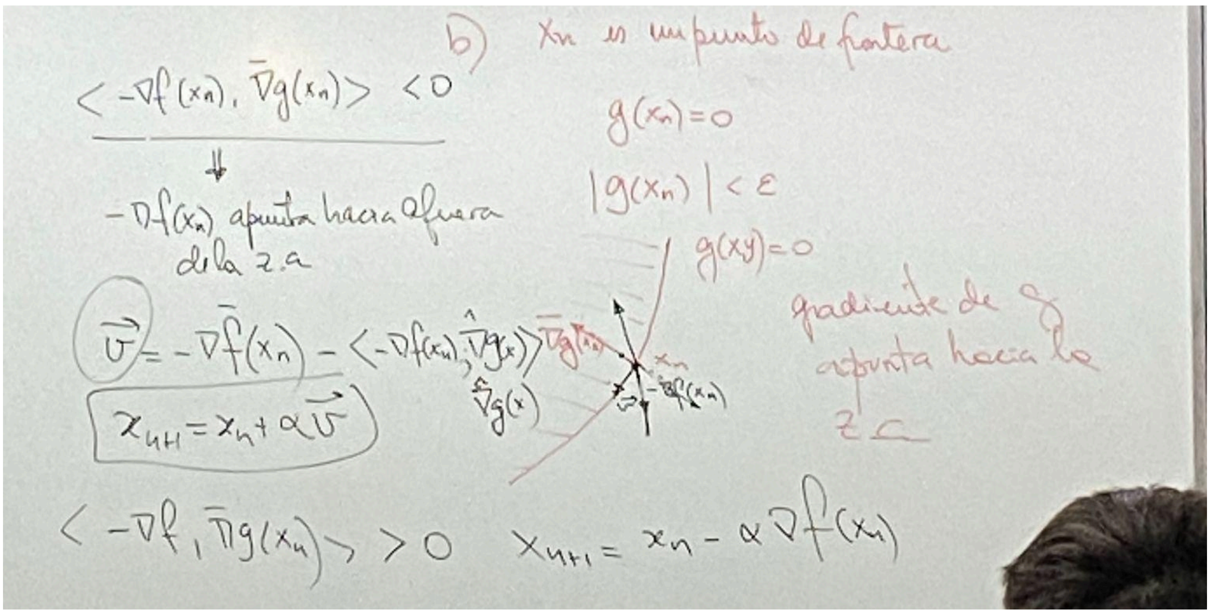
Lo puedo escribir como $(1,x,x^2)$(1 fila 3 columnas) * [-4, 0, 1, 0, -1, 0, 1, 0, 0] * $[1, y, y^2]$(3 filas 1 columna)





Tenemos esa elipse, que representa una curva de nivel de f(x,y).
Buscamos los mínimos y máximos locales.





31/05/2024

Vector

Puede estar en forma de columna o fila.
__init__(self, elems)
Llamo a super y le paso al init de matrix (elems, len(elems), 1)

inner(self,other) Producto interno
outer(self,other) producto matricial de 1 fila por uno de 1 columna

03/06/2024

Portafolio de dos activos

Se puede escribir en función de la cantidad que tenemos de cada papel (n_1, n_2) o se puede escribir como el valor del mismo ⇒ $V = N_1 P_1 + N_2 P_2$

$dv/dt = N_1\, dp_1/dt + N_r\, dpt/dt$ ⇒ Buscamos la variación porcentual del portafolio en el tiempo en porcentaje ⇒ **d representa delta**

$1/v\, dv/dt = N_1/v\, dp_1/dt + N_2/v\, dp_2/dt$

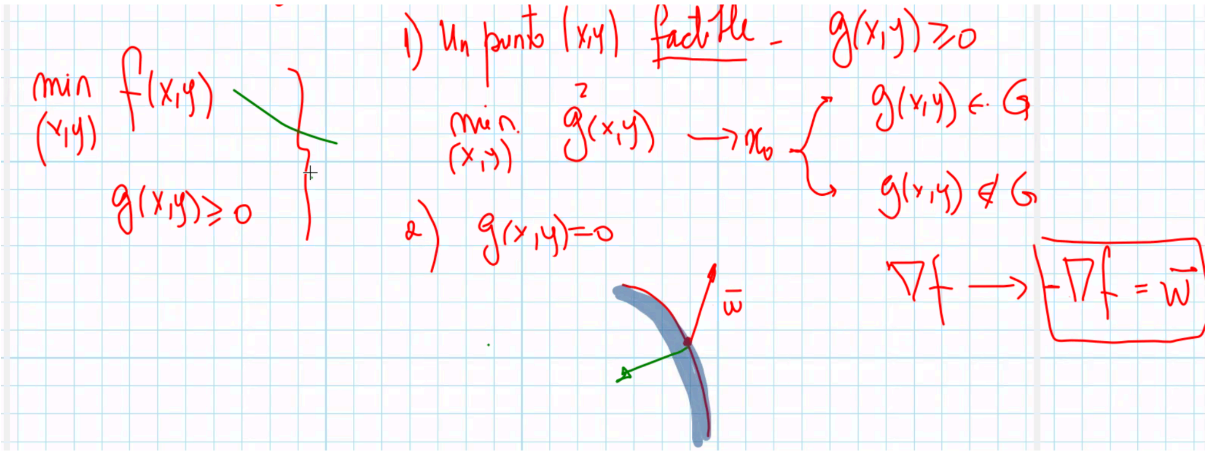
$Return^p = N_1P_1/v *R_1 + N_2P_2/v *R_2$ ⇒ **N_1P_1/v** y **N_2P_2/v** son los weights **W_1 y W_2** del portafolio. A más weight, más importancia.

