Regresión Lineal con múltiples variables

Aprendiendo a predecir

Caso univariable

Tamaño (m²) x	Precio ($1000) y
2104	460
1416	232
1534	315
852	178
…	…

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Caso Multivariable!

Tamaño(m^2)	Número de camas	Número de pisos	Tiempo de uso(años)	Precio ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Múltiples variables (*features*).

Tamaño(m^2)	Número de camas	Número de pisos	Tiempo de uso(años)	Precio ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Notación:

n = número de features

$x^{(i)}$ = features de dato de entrenamiento *i*.

$x_j^{(i)}$ = valor de feature *j* de dato de entrenamiento *i*.

Hipótesis:

Previamente: $\underline{h_\theta(x) = \cancel{\theta_0 + \theta_1 x}}$

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$

E.g. $\underline{h_\theta(x)} = \underline{80} + \underline{0.1} x_1 + \underline{0.01} x_2 + 3 x_3 - 2 \underbrace{x_4}_{\text{age}}$

Hipótesis:

Previamente: $h_\theta(x) = \theta_0 + \theta_1 x$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Por conveniencia de notación, defina $x_0 = 1$

Regresión lineal multivariable
(Multivariate linear regression)

Regresión lineal con multiples variables

Descenso de gradiente para multiples variables

Hipotesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parametros: $\theta_0, \theta_1, \ldots, \theta_n$

Función de costo:

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Descenso de gradiente:

Repetir $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

$\}$ (simultaneamente actualizar para $j = 0, \ldots, n$

Descenso de gradiente

Previamente (n=1):

Repeat $\{$

$$\theta_0 := \theta_0 - \alpha \underbrace{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial \theta_0} J(\theta)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

(simultaneamente actualizar θ_0, θ_1)

$\}$

Nuevo algoritmo $(n \geq 1)$:

Repetir $\{$

$$\theta_j := \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

(simultaneamente actualizar θ_j para $j = 0, \ldots, n$)

$\}$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)}$$

...

Regresion lineal con multiples variables

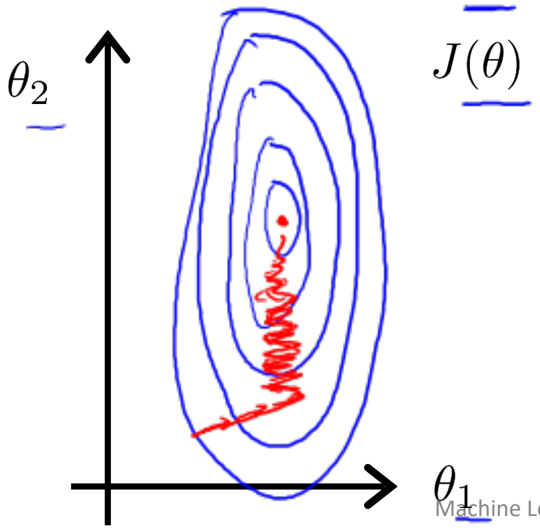
Descenso de Gradiente en la practica I: Escalamiento de Features

Escalamiento de features

Idea: Asegurarse que features estan en escala similar.

E.g. x_1 = tamaño (0-2000 m2)

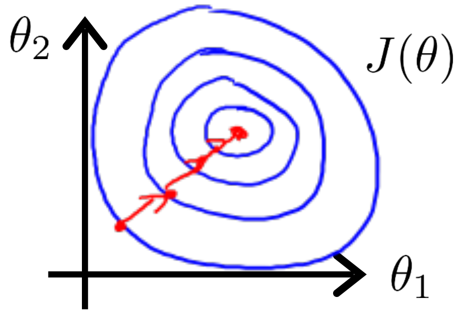
x_2 = número de camas(1-5)



$$x_1 = \frac{\text{tamaño (m2)}}{2000}$$

$$x_2 = \frac{\text{número de camas}}{5}$$

$0 \leq x_1 \leq 1 \qquad 0 \leq x_2 \leq 1$



Escalamiento de features (2)

Transformar cada feature a rango aproximado de:

$$-1 \leq x_i \leq 1$$

De esta forma el descenso de gradiente funciona!.

Otros tipos: *Normalización de media*

Normalización de media

Reemplazar x_i con $x_i - \mu_i$ para obtener features con media igual a cero.

E.g. $x_1 = \frac{size-1000}{2000}$

$x_2 = \frac{\#bedrooms-2}{5}$

$$-0.5 \le x_1 \le 0.5, -0.5 \le x_2 \le 0.5$$

Regresion lineal con multiples variables

Descenso de gradiente en la practica II: Tasa de aprendizaje

Descenso de gradiente

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- “Debugging”: ¿Cómo asegurarse que descenso de gradiente trabaja correctamente?
- ¿Cómo escoger la tasa de aprendizaje α?

