APRENDIZAJE DE MÁQUINA

REGRESIÓN LINEAL

AGENDA

01 Aprendizaje supervisado

Datos de entrenamiento, hipótesis, un ejemplo.

02 Algortimo LMS

Modelo lineal, función de costo, descenso por gradiente.

03 Ecuaciones normales

Modelo lineal matricial, derivando las ecuaciones normales

04 Interpretación probabilística

Suposiciones para errores, función de verosimiltud

05 Funciones de base

Derivación, funciones, maxima verosimilitud

APRENDIZAJE SUPERVISADO

DEFINICIONES

NOMENCLATURA

APRENDIZAJE SUPERVISADO

DEFINICIONES Y NOMENCLATURA

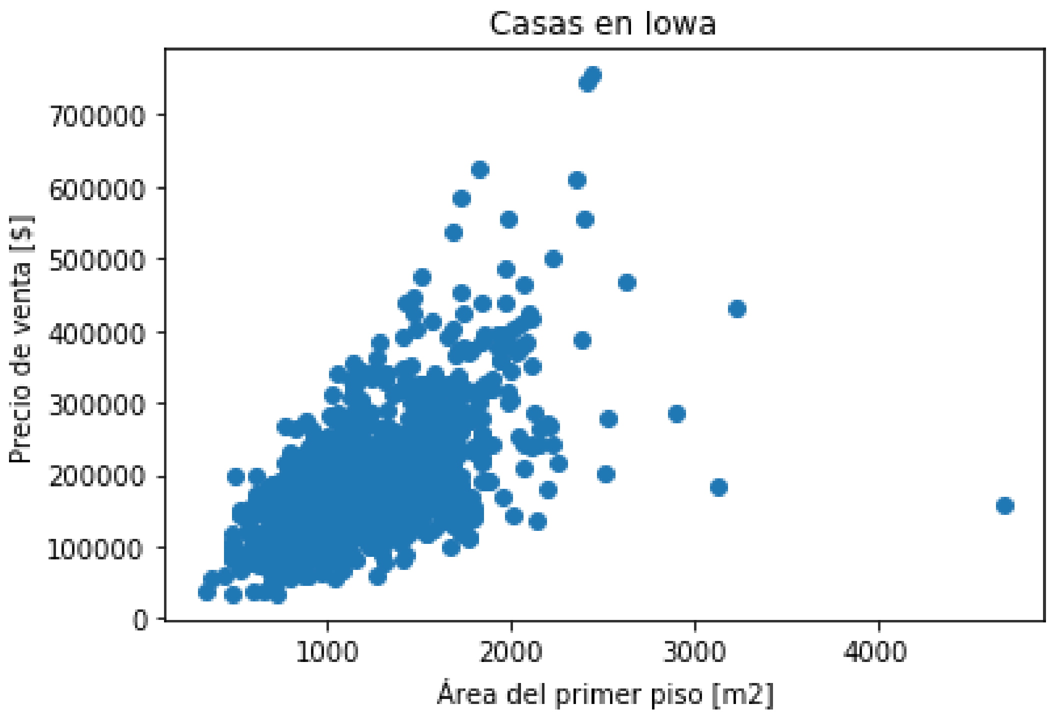
Comenzamos con un conjunto de datos que representan las superficies habitacionales del primer piso y los precios de venta 1,460 casas en Ames, Iowa.

1^{st} floor square feet	Sale Price
856	208500
1262	181500
920	223500
961	140000
1145	250000
⋮	⋮

APRENDIZAJE SUPERVISADO

DEFINICIONES Y NOMENCLATURA

Graficamos los 1,460 datos con Python:



APRENDIZAJE SUPERVISADO

DEFINICIONES Y NOMENCLATURA

La pregunta que surge naturalmente sería:

¿CÓMO PREDECIMOS LOS PRECIOS DE OTRAS CASAS EN AMES, IOWA?

APRENDIZAJE SUPERVISADO

DEFINICIONES Y NOMENCLATURA

Para responder la pregunta primero se establecerá la nomenclatura que se estará usando:

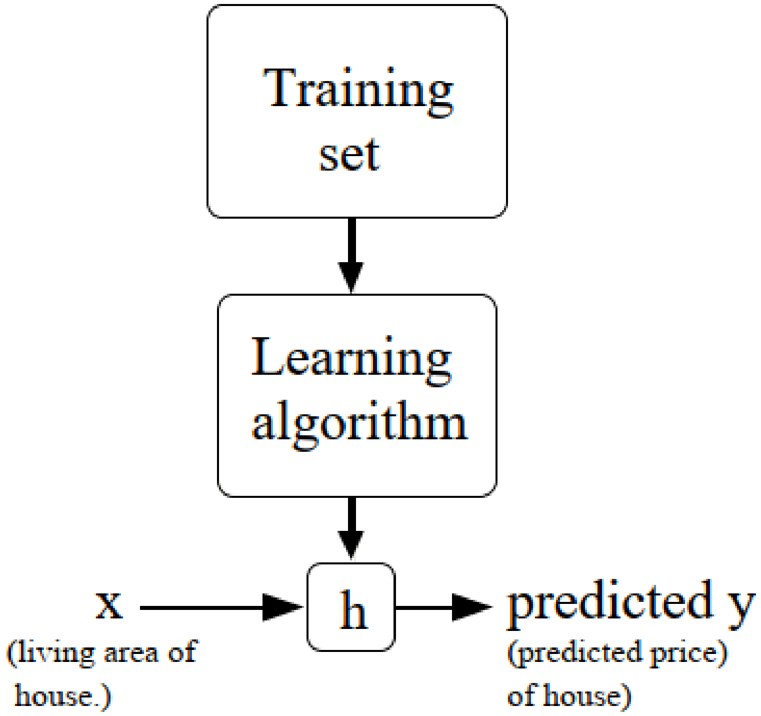
Variable / Símbolo	Descripción	Ejemplo
$\boldsymbol{x^{(i)}}$	**Variable de entrada o características**	**El área del primer piso de la casa.**
$\boldsymbol{y^{(i)}}$	**Variable de salida, objetivo o de respuesta**	**El precio de venta de la casa.**
$\boldsymbol{(x^{(i)}, y^{(i)})}$	**Ejemplo de entrenamiento**	**(área, precio)**
$\boldsymbol{m}$	**Número de ejemplos de entrenamiento**	**1460 casas con su área del 1° piso y el precio.**
$\boldsymbol{\{(x^{(i)}, y^{(i)}); i = 1, \dots, m\}}$	**Conjunto de datos de entrenamiento**	**NA**
$\boldsymbol{\chi}$	**Espacio de los valores de entrada**	**NA**
$\boldsymbol{\gamma}$	**Espacio de los valores de salida**	

APRENDIZAJE SUPERVISADO

DEFINICIONES Y NOMENCLATURA

El **objetivo principal** de cualquier **algoritmo supervisado** es **aprender** una **función** $\boldsymbol{h}: \chi \rightarrow \Upsilon$, de tal manera que $\boldsymbol{h}(\boldsymbol{x})$ pueda **predecir** el correspondiente valor de $\boldsymbol{y}$.

Donde $\boldsymbol{h}$ se define como la **hipótesis**.



APRENDIZAJE SUPERVISADO

DEFINICIONES Y NOMENCLATURA

Existen **2 tipos** de **problemas** de **aprendizaje supervisado**:

Tipo de problema	Descripción
Problema de regresión	y **toma valores continuos.**
Problema de clasificación	y **toma valores discretos.**

ALGORITMO LMS

MODELO LINEAL

FUNCIÓN DE COSTO

DESCENSO POR GRADIENTE

ALGORITMO LMS

MODELO LINEAL

Ahora veamos el mismo modelo pero con otra variable más: la superficie habitacional del segundo piso.

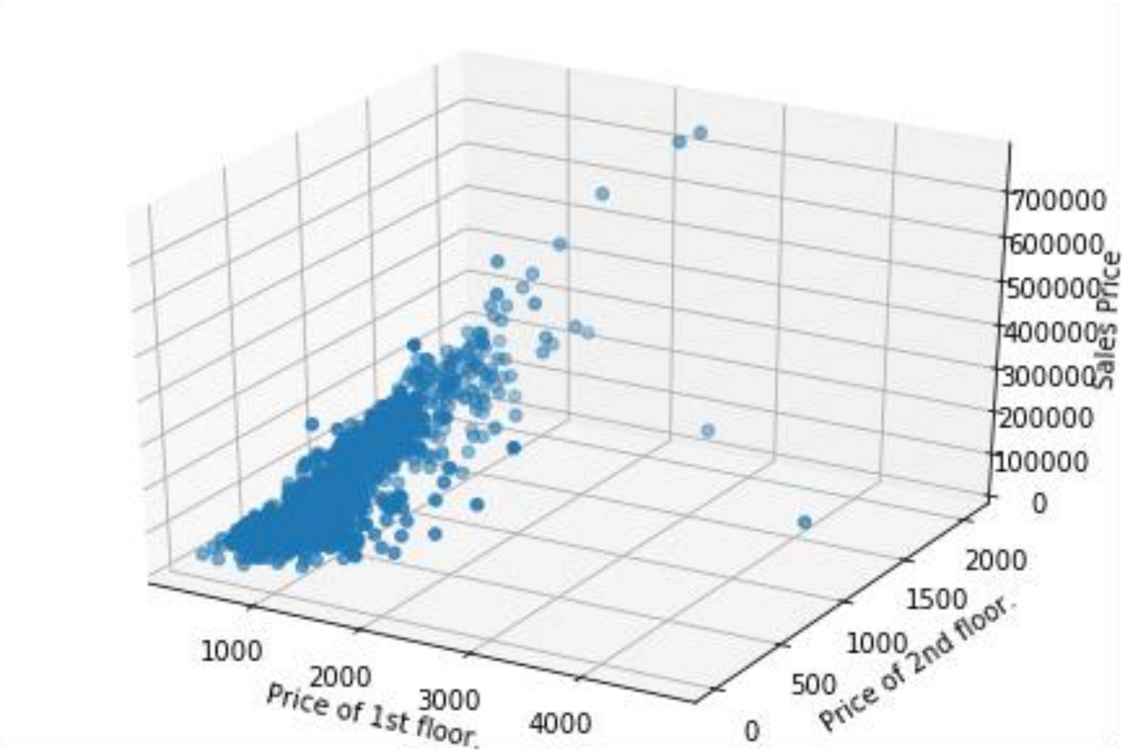
1^{st} floor square feet	2^{nd} floor square feet	Sale Price
856	854	208500
1262	0	181500
920	866	223500
961	756	140000
1145	1053	250000
$\vdots$	$\vdots$	$\vdots$

En este caso cada $\boldsymbol{x}^{(i)} \in \mathbb{R}^2$, por lo que se define como un vector $\boldsymbol{x}^{(i)} = [\boldsymbol{x}^{(i)}{}_1 \;\; \boldsymbol{x}^{(i)}{}_2]$.

