

## Tema 2 – Optimización y Regularización (Parte 1)

Aprendizaje Automático II - Grado en Inteligencia Artificial  
Universidad Rey Juan Carlos

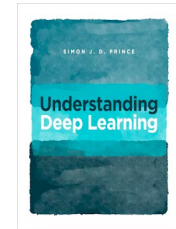
Iván Ramírez Díaz  
[ivan.ramirez@urjc.es](mailto:ivan.ramirez@urjc.es)

José Miguel Buenaposada Biencinto  
[josemiguel.buenaposada@urjc.es](mailto:josemiguel.buenaposada@urjc.es)

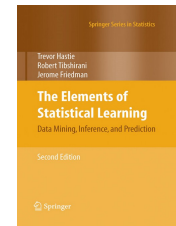
### 2.1 Planteamiento del problema

## Bibliografía

- **Understanding Deep Learning**  
Capítulo 5.



- **The Elements of Statistical Learning**  
Secciones 4.5 y 10.6.



## Aprendizaje supervisado

- **Modelo.** Función que lleva una o más entradas a una o más salidas. El modelo es una ecuación matemática:

$$y = f(\mathbf{x}; \mathbf{w})$$

Parámetros del modelo

**Nota importante:** Usaremos  $\mathbf{w}$  ó  $\Phi$  indistintamente para denominar a los parámetros del modelo

## Aprendizaje supervisado

- **Modelo.** Función que lleva una o más entradas a una o más salidas. El modelo es una ecuación matemática:

$$y = f(\mathbf{x}; \mathbf{w})$$

- **Conjunto de datos de entrenamiento.** N pares de muestras entrada/salida:

$$D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$$

## Función de pérdida

- **Función de pérdida** mide cómo de malo es el modelo para un único dato:

$$L(\mathbf{y}_i, \underbrace{f(\mathbf{x}_i; \mathbf{w})}_{\hat{\mathbf{y}}_i})$$

**Ejemplo:** error cuadrático


$$L(\mathbf{y}_i, \hat{\mathbf{y}}_i) = (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2$$

## Pérdida esperada

- **Función de coste ideal.** Pérdida esperada sobre la distribución de probabilidad  $p(\mathbf{x}, \mathbf{y})$ :

$$J(\mathbf{w}) = E_{(\mathbf{x}, \mathbf{y}) \sim p(\mathbf{x}, \mathbf{y})} [L(f(\mathbf{x}; \mathbf{w}), \mathbf{y})]$$

Esperanza o  
valor esperado




## Pérdida esperada

- **Función de coste ideal.** Pérdida esperada sobre la distribución de probabilidad  $p(\mathbf{x}, \mathbf{y})$ :

$$J(\mathbf{w}) = E_{(\mathbf{x}, \mathbf{y}) \sim p(\mathbf{x}, \mathbf{y})} [L(f(\mathbf{x}; \mathbf{w}), \mathbf{y})]$$

Esperanza o  
valor esperado



**Problema:**  $p(\mathbf{x}, \mathbf{y})$  es desconocida

## Pérdida esperada

- **Función de coste empírica.** Pérdida esperada sobre la distribución de probabilidad de  $(\mathbf{x}, \mathbf{y})$  ahora definida sobre el conjunto de entrenamiento:

$$J(\mathbf{w}) = E_{(\mathbf{x}, \mathbf{y}) \sim \hat{p}(\mathbf{x}, \mathbf{y})} [L(f(\mathbf{x}; \mathbf{w}), \mathbf{y})]$$

Esperanza o  
valor esperado

Distribución de  
probabilidad  
sacada de una  
muestra de datos

## Pérdida esperada

- **Función de coste empírica.** Pérdida esperada sobre la distribución de probabilidad de  $(\mathbf{x}, \mathbf{y})$  ahora definida sobre el conjunto de entrenamiento:

$$J(\mathbf{w}) = E_{(\mathbf{x}, \mathbf{y}) \sim \hat{p}(\mathbf{x}, \mathbf{y})} [L(f(\mathbf{x}; \mathbf{w}), \mathbf{y})]$$

Esperanza o  
valor esperado



$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N L(f(\mathbf{x}_i, \mathbf{w}), \mathbf{y}_i)$$

## Pérdida esperada

- **Función de coste empírica.** Pérdida esperada sobre la distribución de probabilidad de  $(\mathbf{x}, \mathbf{y})$  ahora definida sobre el conjunto de entrenamiento:

*Idealmente queremos encontrar el  $\mathbf{w}$  que optimiza una medida de rendimiento  $\mathbf{P}$  sobre el conjunto de test (o los infinitos datos que no hemos visto en entrenamiento). Sin embargo, en Aprendizaje automático optimizamos  $\mathbf{J}$  (sobre el subconjunto de datos de entrenamiento) con la esperanza de que optimicen también  $\mathbf{P}$ .*



$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N L(f(\mathbf{x}_i, \mathbf{w}), \mathbf{y}_i)$$

## Pérdida media

- **Función de coste.** Mide cómo de malo es el modelo:

$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N L(f(\mathbf{x}_i, \mathbf{w}), \mathbf{y}_i)$$

*En general el aprendizaje supervisado minimizará la pérdida media sobre todo el conjunto de entrenamiento  $D$*

## Pérdida media

- **Función de coste.** Mide cómo de malo es el modelo:

$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N L(f(\mathbf{x}_i, \mathbf{w}), y_i)$$

*En general el aprendizaje supervisado minimizará la pérdida media sobre todo el conjunto de entrenamiento  $D$*

**Problema:** ¿Cómo encontrar la función de pérdida,  $L$ , adecuada a cada problema de aprendizaje?

## 2.2 Funciones de pérdida

### 2.2.1 Funciones de pérdida por Máxima verosimilitud

#### 2.2.2 Funciones de pérdida basadas en el margen

#### 2.2.3 Funciones de pérdida para regresión

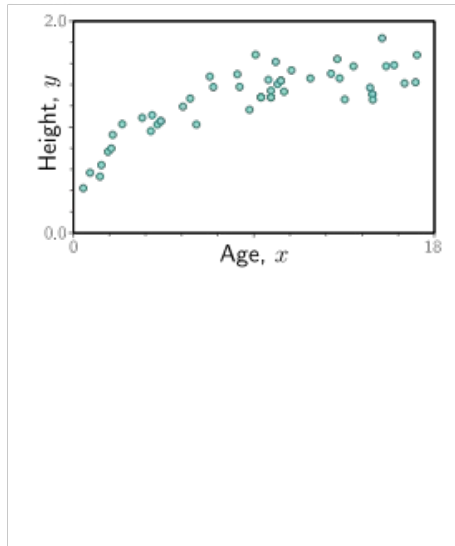
### 2.2.1 Funciones de pérdida por máxima verosimilitud

- **Máxima verosimilitud (likelihood)**
- Receta para funciones de pérdida
- Ejemplo 1: regresión univariante
- Ejemplo 2: clasificación binaria
- Ejemplo 3: clasificación multiclase
- Otros tipos de datos
- Múltiples salidas
- entropía cruzada (cross entropy).

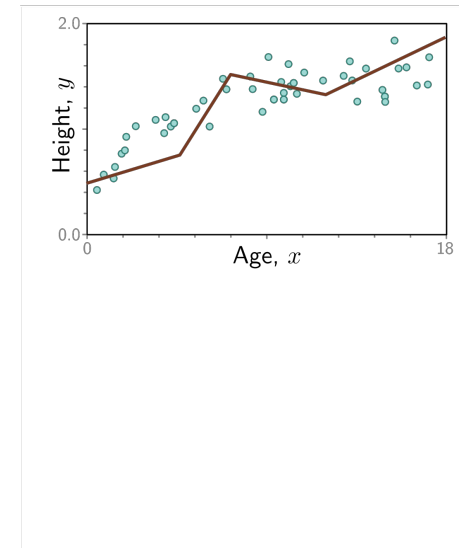
## Como construir la función de pérdida

- El modelo predice la salida  $\mathbf{y}$ , dada la entrada  $\mathbf{x}$

## Ejemplos: Regresión 1D



## Ejemplos: Regresión 1D



## Cómo construir la función de pérdida

- El modelo predice la salida  $y$ , dada la entrada,  $x$

## Cómo construir la función de pérdida

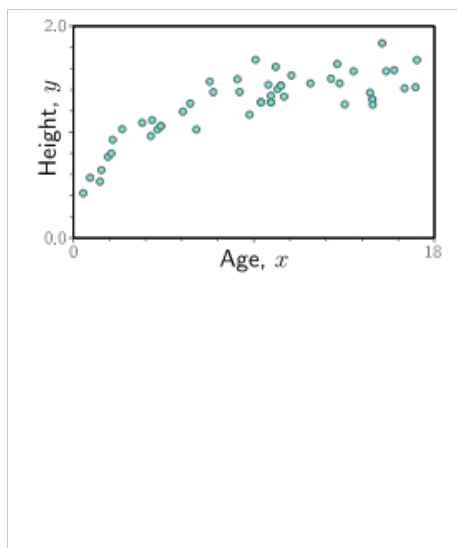
- El modelo predice la salida  $y$ , dada la entrada,  $x$
- El modelo predice la distribución de probabilidad condicional:

$$Pr(y|x)$$

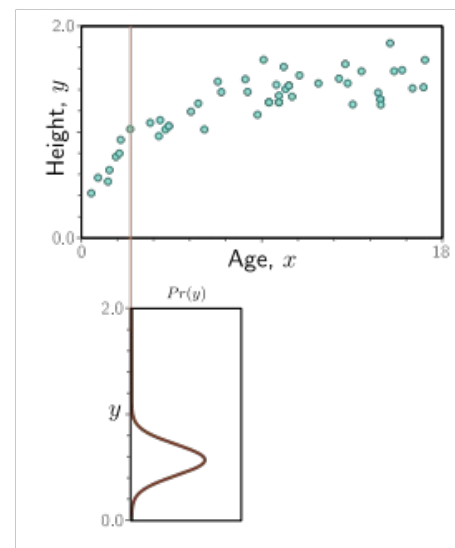
sobre las salidas,  $y$ , dadas las entradas,  $x$ .

- La función de pérdida intenta que las salidas tengan probabilidad alta.