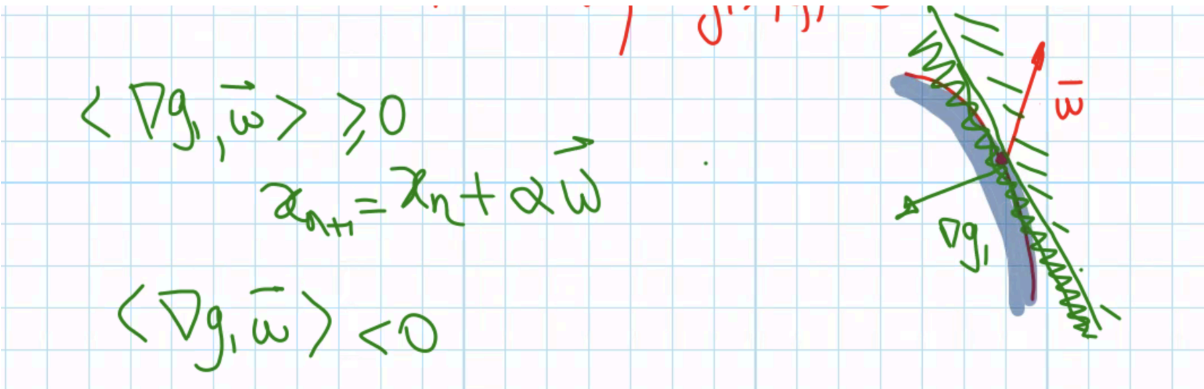
Buscamos la esperanza del $return^p$ ⇒ $E(R^p) = W_1\, E(R_1) + W_2\, E(R_2)$

07/04/2024

Optimización multivariada con restricciones de desigualdad

- Buscar un punto (x,y) factible → g(x,y) ≥ 0
  g^2(x,y) tiene dos casos posibles. g(x,y) ∈ G ó g(x,y) ∉ G
- g(x,y) = 0 (∈ a G) ∇ f → **-∇ f = W**