A L G O R I T M O L M S

M O D E L O L I N E A L

Graficamos de nuevo los 1,460 datos con Python:



ALGORITMO LMS

MODELO LINEAL

Se plantea la **hipótesis** de que los datos se distribuyen de **manera lineal**, por lo tanto:

$$\boldsymbol{h_w(x) = w_0 + w_1 x_1 + w_2 x_2}$$

En este caso las **variables** $\boldsymbol{w_i}$ se definen como los **parámetros** o **pesos** que **parametrizan** el **espacio** de todas las **funciones lineales** que mapean de $\boldsymbol{\chi}$ a $\boldsymbol{\Upsilon}$.

Se **simplifica** la expresión en **forma vectorial**, donde $\boldsymbol{n}$ representa el **número** de **variables** de **entrada** (sin contar a x_0), :

$$\boldsymbol{h(x) = \sum_{i=1}^{n} w_i x_i = w^T x}$$

ALGORITMO LMS

FUNCIÓN DE COSTO

Ahora la pregunta que surge sería:

¿CUÁL ES LA MEJOR COMBINACIÓN DE PARÁMETROS w QUE RESULTE EN LA MEJOR HIPÓTESIS h?

ALGORITMO LMS

FUNCIÓN DE COSTO

Para **resolver** la **pregunta**, necesitamos **medir** el **error** que producen nuestras **estimaciones** realizadas por $\boldsymbol{h}$.

Sabemos que podemos **medir** el **error** de una sola **estimación** $\boldsymbol{i}$ como una **diferencia entre** la estimación $\boldsymbol{h_w}(\boldsymbol{x}^{(i)})$ y la variable de respuesta $\boldsymbol{y}^{(i)}$.

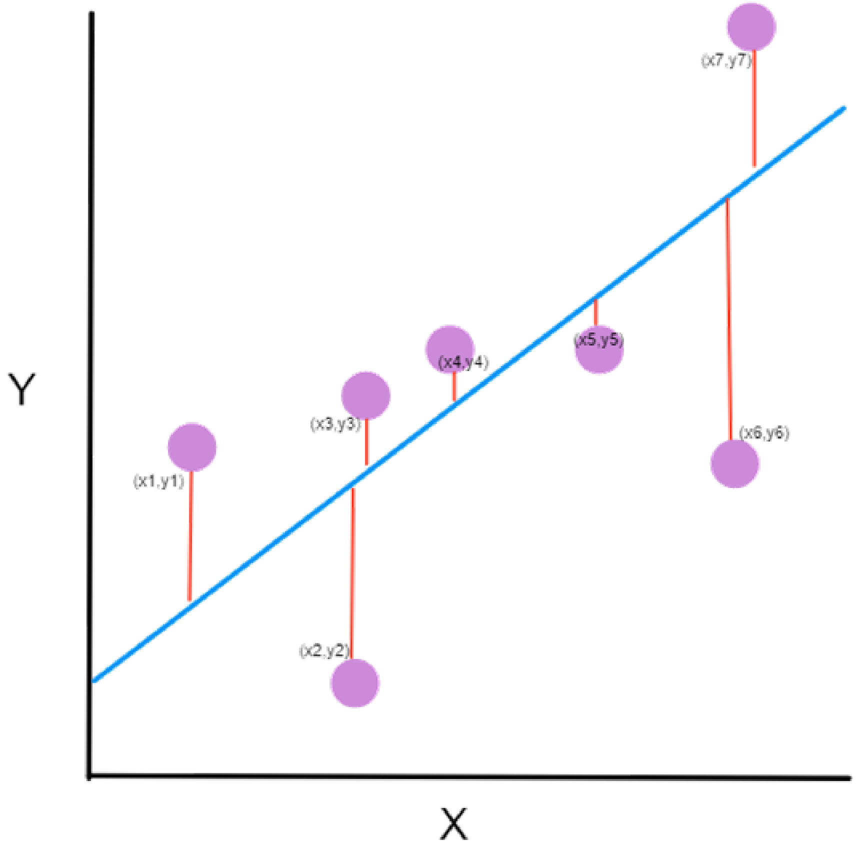
$$\boldsymbol{e} = \boldsymbol{h_w}\left(\boldsymbol{x}^{(i)}\right) - \boldsymbol{y}^{(i)}$$

Para tener siempre cantidades positivas de error, se eleva al cuadrado.

$$\boldsymbol{e}^2 = (\boldsymbol{h_w}\left(\boldsymbol{x}^{(i)}\right) - \boldsymbol{y}^{(i)})^2$$

ALGORITMO LMS
FUNCIÓN DE COSTO

Se **interpreta** el **error** que estamos calculando de **manera gráfica** con un **ejemplo**:



El modelo lineal representado por h

Variables de respuesta $y^{(i)}$

ALGORITMO LMS

FUNCIÓN DE COSTO

Hasta ahora hemos calculado el **error cuadrático** de **un** solo **dato** de **entrenamiento**. Se calcula el **promedio** de **error cuadrático medio** $\boldsymbol{MSE}$ para **todos** los **datos** de **entrenamiento**.

$$\boldsymbol{MSE = \frac{1}{2m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)^2}$$

$$\boldsymbol{MSE = J(w)}$$

Donde $\boldsymbol{J}(\boldsymbol{w})$ es la **función** de **costo**.

A L G O R I T M O   L M S

D E S C E N S O   P O R   G R A D I E N T E

Por lo tanto la **mejor combinación** de pesos $\boldsymbol{w}$**,** es aquella que **minimice** la **función** de **costo** $\boldsymbol{J}(\boldsymbol{w})$, que **mide** nuestro **error cuadrático medio**.

Para **buscar** la **mejor combinación**, diseñemos un **algoritmo** de **búsqueda** que **comience** un **valor inicial** aleatorio de $\boldsymbol{w}$, y que vaya **actualizando** los **valores** de $\boldsymbol{w}$ hasta que **converja** a un **valor mínimo** de $\boldsymbol{J}(\boldsymbol{w})$. La **constante** $\boldsymbol{\alpha}$ se define como la **tasa** de **aprendizaje**.

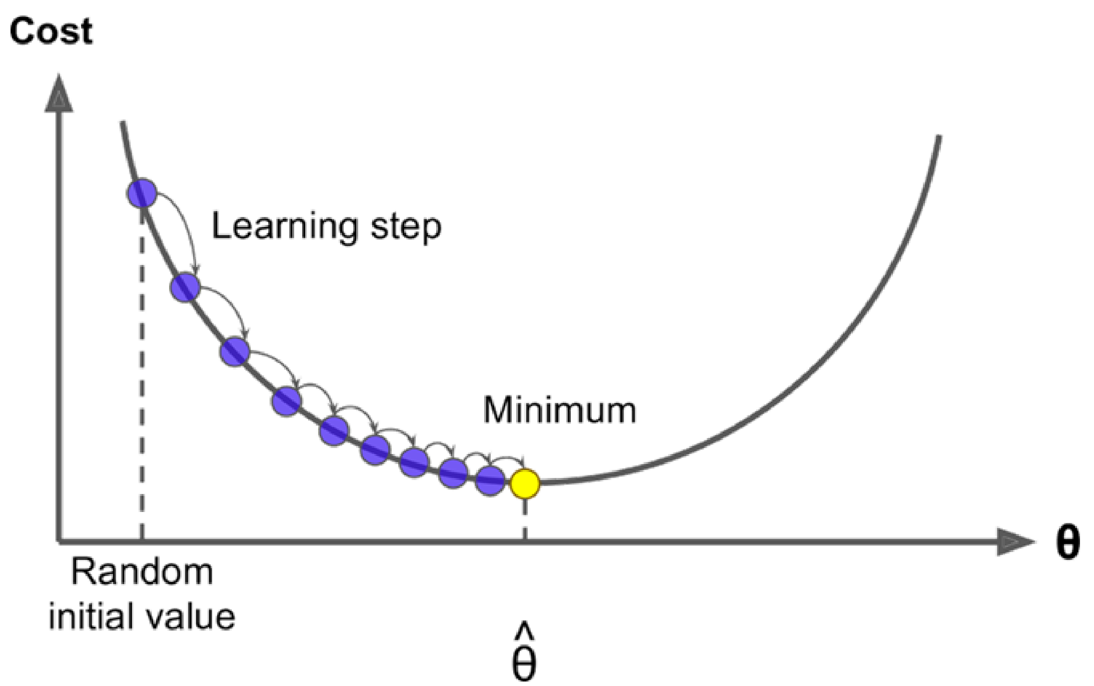
$$\boldsymbol{w_j} := \boldsymbol{w_j} - \boldsymbol{\alpha}\frac{\boldsymbol{\partial}}{\boldsymbol{\partial w_j}}\boldsymbol{J}(\boldsymbol{w})$$

NOTA: la ecuación de arriba **solo actualiza** el **valor** de **un** solo **peso** $\boldsymbol{w_j}$ de los $\boldsymbol{n}$ pesos que **parametrizan** el **modelo lineal**. En la realidad se **actualizan todos** los **pesos** $\boldsymbol{w_j}$ simultáneamente.

ALGORITMO LMS

DESCENSO POR GRADIENTE

Lo que se hace en el **algoritmo** de **optimización** por **descenso** de **gradiente** es **actualizar** los **pesos** en la **dirección** con **mayor decremento** de $\boldsymbol{J}(\boldsymbol{w})$.



ALGORITMO LMS

DESCENSO POR GRADIENTE

Se calcula la **derivada** de la **función** de **costo** $\boldsymbol{J}(\boldsymbol{w})$ con respecto a un **peso específico** $\boldsymbol{w_j}$.

Derivar el resultado:

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m}(h(x) - y)x_j$$

ALGORITMO LMS

DESCENSO POR GRADIENTE

Por lo tanto, la **actualización** de **pesos** por **descenso** de **gradiente** para **un solo dato** de **entrenamiento** quedaría así:

$$\boldsymbol{w_j := w_j + \frac{\alpha}{m}\left(y^{(i)} - h\left(x^{(i)}\right)\right)x_j}$$

A esta **ecuación** se le denomina la **regla** de **actualización LMS** ("**Least Mean Squares**") o también la **regla** de **aprendizaje Widrow-Hoff.**

Este **método** también se llama **descenso** de **gradiente** por **lotes**, en donde se **analizan todos** los **datos** de **entrenamiento simultáneamente**.