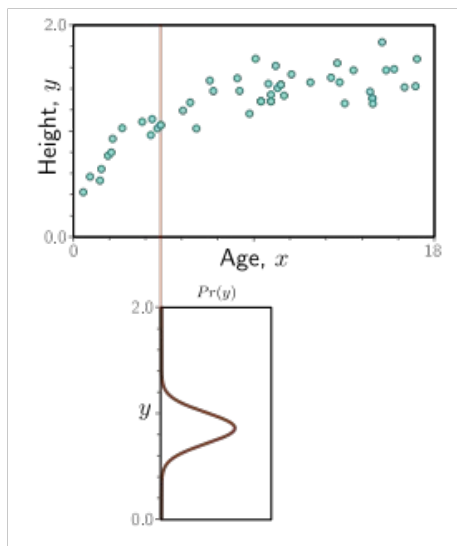
## Ejemplos: Regresión 1D



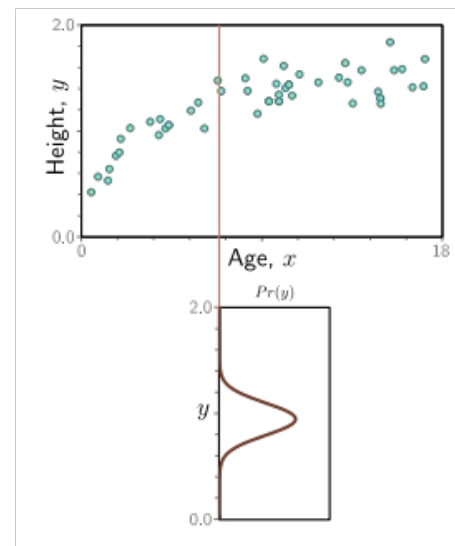
## Ejemplos: Regresión 1D



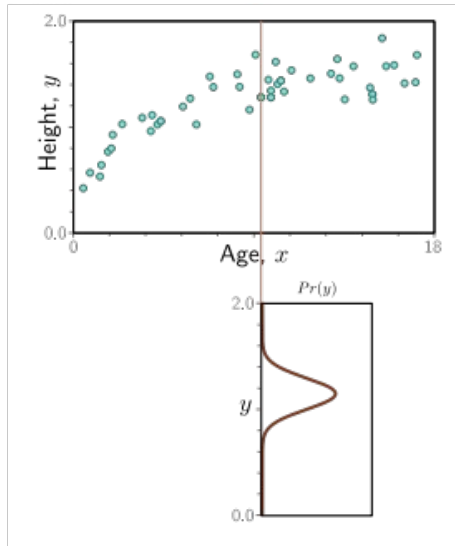
## Ejemplos: Regresión 1D



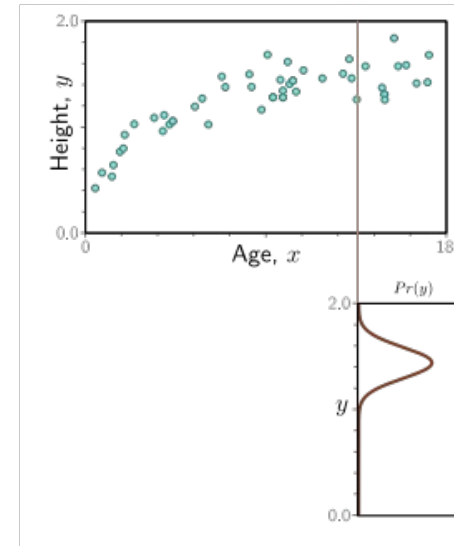
## Ejemplos: Regresión 1D



## Ejemplos: Regresión 1D



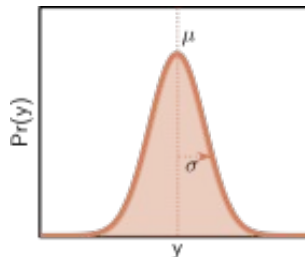
## Ejemplos: Regresión 1D



## Predecir una distribución de probabilidad con $f[x, \Phi]$

1. Asumimos una distribución conocida para modelar la salida,  $y$ , con parámetros  $\theta$ .

p.ej. Distribución normal  $\theta = \{\mu, \sigma^2\}$



2. Se utiliza el modelo,  $f[x, \Phi]$ , para predecir los parámetros,  $\theta$ , de la distribución de probabilidad.

## Criterio de máxima verosimilitud

$$D = \{x_i, y_i\}_{i=1}^N$$

## Criterio de máxima verosimilitud

$$D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N \rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{x}_i) \right]$$

## Criterio de máxima verosimilitud

$$D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N \rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{x}_i) \right]$$

$$= \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \theta_i) \right]$$

Cuando consideramos la probabilidad como una función de los parámetros  $\theta$ , se denomina **función de verosimilitud** (likelihood).

## Criterio de máxima verosimilitud

$$D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N \rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{x}_i) \right]$$

$$= \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \theta_i) \right]$$

Con la distribución normal tendríamos:  $\theta_i = \{\mu_i, \sigma_i^2\}$

Cuando consideramos la probabilidad como una función de los parámetros  $\theta$ , se denomina **función de verosimilitud** (likelihood).

## Criterio de máxima verosimilitud

$$D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N \rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{x}_i) \right]$$

$$= \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \theta_i) \right]$$

$$= \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | f[\mathbf{x}_i, \phi]) \right]$$

Nuestro modelo estima los parámetros de la distribución,  $\theta$ , para cada posible valor de entrada,  $\mathbf{x}$



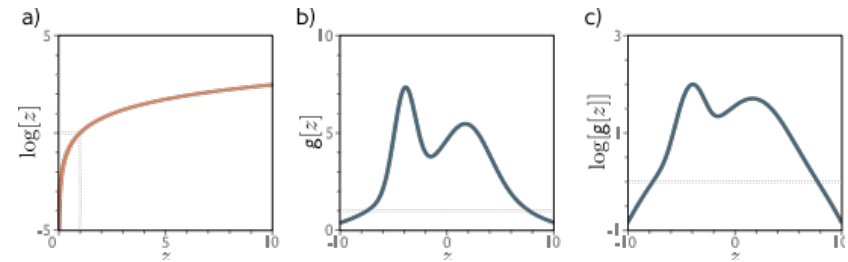
## Criterio de máxima verosimilitud

- Problema:

$$\hat{\phi} = \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right]$$

- Los términos en este producto pueden ser muy pequeños
- El producto puede ser tan pequeño que tengamos problemas de representación numérica.

## La función logaritmo es monótona



- El máximo del logaritmo de una función está en el mismo sitio que el de la función

## Criterio de máxima verosimilitud

$$D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N \rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right]$$

## Criterio de máxima verosimilitud

$$D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N \rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right]$$
$$= \operatorname{argmax}_{\phi} \left[ \log \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right] \right]$$

## Criterio de máxima verosimilitud

$$\begin{aligned}
 D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N &\rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right] \\
 &= \operatorname{argmax}_{\phi} \left[ \log \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right] \right] \\
 &\quad \uparrow \\
 &\quad \log[a \cdot b] = \log[a] + \log[b]
 \end{aligned}$$

## Criterio de máxima verosimilitud

$$\begin{aligned}
 D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N &\rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right] \\
 &= \operatorname{argmax}_{\phi} \left[ \log \left[ \prod_{i=1}^N Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right] \right] \\
 &= \operatorname{argmax}_{\phi} \left[ \sum_{i=1}^N \log [Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi])] \right]
 \end{aligned}$$

## Minimizar el log. de la verosimilitud

$$D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N \rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \sum_{i=1}^N \log [Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi])] \right]$$

## Minimizar el log. de la verosimilitud

$$\begin{aligned}
 D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N &\rightarrow \hat{\phi} = \operatorname{argmax}_{\phi} \left[ \sum_{i=1}^N \log [Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi])] \right] \\
 &= \operatorname{argmin}_{\phi} \left[ - \sum_{i=1}^N \log [Pr(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi])] \right]
 \end{aligned}$$

Cambios necesarios para minimizar la función de coste  
(en lugar de maximizar)

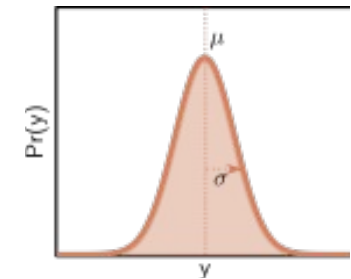
## Minimizar el log. de la verosimilitud

$$\boxed{D = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N} \rightarrow \hat{\phi} = \underset{\phi}{\operatorname{argmax}} \left[ \sum_{i=1}^N \log \left[ \operatorname{Pr}(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right] \right]$$
$$= \underset{\phi}{\operatorname{argmin}} \left[ \underbrace{- \sum_{i=1}^N \log \left[ \operatorname{Pr}(\mathbf{y}_i | \mathbf{f}[\mathbf{x}_i, \phi]) \right]}_{J(\phi)} \right]$$

## Inferencia (estimación del modelo)

- Ahora predecimos una distribución de probabilidad (sus parámetros  $\theta$ )
- ¡Necesitamos la predicción puntual!
- Encontrar el valor de máxima probabilidad (p.ej. La media en una normal)

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} [\operatorname{Pr}(\mathbf{y} | \mathbf{f}[\mathbf{x}, \phi])]$$



### 2.2.1 Funciones de pérdida por máxima verosimilitud

- Máxima verosimilitud (likelihood)
- Receta para funciones de pérdida
- Ejemplo 1: regresión univariante
- Ejemplo 2: clasificación binaria
- Ejemplo 3: clasificación multiclase
- Otros tipos de datos
- Múltiples salidas

## Receta para funciones de pérdida

1. Elegir la distribución de probabilidad  $\operatorname{Pr}(\mathbf{y} | \theta)$ , definida sobre el dominio de las predicciones  $\mathbf{y}$ , con parámetros  $\theta$

## Receta para funciones de pérdida

1. Elegir la distribución de probabilidad  $\Pr(\mathbf{y} | \boldsymbol{\theta})$ , definida sobre el dominio de las predicciones  $\mathbf{y}$ , con parámetros  $\boldsymbol{\theta}$
2. Definir el modelo  $f[\mathbf{x}, \boldsymbol{\Phi}]$  para predecir uno o más parámetros, tal que  $\boldsymbol{\theta} = f[\mathbf{x}, \boldsymbol{\Phi}]$  y  $\Pr(\mathbf{y} | \boldsymbol{\theta}) = \Pr(\mathbf{y} | f[\mathbf{x}, \boldsymbol{\Phi}])$

