Tema 2

Regresión Lineal

Gonzalo A. Aranda-Corral

Ciencias de la Computación e Inteligencia Artificial
Universidad de Huelva

19 de octubre de 2020

Asistencia

Por favor, regístrar vuestra asistencia en la siguiente dirección:

`http://opendatalab.uhu.es/aplicaciones/asistencia`

Contenido

Índice

Introducción

- Aprendizaje numérico .vs. simbólico
- Predicción de valores continuos (reales)
- Aproximación lineal (multidimensional)

Introducción

- Aprendizaje numérico .vs. simbólico
- Predicción de valores continuos (reales)
- Aproximación lineal (multidimensional)

Introducción

- Aprendizaje numérico .vs. simbólico
- Predicción de valores continuos (reales)
- Aproximación lineal (multidimensional)

Introducción

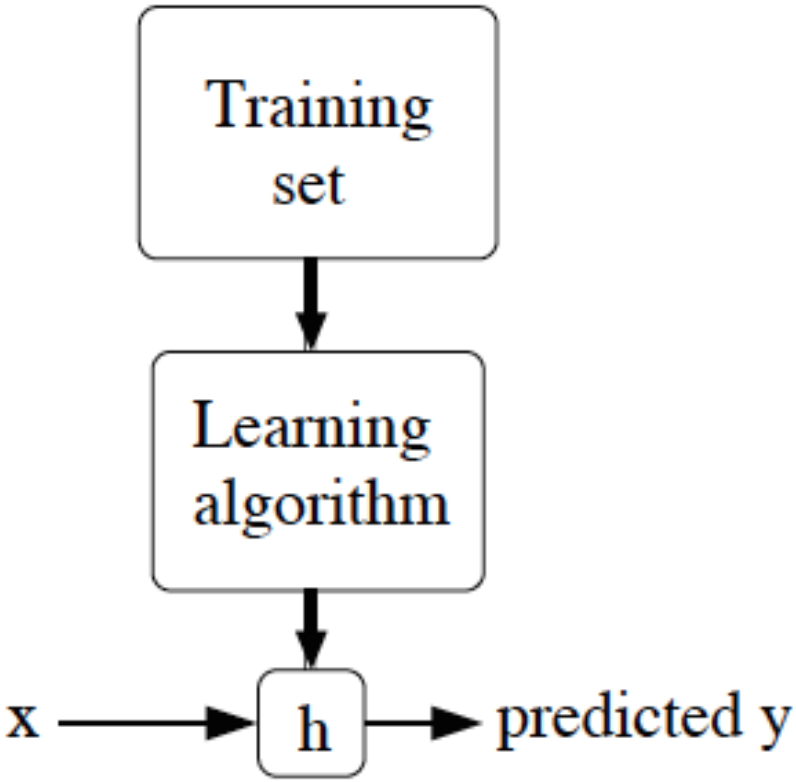
Un problema de aprendizaje numérico se podría enunciar como:

"dado un conjunto de datos de entrenamiento, encontrar la función $h(x) : y' = f(x)$, tal que $h(x)$ se ajusta lo más posible a los valores de y."

A **h**(**x**) se le denomina **Hipótesis**

Introducción

Gráficamente:

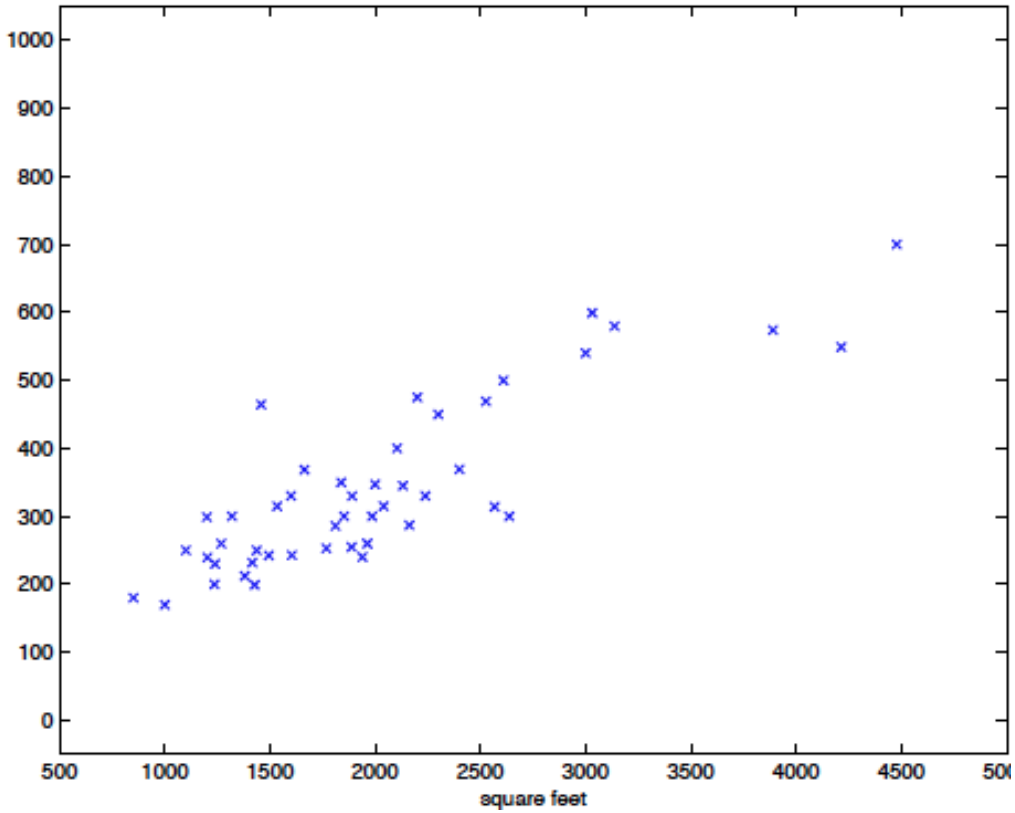


Introducción

Planteamos un ejemplo sobre la estimación del precio de inmuebles en función de su tamaño, en una zona.

Tenemos la siguiente tabla:

Tamaño	Precio
2104	400
1416	232
1539	315
852	178
1940	240
⋮	⋮



Introducción

Por tanto, la pregunta es:

¿ Cómo averigüamos $h(x)$?

Índice

Notación

Antes de empezar vamos a establecer la notación a usar a partir de ahora.

- m $\equiv$ número de ejemplos de entrenamiento
- n $\equiv$ número de atributos
- $(x, y) \equiv$ Ejemplo de entrenamiento
- $(x^j, y^j) \equiv$ Ejemplo de la fila j
- x $\equiv$ variable de entrada / característica / atributo / ...
- y $\equiv$ variable de salida / objetivo / target / ...

Notación

Antes de empezar vamos a establecer la notación a usar a partir de ahora.

- m $\equiv$ número de ejemplos de entrenamiento
- n $\equiv$ número de atributos
- (x, y) $\equiv$ Ejemplo de entrenamiento
- (x^j, y^j) $\equiv$ Ejemplo de la fila j
- x $\equiv$ variable de entrada / característica / atributo / ...
- y $\equiv$ variable de salida / objetivo / target / ...

Notación

Antes de empezar vamos a establecer la notación a usar a partir de ahora.

- m $\equiv$ número de ejemplos de entrenamiento
- n $\equiv$ número de atributos
- (x, y) $\equiv$ Ejemplo de entrenamiento
- (x^j, y^j) $\equiv$ Ejemplo de la fila j
- x $\equiv$ variable de entrada / característica / atributo / ...
- y $\equiv$ variable de salida / objetivo / target / ...

Notación

Antes de empezar vamos a establecer la notación a usar a partir de ahora.

- m $\equiv$ número de ejemplos de entrenamiento
- n $\equiv$ número de atributos
- $(x, y) \equiv$ Ejemplo de entrenamiento
- $(x^j, y^j) \equiv$ Ejemplo de la fila j
- x $\equiv$ variable de entrada / característica / atributo / . . .
- y $\equiv$ variable de salida / objetivo / target / . . .

Notación

Antes de empezar vamos a establecer la notación a usar a partir de ahora.

- m $\equiv$ número de ejemplos de entrenamiento
- n $\equiv$ número de atributos
- $(x, y) \equiv$ Ejemplo de entrenamiento
- $(x^j, y^j) \equiv$ Ejemplo de la fila j
- x $\equiv$ variable de entrada / característica / atributo / ...
- y $\equiv$ variable de salida / objetivo / target / ...