ALGORITMO LMS

DESCENSO POR GRADIENTE

Para $\boldsymbol{m}$ **datos** de **entrenamiento**, y definiendo como **vectores** a $\boldsymbol{x}^{(i)}$ y a $\boldsymbol{w}$, la regla de aprendizaje quedaría así:

$$\boldsymbol{w} := \boldsymbol{w} + \frac{\boldsymbol{\alpha}}{\boldsymbol{m}} \sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \left(\boldsymbol{y}^{(\boldsymbol{i})} - \boldsymbol{h}\left(\boldsymbol{x}^{(\boldsymbol{i})}\right)\right) \boldsymbol{x}^{(\boldsymbol{i})}$$

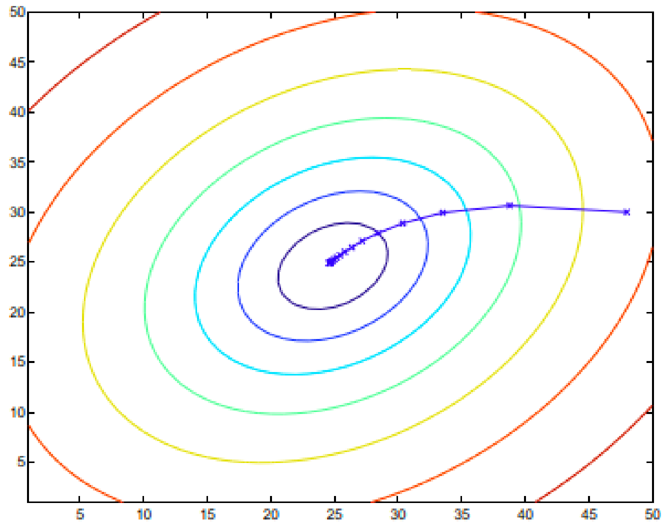
A esta **ecuación** se le denomina la **regla** de **actualización LMS** (“**Least Mean Squares**”) o también la **regla** de **aprendizaje Widrow-Hoff.**

A L G O R I T M O   L M S

D E S C E N S O   P O R   G R A D I E N T E

Por lo general, el **método** de **descenso** por **gradiente** puede sufrir de **varios mínimos locales**. En este caso, **para modelos** de regresión **lineal solo existe un único mínimo global**.

En consecuencia, el **algoritmo siempre convergerá** asumiendo que la **tasa** de **aprendizaje no** sea muy **alta**.



ALGORITMO LMS

DESCENSO POR GRADIENTE

Cuando el **método** solo **observa un dato** de **entrenamiento** a la vez, y **actualiza** los **pesos con un solo dato**, se le denomina como **descenso por gradiente estocástico** o **incremental**.

$$\boldsymbol{w} := \boldsymbol{w} + \boldsymbol{\alpha}\left(\boldsymbol{y}^{(\boldsymbol{i})} - \boldsymbol{h}\left(\boldsymbol{x}^{(\boldsymbol{i})}\right)\right)\boldsymbol{x}^{(\boldsymbol{i})}$$

Esta **variante** del algoritmo **se utiliza** cuando es muy **costosos evaluar** la **actualización** para **conjunto** de **datos muy grandes** (cuando $\boldsymbol{m}$ es **muy grande**), pero tiene el problema de **divergencia** en el mínimo.

ALGORITMO LMS

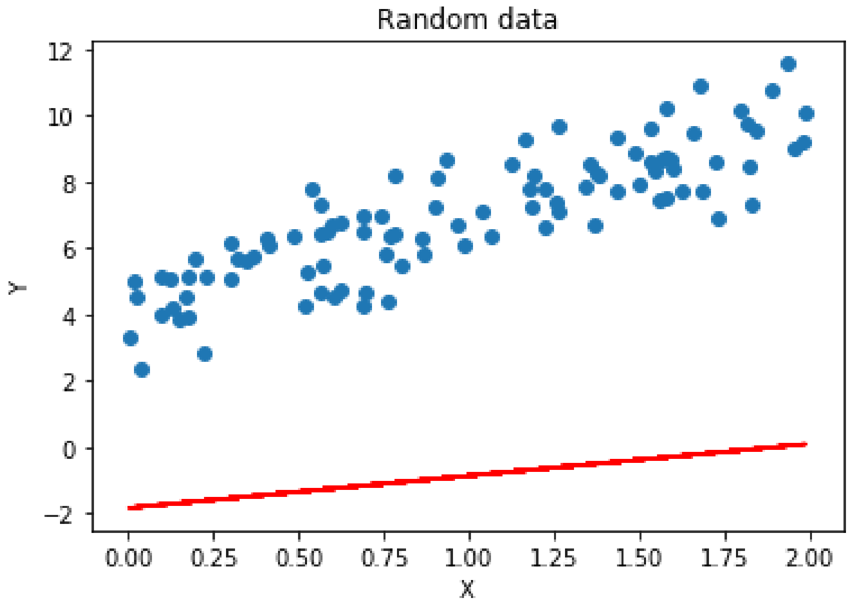
DESCENSO POR GRADIENTE

Ejemplo:

Se generaron datos de entrenamiento aleatorios con cierto grado de error e:

$$y = 3x + 4 + e$$

y se inicializaron aleatoriamente ambos pesos: w_0 y w_1

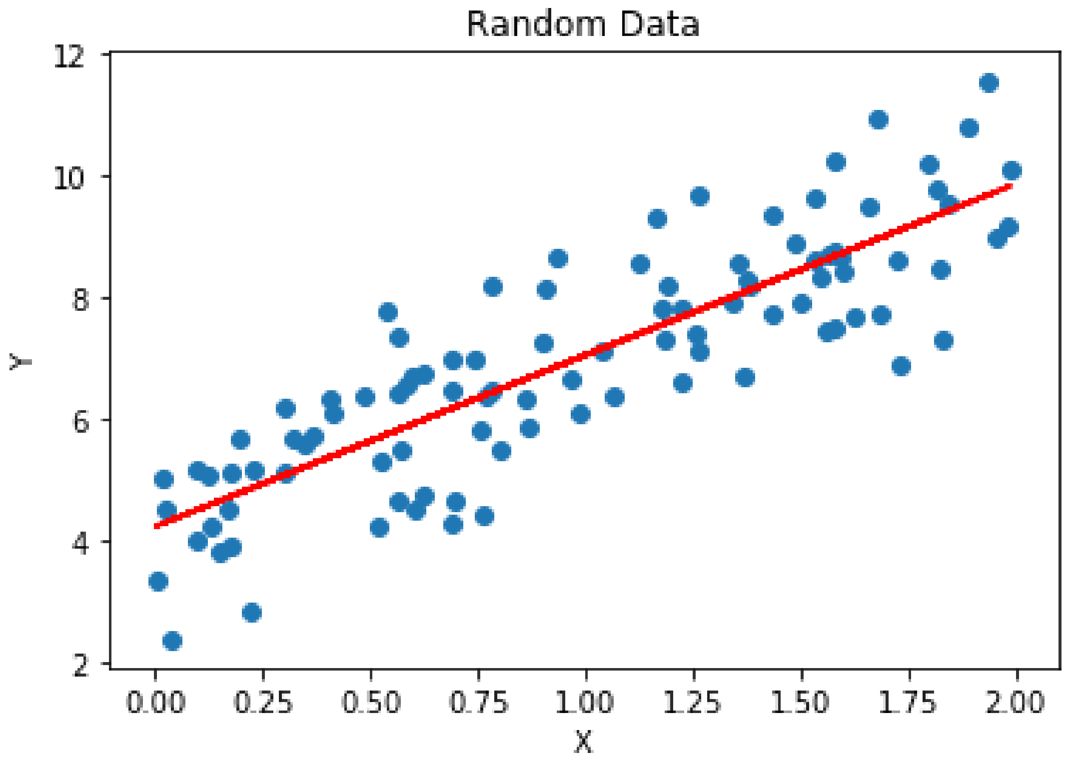


ALGORITMO LMS

DESCENSO POR GRADIENTE

Ejemplo:

Después de 1000 iteraciones se obtuvo el siguiente modelo:



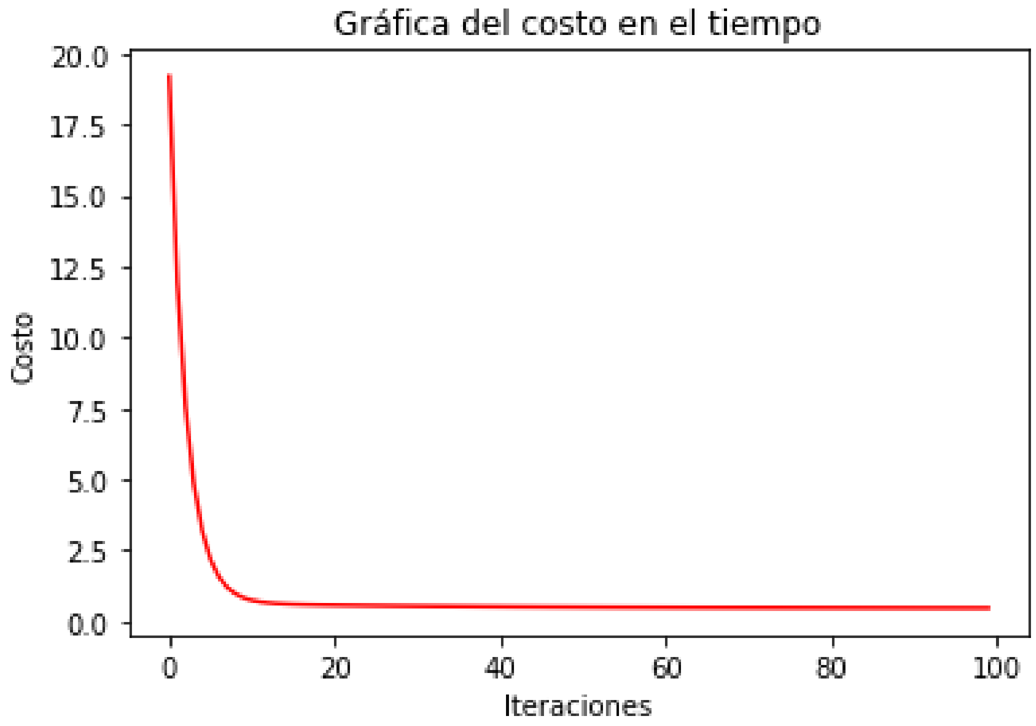
$$w_0 = 4.2174$$
$$w_1 = 2.816$$

A L G O R I T M O   L M S

D E S C E N S O   P O R   G R A D I E N T E

Ejemplo:

Error en función de 100 iteraciones.