## Receta para funciones de pérdida

1. Elegir la distribución de probabilidad  $\Pr(\mathbf{y} | \boldsymbol{\theta})$ , definida sobre el dominio de las predicciones  $\mathbf{y}$ , con parámetros  $\boldsymbol{\theta}$
2. Definir el modelo  $f[\mathbf{x}, \boldsymbol{\Phi}]$  para predecir uno o más parámetros, tal que  $\boldsymbol{\theta} = f[\mathbf{x}, \boldsymbol{\Phi}]$  y  $\Pr(\mathbf{y} | \boldsymbol{\theta}) = \Pr(\mathbf{y} | f[\mathbf{x}, \boldsymbol{\Phi}])$
3. Entrenar el modelo encontrando los parámetros  $\hat{\boldsymbol{\phi}}$ , que minimizan el logaritmo de la verosimilitud en D:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} [J(\boldsymbol{\Phi})] = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log \left[ \Pr(\mathbf{y}_i | f[\mathbf{x}_i, \boldsymbol{\phi}]) \right] \right]$$

## Receta para funciones de pérdida

1. Elegir la distribución de probabilidad  $\Pr(\mathbf{y} | \boldsymbol{\theta})$ , definida sobre el dominio de las predicciones  $\mathbf{y}$ , con parámetros  $\boldsymbol{\theta}$
2. Definir el modelo  $f[\mathbf{x}, \boldsymbol{\Phi}]$  para predecir uno o más parámetros, tal que  $\boldsymbol{\theta} = f[\mathbf{x}, \boldsymbol{\Phi}]$  y  $\Pr(\mathbf{y} | \boldsymbol{\theta}) = \Pr(\mathbf{y} | f[\mathbf{x}, \boldsymbol{\Phi}])$
3. Entrenar el modelo encontrando los parámetros  $\hat{\boldsymbol{\phi}}$ , que minimizan el logaritmo de la verosimilitud en D:

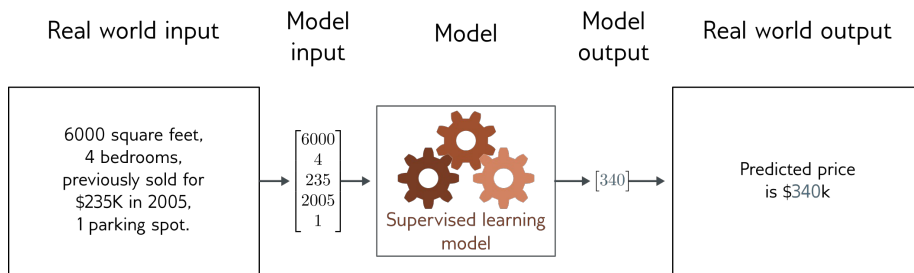
$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} [J(\boldsymbol{\Phi})] = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log \left[ \Pr(\mathbf{y}_i | f[\mathbf{x}_i, \boldsymbol{\phi}]) \right] \right]$$

4. La inferencia en un ejemplo nuevo de test,  $\mathbf{x}$ , devuelve la distribución completa  $\Pr(\mathbf{y} | f[\mathbf{x}, \hat{\boldsymbol{\phi}}])$  o el máximo de esta distribución.

### 2.2.1 Funciones de pérdida por máxima verosimilitud

- Máxima verosimilitud (likelihood)
- Receta para funciones de pérdida
- Ejemplo 1: regresión univariante
- Ejemplo 2: clasificación binaria
- Ejemplo 3: clasificación multiclase
- Otros tipos de datos
- Múltiples salidas

## Ejemplo 1: Regresión univariante

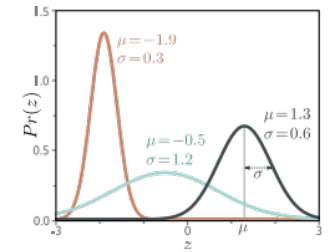


## Ejemplo 1: Regresión univariante

1. Elegir la distribución de probabilidad  $\Pr(\mathbf{y} | \boldsymbol{\theta})$ , definida sobre el dominio de las predicciones  $\mathbf{y}$ , con parámetros  $\boldsymbol{\theta}$

- Predecir la salida (un escalar):  $y \in \mathbb{R}$
- Distribución de probabilidad razonable:
  - Distribución normal:

$$\Pr(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y - \mu)^2}{2\sigma^2} \right]$$



## Ejemplo 1: Regresión univariante

2. Definir el modelo  $f[\mathbf{x}, \boldsymbol{\Phi}]$  para predecir uno o más parámetros, tal que  $\boldsymbol{\theta} = f[\mathbf{x}, \boldsymbol{\Phi}]$  y  $\Pr(\mathbf{y} | \boldsymbol{\theta}) = \Pr(\mathbf{y} | f[\mathbf{x}, \boldsymbol{\Phi}])$

$$\Pr(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y - \mu)^2}{2\sigma^2} \right]$$

$$\Pr(y|f[\mathbf{x}, \boldsymbol{\Phi}], \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y - f[\mathbf{x}, \boldsymbol{\Phi}])^2}{2\sigma^2} \right]$$

## Ejemplo 1: Regresión univariante

3. Entrenar el modelo encontrando los parámetros,  $\boldsymbol{\Phi}$ , que minimizan el logaritmo de la verosimilitud en D:

$$\begin{aligned} J(\boldsymbol{\Phi}) &= - \sum_{i=1}^N \log [\Pr(y_i | f[\mathbf{x}_i, \boldsymbol{\Phi}], \sigma^2)] \\ &= - \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_i - f[\mathbf{x}_i, \boldsymbol{\Phi}])^2}{2\sigma^2} \right] \right] \end{aligned}$$

## Ejemplo 1: Regresión univariante

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \right] \right]$$

## Ejemplo 1: Regresión univariante

$$\begin{aligned} \hat{\phi} &= \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \right] \right] \\ &= \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \right] - \frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \end{aligned}$$

## Ejemplo 1: Regresión univariante

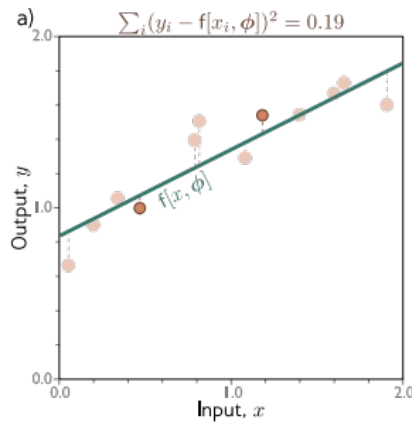
$$\begin{aligned} \hat{\phi} &= \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \right] \right] \\ &= \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \right] - \frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \\ &= \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N - \frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \end{aligned}$$

## Ejemplo 1: Regresión univariante

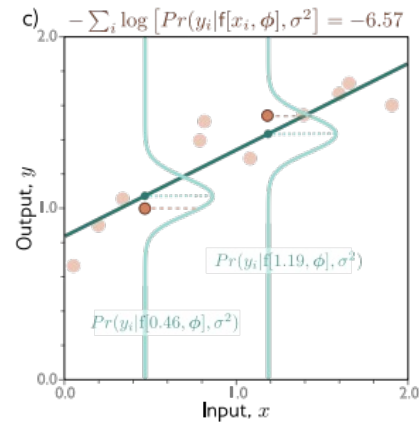
$$\begin{aligned} \hat{\phi} &= \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \right] \right] \\ &= \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \right] - \frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \\ &= \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N - \frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \\ &= \underset{\phi}{\operatorname{argmin}} \left[ \sum_{i=1}^N (y_i - f[\mathbf{x}_i, \phi])^2 \right] \end{aligned}$$

¡Mínimos cuadrados!  
(least squares)

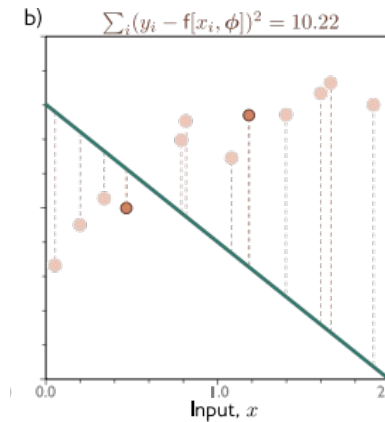
## Ejemplo 1: Regresión univariante



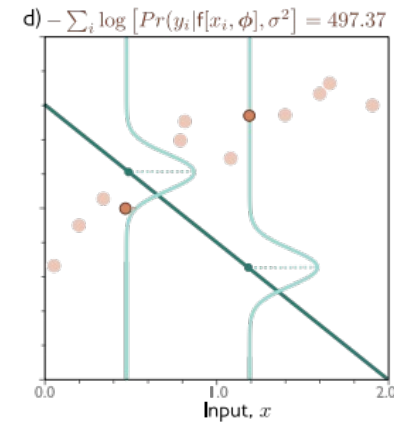
Mínimos cuadrados



Máxima verosimilitud



Mínimos cuadrados



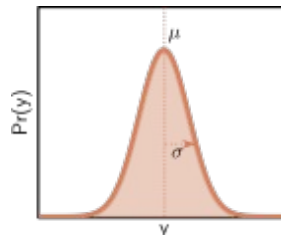
Máxima verosimilitud

## Ejemplo 1: Regresión univariante

4. La inferencia en un ejemplo nuevo de test,  $\mathbf{x}$ , devuelve la distribución completa  $Pr(\mathbf{y} | f[\mathbf{x}, \hat{\phi}])$  o el máximo de esta distribución.

$$Pr(y | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y - \mu)^2}{2\sigma^2} \right]$$

$$Pr(y | f[\mathbf{x}, \phi], \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y - f[\mathbf{x}, \phi])^2}{2\sigma^2} \right]$$



## Estimación de la varianza

- La varianza desaparece de la optimización final:

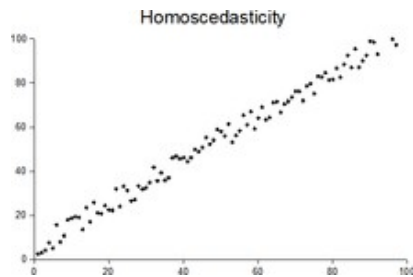
$$\begin{aligned} \hat{\phi} &= \underset{\phi}{\operatorname{argmin}} \left[ -\sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \right] \right] \\ &= \underset{\phi}{\operatorname{argmin}} \left[ \sum_{i=1}^N (y_i - f[\mathbf{x}_i, \phi])^2 \right] \end{aligned}$$

- Pero podríamos aprenderla:

$$\hat{\phi}, \hat{\sigma}^2 = \underset{\phi, \sigma^2}{\operatorname{argmin}} \left[ -\sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_i - f[\mathbf{x}_i, \phi])^2}{2\sigma^2} \right] \right] \right]$$

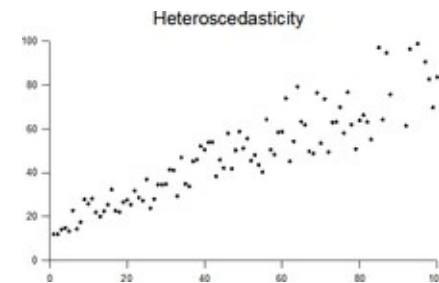
## Regresión heterocedástica

- Hasta ahora hemos asumido homocedasticidad (misma varianza para todo valor de  $\mathbf{x}$ )



## Regresión heterocedástica

- Hasta ahora hemos asumido homocedasticidad (misma varianza para todo valor de  $\mathbf{x}$ )
- Podríamos tener un problema donde el ruido dependa del valor de  $\mathbf{x}$



## Regresión heterocedástica

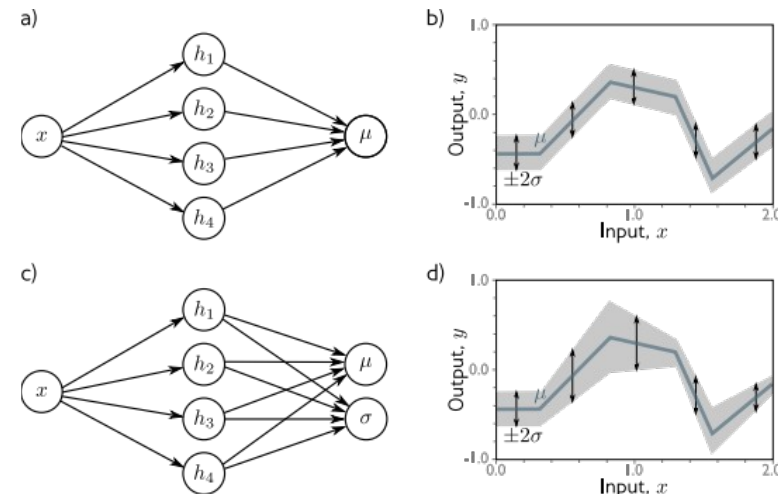
- Hasta ahora hemos asumido homocedasticidad (misma varianza para todo valor de  $\mathbf{x}$ )
- Podríamos tener un problema donde el ruido dependa del valor de  $\mathbf{x}$
- Construiríamos un modelo con dos salidas:

$$\mu = f_1[\mathbf{x}, \phi]$$

$$\sigma^2 = f_2[\mathbf{x}, \phi]^2$$

$$\hat{\phi} = \operatorname{argmin}_{\phi} \left[ -\sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi f_2[\mathbf{x}_i, \phi]^2}} \right] - \frac{(y_i - f_1[\mathbf{x}_i, \phi])^2}{2f_2[\mathbf{x}_i, \phi]^2} \right]$$

## Regresión heterocedástica

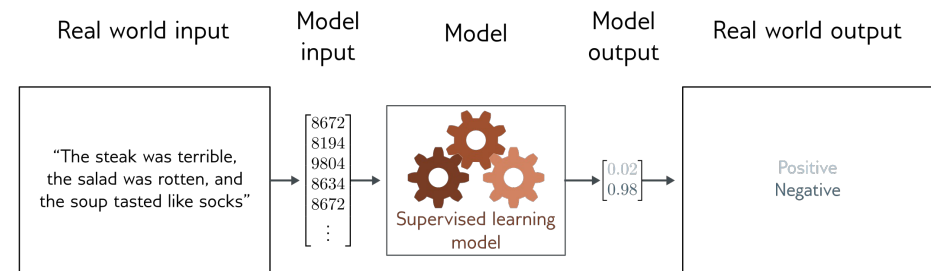




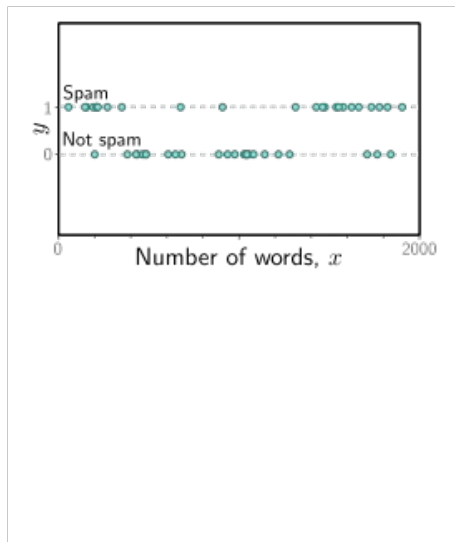
### 2.2.1 Funciones de pérdida por máxima verosimilitud

- Máxima verosimilitud (likelihood)
- Receta para funciones de pérdida
- Ejemplo 1: regresión univariante
- **Ejemplo 2: clasificación binaria**
- Ejemplo 3: clasificación multiclase
- Otros tipos de datos
- Múltiples salidas

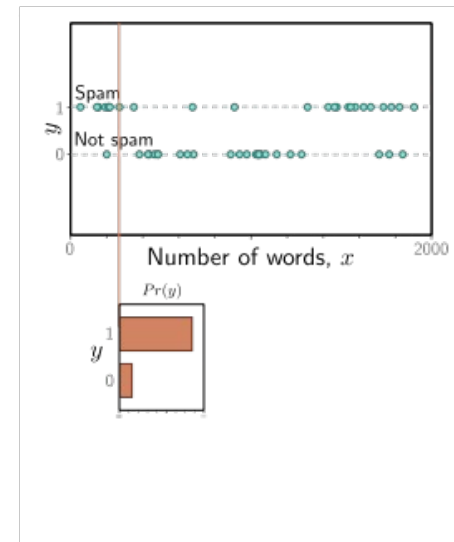
### Ejemplo 2: Clasificación binaria



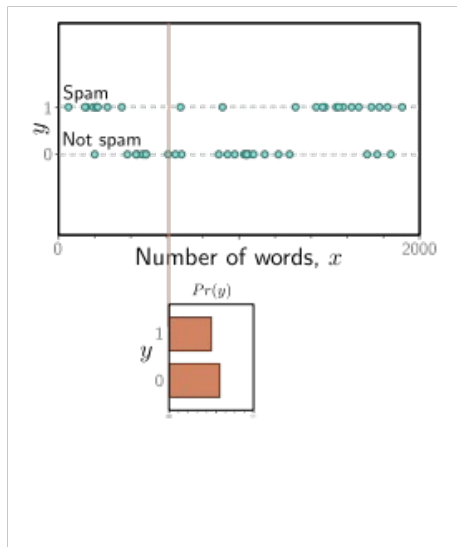
### Ejemplo 2: Clasificación binaria



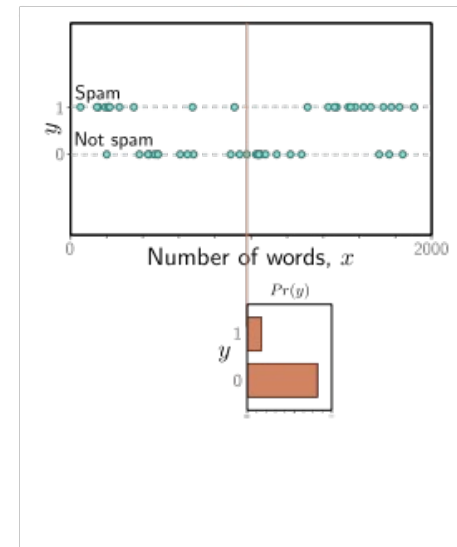
### Ejemplo 2: Clasificación binaria



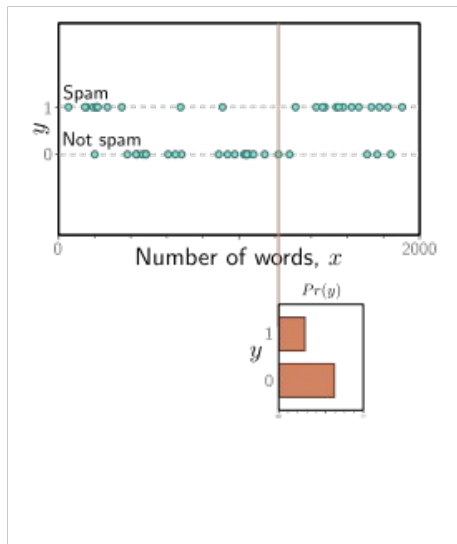
## Ejemplo 2: Clasificación binaria



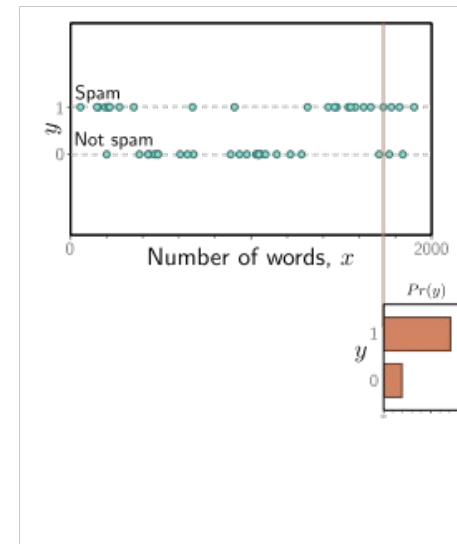
## Ejemplo 2: Clasificación binaria



## Ejemplo 2: Clasificación binaria



## Ejemplo 2: Clasificación binaria



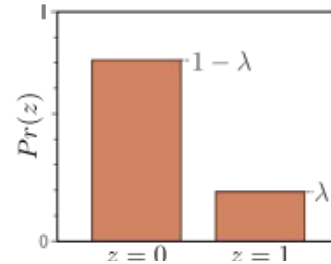
## Ejemplo 2: Clasificación binaria

1. Elegir la distribución de probabilidad  $\Pr(y | \theta)$ , definida sobre el dominio de las predicciones  $y$ , con parámetros  $\theta$