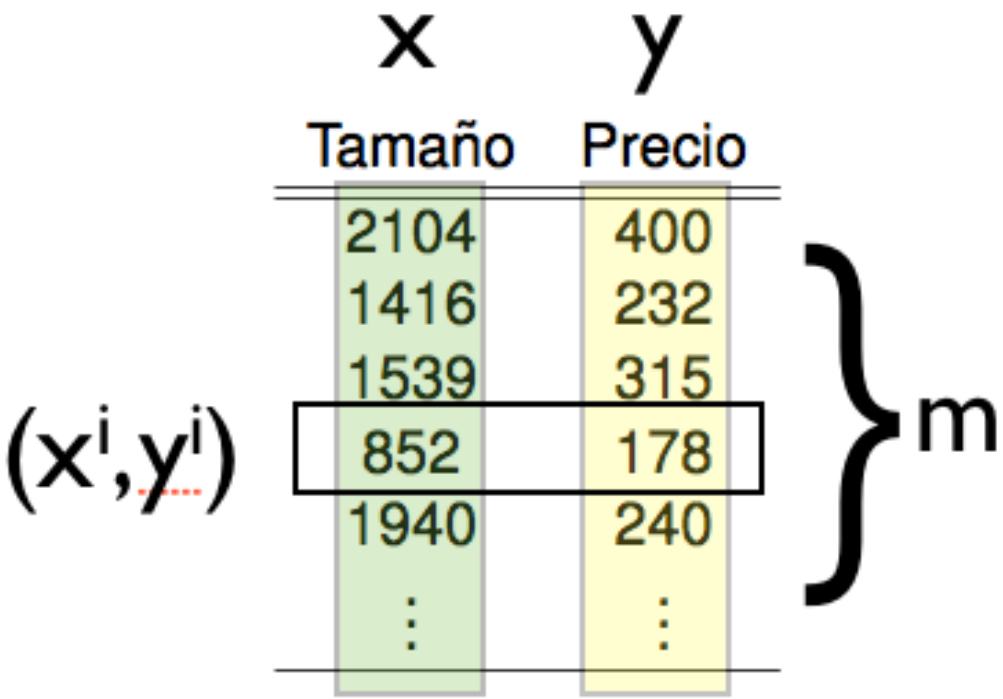
Notación

Antes de empezar vamos a establecer la notación a usar a partir de ahora.

- m $\equiv$ número de ejemplos de entrenamiento
- n $\equiv$ número de atributos
- (x, y) $\equiv$ Ejemplo de entrenamiento
- (x^j, y^j) $\equiv$ Ejemplo de la fila j
- x $\equiv$ variable de entrada / característica / atributo / ...
- y $\equiv$ variable de salida / objetivo / target / ...

Notación

En el ejemplo:



Teoría

Para determinar la hipótesis, **suponemos que es lineal**

$$h(x) = \theta_0 + \theta_1 x$$

Para un dataset con n atributos, podemos escribir:

$$h(\vec{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$$

Por comodidad, consideraremos $\mathbf{x_0} = \mathbf{1}$: $h(\vec{x}) = \theta_0 \mathbf{x_0} + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$

Finalmente, se puede escribir la hipótesis como:

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i = \theta^T \vec{x}$$

Teoría

Para determinar la hipótesis, **suponemos que es lineal**

$$h(x) = \theta_0 + \theta_1 x$$

Para un dataset con n atributos, podemos escribir:

$$h(\vec{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$$

Por comodidad, consideraremos $\mathbf{x_0} = \mathbf{1}$: $h(\vec{x}) = \theta_0 \mathbf{x_0} + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$

Finalmente, se puede escribir la hipótesis como:

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i = \theta^T \vec{x}$$

Teoría

Para determinar la hipótesis, **suponemos que es lineal**

$$h(x) = \theta_0 + \theta_1 x$$

Para un dataset con *n* atributos, podemos escribir:

$$h(\vec{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$$

Por comodidad, consideraremos $\mathbf{x_0} = \mathbf{1}$: $h(\vec{x}) = \theta_0 \mathbf{x_0} + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$

Finalmente, se puede escribir la hipótesis como:

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i = \theta^T \vec{x}$$

Teoría

Para determinar la hipótesis, **suponemos que es lineal**

$$h(x) = \theta_0 + \theta_1 x$$

Para un dataset con n atributos, podemos escribir:

$$h(\vec{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$$

Por comodidad, consideraremos $\mathbf{x_0} = \mathbf{1}$: $h(\vec{x}) = \theta_0 \mathbf{x_0} + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$

Finalmente, se puede escribir la hipótesis como:

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i = \theta^T \vec{x}$$

Teoría

Para determinar la hipótesis, **suponemos que es lineal**

$$h(x) = \theta_0 + \theta_1 x$$

Para un dataset con n atributos, podemos escribir:

$$h(\vec{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$$

Por comodidad, consideraremos $\mathbf{x_0} = \mathbf{1}$: $h(\vec{x}) = \theta_0 \mathbf{x_0} + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$

Finalmente, se puede escribir la hipótesis como:

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i = \theta^T \vec{x}$$

Teoría

Para determinar la hipótesis, **suponemos que es lineal**

$$h(x) = \theta_0 + \theta_1 x$$

Para un dataset con n atributos, podemos escribir:

$$h(\vec{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$$

Por comodidad, consideraremos $\mathbf{x_0} = \mathbf{1}$: $h(\vec{x}) = \theta_0 \mathbf{x_0} + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n$

Finalmente, se puede escribir la hipótesis como:

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i = \theta^T \vec{x}$$

Teoría

Por tanto, ya conocemos $h(\vec{x})$,

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i \quad = \quad \theta^T \vec{x}$$

sólo nos queda determinar los coeficientes: θ_i.

¿ Cómo ?

Los coeficientes deben de ser elegidos de forma que **se minimize el error.**

Teoría

Por tanto, ya conocemos $h(\vec{x})$,

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i \quad = \quad \theta^T \vec{x}$$

sólo nos queda determinar los coeficientes: θ_i.

¿ Cómo ?

Los coeficientes deben de ser elegidos de forma que **se minimize el error.**

Por tanto, ya conocemos $h(\vec{x})$,

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i \quad = \quad \theta^T \vec{x}$$

sólo nos queda determinar los coeficientes: θ_i.

¿ Cómo ?

Los coeficientes deben de ser elegidos de forma que **se minimize el error.**

Por tanto, ya conocemos $h(\vec{x})$,

$$h(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i \quad = \quad \theta^T \vec{x}$$

sólo nos queda determinar los coeficientes: θ_i.

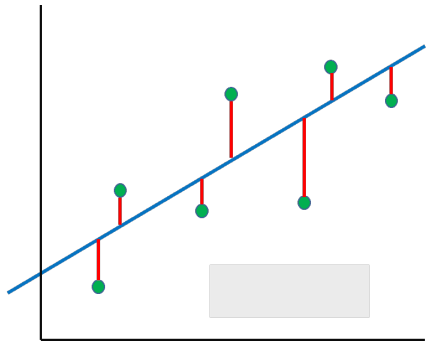
¿ Cómo ?