Asegurar que descenso de gradiente está trabajando OK

$$\min_{\theta} J(\theta)$$

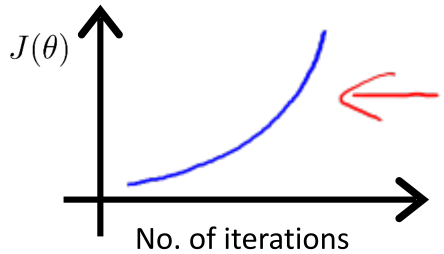
0 100 200 300 400

No. of iterations

Ejemplo de convergencia “automatica”:

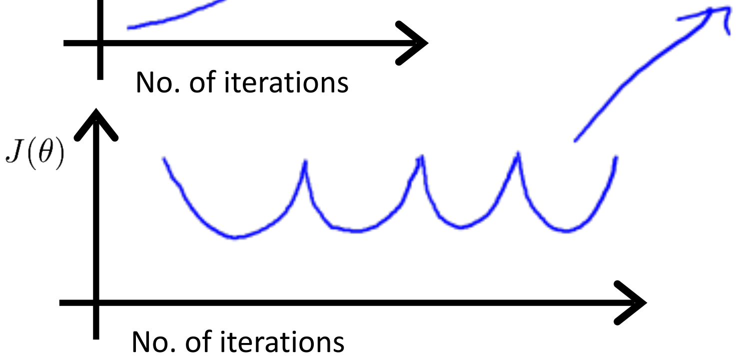
Declarar convergencia, si $J(\theta)$ decrementa por menos de 10^{-3} en una iteración.

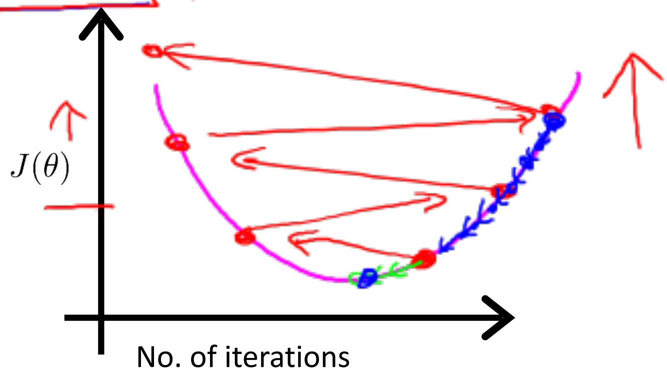
Asegurando que descenso de gradiente trabaje OK.



Descenso de gradiente no trabajando.

Use α mas pequeño.





- Para una tasa correcta α, $J(\theta)$ deberia decrementar.
- Pero si α es muy pequeña, el descenso sera muy lento para converger.

Resumen:

- Si α es muy pequeña: lenta convergencia.
- Si α es muy grande: $J(\theta)$ podria no decrementar en cada iteración, por lo que podría no converger.

Para escoger α, try

$$\ldots, 0.001, \qquad , 0.01, \qquad , 0.1, \qquad , 1, \ldots$$

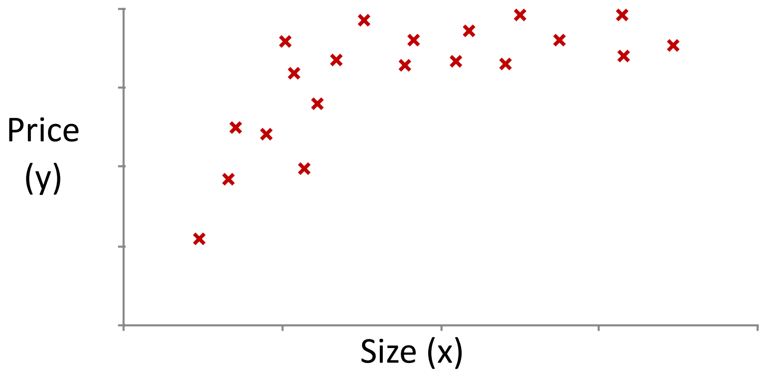
Usted implementa el descenso de gradiente a una base de datos de 100 millones de registros, la curva de error decae adecuadamente pero demora demasiado. ¿Que sugeriría para acelerar el aprendizaje de regreso lineal?

- ○ Incrementar la tasa de aprendizaje.
- ○ Decrementar la tasa de aprendizaje.
- ○ Tomar una muestra aleatoria y ajustar curva, para inicializar regresor.
- ○ Cambiar de signo al gradiente en el código.

