Programación dinámica: Mínimos cuadrados segmentados

Teoría de Algoritmos I (75.29 / 95.06)

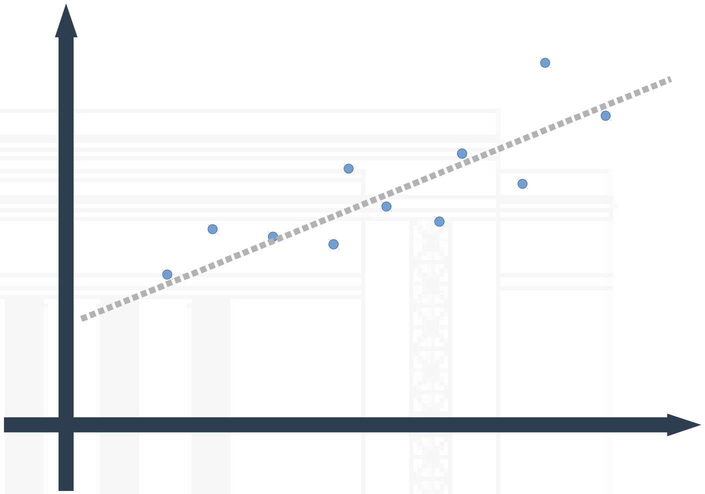
Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Mínimos cuadrados segmentados

Sea

Un set de “n” puntos P=(x,y) en el plano

p_i = (xi,yi) ∈ P y $x_i > x_a$ para todo a>i



Queremos

aproximar mediante segmentos los puntos de P

minimizando el error cometido

con la menor cantidad posibles de segmentos

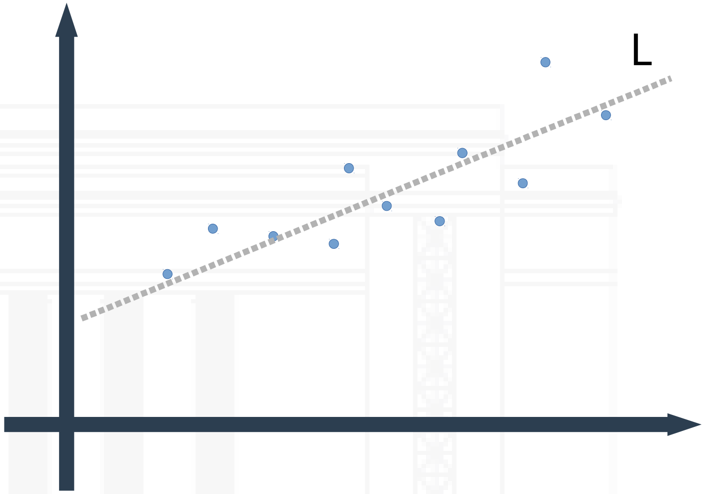
Segmentos

Recta de aproximación

$$L = a x + b$$

$$a = \frac{n \sum_i x_i y_i - \left(\sum_i x_i\right)\left(\sum_i y_i\right)}{n \sum_i x_i^2 - \left(\sum_i x_i\right)^2}$$

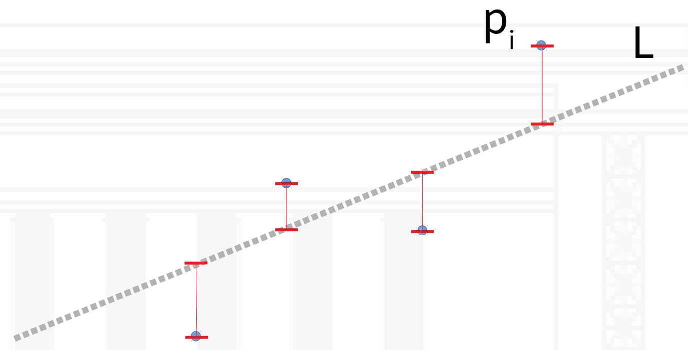
$$b = \frac{\sum_i y_i - a \sum_i x_i}{n}$$