Los coeficientes deben de ser elegidos de forma que **se minimize el error.**

Minimización del error

¿Qué es el error?

$$Error = Prediccion - ValorReal = h(x) - y(x)$$

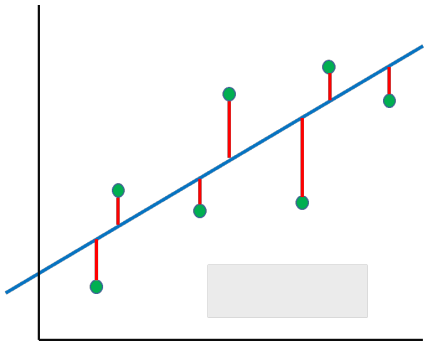


MALA DEFINICIÓN

Minimización del error

¿Qué es el error?

$$Error = Prediccion - ValorReal = h(x) - y(x)$$

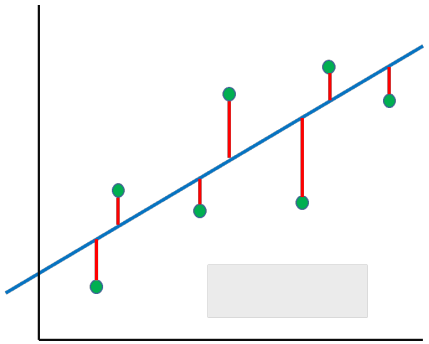


MALA DEFINICIÓN

Minimización del error

¿Qué es el error?

$$Error = Prediccion - ValorReal = h(x) - y(x)$$

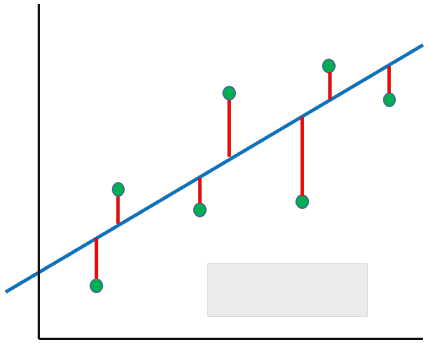


MALA DEFINICIÓN

Minimización del error

¿Qué es el error?

$$Error = Prediccion - ValorReal = h(x) - y(x)$$

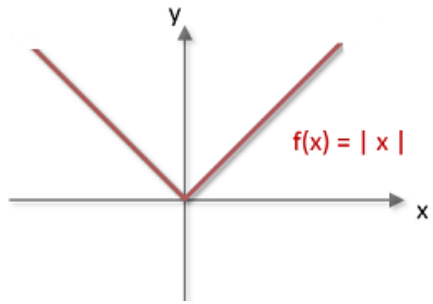


MALA DEFINICIÓN

Minimización del error

¿Qué es el error?

$$Error = Prediccion - ValorReal = |h(x) - y(x)|$$

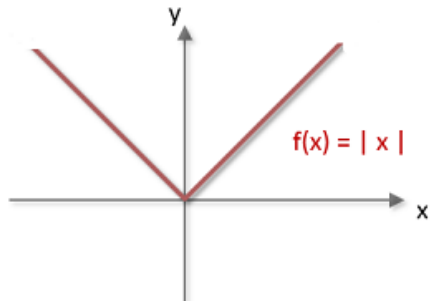


MALA DEFINICIÓN

Minimización del error

¿Qué es el error?

$$Error = Prediccion - ValorReal = |h(x) - y(x)|$$

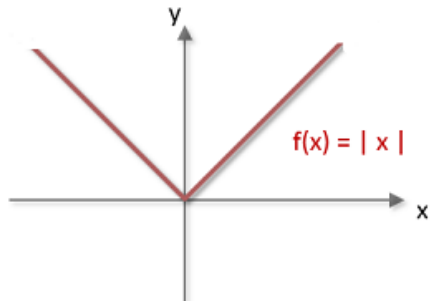


MALA DEFINICIÓN

Minimización del error

¿Qué es el error?

$$Error = Prediccion - ValorReal = |h(x) - y(x)|$$

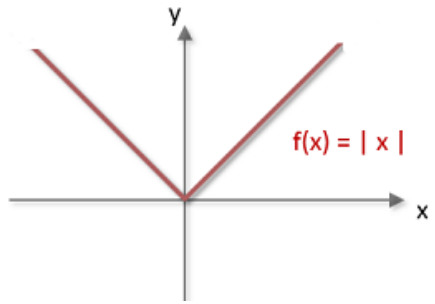


MALA DEFINICIÓN

Minimización del error

¿Qué es el error?

$$Error = Prediccion - ValorReal = |h(x) - y(x)|$$



MALA DEFINICIÓN

Minimización del error

¿Qué es el error?

$$ErrorCuadratico = \frac{1}{2}(h(x) - y(x))^2$$

Existen más definiciones de errores, pero nosotros vamos a usar esta.

Minimización del error

¿Qué es el error?

$$ErrorCuadratico = \frac{1}{2}(h(x) - y(x))^2$$

Existen más definiciones de errores, pero nosotros vamos a usar esta.

Minimización del error

Para obtener el error total, sumamos para todos los ejemplos(m):

$$ErrorCuadraticoTotal = \frac{1}{2}\sum_{j=1}^{m}(h_\theta(x^j) - y^j)^2$$

ya que nuestro objetivo es minimizar el error total:

$$min_\theta \left(\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)^2\right)$$

Minimización del error

Para obtener el error total, sumamos para todos los ejemplos(m):

$$ErrorCuadraticoTotal = \frac{1}{2}\sum_{j=1}^{m}(h_\theta(x^j) - y^j)^2$$

ya que nuestro objetivo es minimizar el error total:

$$min_\theta \left(\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)^2\right)$$

Minimización del error

Si llamamos $J(\theta) = \frac{1}{2}\sum_{j=1}^{m}(h_\theta(x^j) - y^j)^2$

por lo que buscamos:

Los valores de θ que hagan que se cumpla $min_\theta\ J(\theta)$

Minimización del error

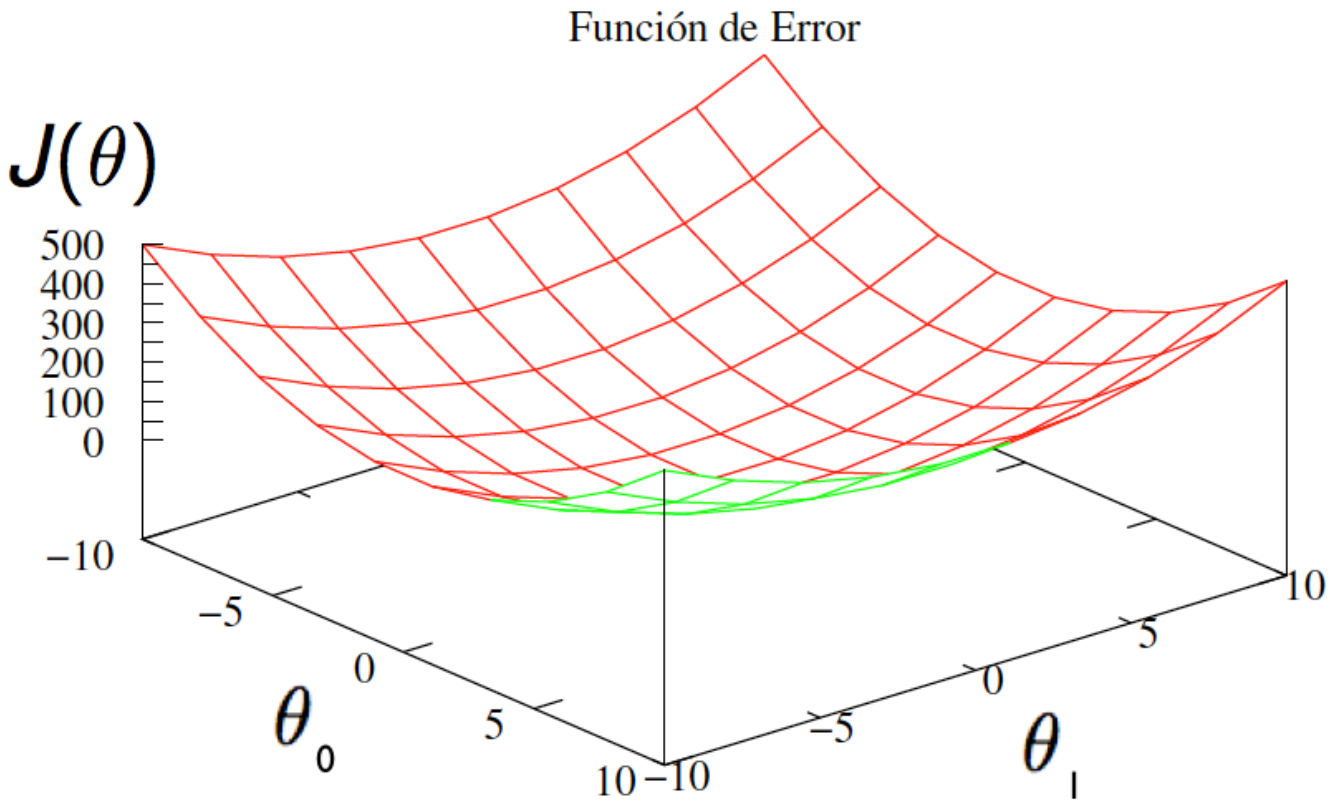
Si llamamos $J(\theta) = \frac{1}{2}\sum_{j=1}^{m}(h_\theta(x^j) - y^j)^2$

por lo que buscamos:

Los valores de θ que hagan que se cumpla $min_\theta\ J(\theta)$

Minimización del error

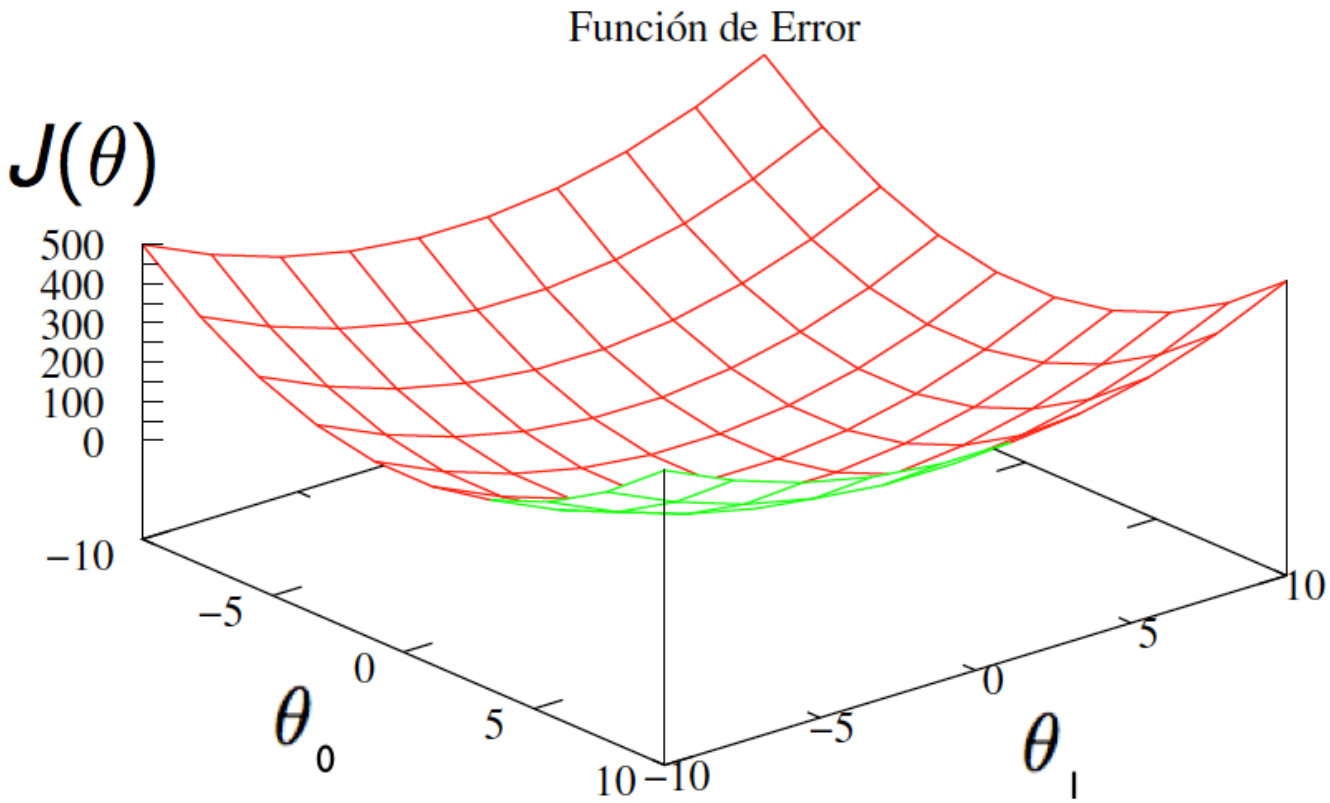
Podemos dibujar el error como una superficie, en el espacio de $\vec{\theta}$



¿Dónde está el mínimo?

Minimización del error

Podemos dibujar el error como una superficie, en el espacio de $\vec{\theta}$



¿Dónde está el mínimo?

Índice

Algoritmo

Debemos de construir un **algoritmo** para esa minimización

- La 1^a aproximación la realizamos mediante **Búsqueda**
- Comenzamos inicializando θ con algún valor. Ej: $\vec{\theta} = \vec{0}$
- Vamos cambiando los valores de θ de forma que disminuya $J(\theta)$
  - Hay que asegurar que los nuevos valores de $\vec{\theta}$ hacen disminuir J
- Repetimos el cambio de valores hasta alcanzar el mínimo.

Esta es la idea del algoritmo "**Descenso de Gradiente**"

Algoritmo

Debemos de construir un **algoritmo** para esa minimización

- La 1^a aproximación la realizamos mediante **Búsqueda**
- Comenzamos inicializando θ con algún valor. Ej: $\vec{\theta} = \vec{0}$
- Vamos cambiando los valores de θ de forma que disminuya $J(\theta)$
  - Hay que asegurar que los nuevos valores de $\vec{\theta}$ hacen disminuir J
- Repetimos el cambio de valores hasta alcanzar el mínimo.

Esta es la idea del algoritmo "**Descenso de Gradiente**"

Algoritmo

Debemos de construir un **algoritmo** para esa minimización

- La 1^a aproximación la realizamos mediante **Búsqueda**
- Comenzamos inicializando θ con algún valor. Ej: $\vec{\theta} = \vec{0}$
- Vamos cambiando los valores de θ de forma que disminuya $J(\theta)$
  - Hay que asegurar que los nuevos valores de $\vec{\theta}$ hacen disminuir J
- Repetimos el cambio de valores hasta alcanzar el mínimo.

Esta es la idea del algoritmo "**Descenso de Gradiente**"

Algoritmo

Debemos de construir un **algoritmo** para esa minimización

- La 1^a aproximación la realizamos mediante **Búsqueda**
- Comenzamos inicializando θ con algún valor. Ej: $\vec{\theta} = \vec{0}$
- Vamos cambiando los valores de θ de forma que disminuya $J(\theta)$
  - Hay que asegurar que los nuevos valores de $\vec{\theta}$ hacen disminuir J
- Repetimos el cambio de valores hasta alcanzar el mínimo.

Esta es la idea del algoritmo "**Descenso de Gradiente**"

Algoritmo

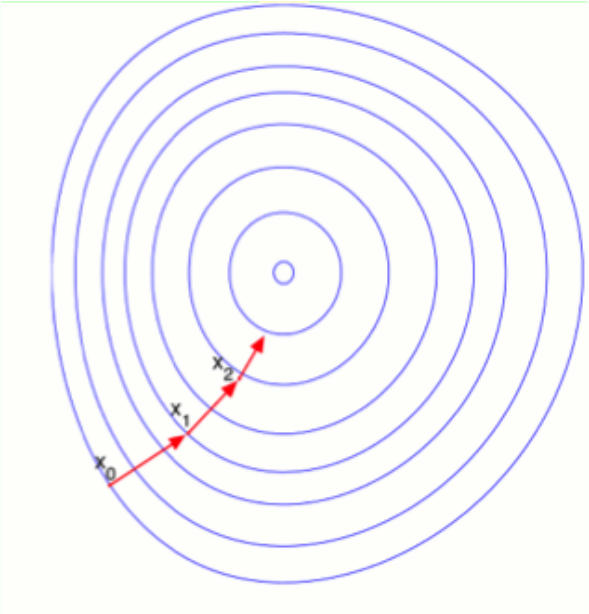
Debemos de construir un **algoritmo** para esa minimización

- La 1^a aproximación la realizamos mediante **Búsqueda**
- Comenzamos inicializando θ con algún valor. Ej: $\vec{\theta} = \vec{0}$
- Vamos cambiando los valores de θ de forma que disminuya $J(\theta)$
  - Hay que asegurar que los nuevos valores de $\vec{\theta}$ hacen disminuir J
- Repetimos el cambio de valores hasta alcanzar el mínimo.

Esta es la idea del algoritmo "**Descenso de Gradiente**"

Algoritmo

La variación de parámetros representa **un camino** dentro del espacio de los parámetros.



Si existen **mínimos locales** en la superficie se puede dar que, en función de los valores iniciales de θ, podemos llegar a distintas soluciones.

Gradiente

pero,... ¿Cómo variamos los coeficientes ?

- Debemos de actualizar los coeficientes en la dirección de máximo decremento
- Usaremos la interpretación física del gradiente$= \vec{\nabla}_\theta J(\theta)$

- y los coeficientes los vamos aproximando poco a poco:[1]

$$\theta_{nuevo} := \theta_{antiguo} + \alpha \, \vec{\nabla}_\theta J(\theta)$$

[1] α modula la velocidad de acercamiento

Gradiente

pero,... ¿Cómo variamos los coeficientes ?