Regresion lineal con multiples variables

Features y regresión polinomial

Regresión polinomial



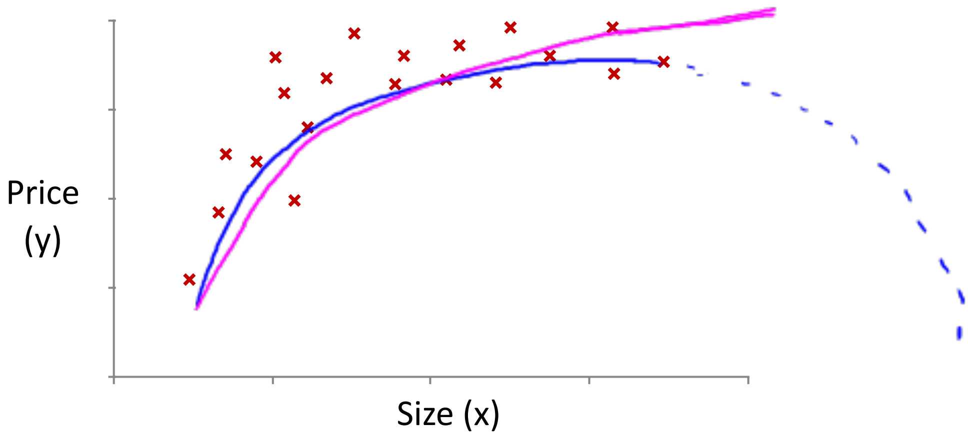
$$\theta_0 + \theta_1 x + \theta_2 x^2$$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$
$$= \theta_0 + \theta_1 (size) + \theta_2 (size)^2 + \theta_3 (size)^3$$

$$x_1 = (size)$$
$$x_2 = (size)^2$$
$$x_3 = (size)^3$$

Elección de features



$$h_\theta(x) = \theta_0 + \theta_1 (size) + \theta_2 (size)^2$$

$$h_\theta(x) = \theta_0 + \theta_1 (size) + \theta_2 \sqrt{(size)}$$

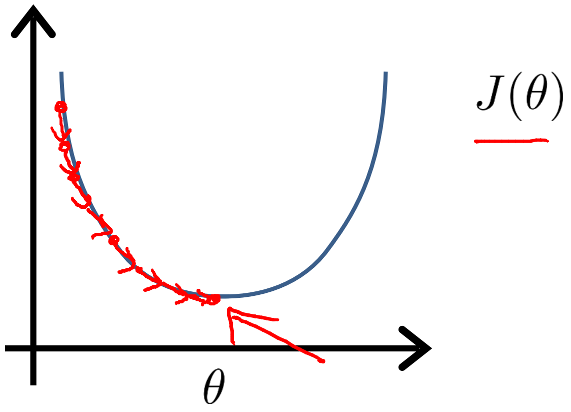
En Asgard, el tiempo que demora en caer un objeto en caida libre es dependiente del cubo de la altura. Usted documenta miles de ejemplos. ¿Cómo obtendría una fórmula mas precisa dado que usted desconoce la física de Asgard usando una regresión lineal?.

- ○ Ingresar como entrada el cubo de altura y la altura.
- ○ Ingresar como entrada el cubo y el cuadrado de altura así como la propia altura.
- ○ Ingresa el cubo del tiempo de demora de objetos.
- ○ Ingresar como entrada lo mismo que segunda opción además del product de altura y tiempo.

Regresión lineal con múltiples variables

Ecuación Normal

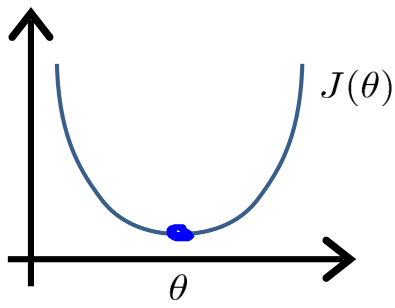
Descenso
de gradiente



Ecuación Normal: Metodo para resolver θ analíticamente.

Intuición: Si 1D$(\theta \in \mathbb{R})$

$$J(\theta) = a\theta^2 + b\theta + c$$



$\theta \in \mathbb{R}^{n+1}$ $\quad J(\theta_0, \theta_1, \ldots, \theta_m) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

$\frac{\partial}{\partial \theta_j} J(\theta) = \cdots = 0$ (for todo j)

Resolver $\theta_0, \theta_1, \ldots, \theta_n$

Ejemplo: $m = 4.$

x_0	Tamaño (m²) x_1	Número de camas x_2	Número de pisos x_3	Tiempo de uso (años) x_4	Precio ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

$$\theta = (X^T X)^{-1} X^T y$$

$$\theta = (X^T X)^{-1} X^T y$$

$(X^T X)^{-1}$ es inversa de matriz $X^T X$.

Octave: `pinv(X'*X)*X'*y`

pinv$(X^T * X) * X^T * y$

$\theta = (X^T X)^{-1} X^T y$

$\min_{\theta} J(\theta)$

~~Feature Scaling~~

$0 \leq x_1 \leq 1$

$0 \leq x_2 \leq 1000$

$0 \leq x_3 \leq 10^{-5}$ ✓

m **ejemplos de entrenamiento,** n **features.**

Descenso de gradiente

- Necesita escoger α.
- Necesita muchas iteraciones.
- Trabaja bien aun cuando n es muy grande (10^6).

Ecuación normal

- No necesita escoger α.
- No necesita iterar.
- Necesita computar $(X^T X)^{-1}$
- Lento si n es grande (10^4 , 10^5).