- Dominio de salida:  $y \in \{0, 1\}$
- Distribución de Bernoulli
- Un parámetro en  $[0, 1]$

$$\Pr(y|\lambda) = \begin{cases} 1 - \lambda & y = 0 \\ \lambda & y = 1 \end{cases}$$

$$\Pr(y|\lambda) = (1 - \lambda)^{1-y} \cdot \lambda^y$$



## Ejemplo 2: Clasificación binaria

2. Definir el modelo  $f[x, \Phi]$  para predecir uno o más parámetros, tal que  $\theta = f[x, \Phi]$  y  $\Pr(y | \theta) = \Pr(y | f[x, \Phi])$

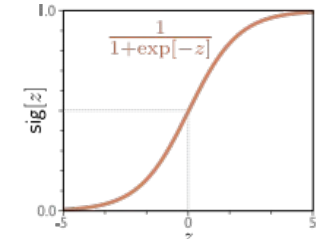
Problema:

- $\theta = f[x, \Phi]$  puede tomar cualquier valor.
- El parámetro de la Bernoulli debe estar en  $[0, 1]$

Solución:

- Una función que pase el valor al intervalo  $[0, 1]$ :

$$\text{sig}[z] = \frac{1}{1 + \exp[-z]}$$



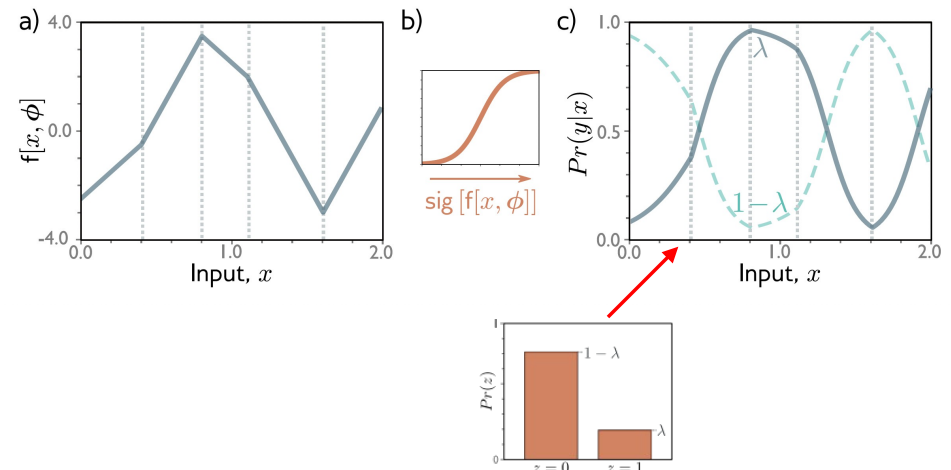
## Ejemplo 2: Clasificación binaria

2. Definir el modelo  $f[x, \Phi]$  para predecir uno o más parámetros, tal que  $\theta = f[x, \Phi]$  y  $\Pr(y | \theta) = \Pr(y | f[x, \Phi])$

$$\Pr(y|\lambda) = (1 - \lambda)^{1-y} \cdot \lambda^y$$

$$\Pr(y|x) = (1 - \text{sig}[f[x, \Phi]])^{1-y} \cdot \text{sig}[f[x, \Phi]]^y$$

## Ejemplo 2: Clasificación binaria



## Ejemplo 2: Clasificación binaria

3. Entrenar el modelo encontrando los parámetros,  $\phi$ , que minimizan el logaritmo de la verosimilitud en D:

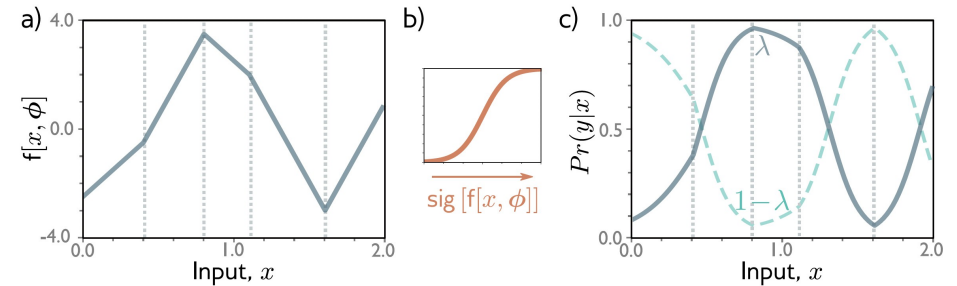
$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} [J(\Phi)] = \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log [Pr(y_i | f[x_i, \phi])] \right]$$

$$Pr(y|x) = (1 - \operatorname{sig}[f[x|\phi]])^{1-y} \cdot \operatorname{sig}[f[x|\phi]]^y$$

$$J(\Phi) = \sum_{i=1}^N -(1 - y_i) \log [1 - \operatorname{sig}[f[x_i|\phi]]] - y_i \log [\operatorname{sig}[f[x_i|\phi]]]$$

## Ejemplo 2: Clasificación binaria

4. La inferencia en un ejemplo nuevo de test,  $x$ , devuelve la distribución completa  $Pr(y|x, \hat{\phi})$  o el máximo de esta distribución.

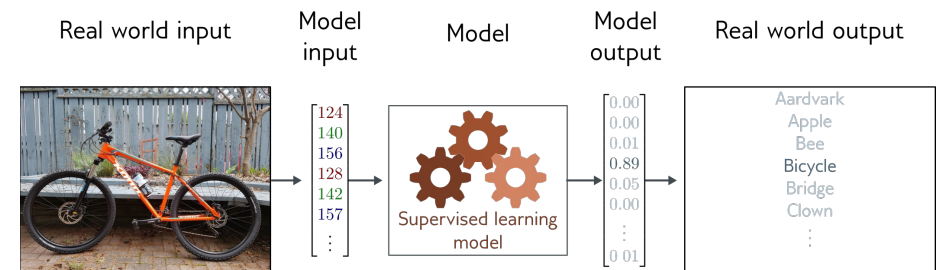


Elegir  $y=1$  donde  $Pr(y|x) > 0.5$ , y en otro caso 0

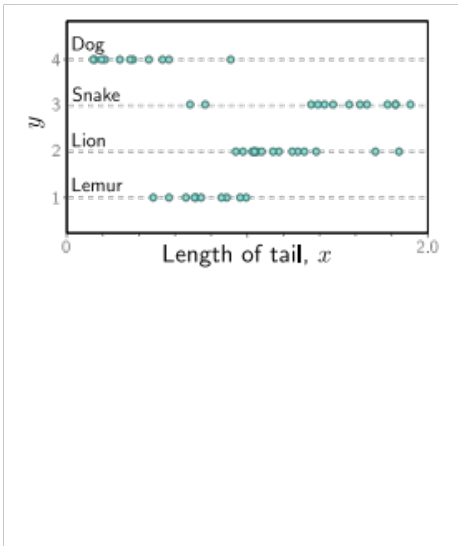
### 2.2.1 Funciones de pérdida por máxima verosimilitud

- Máxima verosimilitud (likelihood)
- Receta para funciones de pérdida
- Ejemplo 1: regresión univariante
- Ejemplo 2: clasificación binaria
- **Ejemplo 3: clasificación multiclase**
- Otros tipos de datos
- Múltiples salidas