Cálculo de error

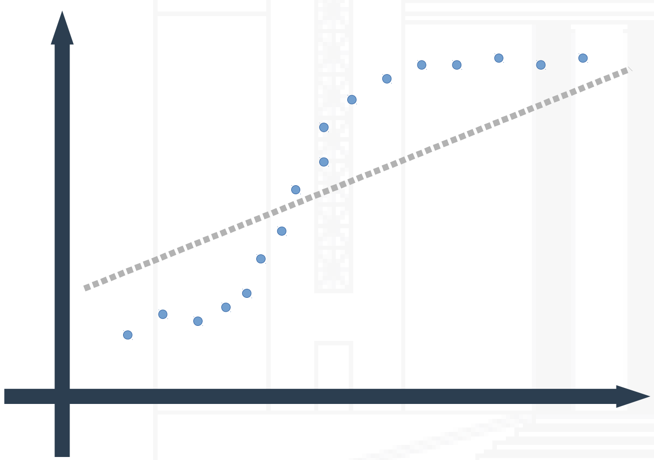
Error cometido

$$E(L,P)=\sum_{i}^{n}\left(y_i-ax_i-b\right)^2$$

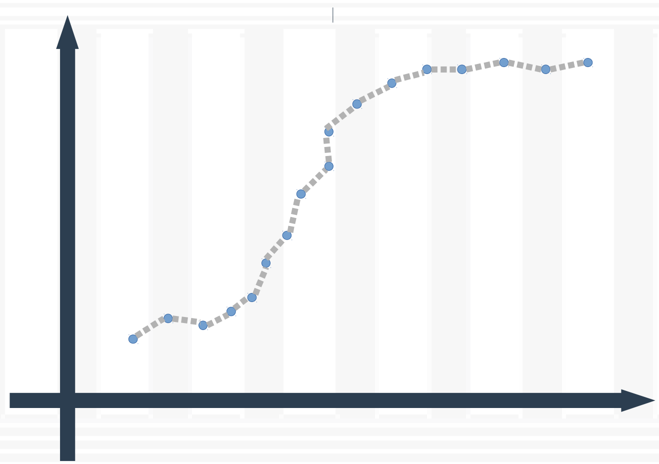


Ejemplo

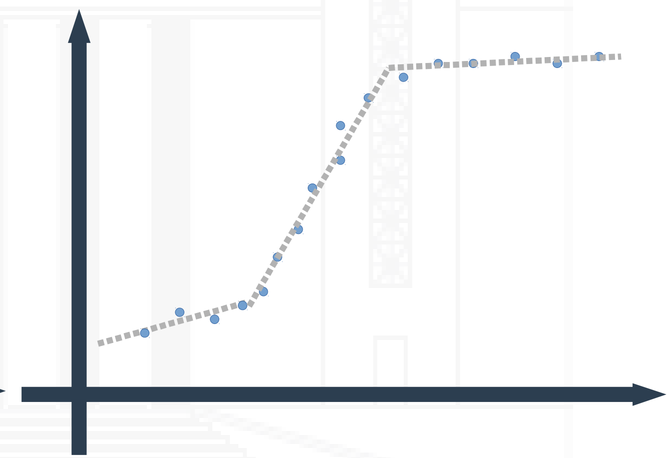
Diferentes soluciones a un mismo set de puntos



1 segmento. Error grande



n-1 segmentos. Cero error



3 segmentos. Error aceptable

Penalización por nuevo segmento

Proponemos un parámetro “C” > 0

Penalidad por cada segmento añadido

El ajuste del mismo establecerá un equilibrio entre segmentos y error cometido

A mayor “C” → menos segmentos

A menor “C” → menos error

Solución por fuerza bruta

Sobre n puntos

Cada punto i>1 puede ser el final de un segmento

Podemos representarlo como un vector de tamaño n

El valor 0 o 1 (es el final del segmento o no)

Existen 2^n combinaciones posibles

Para cada una, calcular los segmentos O(n) y el error cometido O(n)

Complejidad $O(2^n * n)$

Análisis

El punto p_n pertenece al último segmento

Este segmento inicia en un p_x anterior

Si en la solución optima conocemos el último segmento

Conocemos el error $e_{x,n}$

El problema se reduce al subproblema entre los puntos 1 y x-1

$$OPT(n) = e_{x,n} + C + OPT(x-1)$$

Análisis (cont.)

Pero no conocemos el óptimo…

Pero queremos elegir como último segmento aquel que minimice el error general

$$OPT(n) = min_{1 \le x \le n}\left(e_{x,n} + C + OPT(x-1)\right)$$

Podemos generalizar el problema como

$$\begin{cases} OPT(i) = min_{1 \le x \le i}\left(e_{x,i} + C + OPT(x-1)\right) \\ OPT(0) = 0 \end{cases}$$

Solución iterativa

Complejidad Temporal

El cálculo de cada optimo es O(n)

Se calculan n óptimos

→ $O(n^2)$

El calculo de los errores se realizara para todos los pares posibles. Es O(n2)

El calculo del error es O(n)

→ $O(n^3)$ que es la complejidad total

```
OPT[0] = 0

Para todo par i,j con i<=j
        Calcular e[i][j]

Desde j=1 a n

    OPTIMO[j] = +∞

    Desde i=1 a n
        segmento = e[i][j] + C + OPT[i-1]

        si OPTIMO[j] > segmento
            OPTIMO[j] = segmento

Retornar OPT[n]
```

Solución iterativa (cont.)

Complejidad espacial

Almacenamiento de errores: $O(n^2)$

Almacenamiento del optimo $O(n)$

Total: $O(n^2)$

```
OPT[0] = 0

Para todo par i,j con i<=j
        Calcular e[i][j]

Desde j=1 a n

    OPTIMO[j] = +∞

    Desde i=1 a n
        segmento = e[i][j] + C + OPT[i-1]

        si OPTIMO[j] > segmento
            OPTIMO[j] = segmento

Retornar OPT[n]
```

.UBA fiuba
FACULTAD DE INGENIERÍA

Presentación realizada en Abril de 2020