$$\min f(x,y)$$

$$g_1(x,y) \geq 0$$

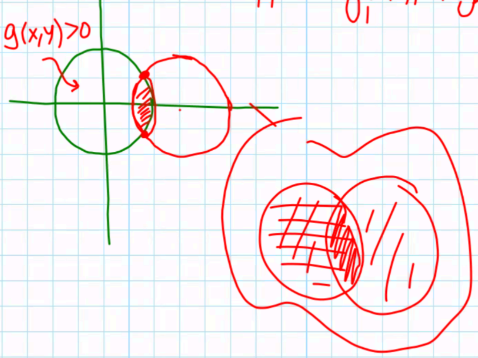
$$g_2(x,y) \geq 0$$

1) Encontrar (x,y) feasible

$$\min^2(g_1(x,y),0) + \min^2(g_2(x,y),0)$$

$$4 - (x-x_0)^2 - (y-y_0)^2 \geq 0$$

$$H(x,y) = g_1^2(x,y) + g_2^2(x,y)$$



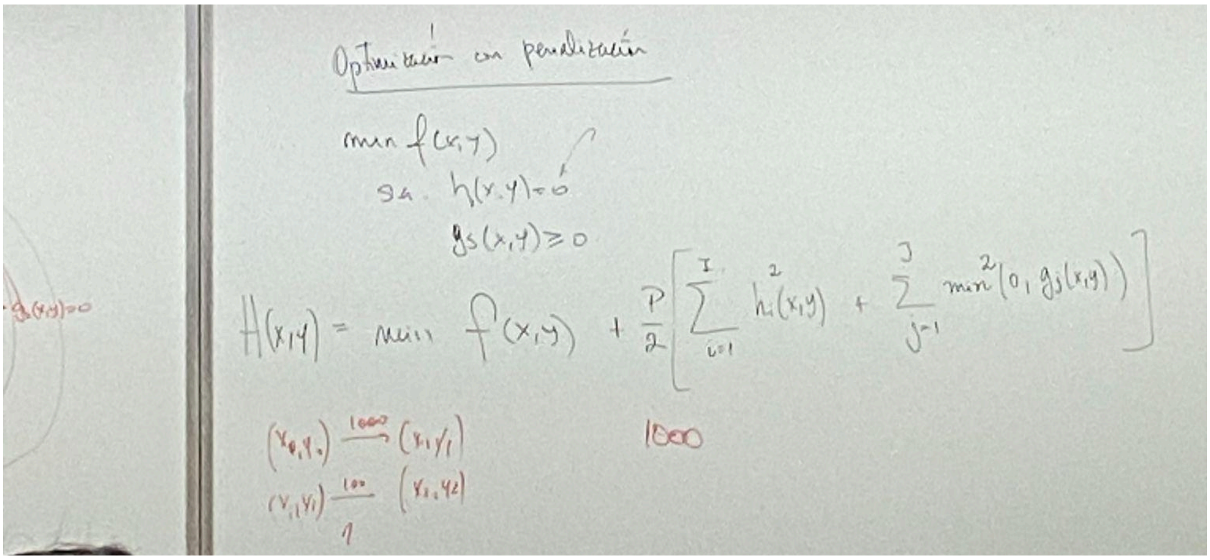
Hay q construir una H en base a todas las restricciones dadas y buscar el mínimo de esa función.

10/06/2024

Cobb Douglas

Optimización sin restricciones

Optimización con penalización

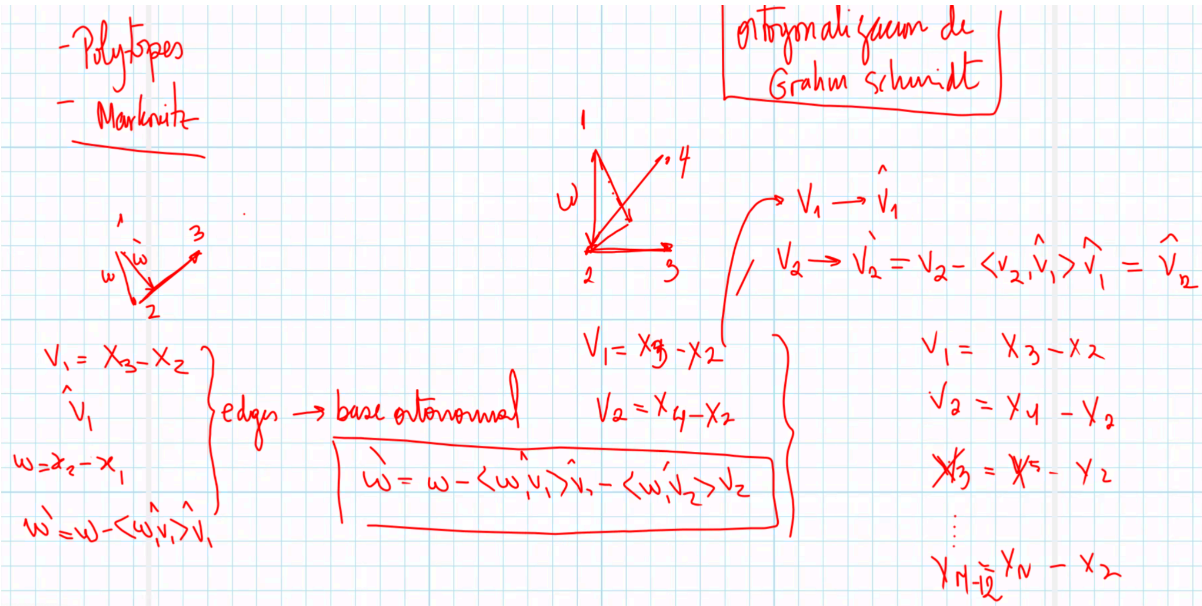


14/06/2024

<u>Polytopes en ℝ^n</u>

En ℝ^2 tenemos 3 puntos. Queremos reflejar el punto 1, perpendicularmente al conjunto que forman los otros dos puntos, que en este caso es la recta que forma los puntos 2, 3.
$X_3 - X_2 = V_1$

Buscamos contruisr una base orto-normal. Los vectores que quedan (en este caso es solo uno), los llamamos **edges**



[ortogonalización de Gram-Schmidt](#) ⇒ buscamos armar una base ortogonal a partir de x vectores.