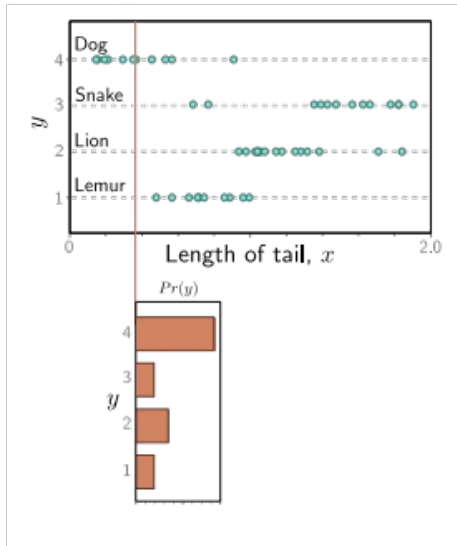
## Ejemplo 3: Clasificación multiclase



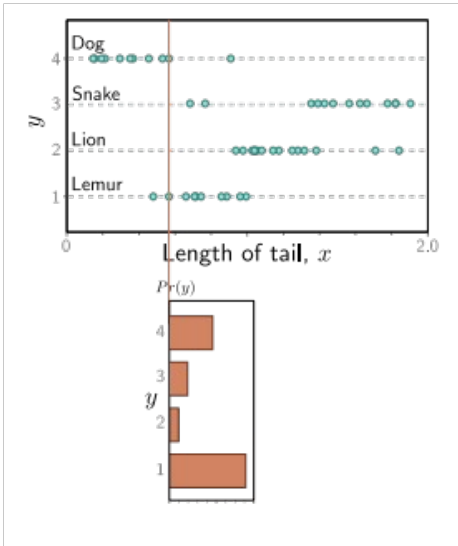
Ejemplo 3: Clasificación multiclase



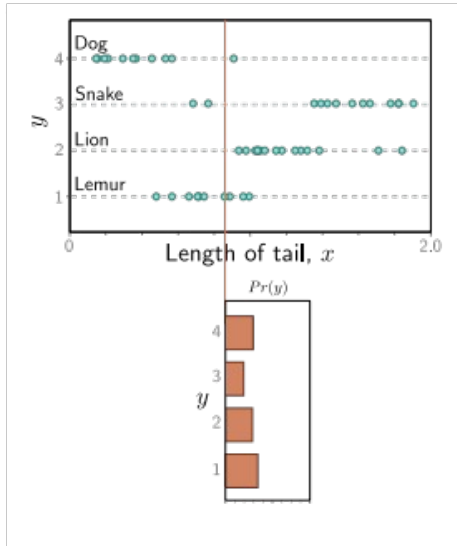
Ejemplo 3: Clasificación multiclase



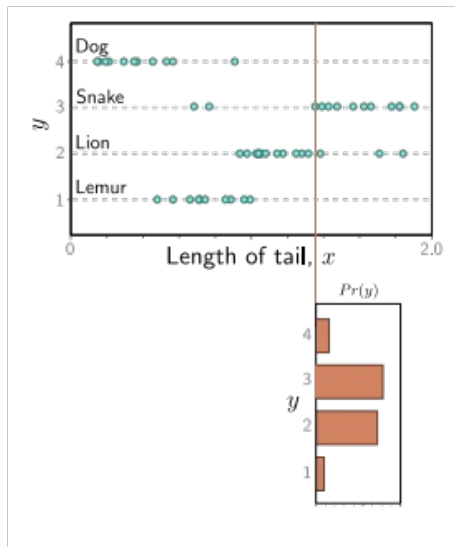
Ejemplo 3: Clasificación multiclase



Ejemplo 3: Clasificación multiclase



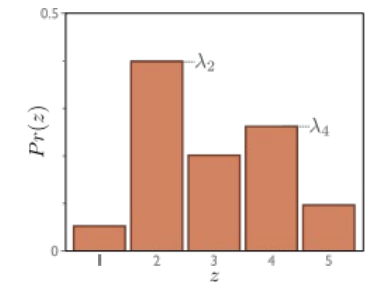
## Ejemplo 3: Clasificación multiclase



## Ejemplo 3: Clasificación multiclase

1. Elegir la distribución de probabilidad  $Pr(y | \theta)$ , definida sobre el dominio de las predicciones  $y$ , con parámetros  $\theta$

- Dominio de salida:  $y \in \{1, 2, \dots, K\}$
- Distribución categórica (Bernoulli generalizada)
- $K$  parámetros en  $[0, 1]$
- Los parámetros suman 1



$$Pr(y = k) = \lambda_k$$

## Ejemplo 3: Clasificación multiclase

2. Definir el modelo  $f[x, \Phi]$  para predecir uno o más parámetros, tal que  $\theta = f[x, \Phi]$  y  $Pr(y | \theta) = Pr(y | f[x, \Phi])$

### Problema:

- $\theta = f[x, \Phi]$  puede tomar cualquier valor.
- Parámetros en  $[0, 1]$  y que sumen 1

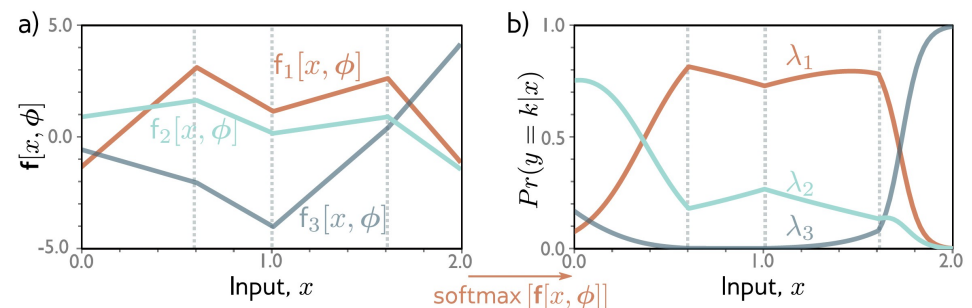
### Solución:

- Una función que pase el valor al intervalo  $[0, 1]$  y los haga sumar 1:

$$\text{softmax}_k[\mathbf{z}] = \frac{\exp[z_k]}{\sum_{k'=1}^K \exp[z_{k'}]}$$

$$Pr(y = k | \mathbf{x}) = \text{softmax}_k[\mathbf{f}[\mathbf{x}, \Phi]]$$

## Ejemplo 3: Clasificación multiclase



$$Pr(y = k | \mathbf{x}) = \text{softmax}_k[\mathbf{f}[\mathbf{x}, \Phi]]$$

## Ejemplo 3: Clasificación multiclase

3. Entrenar el modelo encontrando los parámetros,  $\phi$ , que minimizan el logaritmo de la verosimilitud en D:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} [J(\Phi)] = \underset{\phi}{\operatorname{argmin}} \left[ - \sum_{i=1}^N \log [Pr(y_i | f[x_i, \phi])] \right]$$

$$J(\Phi) = - \sum_{i=1}^N \log [\operatorname{softmax}_{y_i} [f[x_i, \phi]]]$$

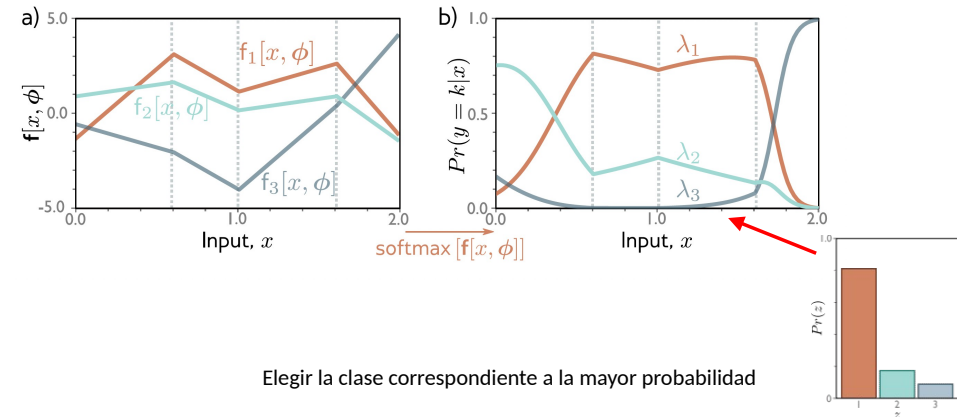
$$\operatorname{softmax}_k [z] = \frac{\exp[z_k]}{\sum_{k'=1}^K \exp[z_{k'}]}$$

$$= - \sum_{i=1}^N f_{y_i} [x_i, \phi] - \log \left[ \sum_{k=1}^K \exp [f_k [x_i, \phi]] \right]$$

“Multiclass cross-entropy loss”

## Ejemplo 3: Clasificación multiclase

4. La inferencia en un ejemplo nuevo de test,  $x$ , devuelve la distribución completa  $Pr(y | f[x, \hat{\phi}])$  o el máximo de esta distribución.



### 2.2.1 Funciones de pérdida por máxima verosimilitud

- Máxima verosimilitud (likelihood)
- Receta para funciones de pérdida
- Ejemplo 1: regresión univariante
- Ejemplo 2: clasificación binaria
- Ejemplo 3: clasificación multiclase
- Otros tipos de datos
- Múltiples salidas

## Otros tipos de datos

Data Type	Domain	Distribution	Use
univariate, continuous, unbounded	$y \in \mathbb{R}$	univariate normal	regression
univariate, continuous, unbounded	$y \in \mathbb{R}$	Laplace or t-distribution	robust regression
univariate, continuous, unbounded	$y \in \mathbb{R}$	mixture of Gaussians	multimodal regression
univariate, continuous, bounded below	$y \in \mathbb{R}^+$	exponential or gamma	predicting magnitude
univariate, continuous, bounded	$y \in [0, 1]$	beta	predicting proportions
multivariate, continuous, unbounded	$\mathbf{y} \in \mathbb{R}^K$	multivariate normal	multivariate regression
symmetric positive definite matrix	$\mathbf{Y} \in \mathbb{R}^{K \times K}$ $\mathbf{z}^T \mathbf{Y} \mathbf{z} > 0 \quad \forall \mathbf{z} \in \mathbb{R}^K$	Wishart	predicting covariances
univariate, continuous, circular	$y \in (-\pi, \pi]$	von Mises	predicting direction
univariate, discrete, binary	$y \in \{0, 1\}$	Bernoulli	binary classification
univariate, discrete, bounded	$y \in \{1, 2, \dots, K\}$	categorical	multiclass classification
univariate, discrete, bounded below	$y \in [0, 1, 2, 3, \dots]$	Poisson	predicting event counts
multivariate, discrete, permutation	$\mathbf{y} \in \operatorname{Perm}[1, 2, \dots, K]$	Plackett-Luce	ranking

Figure 5.10 Distributions for loss functions for different prediction types.

En estos casos la aproximación de máxima verosimilitud permite diseñar la función de pérdida.

### 2.2.1 Funciones de pérdida por máxima verosimilitud

- Máxima verosimilitud (likelihood)
- Receta para funciones de pérdida
- Ejemplo 1: regresión univariante
- Ejemplo 2: clasificación binaria
- Ejemplo 3: clasificación multiclase
- Otros tipos de datos
- **Múltiples salidas**

## Múltiples salidas

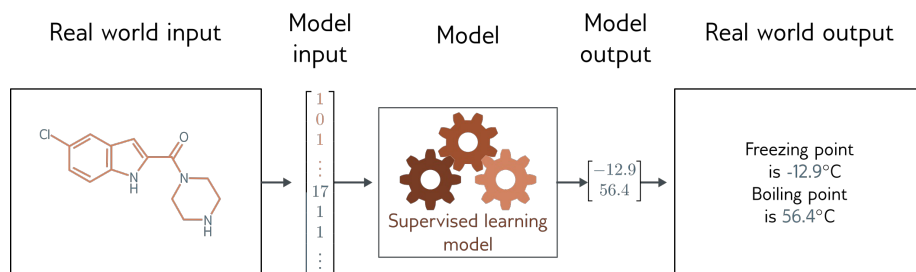
- Suponer todas las salidas,  $y_d$ , independientes:

$$Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}_i, \phi]) = \prod_d Pr(y_d|\mathbf{f}_d[\mathbf{x}_i, \phi])$$

- El logaritmo de la verosimilitud es una suma:

$$\mathbf{J}(\Phi) = -\sum_{i=1}^N \log [Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}_i, \phi])] = -\sum_{i=1}^N \sum_d \log [Pr(y_{id}|\mathbf{f}_d[\mathbf{x}_i, \phi])]$$