ECUACIONES NORMALES

MODELO LINEAL MATRICIAL

DERIVACIÓN

ECUACIONES NORMALES

NOTACIÓN MATRICIAL

Ahora se quiere **calcular** el **mínimo** de la **función** de **costo** de manera **analítica**. Para esto, se introduce la **notación matricial**, lo que **permite expresar** de manera **elegante** las **derivadas** de la **función** de **costo** sin entrar en tanta **verborrea**.

Se representan los $\boldsymbol{m}$ datos de **entrenamiento** $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}); \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{m}\}$ como una matriz $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{m} \times \boldsymbol{n}}$, al conjunto de **salidas** como un **vector** $\vec{\boldsymbol{y}} \in \mathbb{R}^{\boldsymbol{m}}$ y al los $\boldsymbol{n}$ **pesos** $\boldsymbol{w_j}$ como un vector $\vec{\boldsymbol{w}} \in \mathbb{R}^{\boldsymbol{n}}$.

$$\boldsymbol{x}^{(i)} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \vdots \\ \boldsymbol{x_n} \end{bmatrix} \qquad \boldsymbol{X} = \begin{bmatrix} -\left(\boldsymbol{x}^{(1)}\right)^{\boldsymbol{T}} - \\ -\left(\boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}} - \\ \vdots \\ -\left(\boldsymbol{x}^{(m)}\right)^{\boldsymbol{T}} - \end{bmatrix} \qquad \vec{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \boldsymbol{y}^{(2)} \\ \vdots \\ \boldsymbol{y}^{(m)} \end{bmatrix} \qquad \vec{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{w_1} \\ \boldsymbol{w_2} \\ \vdots \\ \boldsymbol{w_n} \end{bmatrix}$$

ECUACIONES NORMALES

NOTACIÓN MATRICIAL

Se representa la **función** de **error absoluto** en **notación matricial** donde $\boldsymbol{h_w}\left(\boldsymbol{x}^{(i)}\right) = \left(\boldsymbol{x}^{(i)}\right)^{\boldsymbol{T}}\vec{\boldsymbol{w}}$:

$$\boldsymbol{X}\vec{\boldsymbol{w}} - \vec{\boldsymbol{y}} = \begin{bmatrix} -\left(\boldsymbol{x}^{(1)}\right)^{\boldsymbol{T}}\vec{\boldsymbol{w}} - \\ -\left(\boldsymbol{x}^{(2)}\right)^{\boldsymbol{T}}\vec{\boldsymbol{w}} - \\ \vdots \\ -\left(\boldsymbol{x}^{(m)}\right)^{\boldsymbol{T}}\vec{\boldsymbol{w}} - \end{bmatrix} - \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \boldsymbol{y}^{(2)} \\ \vdots \\ \boldsymbol{y}^{(m)} \end{bmatrix}$$

$$\boldsymbol{X}\vec{\boldsymbol{w}} - \vec{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{h_w}\left(\boldsymbol{x}^{(1)}\right) - \boldsymbol{y}^{(1)} \\ \boldsymbol{h_w}\left(\boldsymbol{x}^{(2)}\right) - \boldsymbol{y}^{(2)} \\ \vdots \\ \boldsymbol{h_w}\left(\boldsymbol{x}^{(m)}\right) - \boldsymbol{y}^{(m)} \end{bmatrix}$$

ECUACIONES NORMALES

NOTACIÓN MATRICIAL

El **término** del **error cuadrático medio** (función de costo):

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)^2$$

puede ser **representado** en forma **vectorial** usando la **propiedad** $z^T z = \sum_i {z_i}^2$

$$J(w) = \frac{1}{2m}(X\vec{w} - \vec{y})^T(X\vec{w} - \vec{y})$$

ECUACIONES NORMALES

DERIVACIÓN DE LAS ECUACIONES

Se tienen que **obtener** las **derivadas** de $\boldsymbol{J}(\boldsymbol{w})$ con respecto al vector $\boldsymbol{w}$.

$$\nabla_w J(w) = \nabla_w \frac{1}{2m}(X\vec{w} - \vec{y})^T (X\vec{w} - \vec{y})$$

El **gradiente** nos **calcula** todas las **derivadas** con **respecto** a **todos** los pesos $\boldsymbol{w_j}$ de manera **simultánea**.

Además, gracias a que **representamos** al **conjunto** de **datos** de **entrenamiento** en forma **matricial**, podemos **calcular** el **gradiente** para **todos** los **datos** de **entrenamiento**.

ECUACIONES NORMALES

DERIVACIÓN DE LAS ECUACIONES

Se tiene entonces que:

$$\nabla_w J(w) = \nabla_w \frac{1}{2m}(X\vec{w} - \vec{y})^T (X\vec{w} - \vec{y})$$

DERIVAR LA IGUALDAD

$$\nabla_w J(w) = \frac{1}{m}\left(X^T X\vec{w} - X^T\vec{y}\right)$$

ECUACIONES NORMALES

DERIVACIÓN DE LAS ECUACIONES

Igualando a **cero** la **derivada** para encontrar el **mínimo** se obtiene el **vector** de **pesos** que da ese **mínimo**:

$$\nabla_w J(w) = \frac{1}{m}\left(X^T X\vec{w} - X^T\vec{y}\right) = 0$$

$$\vec{w} = \left(X^T X\right)^{-1} X^T\vec{y}$$

INTERPRETACIÓN PROBABILÍSTICA

SUPOSICIONES PARA ERRORES

FUNCIÓN DE VEROSIMILTUD

INTERPRETACIÓN PROBABILÍSTICA

MOTIVACIÓN DEL DESARROLLO

¿PORQUÉ MINIMIZAMOS LA FUNCIÓN DE ERROR CUADRÁTICO MEDIO Y NO OTRA FUNCIÓN?

INTERPRETACIÓN PROBABILÍSTICA

SUPOSICIÓN DE ERRORES

Se pueden definir a las **variables** de **salida** $\boldsymbol{y}^{(i)}$ como la **hipótesis planteada** $\boldsymbol{h_w}$ **más** un **error** $\boldsymbol{\varepsilon}^{(i)}$ de **estimación** que captura los **efectos** que **no consideramos** en nuestra **hipótesis** o contempla los **errores aleatorios**.

$$\boldsymbol{y}^{(i)} = \boldsymbol{h_w}\left(\boldsymbol{x}^{(i)}\right) + \boldsymbol{\varepsilon}^{(i)}$$

INTERPRETACIÓN PROBABILÍSTICA

SUPOSICIÓN DE ERRORES

Se hacen las **siguientes suposiciones** para los **errores $\varepsilon^{(i)}$** (**muestreo aleatorio**): **IID**

Errores independientes: la probabilidad de que suscite un error no afecta a la probabilidad de los demás errores.

Errores idénticamente distribuidos: el muestreo de los errores se realiza de la misma distribución de probabilidad.

Distribuidos por una distribución de probabilidad Gaussiana con media $\mu = 0$ y varianza σ^2.

$$\varepsilon^{(i)} \sim N(0, \sigma^2)$$

INTERPRETACIÓN PROBABILÍSTICA

SUPOSICIÓN DE ERRORES

Por lo tanto, la **función** de **densidad** de **probabilidad** de $\boldsymbol{\varepsilon}^{(i)}$ está **dada** por:

$$\boldsymbol{p\left(\varepsilon^{(i)}\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{\left(\varepsilon^{(i)}\right)^2}{2\sigma^2}\right)}}$$

INTERPRETACIÓN PROBABILÍSTICA

SUPOSICIÓN DE ERRORES

¿PORQUÉ PROPONER UN MODELO QUE ESTÁ DISTRIBUIDO NORMALMENTE?

INTERPRETACIÓN PROBABILÍSTICA

SUPOSICIÓN DE ERRORES

Además, se establece a las **salidas** $\boldsymbol{y^{(i)}}$ y a las **entradas** $\boldsymbol{x^{(i)}}$ como **variables aleatorias**.

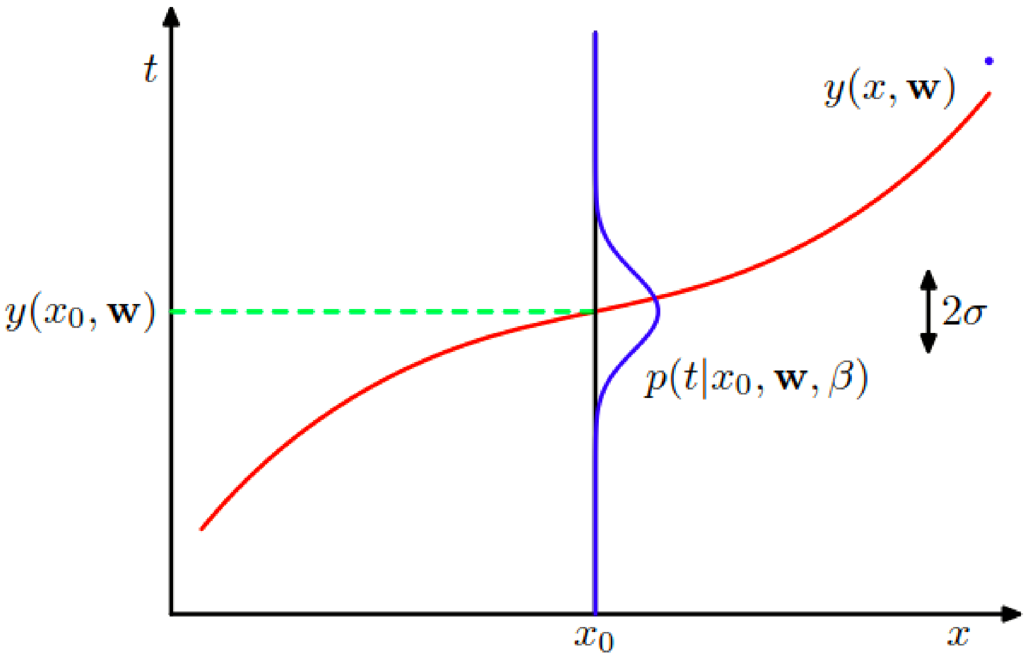
Por lo tanto, se realiza la siguiente pregunta:

Dado mi conjunto de datos de entrada $\boldsymbol{X}$ y mi vector de pesos $\vec{\boldsymbol{w}}$ ¿cuál es la distribución de probabilidad que modela a mis salidas $\vec{\boldsymbol{y}}$?

INTERPRETACIÓN PROBABILÍSTICA

SUPOSICIÓN DE ERRORES

Siguiendo las **suposiciones** de la **distribución normal** de **errores**, surge naturalmente que **la distribución** de las salidas $\boldsymbol{y^{(i)}}$ siga una **forma normal** con media $\boldsymbol{h_w\left(x^{(i)}\right)}$ y **varianza** $\boldsymbol{\sigma^2}$.



INTERPRETACIÓN PROBABILÍSTICA

SUPOSICIÓN DE ERRORES

Formalmente quedaría así:

$$p(y^{(i)}/x^{(i)};w) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{\left(y^{(i)}-h_w(x^{(i)})\right)^2}{2\sigma^2}\right)}$$

$$p(y^{(i)}/x^{(i)};w) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{\left(y^{(i)}-w^T x^{(i)}\right)^2}{2\sigma^2}\right)}$$

INTERPRETACIÓN PROBABILÍSTICA

SUPOSICIÓN DE ERRORES

Definimos la **probabilidad condicionada** para **todos** los **datos** de **entrenamiento** (suponiendo que las **muestras** se **recolectaron** de manera **independiente**):

$$\boldsymbol{p(\vec{y}/X; w) = \prod_{i=1}^{m} p(y^{(i)}/x^{(i)}; w) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{\left(y^{(i)} - w^T x^{(i)}\right)^2}{2\sigma^2}\right)}}$$

INTERPRETACIÓN PROBABILÍSTICA

FUNCIÓN DE VEROSIMILTUD

A está ecuación se le llama función de **verosimilitud**, porque describe la **probabilidad** de que nuestras **creencias** sobre la **realidad** de las **observaciones** (datos) sean **ciertas**. Es decir que la **hipótesis** que **planteamos** sea la **correcta**.

$$p(\vec{y}/X;\vec{w}) = \prod_{i=1}^{m} p(y^{(i)}/x^{(i)};\vec{w}) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma}} e^{\left(-\frac{\left(y^{(i)}-w^T x^{(i)}\right)^2}{2\sigma^2}\right)}$$

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

INTERPRETACIÓN PROBABILÍSTICA

FUNCIÓN DE VEROSIMILTUD

Por lo tanto, nosotros queremos **maximizar** la **probabilidad** de que **nuestras creencias** sobre como se **distribuyen** los **datos** (de manera **normal**).

Es decir queremos **maximizar** la **función** de **verosimilitud**. Desde una perspectiva **frecuentista** (el valor de $\vec{w}$ no es **aleatorio**), queremos **encontrar** el **valor** de $\vec{w}$ que **maximice** la **probabilidad** de que las **observaciones** que se hacen de la **realidad** sean **ciertas**.

$$\arg\max_{\vec{w}} L(\vec{w}; X, \vec{y}) = \arg\max_{\vec{w}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma}} e^{\left(-\frac{\left(y^{(i)} - w^{T} x^{(i)}\right)^{2}}{2\sigma^{2}}\right)}$$

INTERPRETACIÓN PROBABILÍSTICA

VEROSIMILTUD LOGARÍTMICA

Es mucho más fácil **maximizar** una **sumatoria** que una **multiplicación**

$$\underset{\vec{w}}{\arg\max}\ log\left(L(\vec{w})\right) =_{\underset{\vec{w}}{\arg\max}} log\left(\prod_{i=1}^{m} p\left(y^{(i)}/x^{(i)};\vec{w}\right)\right)$$

$$\underset{\vec{w}}{\arg\max}\ log(L(\vec{w})) =_{\underset{\vec{w}}{\arg\max}} \sum_{i=1}^{m} log\left(p\left(y^{(i)}/x^{(i)};\vec{w}\right)\right)$$

INTERPRETACIÓN PROBABILÍSTICA

PÉRDIDA LOGARÍTMICA

Se **convierte** el **problema** de **maximización** en uno de **minimización** al **escalar** la función por un **signo menos**.

A esta **función** $-\boldsymbol{log}(\boldsymbol{L}(\vec{\boldsymbol{w}}))$ se le llama **pérdida logarítmica**.

$$\underset{\vec{w}}{\arg\min}\ -\boldsymbol{log}(\boldsymbol{L}(\vec{\boldsymbol{w}})) =_{\underset{\vec{w}}{\arg\min}} - \sum_{i=1}^{m} \log(\boldsymbol{p}(\boldsymbol{y}^{(i)}/\boldsymbol{x}^{(i)};\vec{\boldsymbol{w}}))$$

$$\underset{\vec{w}}{\arg\min}\ -\boldsymbol{log}(\boldsymbol{L}(\vec{\boldsymbol{w}})) =_{\underset{\vec{w}}{\arg\min}} - \sum_{i=1}^{m} \log(\frac{1}{\sqrt{2\pi\sigma}} e^{\left(-\frac{\left(y^{(i)} - w^{T}x^{(i)}\right)^{2}}{2\sigma^{2}}\right)})$$

INTERPRETACIÓN PROBABILÍSTICA

PÉRDIDA LOGARÍTMICA

Desarrollamos la ecuación:

$$\underset{\vec{w}}{\arg\min}\ -log(L(\vec{w})) =_{\underset{\vec{w}}{\arg\min}} -\sum_{i=1}^{m} log\left(\frac{1}{\sqrt{2\pi\sigma}}\right) + log\left(e^{\left(-\frac{\left(y^{(i)} - w^T x^{(i)}\right)^2}{2\sigma^2}\right)}\right)$$

$$\underset{\vec{w}}{\arg\min}\ -log(L(\vec{w})) =_{\underset{\vec{w}}{\arg\min}} -m\, log\left(\frac{1}{\sqrt{2\pi\sigma}}\right) - \sum_{i=1}^{m} -\frac{\left(y^{(i)} - w^T x^{(i)}\right)^2}{2\sigma^2}$$

INTERPRETACIÓN PROBABILÍSTICA

PÉRDIDA LOGARÍTMICA

Desarrollamos la ecuación:

$$\operatorname*{arg\,min}_{\vec{w}} -log(L(\vec{w})) =_{\operatorname*{arg\,min}_{\vec{w}}} c + \sum_{i=1}^{m} \frac{\left(y^{(i)} - w^T x^{(i)}\right)^2}{2\sigma^2}$$

$$\operatorname*{arg\,min}_{\vec{w}} -log(L(\vec{w})) =_{\operatorname*{arg\,min}_{\vec{w}}} \frac{1}{2\sigma^2} \sum_{i=1}^{m} \left(y^{(i)} - w^T x^{(i)}\right)^2$$

INTERPRETACIÓN PROBABILÍSTICA

ERROR CUADRÁTICO MEDIO

Minimizar la **pérdida logarítmica** es lo **mismo** que **minimizar** el **error cuadrático medio** $MSE = J(\vec{w})$:

$$\arg\min_{\vec{w}} MSE = \arg\min_{\vec{w}} \frac{1}{2m} \sum_{i=1}^{m} (y^{(i)} - w^T x^{(i)})^2$$

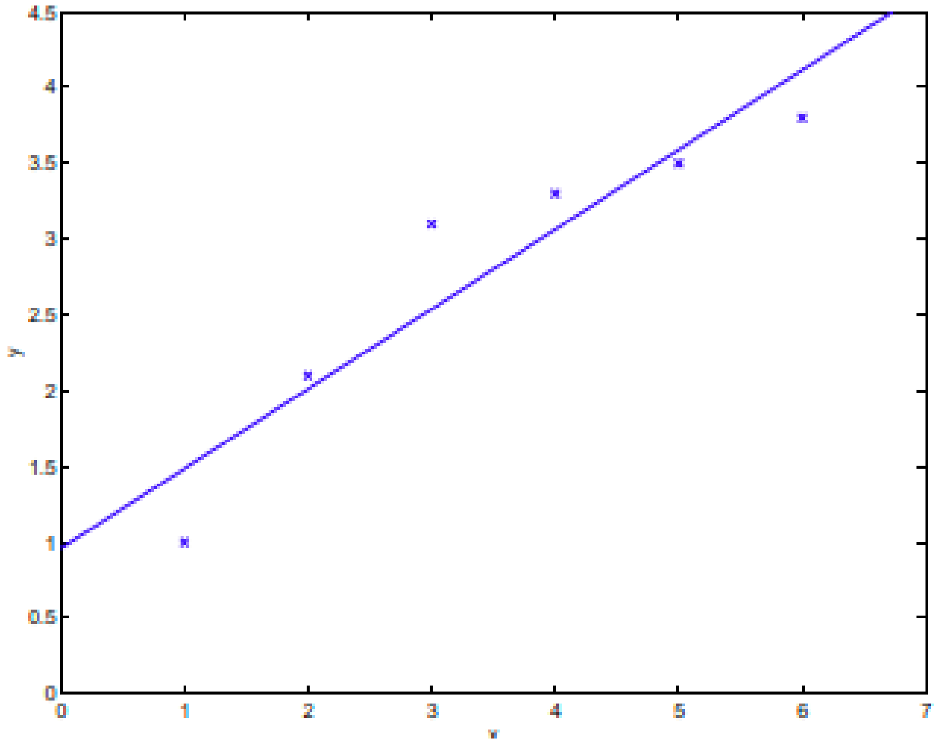
$$\arg\min_{\vec{w}} -log(L(\vec{w})) =_{\arg\min_{\vec{w}}} \frac{1}{2\sigma^2} \sum_{i=1}^{m} \left(y^{(i)} - w^T x^{(i)}\right)^2$$

FUNCIONES DE BASE

FUNCIONES DE BASE

PROBLEMA DE LINEALIDAD

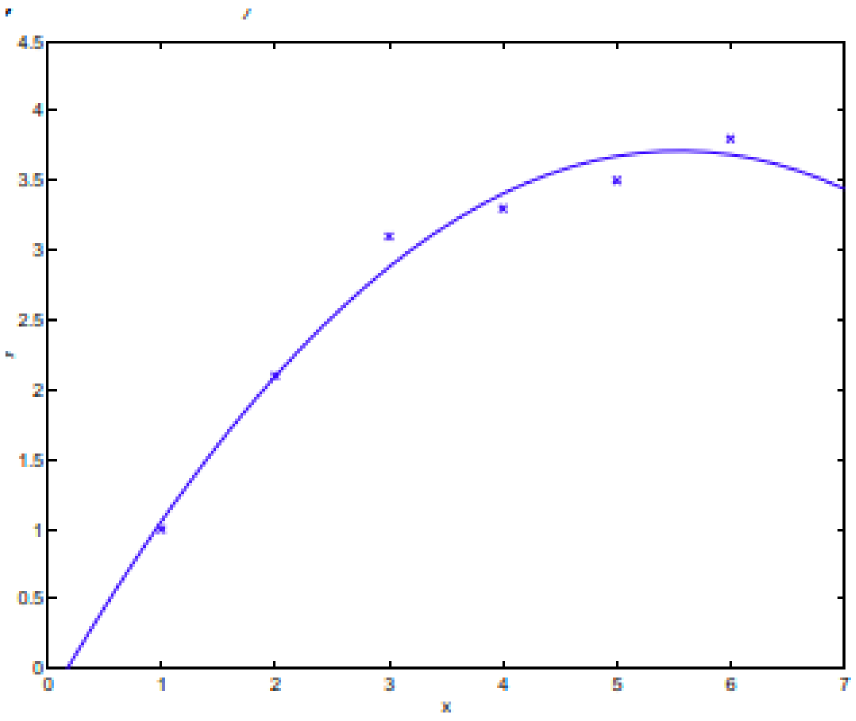
El **mundo real** tiene comportamientos **no lineales,** por lo que los **modelos de regresión lineal** vistos hasta ahora tienen sus **limitantes.**



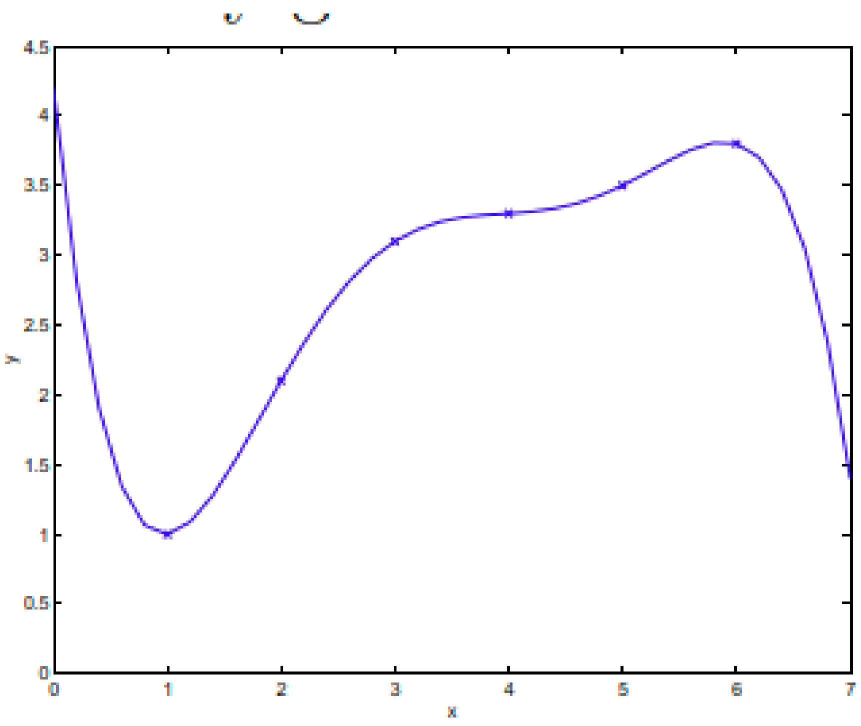
FUNCIONES DE BASE

PROBLEMA DE LINEALIDAD

Podemos **establecer** nuestra **hipótesis** como un **modelo polinómico**.



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



$$y = \sum_{j=0}^{5} \theta_j x^j$$

FUNCIONES DE BASE

INTRODUCCIÓN A FUNCIONES DE BASE

Si llevamos este tipo de pensamiento más allá, podemos **construir** modelos como **combinaciones lineales** de **funciones no lineales** $\boldsymbol{\phi_j(x)}$, llamadas **funciones de base**. Donde $\boldsymbol{\phi}: \mathbb{R}^{\boldsymbol{n}} \rightarrow \mathbb{R}^{\boldsymbol{k}}$

$$\boldsymbol{h_w\left(x^{(i)}\right) = \sum_{j=1}^{k} w_j \phi_j\left(x^{(i)}\right)}$$

$$\boldsymbol{h_w\left(x^{(i)}\right) = w^T \phi\left(x^{(i)}\right)}$$

NOTA
$\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{k}}$
$\boldsymbol{x^{(i)}} \in \mathbb{R}^{\boldsymbol{n}}$

Así logramos **construir** un **modelo no lineal**, pero que sigue estando **parametrizado** por **pesos lineales** w.

FUNCIONES DE BASE

INTRODUCCIÓN A FUNCIONES DE BASE

De manera más detallada, se tiene que:

$$\boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right) = \begin{bmatrix} \boldsymbol{\phi}_1\left(\boldsymbol{x}^{(i)}\right) \\ \boldsymbol{\phi}_j\left(\boldsymbol{x}^{(i)}\right) \\ \vdots \\ \boldsymbol{\phi}_k\left(\boldsymbol{x}^{(i)}\right) \end{bmatrix}$$

FUNCIONES DE BASE

REGRESIÓN LINEAL CLÁSICA

En el caso de **regresión lineal clásica** se tiene **para un solo** dato de entrenamiento $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^{\boldsymbol{k}}$ en este caso $\boldsymbol{k} = \boldsymbol{n}$:

$$\boldsymbol{\phi}_j\left(\boldsymbol{x}^{(i)}\right) = \boldsymbol{x}^{(i)} \qquad \boldsymbol{x}^{(i)} = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_k \end{bmatrix}$$

FUNCIONES DE BASE

REGRESIÓN LINEAL CLÁSICA

En el caso de **regresión lineal clásica** se tiene **para un solo** dato de entrenamiento $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^{\boldsymbol{k}}$ en este caso $\boldsymbol{k} = \boldsymbol{n}$:

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\phi}_0(\boldsymbol{x}^{(1)}) \\ \boldsymbol{\phi}_1(\boldsymbol{x}^{(i)}) \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \boldsymbol{x}^{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix}$$

$$\boldsymbol{h}_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right) = \boldsymbol{w}^{T}\boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right) = \boldsymbol{w}_0 + \boldsymbol{w}_1(\boldsymbol{x}_1) + \cdots + \boldsymbol{w}_k(\boldsymbol{x}_n)$$

FUNCIONES DE BASE

REGRESIÓN LINEAL CLÁSICA

En el caso de **regresión lineal clásica** se tiene **para m datos** de entrenamiento:

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{bmatrix} \phi_0(x^{(1)}) & \phi_1(x^{(1)}) \\ \vdots & \vdots \\ \phi_0(x^{(m)}) & \phi_1(x^{(m)}) \end{bmatrix} = \begin{bmatrix} 1 & {x^{(1)}}^T \\ \vdots & \vdots \\ 1 & {x^{(m)}}^T \end{bmatrix}$$

$$\boldsymbol{w} = [w_0 \quad \cdots \quad w_k]$$

FUNCIONES DE BASE

REGRESIÓN LINEAL CLÁSICA

En el caso de **regresión lineal clásica** se tiene **para m datos** de entrenamiento:

$$\boldsymbol{h_w(x) = w^T \phi(x) = \begin{bmatrix} w_0 + \cdots + w_k x_n{}^{(1)} \\ \vdots \\ w_0 + \cdots + w_k x_n{}^{(m)} \end{bmatrix}}$$

$$\boldsymbol{h_w(x) = \begin{bmatrix} h_w\left(x^{(1)}\right) \\ \vdots \\ h_w\left(x^{(m)}\right) \end{bmatrix}}$$

FUNCIONES DE BASE

MATRIZ DE DISSEÑO

Por lo tanto, la **matriz** de **diseño** para cualquier $\boldsymbol{\phi}(\boldsymbol{x})$ estaría dada por:

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\phi}_0(\boldsymbol{x}^{(1)}) & \cdots & \boldsymbol{\phi}_{k-1}(\boldsymbol{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\phi}_0(\boldsymbol{x}^{(m)}) & \cdots & \boldsymbol{\phi}_{k-1}(\boldsymbol{x}^{(m)}) \end{bmatrix}$$

$$\boldsymbol{x}^{(i)} \in \mathbb{R}^{\boldsymbol{n}}$$

Donde cada fila de la matriz $\boldsymbol{\phi}(\boldsymbol{x})$ está dada por $\boldsymbol{\phi}_{\boldsymbol{j}} = \boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)^{\boldsymbol{T}}$

FUNCIONES DE BASE

FUNCIONES DE BASE LINEALES

Funciones lineales clásicas:

$$\boldsymbol{\phi_j\left(x^{(i)}\right) = x^{(i)}; x^{(i)} \in \mathbb{R}^{n+1}}$$

$$\boldsymbol{w = \begin{bmatrix} w_0 \\ \vdots \\ w_k \end{bmatrix}; k = n}$$

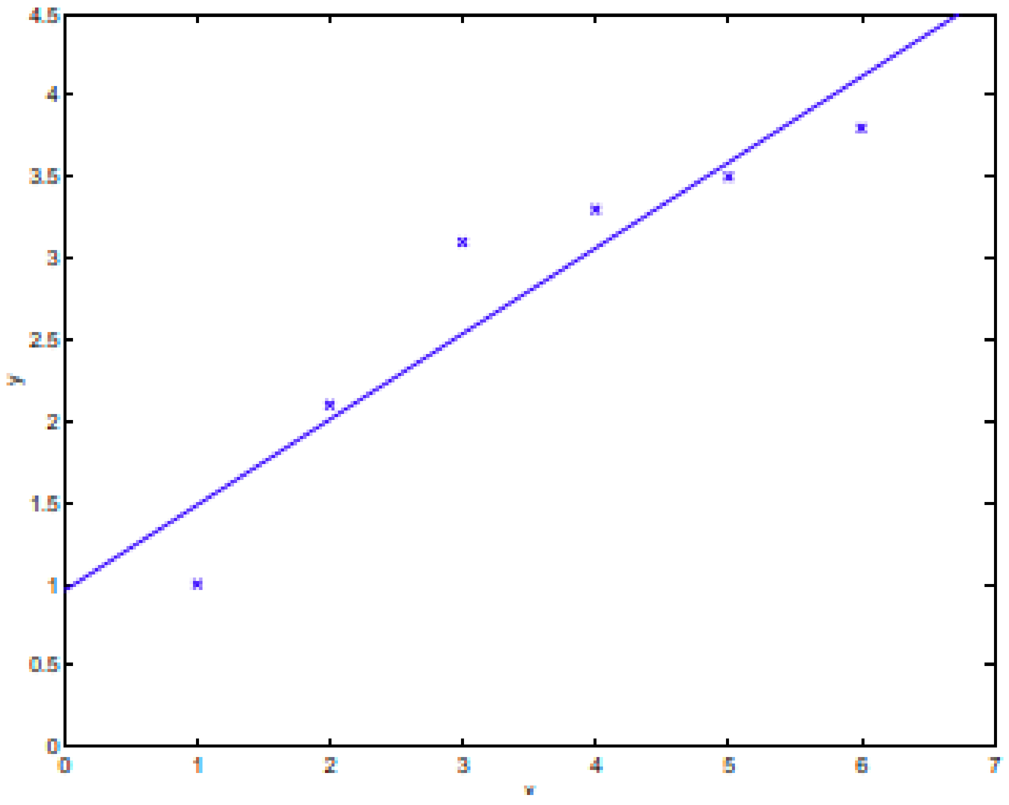
$$\boldsymbol{\phi(x) = \begin{bmatrix} 1 & {x^{(1)}}^T \\ \vdots & \vdots \\ 1 & {x^{(m)}}^T \end{bmatrix} = \begin{bmatrix} 1 & \cdots & {x_n}^{(1)} \\ \vdots & \ddots & \vdots \\ 1 & \cdots & {x_n}^{(m)} \end{bmatrix}}$$

$$\boldsymbol{h_w(x) = w^T\phi(x) = \begin{bmatrix} w_0 + w_1{x_1}^{(1)} + \cdots + w_k{x_n}^{(1)} \\ \vdots \\ w_0 + w_1{x_1}^{(m)} + \cdots + w_k{x_n}^{(m)} \end{bmatrix}}$$

FUNCIONES DE BASE

FUNCIONES DE BASE LINEALES

Funciones lineales clásicas:



FUNCIONES DE BASE

FUNCIONES DE BASE POLINÓMICAS

Funciones de base polinómicas:

$$\phi_j(x) = \left(x^{(i)}\right)^j;\, x^{(i)} \in \mathbb{R}^n$$

$$w = \begin{bmatrix} w_0 \\ \vdots \\ w_{k*n+1} \end{bmatrix}$$

$$\phi(x) = \begin{bmatrix} 1 & {x^{(1)}}^T & \dots & \left({x^{(1)}}^T\right)^{(k-1)} \\ \vdots & \vdots & & \vdots \\ 1 & {x^{(m)}}^T & \dots & \left({x^{(m)}}^T\right)^{(k-1)} \end{bmatrix}$$

$$h_w(x) = w^T\phi(x) = \begin{bmatrix} w_0 + w_1 {x_1}^{(1)} + \dots + w_{k*n}\left({x_n}^{(1)}\right)^{k-1} \\ \vdots \\ w_0 + w_1 {x_1}^{(m)} + \dots + w_{k*n}\left({x_n}^{(m)}\right)^{k-1} \end{bmatrix}$$

FUNCIONES DE BASE

FUNCIONES DE BASE POLINÓMICAS

Funciones de base polinómicas: ejemplo concreto polinomio grado 2 y dos datos de entrenamiento $\boldsymbol{m} = \mathbf{2}$

$$\boldsymbol{x}^{(i)} = \begin{bmatrix} \mathbf{1} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_0 \\ \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \boldsymbol{w}_3 \\ \boldsymbol{w}_4 \end{bmatrix}$$

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{1} & \boldsymbol{x}_1{}^{(1)} & \boldsymbol{x}_2{}^{(1)} & \left(\boldsymbol{x}_1{}^{(1)}\right)^2 & \left(\boldsymbol{x}_2{}^{(1)}\right)^2 \\ \mathbf{1} & \boldsymbol{x}_1{}^{(2)} & \boldsymbol{x}_2{}^{(2)} & \left(\boldsymbol{x}_1{}^{(2)}\right)^2 & \left(\boldsymbol{x}_2{}^{(2)}\right)^2 \end{bmatrix}$$

FUNCIONES DE BASE

FUNCIONES DE BASE POLINÓMICAS

Funciones de base polinómicas: ejemplo concreto **polinomio grado** 2 y **dos datos** de entrenamiento $\boldsymbol{m} = 2$ con **2 características**.

$$\boldsymbol{h_w(x) = w^T \phi(x)}$$

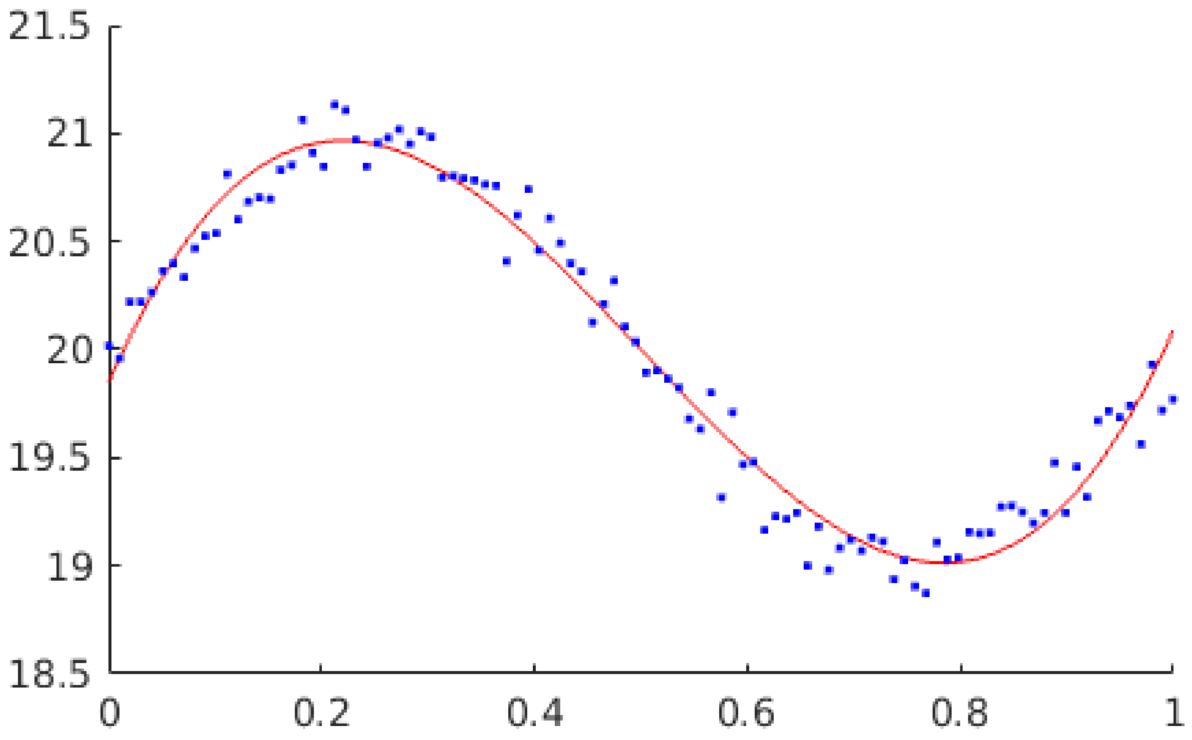
$$\boldsymbol{h_w(x)} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{x_1}^{(1)} & \boldsymbol{x_2}^{(1)} & \left(\boldsymbol{x_1}^{(1)}\right)^2 & \left(\boldsymbol{x_2}^{(1)}\right)^2 \\ \boldsymbol{1} & \boldsymbol{x_1}^{(2)} & \boldsymbol{x_2}^{(2)} & \left(\boldsymbol{x_1}^{(2)}\right)^2 & \left(\boldsymbol{x_2}^{(2)}\right)^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{w_0} \\ \boldsymbol{w_1} \\ \boldsymbol{w_2} \\ \boldsymbol{w_3} \\ \boldsymbol{w_4} \end{bmatrix}$$

$$\boldsymbol{h_w(x)} = \begin{bmatrix} \boldsymbol{w_0 + w_1 x_1}^{(1)} + \boldsymbol{w_2 x_2}^{(1)} + \boldsymbol{w_3}\left(\boldsymbol{x_1}^{(1)}\right)^2 + \boldsymbol{w_4}\left(\boldsymbol{x_2}^{(1)}\right)^2 \\ \boldsymbol{w_0 + w_1 x_1}^{(2)} + \boldsymbol{w_2 x_2}^{(2)} + \boldsymbol{w_3}\left(\boldsymbol{x_1}^{(2)}\right)^2 + \boldsymbol{w_4}\left(\boldsymbol{x_2}^{(2)}\right)^2 \end{bmatrix}$$

F U N C I O N E S D E B A S E

FUNCIONES DE BASE POLINÓMICAS

Funciones de base polinómicas:



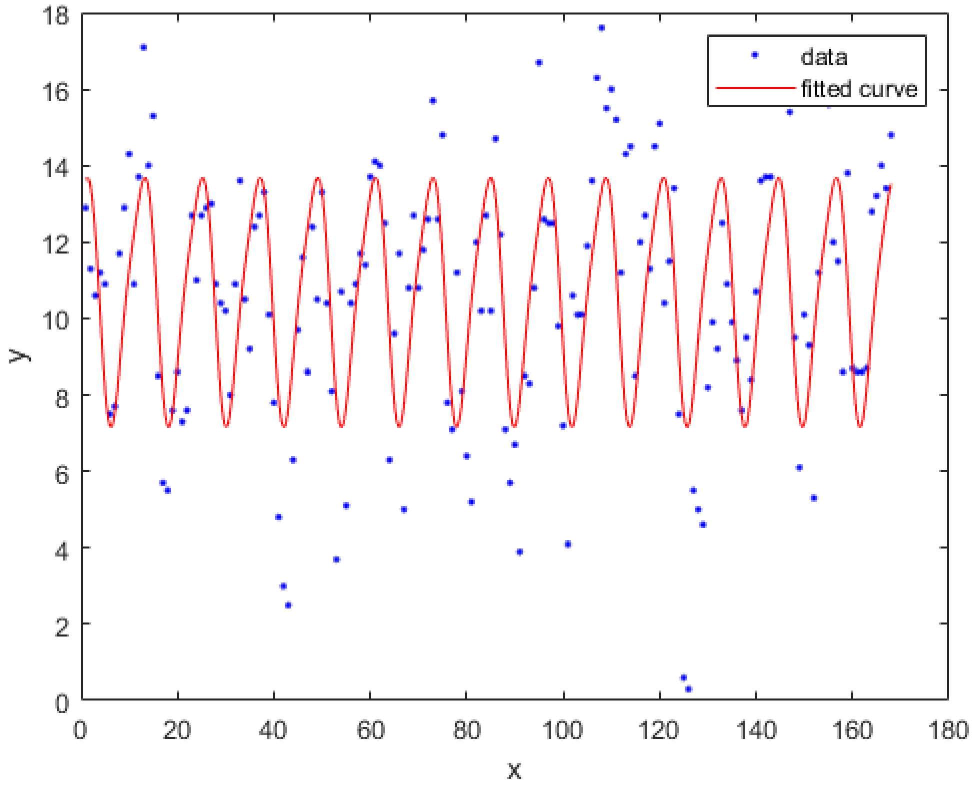
FUNCIONES DE BASE

SERIES DE FOURIER

Series de Fourier

$$\boldsymbol{\phi_0(x) = 1}$$

$$\boldsymbol{\phi_j(x)} = \cos(\boldsymbol{\varpi_j x^{(i)}} + \boldsymbol{\psi_j})\,;\boldsymbol{j > 0}$$

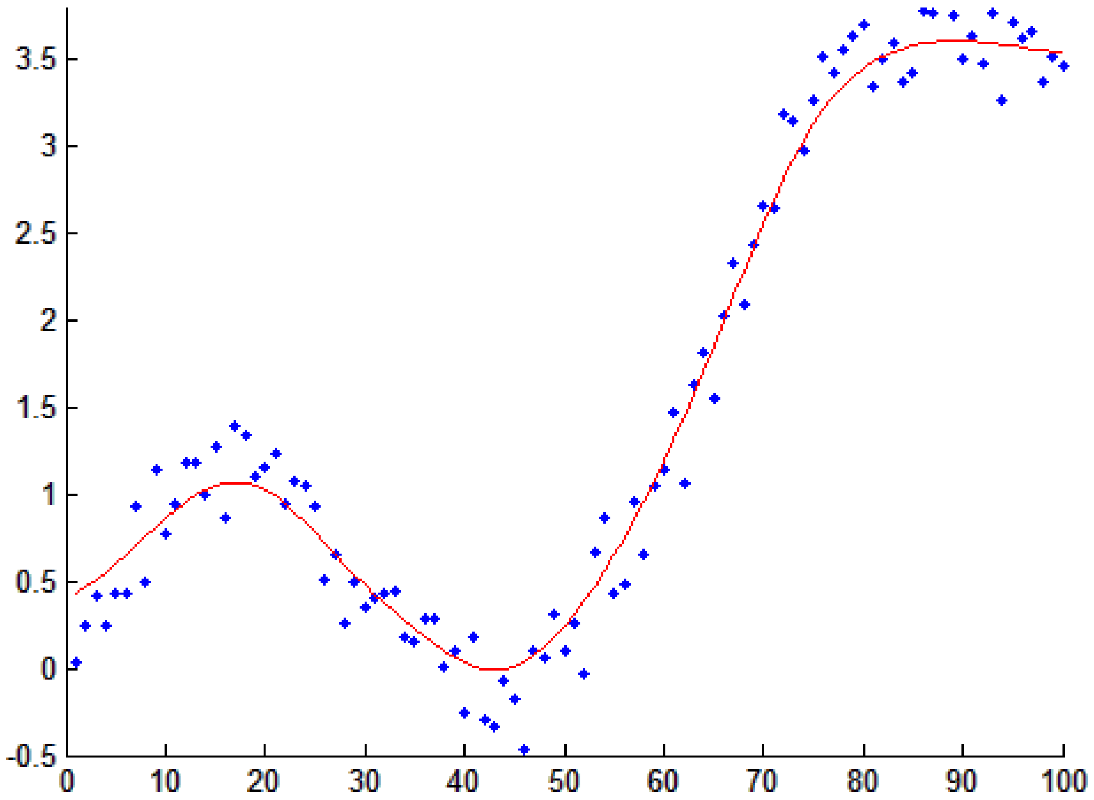


FUNCIONES DE BASE

BASE RADIAL

Funciones de base radial

$$\phi_j(x) = e^{-\frac{1}{2l}\sum_{r=1}^{n}\left(x_r^{(i)}-\mu_{j,r}\right)^2}$$

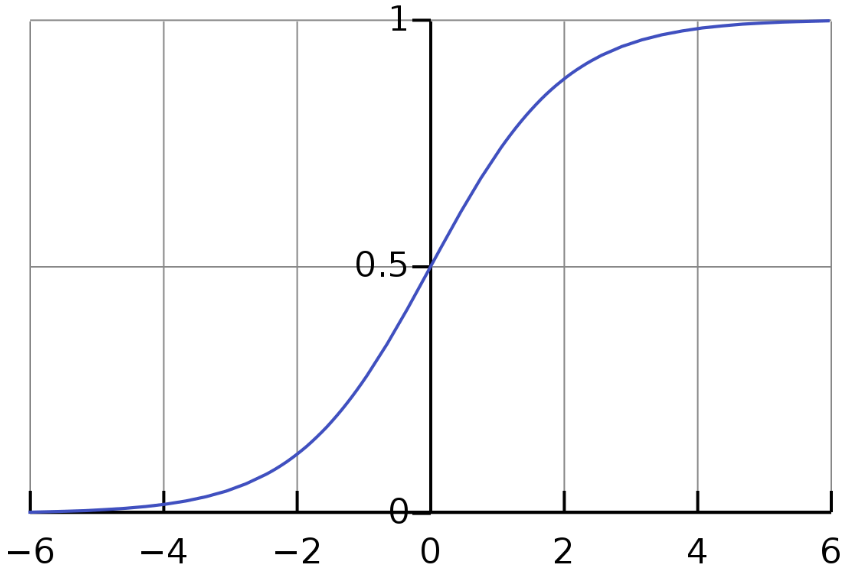


FUNCIONES DE BASE
SIGMOIDALES

Funciones sigmoidales

$$\phi_j\left(x^{(i)}\right) = \sigma\left(\frac{x^{(i)} - \mu_j}{s}\right)$$

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$



FUNCIONES DE BASE

MÁXIMA VEROSIMILTUD

Se propone el **mismo** modelo de **estimación más** un **error**:

$$\boldsymbol{y^{(i)} = h_w\left(x^{(i)}\right) + \varepsilon^{(i)}}$$

donde:

$$\boldsymbol{h_w\left(x^{(i)}\right) = w^T \phi(x)}$$

FUNCIONES DE BASE

MÁXIMA VEROSIMILTUD

Suponiendo otra vez que el **error** se **distribuye** de manera **normal** con **media 0** y **varianza $\boldsymbol{\beta^{-1}}$.** Se define para $\boldsymbol{m}$ datos de **entrenamiento**:

$$p(\vec{y}/X, w, \beta) = \prod_{i=1}^{m} p(y^{(i)}/x^{(i)}; w) = \prod_{i=1}^{m} \sqrt{\frac{\beta}{2\pi}}\, e^{\left(-\frac{\beta}{2}\left(y^{(i)} - w^T \phi\left(x^{(i)}\right)\right)^2\right)}$$

FUNCIONES DE BASE

MÁXIMA VEROSIMILTUD

Como la **función** de **verosimilitud** está **parametrizada** en $\boldsymbol{w}$ y $\boldsymbol{\beta}$ la verosimilitud logarítmica se escribe así:

$$\log p(\vec{y}/w,\beta) = \log \prod_{i=1}^{m} \sqrt{\frac{\beta}{2\pi}}\, e^{\left(-\frac{\beta}{2}\left(y^{(i)}-w^{T}\phi\left(x^{(i)}\right)\right)^{2}\right)}$$

$$\log p(\vec{y}/w,\beta) = \sum_{i=1}^{m} \log \sqrt{\frac{\beta}{2\pi}}\, e^{\left(-\frac{\beta}{2}\left(y^{(i)}-w^{T}\phi\left(x^{(i)}\right)\right)^{2}\right)}$$

FUNCIONES DE BASE

MÁXIMA VEROSIMILTUD

Desarrollando:

$$\log p(\vec{y}/w,\beta) = \sum_{i=1}^{m} \log\sqrt{\frac{\beta}{2\pi}} + \log e^{\left(-\frac{\beta}{2}\left(y^{(i)}-w^T\phi\left(x^{(i)}\right)\right)^2\right)}$$

$$\log p(\vec{y}/w,\beta) = m\log\sqrt{\frac{\beta}{2\pi}} - \sum_{i=1}^{m}\frac{\beta}{2}\left(y^{(i)} - w^T\phi\left(x^{(i)}\right)\right)^2$$

FUNCIONES DE BASE

MÁXIMA VEROSIMILTUD

Desarrollando:

$$\log p(\vec{y}/w,\beta) = \frac{m}{2}\log\beta - \frac{m}{2}\log 2\pi - \sum_{i=1}^{m}\frac{\beta}{2}\left(y^{(i)} - w^{T}\phi\left(x^{(i)}\right)\right)^{2}$$

$$\log p(\vec{y}/w,\beta) = \frac{m}{2}\log\beta - \frac{m}{2}\log 2\pi - \beta E_{D}(w)$$

$$E_{D}(w) = \frac{1}{2}\left(y^{(i)} - w^{T}\phi\left(x^{(i)}\right)\right)^{2}$$

FUNCIONES DE BASE

MÁXIMA VEROSIMILTUD

Desarrollando:

$$-\log p(\vec{y}/w,\beta) = -\frac{m}{2}\log\beta + \frac{m}{2}\log 2\pi + \beta E_D(w)$$

$$\underset{\vec{w}}{\arg\min}\ -log(L(\vec{w})) =_{\underset{\vec{w}}{\arg\min}} \frac{\beta}{2}\sum_{i=1}^{m}\left(y^{(i)} - w^T\phi\left(x^{(i)}\right)\right)^2$$

FUNCIONES DE BASE

ECUACIONES NORMALES

Calculando el **gradiente**, **igualando** a **cero** y resolviendo para el **vector w:**

$$\boldsymbol{w} = \left(\boldsymbol{\phi}^T \boldsymbol{\phi}\right)^{-1} \boldsymbol{\phi}^T \vec{\boldsymbol{y}}$$