- Debemos de actualizar los coeficientes en la dirección de máximo decremento
- Usaremos la interpretación física del gradiente$= \vec{\nabla}_\theta J(\theta)$
- y los coeficientes los vamos aproximando poco a poco:[1]

$$\theta_{nuevo} := \theta_{antiguo} + \alpha \, \vec{\nabla}_\theta J(\theta)$$

[1] α modula la velocidad de acercamiento

Gradiente

pero,... ¿Cómo variamos los coeficientes ?

- Debemos de actualizar los coeficientes en la dirección de máximo decremento
- Usaremos la interpretación física del gradiente$= \vec{\nabla}_\theta J(\theta)$
- y los coeficientes los vamos aproximando poco a poco:[1]

$$\theta_{nuevo} := \theta_{antiguo} + \alpha \, \vec{\nabla}_\theta J(\theta)$$

[1] α modula la velocidad de acercamiento

Gradiente

pero,... ¿Cómo variamos los coeficientes ?

- Debemos de actualizar los coeficientes en la dirección de máximo decremento
- Usaremos la interpretación física del gradiente$= \vec{\nabla}_\theta J(\theta)$

- y los coeficientes los vamos aproximando poco a poco:[1]

$$\theta_{nuevo} := \theta_{antiguo} + \alpha \ \vec{\nabla}_\theta \boldsymbol{J}(\theta)$$

[1] α modula la velocidad de acercamiento

Gradiente

Vamos a desarrollar ahora la fórmula del gradiente:

$$J(\theta) = \frac{1}{2}\sum_{j=1}^{m}(h_\theta(x^j) - y^j)^2$$

aplicado al gradiente:

$$\vec{\nabla}_\theta J(\theta) = \left(\frac{\partial J(\theta)}{\partial \theta_0}, \frac{\partial J(\theta)}{\partial \theta_1}, \ldots, \frac{\partial J(\theta)}{\partial \theta_n}\right)$$

que es el vector de todas las derivadas parciales de J, en las direcciones de θ

$$\frac{\partial}{\partial \theta_i} J(\theta)$$

Gradiente

Vamos a desarrollar ahora la fórmula del gradiente:

$$J(\theta) = \frac{1}{2}\sum_{j=1}^{m}(h_\theta(x^j) - y^j)^2$$

aplicado al gradiente:

$$\vec{\nabla}_\theta J(\theta) = \left(\frac{\partial J(\theta)}{\partial \theta_0}, \frac{\partial J(\theta)}{\partial \theta_1}, \ldots, \frac{\partial J(\theta)}{\partial \theta_n}\right)$$

que es el vector de todas las derivadas parciales de J, en las direcciones de θ

$$\frac{\partial}{\partial \theta_i} J(\theta)$$

Gradiente

Vamos a desarrollar ahora la fórmula del gradiente:

$$J(\theta) = \frac{1}{2}\sum_{j=1}^{m}(h_\theta(x^j) - y^j)^2$$

aplicado al gradiente:

$$\vec{\nabla}_\theta J(\theta) = \left(\frac{\partial J(\theta)}{\partial \theta_0}, \frac{\partial J(\theta)}{\partial \theta_1}, \ldots, \frac{\partial J(\theta)}{\partial \theta_n}\right)$$

que es el vector de todas las derivadas parciales de J, en las direcciones de θ

$$\frac{\partial}{\partial \theta_i} J(\theta)$$

Gradiente

Para un dataset de 1 ejemplo : $(m = 1)\ J(\theta) = \frac{1}{2}\left(h_\theta(x) - y\right)^2$

y para la componente i: $\frac{\partial}{\partial \theta_i} J(\theta) = \frac{\partial}{\partial \theta_i}\left(\frac{1}{2}(h_\theta(x) - y)^2\right) =$

$$= \cancel{2}\frac{1}{\cancel{2}}(h_\theta(x) - y)\ \frac{\partial}{\partial \theta_i}(h_\theta(x) - y) = \quad (1)$$

recordando que: $h_\theta(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i$

$$= (h_\theta(x) - y)\ \frac{\partial}{\partial \theta_i}(\underbrace{\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n}_{h_\theta(x)} - y) =$$

Gradiente

Para un dataset de 1 ejemplo : $(m = 1)\ J(\theta) = \frac{1}{2}\left(h_\theta(x) - y\right)^2$

y para la componente i: $\frac{\partial}{\partial \theta_i} J(\theta) = \frac{\partial}{\partial \theta_i}\ (\frac{1}{2}(h_\theta(x) - y)^2)\ =$

$$= \cancel{2}\frac{1}{\cancel{2}}(h_\theta(x) - y)\ \frac{\partial}{\partial \theta_i}(h_\theta(x) - y)\ = \tag{1}$$

recordando que: $h_\theta(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i$

$$= (h_\theta(x) - y)\ \frac{\partial}{\partial \theta_i}(\underbrace{\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n}_{h_\theta(x)} - y)\ =$$

Gradiente

Para un dataset de 1 ejemplo : $(m = 1)\ J(\theta) = \frac{1}{2}\left(h_\theta(x) - y\right)^2$

y para la componente i: $\frac{\partial}{\partial \theta_i} J(\theta) = \frac{\partial}{\partial \theta_i}\ (\frac{1}{2}(h_\theta(x) - y)^2)\ =$

$$= \cancel{2}\frac{1}{\cancel{2}}(h_\theta(x) - y)\ \frac{\partial}{\partial \theta_i}(h_\theta(x) - y)\ = \tag{1}$$

recordando que: $h_\theta(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i$

$$= \left(h_\theta(x) - y\right)\ \frac{\partial}{\partial \theta_i}(\underbrace{\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n}_{h_\theta(x)} - y)\ =$$

Gradiente

Para un dataset de 1 ejemplo : $(m = 1)\ J(\theta) = \frac{1}{2}\left(h_\theta(x) - y\right)^2$

y para la componente i: $\frac{\partial}{\partial \theta_i} J(\theta) = \frac{\partial}{\partial \theta_i}\ \left(\frac{1}{2}(h_\theta(x) - y)^2\right)\ =$

$$= \cancel{2}\frac{1}{\cancel{2}}(h_\theta(x) - y)\ \frac{\partial}{\partial \theta_i}(h_\theta(x) - y)\ = \tag{1}$$

recordando que: $h_\theta(\vec{x}) = \sum_{i=0}^{n} \theta_i x_i$

$$= \ (h_\theta(x) - y)\ \frac{\partial}{\partial \theta_i}(\underbrace{\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \cdots + \theta_n x_n}_{h_\theta(x)} - y)\ =$$

Gradiente

Calculamos la derivada primero:

$$\frac{\partial}{\partial \theta_i} \left(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n - y\right)$$

Algunas cosillas:

- Es la derivada de sumas y restas. . .
- "**y**" NO dependen de θ_i ¿Seguro?
- **todos los términos** de "h", salvo $\theta_i x_i$, **NO dependen de** θ_i ¿Seguro?

por lo que:

$$\frac{\partial}{\partial \theta_i} \left(\cancelto{0}{\theta_0 x_0} + \cancelto{0}{\theta_1 x_1} + \cdots + \theta_i x_i + \cdots + \cancelto{0}{\theta_n x_n} - \cancelto{0}{y}\right) = \frac{\partial(\theta_i x_i)}{\partial \theta_i} = \mathbf{x_i}$$

Gradiente

Calculamos la derivada primero:

$$\frac{\partial}{\partial \theta_i} \left(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n - y\right)$$

Algunas cosillas:

- Es la derivada de sumas y restas...
- "**y**" NO dependen de θ_i ¿Seguro?
- **todos los términos** de "h", salvo $\theta_i x_i$, **NO dependen de** θ_i ¿Seguro?

por lo que:

$$\frac{\partial}{\partial \theta_i} \left(\cancelto{0}{\theta_0 x_0} + \cancelto{0}{\theta_1 x_1} + \cdots + \theta_i x_i + \cdots + \cancelto{0}{\theta_n x_n} - \cancelto{0}{y}\right) = \frac{\partial(\theta_i x_i)}{\partial \theta_i} = \mathbf{x_i}$$

Gradiente

Calculamos la derivada primero:

$$\frac{\partial}{\partial \theta_i} \left(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n - y\right)$$

Algunas cosillas:

- Es la derivada de sumas y restas...
- "**y**" NO dependen de θ_i ¿Seguro?
- **todos los términos** de "h", salvo $\theta_i x_i$, **NO dependen de** θ_i ¿Seguro?

por lo que:

$$\frac{\partial}{\partial \theta_i} \left(\cancelto{0}{\theta_0 x_0} + \cancelto{0}{\theta_1 x_1} + \cdots + \theta_i x_i + \cdots + \cancelto{0}{\theta_n x_n} - \cancelto{0}{y}\right) = \frac{\partial(\theta_i x_i)}{\partial \theta_i} = \mathbf{x_i}$$

Gradiente

Calculamos la derivada primero:

$$\frac{\partial}{\partial \theta_i} \left(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n - y\right)$$

Algunas cosillas:

- Es la derivada de sumas y restas...
- "**y**" NO dependen de θ_i ¿Seguro?
- **todos los términos** de "h", salvo $\theta_i x_i$, **NO dependen de** θ_i ¿Seguro?

por lo que:

$$\frac{\partial}{\partial \theta_i} \left(\cancelto{0}{\theta_0 x_0} + \cancelto{0}{\theta_1 x_1} + \cdots + \theta_i x_i + \cdots + \cancelto{0}{\theta_n x_n} - \cancelto{0}{y}\right) = \frac{\partial(\theta_i x_i)}{\partial \theta_i} = \mathbf{x_i}$$

Gradiente

Calculamos la derivada primero:

$$\frac{\partial}{\partial \theta_i} \left(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n - y\right)$$

Algunas cosillas:

- Es la derivada de sumas y restas...
- "**y**" NO dependen de θ_i ¿Seguro?
- **todos los términos** de "h", salvo $\theta_i x_i$, **NO dependen de** θ_i ¿Seguro?

por lo que:

$$\frac{\partial}{\partial \theta_i} \left(\cancelto{0}{\theta_0 x_0} + \cancelto{0}{\theta_1 x_1} + \cdots + \theta_i x_i + \cdots + \cancelto{0}{\theta_n x_n} - \cancelto{0}{y}\right) = \frac{\partial(\theta_i x_i)}{\partial \theta_i} = \mathbf{x_i}$$

Gradiente

Calculamos la derivada primero:

$$\frac{\partial}{\partial \theta_i} \left(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n - y\right)$$

Algunas cosillas:

- Es la derivada de sumas y restas...
- "**y**" NO dependen de θ_i ¿Seguro?
- **todos los términos** de "h", salvo $\theta_i x_i$, **NO dependen de** θ_i ¿Seguro?

por lo que:

$$\frac{\partial}{\partial \theta_i} \left(\cancelto{0}{\theta_0 x_0} + \cancelto{0}{\theta_1 x_1} + \cdots + \theta_i x_i + \cdots + \cancelto{0}{\theta_n x_n} - \cancelto{0}{y}\right) = \frac{\partial(\theta_i x_i)}{\partial \theta_i} = \mathbf{x_i}$$

Gradiente

Calculamos la derivada primero:

$$\frac{\partial}{\partial \theta_i} \left(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n - y\right)$$

Algunas cosillas:

- Es la derivada de sumas y restas...
- "**y**" NO dependen de θ_i ¿Seguro?
- **todos los términos** de "h", salvo $\theta_i x_i$, **NO dependen de** θ_i ¿Seguro?

por lo que:

$$\frac{\partial}{\partial \theta_i} \left(\cancelto{0}{\theta_0 x_0} + \cancelto{0}{\theta_1 x_1} + \cdots + \theta_i x_i + \cdots + \cancelto{0}{\theta_n x_n} - \cancelto{0}{y}\right) = \frac{\partial(\theta_i x_i)}{\partial \theta_i} = \mathbf{x_i}$$

Gradiente

La fórmula del gradiente (ec. 1) (en la componente i) queda:

$$\frac{\partial}{\partial \theta_i} J(\theta) = (h_\theta(x) - y)x_i$$

La actualización de los parámetros queda como:

$$\theta_i' := \theta_i + \alpha(h_\theta(x) - y)x_i$$

Extendiendo para m ejemplos:

$$\theta_i' := \theta_i + \alpha \sum_{j=1}^{m} (h_\theta(x^j) - y^j)x_i^j$$

y calculando los "$n+1$" coeficientes para los "n" atributos

Gradiente

La fórmula del gradiente (ec. 1) (en la componente i) queda:

$$\frac{\partial}{\partial \theta_i} J(\theta) = (h_\theta(x) - y)x_i$$

La actualización de los parámetros queda como:

$$\theta_i' := \theta_i + \alpha(h_\theta(x) - y)x_i$$

Extendiendo para m ejemplos:

$$\theta_i' := \theta_i + \alpha \sum_{j=1}^{m} (h_\theta(x^j) - y^j)x_i^j$$

y calculando los "$n+1$" coeficientes para los "n" atributos

Gradiente

La fórmula del gradiente (ec. 1) (en la componente i) queda:

$$\frac{\partial}{\partial \theta_i} J(\theta) = (h_\theta(x) - y)x_i$$

La actualización de los parámetros queda como:

$$\theta_i' := \theta_i + \alpha(h_\theta(x) - y)x_i$$

Extendiendo para m ejemplos:

$$\theta_i' := \theta_i + \alpha \sum_{j=1}^{m} (h_\theta(x^j) - y^j)x_i^j$$

y calculando los "$n+1$" coeficientes para los "n" atributos

Gradiente

La fórmula del gradiente (ec. 1) (en la componente i) queda:

$$\frac{\partial}{\partial \theta_i} J(\theta) = (h_\theta(x) - y)x_i$$

La actualización de los parámetros queda como:

$$\theta_i' := \theta_i + \alpha(h_\theta(x) - y)x_i$$

Extendiendo para m ejemplos:

$$\theta_i' := \theta_i + \alpha \sum_{j=1}^{m} (h_\theta(x^j) - y^j)x_i^j$$

y calculando los "$n + 1$" coeficientes para los "n" atributos.

Algoritmo

Escribiendo esto en una primera aproximación...

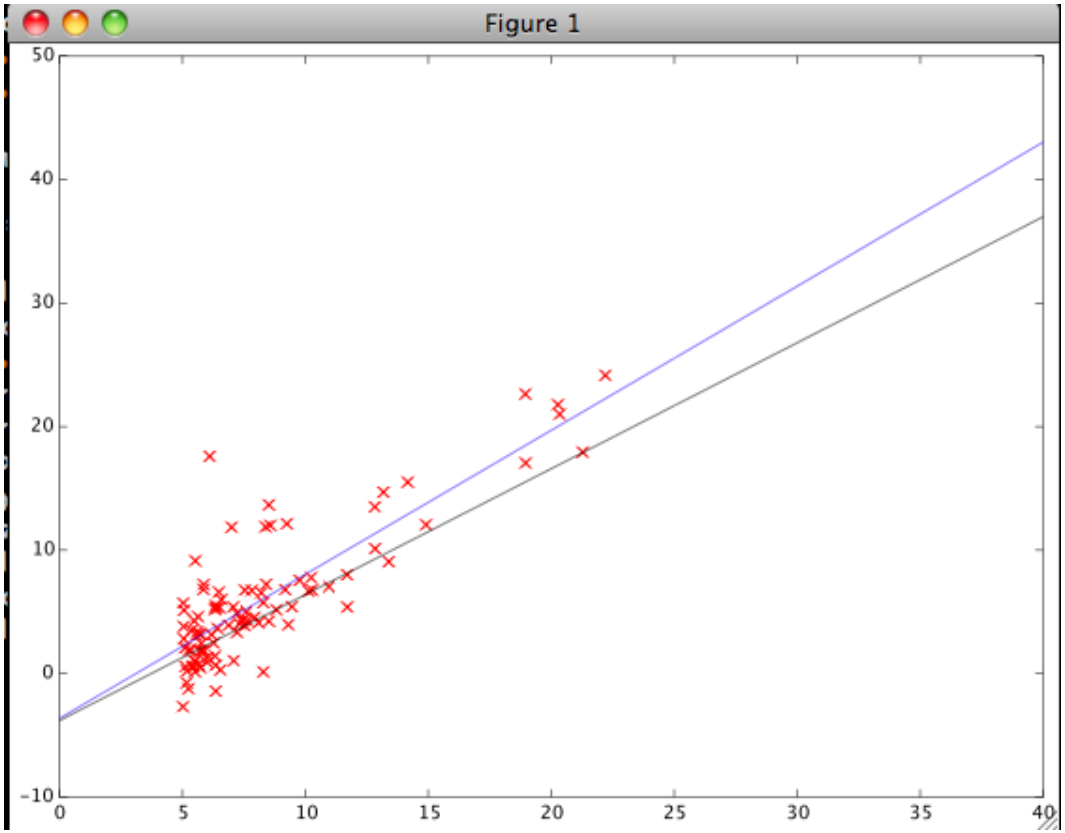
```
1 theta[i] = init()
2 repeat until convergence {
3     for i=0 to n {
4         theta[i] := Actualizacion[i]
5     }
6 }
```

Este es el algoritmo **Descenso por gradiente**

Ejemplo

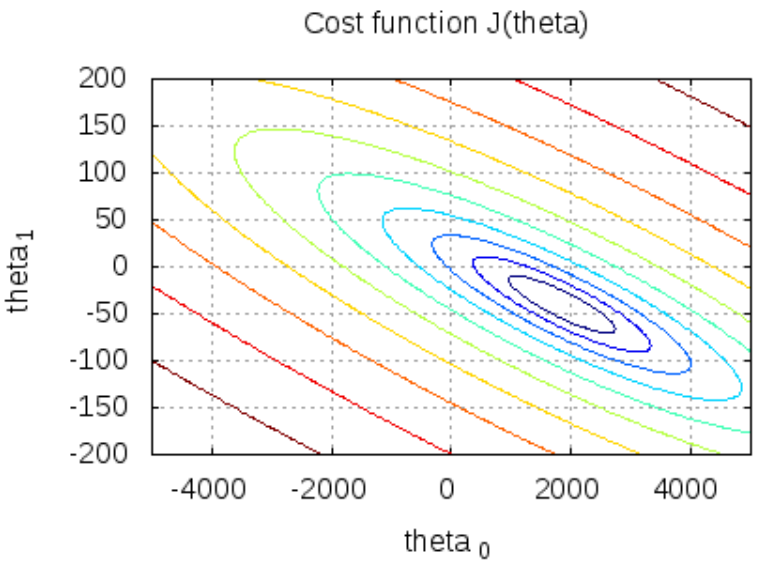
Volviendo al ejemplo del precio de las casas:

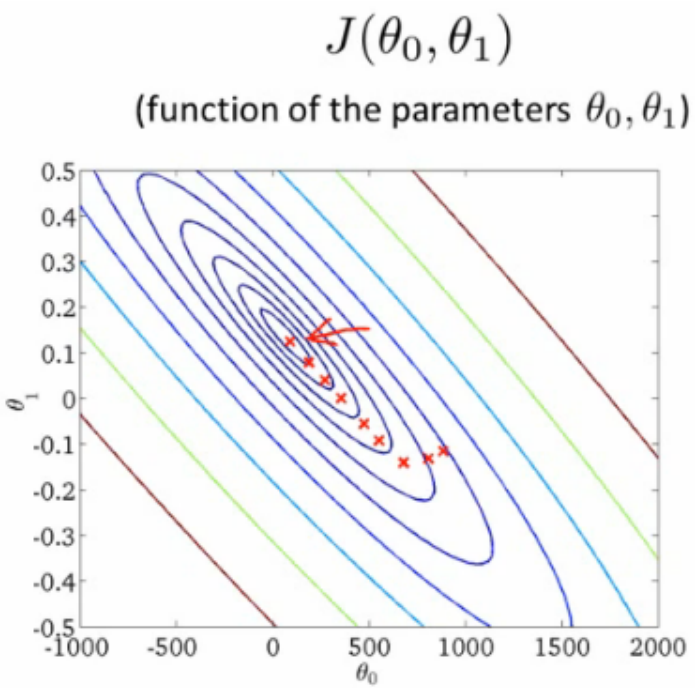
Representamos los valores de $h_\theta(x)$ en función de las distintas iteraciones.



Ejemplo

Si representamos el error:





Características

- Este algoritmo procesa **todos** los ejemplos por **cada** nueva **asignación** de parámetro (θ_i).
- En cada iteración, se hacen **m** $\times$ **n** actualizaciones
- No es aconsejable en datasets grandes
- Con el objeto de evitar mínimos locales, el *init* puede ser a *random*
- Debemos buscar una **ALTERNATIVA**

Características

- Este algoritmo procesa **todos** los ejemplos por **cada** nueva **asignación** de parámetro (θ_i).
- En cada iteración, se hacen **m** $\times$ **n** actualizaciones
- No es aconsejable en datasets grandes
- Con el objeto de evitar mínimos locales, el *init* puede ser a *random*
- Debemos buscar una **ALTERNATIVA**

Características

- Este algoritmo procesa **todos** los ejemplos por **cada** nueva **asignación** de parámetro (θ_i).
- En cada iteración, se hacen **m** $\times$ **n** actualizaciones
- No es aconsejable en datasets grandes
- Con el objeto de evitar mínimos locales, el *init* puede ser a *random*
- Debemos buscar una **ALTERNATIVA**

Características

- Este algoritmo procesa **todos** los ejemplos por **cada** nueva **asignación** de parámetro (θ_i).
- En cada iteración, se hacen **m** $\times$ **n** actualizaciones
- No es aconsejable en datasets grandes
- Con el objeto de evitar mínimos locales, el *init* puede ser a *random*
- Debemos buscar una **ALTERNATIVA**

Características

- Este algoritmo procesa **todos** los ejemplos por **cada** nueva **asignación** de parámetro (θ_i).
- En cada iteración, se hacen **m** $\times$ **n** actualizaciones
- No es aconsejable en datasets grandes
- Con el objeto de evitar mínimos locales, el *init* puede ser a *random*
- Debemos buscar una **ALTERNATIVA**

Índice

(Batch) Descenso de Gradiente Estocástico

- Consiste en actualizar los parámetros θ_i para cada uno de los ejemplos, en lugar de actualizarlos al final.
- El ejemplo es **elegido al azar**
- Si el **número de ejemplos** es bastante **mayor** que el de **atributos**, la solución también **converge**, y , generalment, más rápido

(Batch) Descenso de Gradiente Estocástico

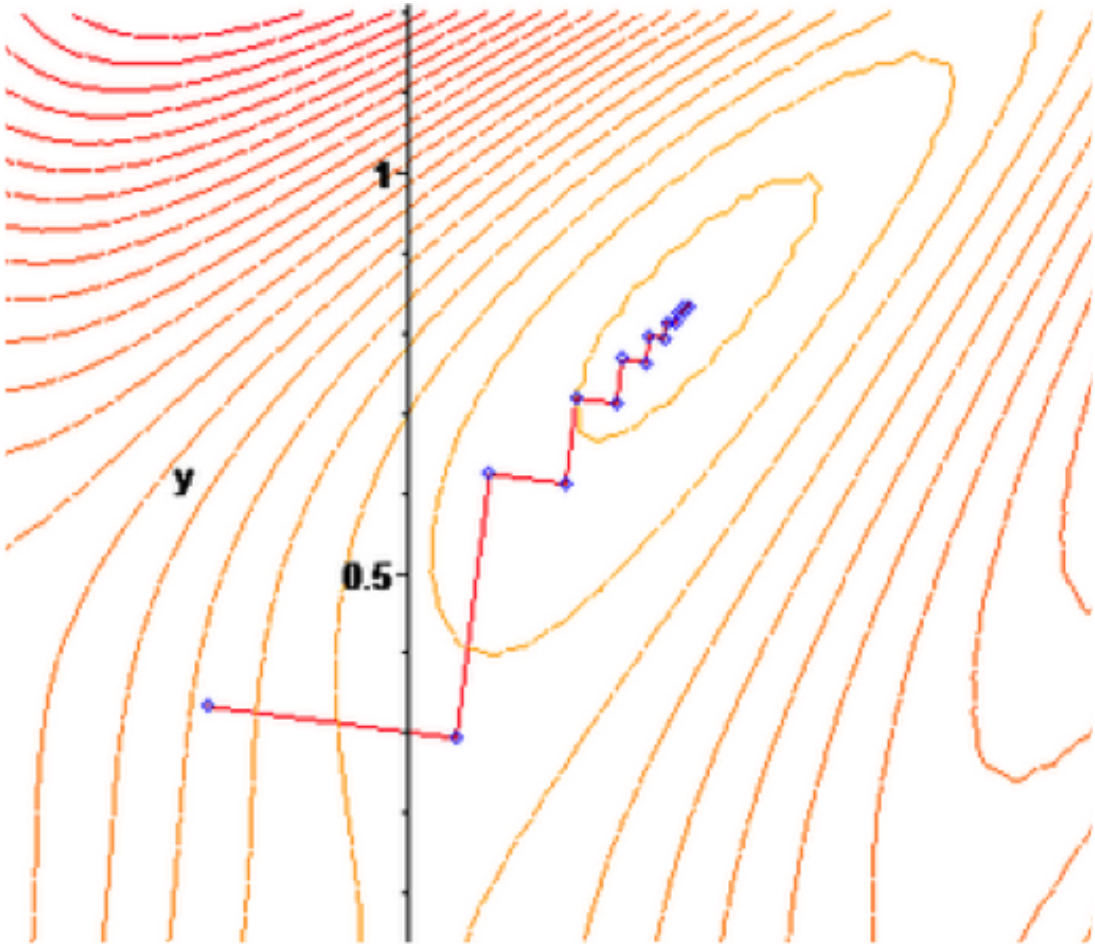
- Consiste en actualizar los parámetros θ_i para cada uno de los ejemplos, en lugar de actualizarlos al final.
- El ejemplo es **elegido al azar**
- Si el **número de ejemplos** es bastante **mayor** que el de **atributos**, la solución también **converge**, y , generalment, más rápido

(Batch) Descenso de Gradiente Estocástico

- Consiste en actualizar los parámetros θ_i para cada uno de los ejemplos, en lugar de actualizarlos al final.
- El ejemplo es **elegido al azar**
- Si el **número de ejemplos** es bastante **mayor** que el de **atributos**, la solución también **converge**, y , generalment, más rápido

Descenso de Gradiente Estocástico

Interpretación física del concepto



Comparativa

Num.	Descenso por Gradiente	Descenso Estocástico
1.	Calcula el gradiente usando toda la muestra de entrenamiento	Calcula el gradiente usando una sola muestra de entrenamiento
2.	Algoritmo lento y computacionalmente costoso	Más rápido y menos computacionalmente costoso que Batch GD
3.	No sugerido para muestras de entrenamiento grandes.	Puede utilizarse para grandes muestras de entrenamiento.
4.	De naturaleza determinista.	De naturaleza estocástica.
5.	Proporciona una solución óptima con el tiempo suficiente para converger.	Da una buena solución pero no óptima.
6.	No se requiere una mezcla aleatoria de puntos.	La muestra de datos debe estar en un orden aleatorio, y es por eso que queremos mezclar el conjunto de entrenamiento para cada *epoch*.
7.	No se puede escapar fácilmente de los mínimos locales poco profundos.	puede escapar de los mínimos locales poco profundos más fácilmente.
8.	La convergencia es lenta.	Alcanza la convergencia mucho más rápido.

Índice

Ecuaciones normales

- Existe una **alternativa** al método iterativo, que es el método **analítico**.
- Buscar la **forma general** de resolución de la ecuación de **minimización**.
- Son los métodos usados, generalmente, en **estadística**
- Resolución general de:

$$\theta := \theta + \alpha \nabla_\theta \boldsymbol{J}(\theta)$$

Un poco de notación

- Fórmula de actualización (para 1 ejemplo):

$$\underbrace{\theta}_{\in \mathbb{R}^{n+1}} := \theta + \underbrace{\alpha}_{\in \mathbb{R}} \underbrace{\nabla_\theta J(\theta)}_{\in \mathbb{R}^{n+1}}$$

- Fórmula general del gradiente:

$$\theta = \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1} \qquad \nabla_\theta J(\theta) = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \vdots \\ \frac{\partial J}{\partial \theta_n} \end{bmatrix} \in \mathbb{R}^{n+1}$$

Un poco de notación

- Fórmula de actualización (para 1 ejemplo):

$$\underbrace{\theta}_{\in \mathbb{R}^{n+1}} := \theta + \underbrace{\alpha}_{\in \mathbb{R}} \underbrace{\nabla_\theta J(\theta)}_{\in \mathbb{R}^{n+1}}$$

- Fórmula general del gradiente:

$$\theta = \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1} \qquad \nabla_\theta J(\theta) = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \vdots \\ \frac{\partial J}{\partial \theta_n} \end{bmatrix} \in \mathbb{R}^{n+1}$$

Desarrollo de Ecuaciones Normales

$$h(\theta) = X \cdot \theta = \overbrace{\begin{bmatrix} x_0^1 & \dots & x_i^1 & \dots & x_n^1 \\ \vdots & \dots & \vdots & \dots & \vdots \\ x_0^m & \dots & x_i^m & \dots & x_n^m \end{bmatrix}}^{dataset} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} = \begin{bmatrix} x^1 \cdot \theta \\ \vdots \\ x^m \cdot \theta \end{bmatrix} =$$

$$\begin{bmatrix} \sum_{i=0}^{n} \theta_i^1 x_i^1 \\ \vdots \\ \sum_{i=0}^{n} \theta_i^m x_i^m \end{bmatrix} = \begin{bmatrix} h_\theta(x^1) \\ \vdots \\ h_\theta(x^m) \end{bmatrix} \in \mathbb{R}^m$$

Desarrollo de Ecuaciones Normales

$$X \cdot \theta - y = \begin{bmatrix} h(x^1) - y^1 \\ \vdots \\ h(x^m) - y^m \end{bmatrix}$$

y sabiendo que $z^t \cdot z = \sum_{j=0}^{n} (z_i^2)$

$$\frac{1}{2}(h_\theta(x) - y)^t (h_\theta(x) - y) = \frac{1}{2} \sum_{j=1}^{m} \left(h(x^j) - y^j \right)^2 = J(\theta)$$

$$J(\theta) = \frac{1}{2}(X\theta - y)^t (X\theta - y)$$

Minimizando $J(\theta)$

Ahora debemos de obtener el mínimo de $J(\theta)$, y para ello se debe de cumplir que:

$$\nabla_\theta J(\theta) = 0$$

Sustituyendo y calculando...

$$\nabla_\theta(\frac{1}{2}(X\theta - y)^t(X\theta - y)) =$$

$$\frac{1}{2}\nabla_\theta(tr(\theta^t X^t X\theta - \theta^t X^t y \dots$$

$$\cdots = 0$$

Minimizando $J(\theta)$

Obteniendo...

$$\nabla_\theta J(\theta) = X^t X\theta - X^t y = 0$$

o lo que es lo mismo, la ecuación normal:

$$X^t X\theta = X^t y$$

Ecuaciones normales

despejando θ, obtenemos su definición:

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

- Es una definición **analítica** de θ,
- **NO** necesitamos un algoritmo iterativo
  - con cualquier **librería** de matrices, es una simple orden
  - en Matlab: $inv(X' * X) * X' * y$

Ecuaciones normales

despejando θ, obtenemos su definición:

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

- Es una definición **analítica** de θ,
- **NO** necesitamos un algoritmo iterativo
  - con cualquier **librería** de matrices, es una simple orden
  - en Matlab: $inv(X' * X) * X' * y$

Ecuaciones normales

despejando θ, obtenemos su definición:

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

- Es una definición **analítica** de θ,
- **NO** necesitamos un algoritmo iterativo
  - con cualquier **librería** de matrices, es una simple orden
  - en Matlab: $inv(X' * X) * X' * y$

Ecuaciones normales

despejando θ, obtenemos su definición:

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

- Es una definición **analítica** de θ,
- **NO** necesitamos un algoritmo iterativo
  - con cualquier **librería** de matrices, es una simple orden
  - en Matlab: $inv(X' * X) * X' * y$

Algunas preguntas

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

- ¿Es invertible la matriz $(X^t X)$?
- Realmente ¿se puede hacer $\nabla_\theta J(\theta) = 0$?

Algunas preguntas

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

- ¿Es invertible la matriz $(X^t X)$?
- Realmente ¿se puede hacer $\nabla_\theta J(\theta) = 0$?

Algunas preguntas

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

- ¿Es invertible la matriz $(X^t X)$?
- Realmente ¿se puede hacer $\nabla_\theta J(\theta) = 0$?