### Ejemplo 4: Regresión multivariante



### Ejemplo 4: Regresión multivariante

- **Objetivo:** predecir una salida multivariante  $\mathbf{y} \in \mathbb{R}^{D_o}$
- Suponer todas las salidas,  $y_d$ , independientes:

$$\begin{aligned} Pr(\mathbf{y}|\boldsymbol{\mu}, \sigma^2) &= \prod_{d=1}^{D_o} Pr(y_d|\mu_d, \sigma^2) \\ &= \prod_{d=1}^{D_o} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_d - \mu_d)^2}{2\sigma^2} \right] \end{aligned}$$

- El modelo predice la media de diferentes dist. normales:

$$Pr(\mathbf{y}|\mathbf{f}[\mathbf{x}, \phi], \sigma^2) = \prod_{d=1}^{D_o} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_d - \mathbf{f}_d[\mathbf{x}, \phi])^2}{2\sigma^2} \right]$$



## 2.2 Funciones de pérdida

2.2.1 Funciones de pérdida por Máxima verosimilitud

2.2.2 Funciones de pérdida basadas en el margen

2.2.3 Funciones de pérdida para regresión

## Modelo lineal

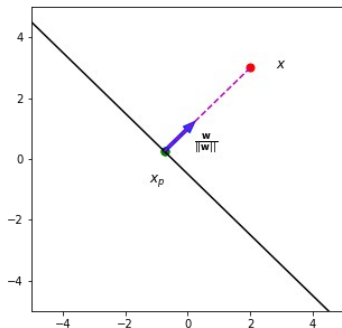
- Modelo lineal en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$

## Distancia a un hiperplano

- Modelo lineal en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$

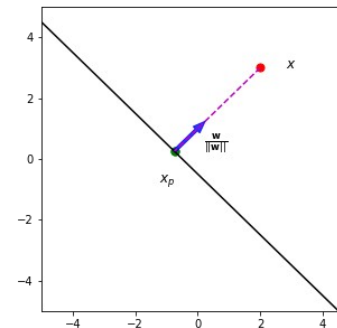


- $f(\mathbf{x}; \Phi) = 0$  define los puntos de la recta (hiperplano en más dimensiones), H

## Distancia a un hiperplano

- Modelo lineal en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$

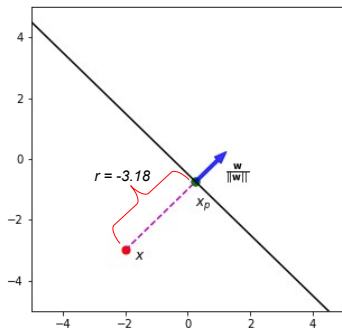


- $f(\mathbf{x}; \Phi) = 0$  define los puntos de la recta (hiperplano en más dimensiones), H
- El vector  $\mathbf{w}$  es normal a H

## Distancia a un hiperplano

- Modelo lineal en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$



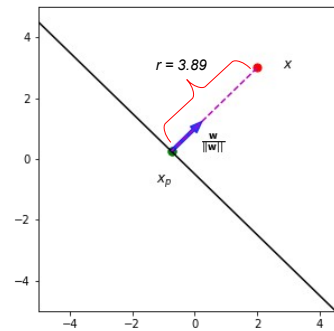
- $f(\mathbf{x}; \Phi) = 0$  define los puntos de la recta (hiperplano en más dimensiones), H
- El vector  $\mathbf{w}$  es normal a H
- Si  $\mathbf{x}_p$  es la proyección de  $\mathbf{x}$  en H

$$\mathbf{x} = \mathbf{x}_p + r \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

## Distancia a un hiperplano

- Modelo lineal en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$



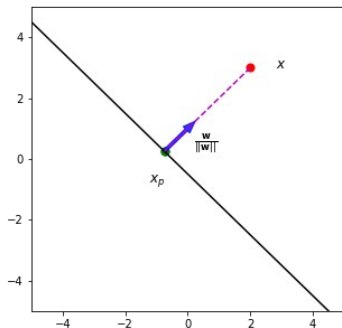
- $f(\mathbf{x}; \Phi) = 0$  define los puntos de la recta (hiperplano en más dimensiones), H
- El vector  $\mathbf{w}$  es normal a H
- Si  $\mathbf{x}_p$  es la proyección de  $\mathbf{x}$  en H

$$\mathbf{x} = \mathbf{x}_p + r \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

## Distancia a un hiperplano

- Modelo lineal en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$



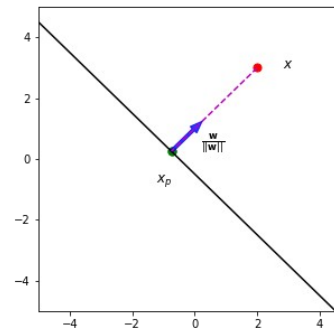
- $f(\mathbf{x}; \Phi) = 0$  define los puntos de la recta (hiperplano en más dimensiones), H
- El vector  $\mathbf{w}$  es normal a H
- Si  $\mathbf{x}_p$  es la proyección de  $\mathbf{x}$  en H

$$f(\mathbf{x}) = f\left(\mathbf{x}_p + r \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}\right)$$

## Distancia a un hiperplano

- Modelo lineal en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$



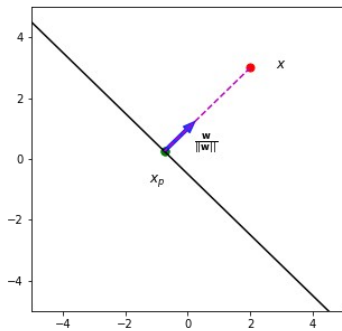
- $f(\mathbf{x}; \Phi) = 0$  define los puntos de la recta (hiperplano en más dimensiones), H
- El vector  $\mathbf{w}$  es normal a H
- Si  $\mathbf{x}_p$  es la proyección de  $\mathbf{x}$  en H

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}_p + r \cdot \frac{\mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|} + \Phi_0$$

## Distancia a un hiperplano

- Modelo lineal en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$



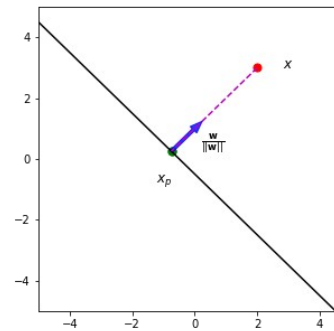
- $f(\mathbf{x}; \Phi) = 0$  define los puntos de la recta (hiperplano en más dimensiones), H
- El vector  $\mathbf{w}$  es normal a H
- Si  $\mathbf{x}_p$  es la proyección de  $\mathbf{x}$  en H

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}_p + r \cdot \frac{\|\mathbf{w}\|^2}{\|\mathbf{w}\|} + \Phi_0$$

## Distancia a un hiperplano

- Modelo lineal en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$



- $f(\mathbf{x}; \Phi) = 0$  define los puntos de la recta (hiperplano en más dimensiones), H
- El vector  $\mathbf{w}$  es normal a H
- Si  $\mathbf{x}_p$  es la proyección de  $\mathbf{x}$  en H

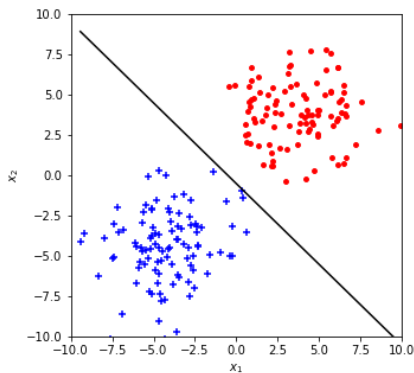
$$f(\mathbf{x}) = r \cdot \|\mathbf{w}\|$$

$f(\mathbf{x}; \Phi)$  es proporcional a  $r$

## Modelo lineal y definición de margen

- Volvamos al modelo lineal como clasificador en 2D:

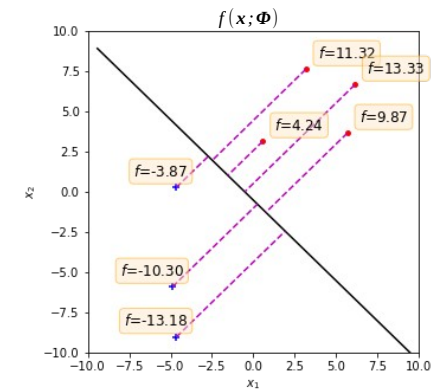
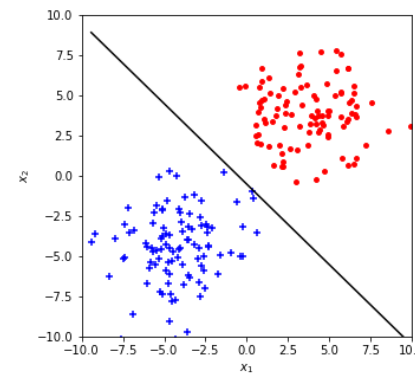
$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$



## Modelo lineal y definición de margen

- Volvamos al modelo lineal como clasificador en 2D:

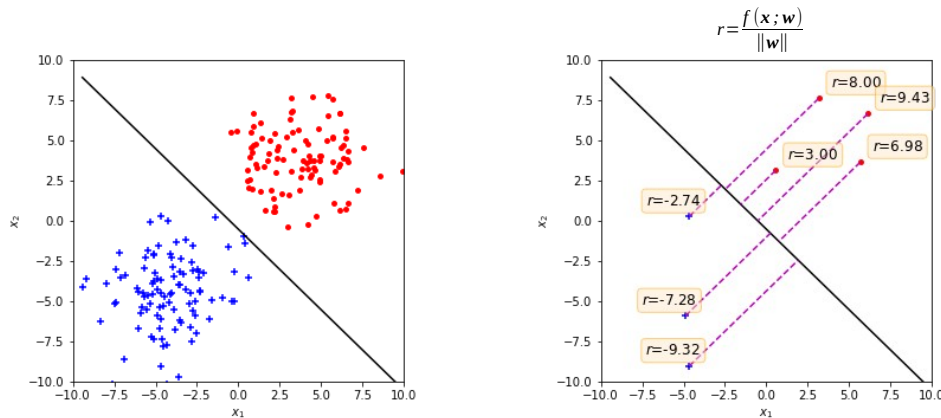
$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$



## Modelo lineal y definición de margen

- Volvamos al modelo lineal como clasificador en 2D:

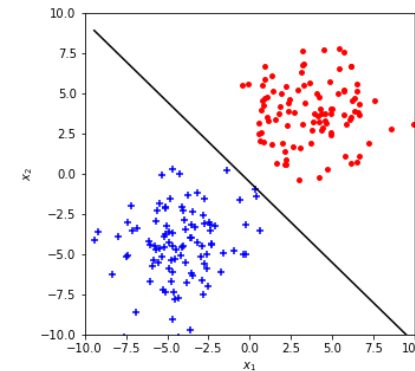
$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$



## Modelo lineal y definición de margen

- Volvamos al modelo lineal como clasificador en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$

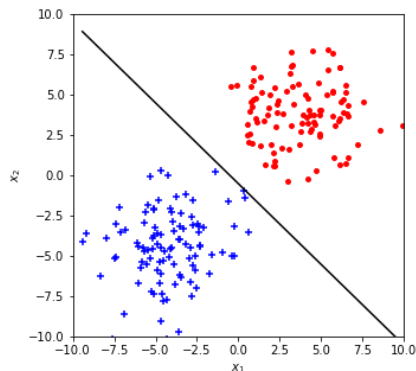


- Problema de clasificación con etiquetas y en  $\{1, -1\}$ .
- Clasificador lineal:  
 $G(\mathbf{x}_i) = \text{signo}(f(\mathbf{x}_i))$
- Entonces para un ejemplo  $\mathbf{x}_i$  el "margen" (*margin* en inglés):  
 $y_i \cdot f(\mathbf{x}_i) > 0$  si acierto  
 $y_i \cdot f(\mathbf{x}_i) < 0$  si fallo

## Uso del margen en funciones de pérdida

- Volvamos al modelo lineal como clasificador en 2D:

$$f(x; \Phi) = \Phi_0 + \Phi_1 x_1 + \Phi_2 x_2 = \Phi_0 + \mathbf{w}^T \mathbf{x}$$

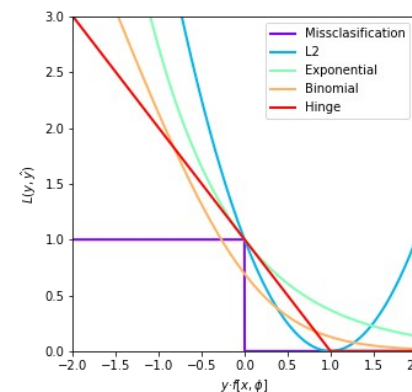


- Existen funciones de pérdida definidas sobre el margen:

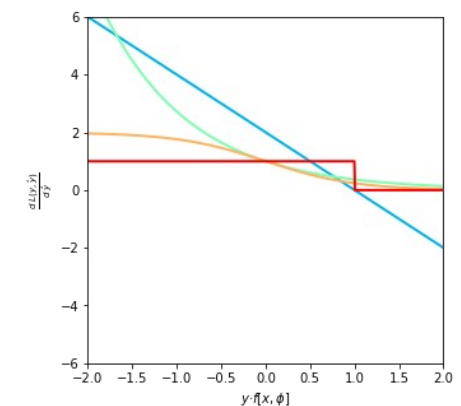
$$L(y_i, \underbrace{f(\mathbf{x}_i; \Phi)}_{\hat{y}_i}) = L(y_i \cdot \hat{y}_i)$$

## Funciones de pérdida para clasificación

- Las más habituales basadas en el margen:

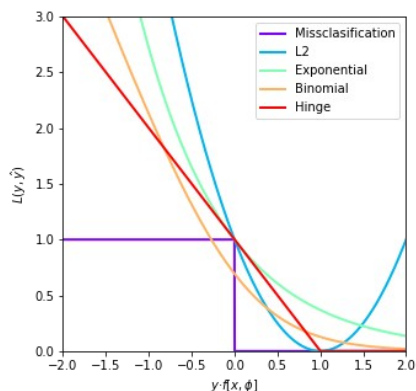


Gradiente de L



## Funciones de pérdida para clasificación

- Las más habituales basadas en el margen:



$$L_{0-1}(y_i, f(x_i; \Phi)) = I(y_i \cdot f(x_i; \Phi) < 0)$$

$$L_{L2}(y_i, f(x_i; \Phi)) = (y_i - f(x_i; \Phi))^2$$

$$L_{\text{exp}}(y_i, f(x_i; \Phi)) = e^{-y_i \cdot f(x_i; \Phi)}$$

$$L_{\text{Bin}}(y_i, f(x_i; \Phi)) = \log(1 + e^{-2 \cdot y_i \cdot f(x_i; \Phi)})$$

$$L_{\text{Hinge}}(y_i, f(x_i; \Phi)) = \max(0, 1 - y_i \cdot f(x_i; \Phi))$$

## 2.2 Funciones de pérdida

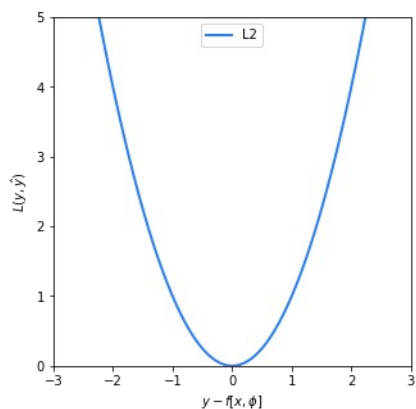
2.2.1 Funciones de pérdida por Máxima verosimilitud

2.2.2 Funciones de pérdida basadas en el margen

2.2.3 Funciones de pérdida para regresión

## Funciones de pérdida para regresión

- Las más habituales:

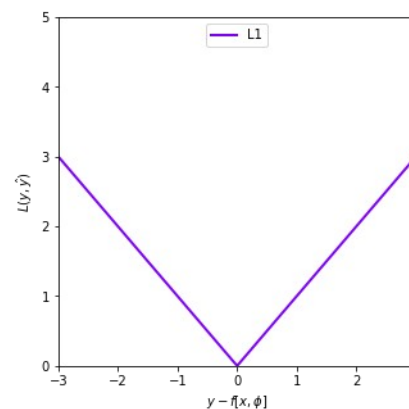


- Para salida univariante:

$$L_{L2}(y, \hat{y}) = (y - \hat{y})^2$$

## Funciones de pérdida para regresión

- Las más habituales:

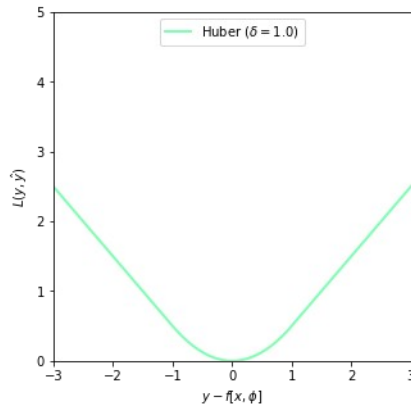


- Para salida univariante:

$$L_{L1}(y, \hat{y}) = |y - \hat{y}|$$

## Funciones de pérdida para regresión

- Las más habituales:

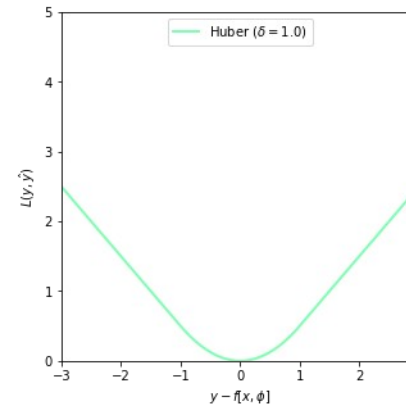


- Para salida univariante:

$$L_{Huber}(y, \hat{y}, \delta) = \begin{cases} \frac{1}{2} \cdot (y - \hat{y})^2, & \text{si } |y - \hat{y}| < \delta \\ \delta \cdot (|y - \hat{y}| - \frac{\delta}{2}), & \text{si } |y - \hat{y}| \geq \delta \end{cases}$$

## Funciones de pérdida para regresión

- Las más habituales:

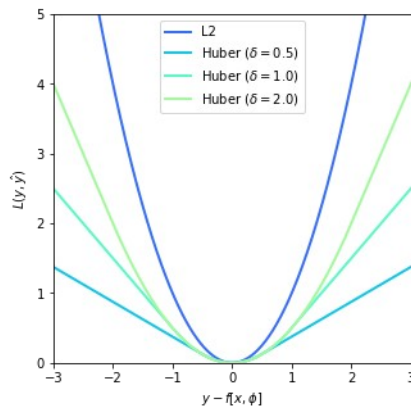


- Para salida univariante:

$$L_{Huber}(y, \hat{y}, \delta) = \begin{cases} \frac{1}{2} \cdot (y - \hat{y})^2, & \text{si } |y - \hat{y}| < \delta \\ \delta \cdot (|y - \hat{y}| - \frac{\delta}{2}), & \text{si } |y - \hat{y}| \geq \delta \end{cases}$$

## Funciones de pérdida para regresión

- Las más habituales:

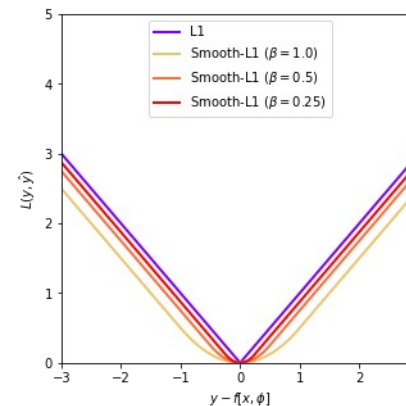


- Para salida univariante:

$$L_{Huber}(y, \hat{y}, \delta) = \begin{cases} \frac{1}{2} \cdot (y - \hat{y})^2, & \text{si } |y - \hat{y}| < \delta \\ \delta \cdot (|y - \hat{y}| - \frac{\delta}{2}), & \text{si } |y - \hat{y}| \geq \delta \end{cases}$$

## Funciones de pérdida para regresión

- Las más habituales:



- Para salida univariante:

$$L_{SmoothL1}(y, \hat{y}, \beta) = \frac{L_{Huber}(y, \hat{y}, \beta)}{\beta}$$

# Funciones de pérdida para regresión

- Las más habituales:

