

INF-398 Intro. al Aprendizaje Automático

Modelo y Conceptos Básicos



Objetivos

- Formalización del problema de aprendizaje.
- Introducir definiciones y conceptos básicos.
- Diferenciar error de predicción del error de entrenamiento.
- Introducir el problema del overfitting.

Definición



“A program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E**”

Tom Mitchell, *Machine Learning*, 1997.

Definiciones Básicas

Tarea (T)

Problema que queremos que el programa resuelva.

- Ejemplos:
 - Determinar si una opinión es positiva o negativa.
 - Detectar la presencia de una cara en una imagen.
 - Producir una descripción (caption) de una imagen.
 - Auto-completar una frase.



Definiciones Básicas

Tarea (T)

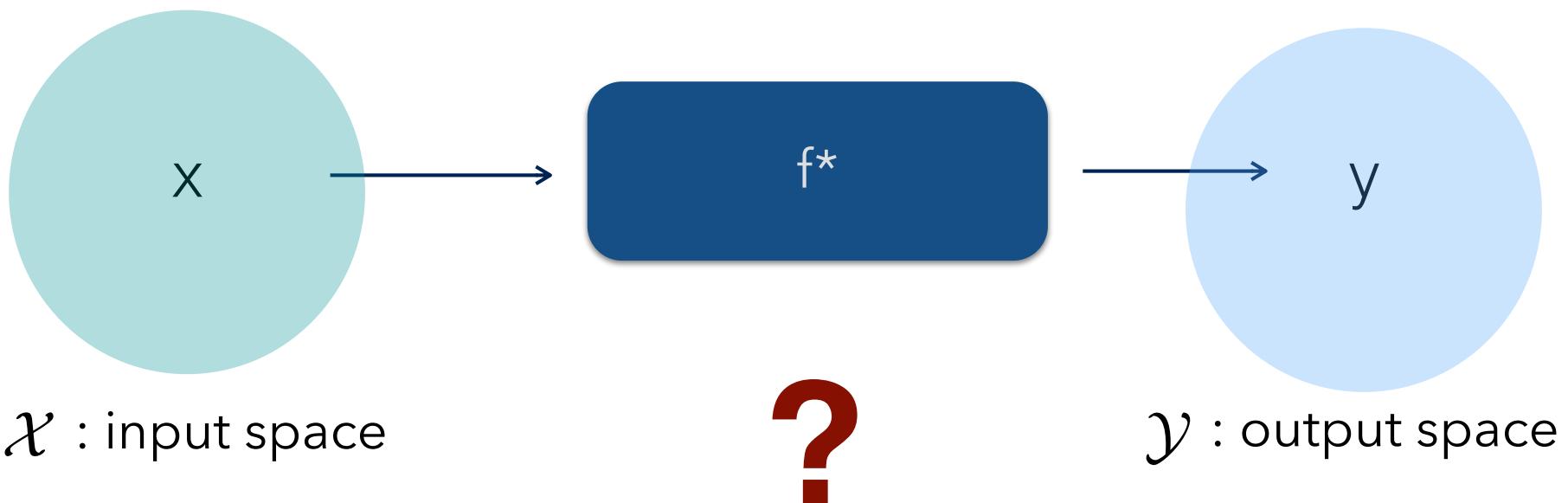
- Formalmente una tarea se puede definir como una función o transformación f^* entre dos espacios: uno que llamaremos espacio de entrada y otro que llamaremos espacio de salida



Definiciones Básicas

Tarea (T)

- Esta función es **desconocida** (sólo sabemos qué cosa toma como input y qué cosa produce como output). Queremos **imitarla al menos de modo aproximado.**



Tipos de Tarea

- **Clasificación:** problema en que queremos que el programa determine la categoría o etiqueta \mathbf{y} que corresponde a un determinado input \mathbf{x} de entre un conjunto finito y predefinido de valores.



\mathcal{X} input space

$$f : \mathcal{X} \longrightarrow \mathcal{Y} = \boxed{\{c_1, \dots, c_k\}}$$

\mathcal{Y} output space

$$x \mapsto \hat{y} = \boxed{f(x)}$$

hipótesis implementada
por el programa

Tipos de Tarea

- **Clasificación Multi-label:** en este caso input x puede pertenecer a más de una categoría simultáneamente.



\mathcal{X} input space

$\mathcal{Z} = 2^{\mathcal{Y}}$ output space

$$f : \mathcal{X} \longrightarrow 2^{\mathcal{Y}}$$

$$x \mapsto \{c_{i_1}, \dots, c_{i_{k_i}}\} = f(x)$$

hipótesis implementada
por el programa

Tipos de Tarea

- **Regresión:** problema en que queremos que el programa prediga un valor continuo (e.g. distancia, velocidad, aceleración, tiempo, precio) \mathbf{y} asociado a \mathbf{x} .



\mathcal{X} input space

\mathcal{Y} output space

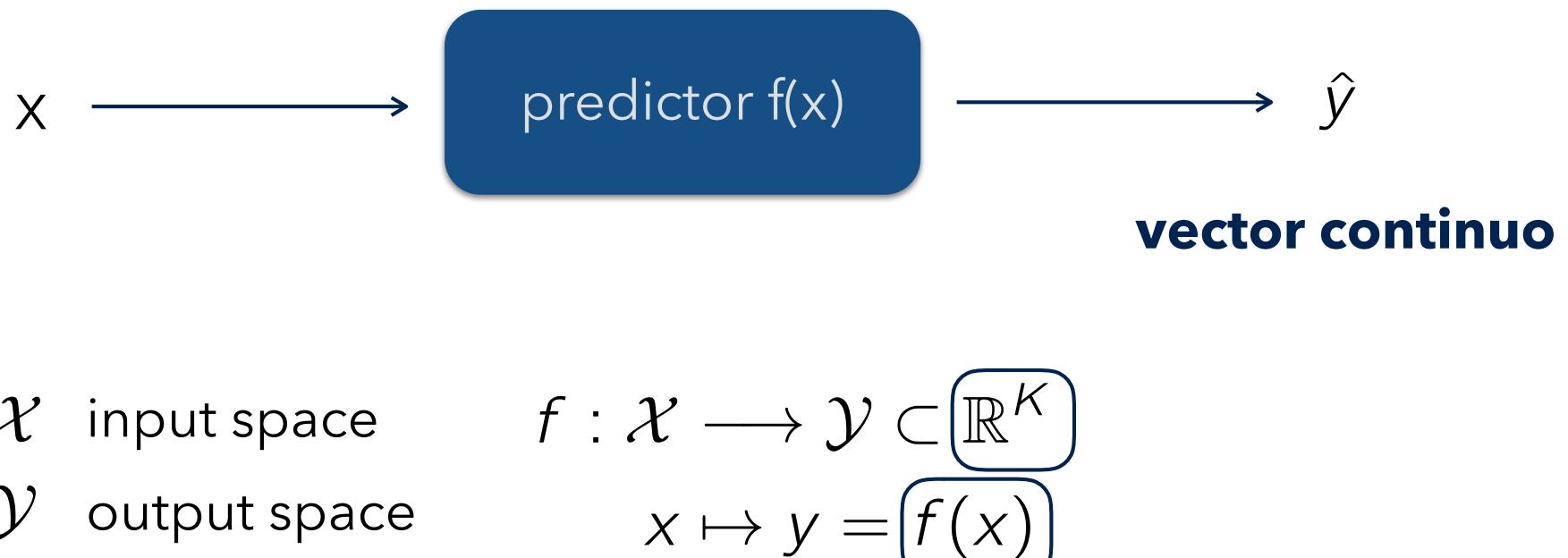
$$f : \mathcal{X} \longrightarrow \mathcal{Y} \subset \mathbb{R}$$

$$x \mapsto \hat{y} = f(x)$$

hipótesis implementada
por el programa

Tipos de Tarea

- **Regresión Múltiple:** problema en que queremos que el programa prediga un vector de valores numéricos.

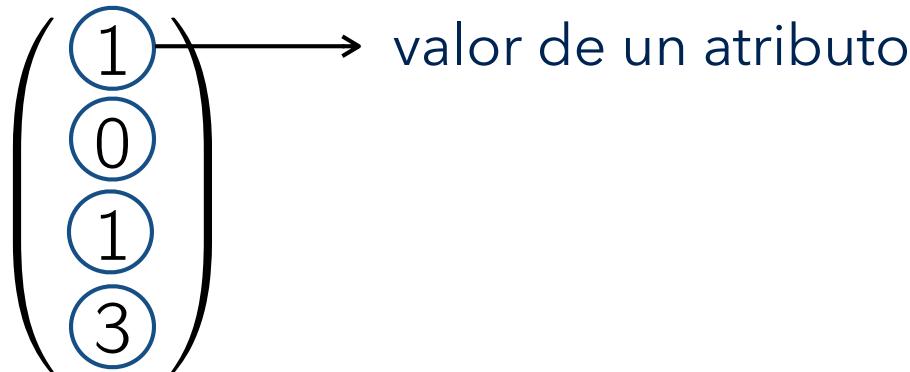


hipótesis implementada
por el programa

Representación

- **Espacio de entrada típico.** La gran mayoría de los modelos/ métodos asumirán que el input al sistema (\mathbf{x}) es un vector de atributos numéricos de dimensionalidad fija d .

$$\mathcal{X} \subset \mathbb{R}^d \quad d : \text{dimensionalidad}$$



Representación

Abstracción típica

| Name | Thread pitch (mm) | Minor diameter tolerance | Nominal diameter (mm) | Head shape | Price for 50 screws | Available at factory outlet? | Number in stock | Flat or Phillips head? |
|------|-------------------|--------------------------|-----------------------|------------|---------------------|------------------------------|-----------------|------------------------|
| M4 | 0.7 | 4g | 4 | Pan | \$10.08 | Yes | 276 | Flat |
| M5 | 0.8 | 4g | 5 | Round | \$13.89 | Yes | 183 | Both |
| M6 | 1 | 5g | 6 | Button | \$10.42 | Yes | 1043 | Flat |
| M8 | 1.25 | 5g | 8 | Pan | \$11.98 | No | 298 | Phillips |
| M10 | 1.5 | 6g | 10 | Round | \$16.74 | Yes | 488 | Phillips |
| M12 | 1.75 | 7g | 12 | Pan | \$18.26 | No | 998 | Flat |
| M14 | 2 | 7g | 14 | Round | \$21.19 | No | 235 | Phillips |
| M16 | 2 | 8g | 16 | Button | \$23.57 | Yes | 292 | Both |
| M18 | 2.1 | 8g | 18 | Button | \$25.87 | No | 664 | Both |
| M20 | 2.4 | 8g | 20 | Pan | \$29.09 | Yes | 486 | Both |
| M24 | 2.55 | 9g | 24 | Round | \$33.01 | Yes | 982 | Phillips |
| M28 | 2.7 | 10g | 28 | Button | \$35.66 | No | 1067 | Phillips |
| M36 | 3.2 | 12g | 36 | Pan | \$41.32 | No | 434 | Both |
| M50 | 4.5 | 15g | 50 | Pan | \$44.72 | No | 740 | Flat |

Ejemplo 1
Ejemplo 2

Ejemplo n

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \dots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{pmatrix}$$

n x d real matrix



Representación

- En muchos problemas prácticos el input al sistema no es un vector y se requiere algún tipo de **extractor de características que lo represente como tal.**

SCENE FROM "DAN'L DRUCE."

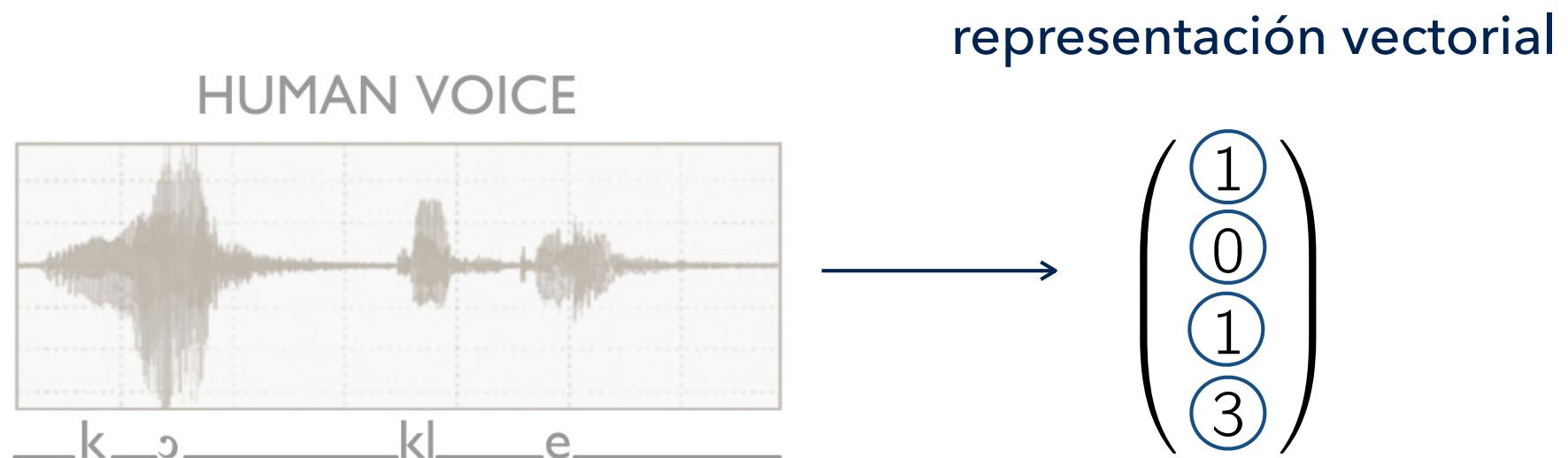
This interesting domestic drama, by Mr. W. S. Gilbert, has continued to engage the sympathies of a nightly sufficient audience at the Haymarket Theatre, where it has now been represented more than sixty times. Its subject and character were described by us, in the ordinary report of theatrical novelties, about two months ago. Our readers will probably not need to be reminded that the hero of the story, Dan'l Druce, the blacksmith, is a solitary recluse dwelling on the coast of Norfolk, where his lone cottage is visited by fugitives from party vengeance during the civil wars of the Commonwealth. His hoard of money is stolen; but a different sort of treasure, a helpless female infant, is left by some mysterious agency, and may be accepted, as in George Eliot's tale of "Silas Marner," for a Divine gift to the sad-hearted misanthrope, far better than riches. In this spirit, at least, he is content to receive the precious human charge; and so to those who would remove it from his home, Dan'l Druce here makes answer with the solemn exclamation, "Touch not the Lord's gift!" This character is well acted by Mr. Hermann Vezin.

representación vectorial

$$\xrightarrow{\hspace{1cm}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 3 \end{pmatrix}$$

Representación

- En muchos problemas prácticos el input al sistema no es un vector y se requiere algún tipo de **extractor de características que lo represente como tal**.



Representación (del Output)

- **Predicción Estructurada:** el output no es un número real o un vector de números reales sino un conjunto valores relacionados entre sí de modo semánticamente relevante.

Inglés▼

I love this university

x —————→

traductor $f(x)$

hipótesis implementada
por el programa

Español▼

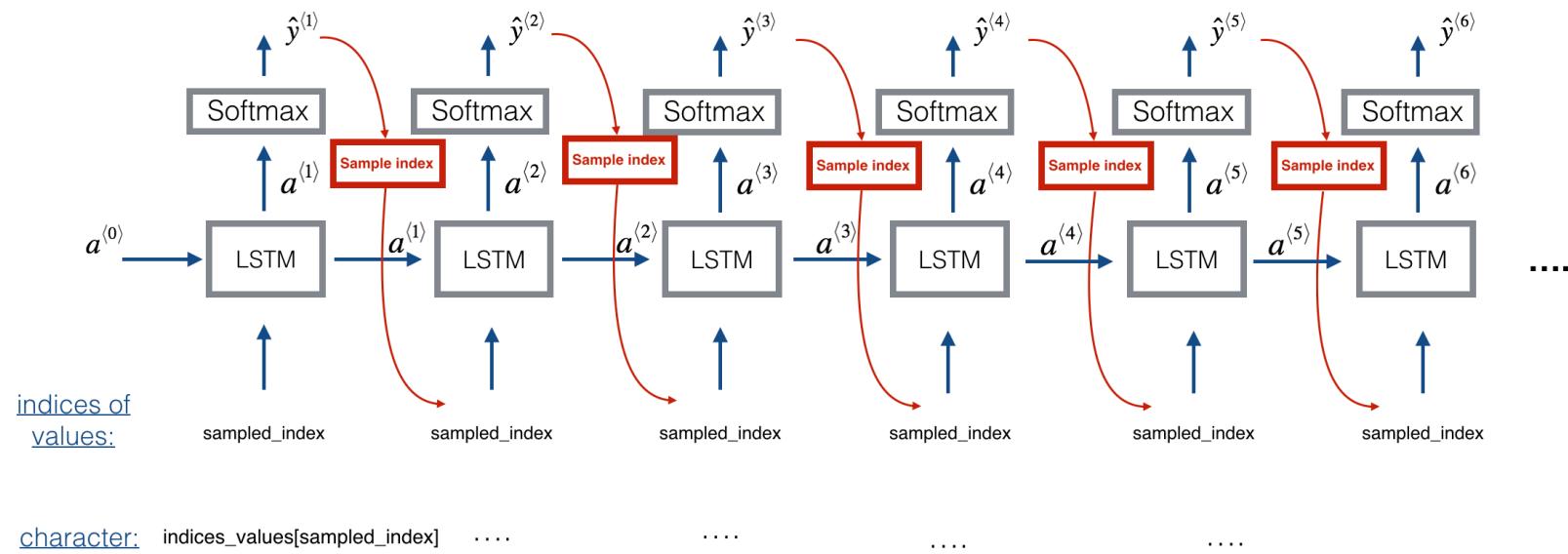
Me encanta esta
universidad

————→ y

estructurado

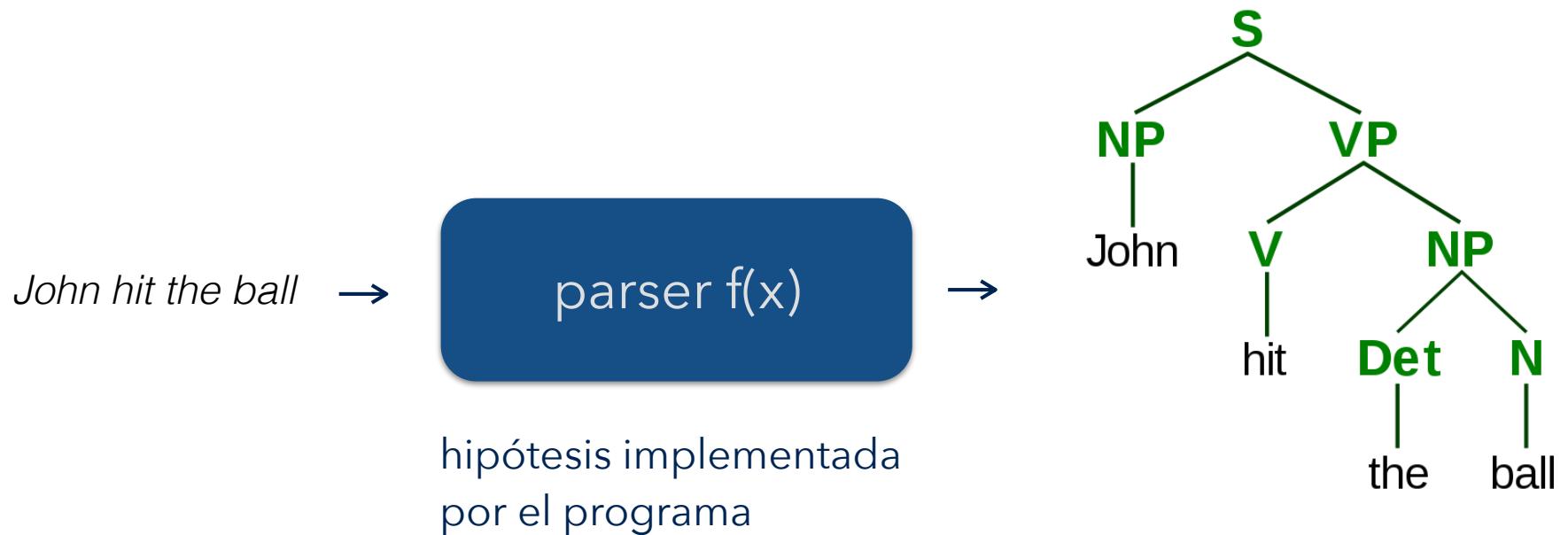


Representación (del Output)



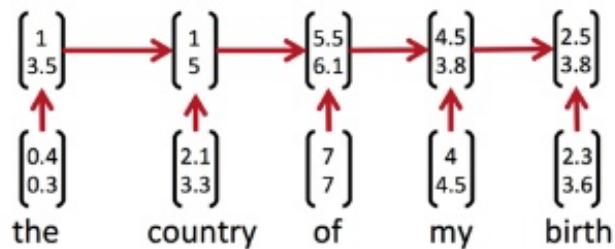
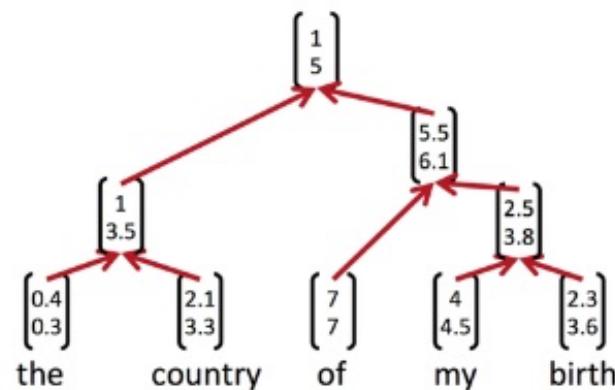
Representación (del Output)

- **Predicción Estructurada:** el output no es un número real o un vector de números reales sino un conjunto valores relacionados entre sí de modo semánticamente relevante.



Representación (del Output)

Recursive vs Recurrent NN



Otras Tareas

- **Detección de Anomalías:** se desea que el programa lance una alarma frente a inputs inusuales ó atípicos.



$$f : \mathcal{X} \longrightarrow \{\emptyset, \text{alarm}\}$$
$$x \mapsto \hat{y} = f(x)$$

Otras Tareas

- **Denoising:** Dada una señal corrupta se desea que el programa reconstruya la versión original

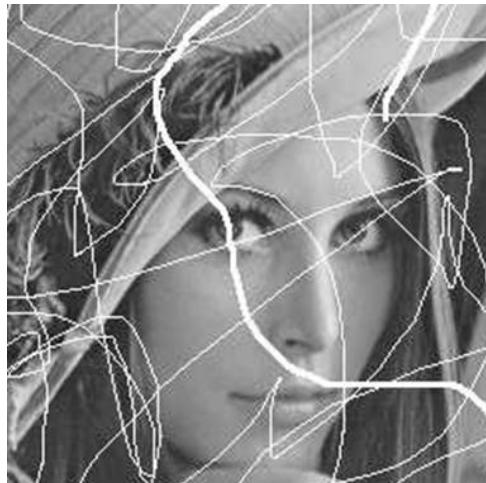


$$f : \mathcal{X} \longrightarrow \mathcal{X}$$
$$\tilde{x} \mapsto x = f(\tilde{x})$$



Otras Tareas

- **Completación de data faltante:** Dada una señal corrupta se desea que el programa reconstruya la versión original



$$f : \mathcal{X} \longrightarrow \mathcal{X}$$

$$\tilde{x} \mapsto x = f(\tilde{x})$$



Otras Tareas

- **Estimación de densidad de probabilidad:** se desea que el programa aproxime la densidad de probabilidad asociada a un input.



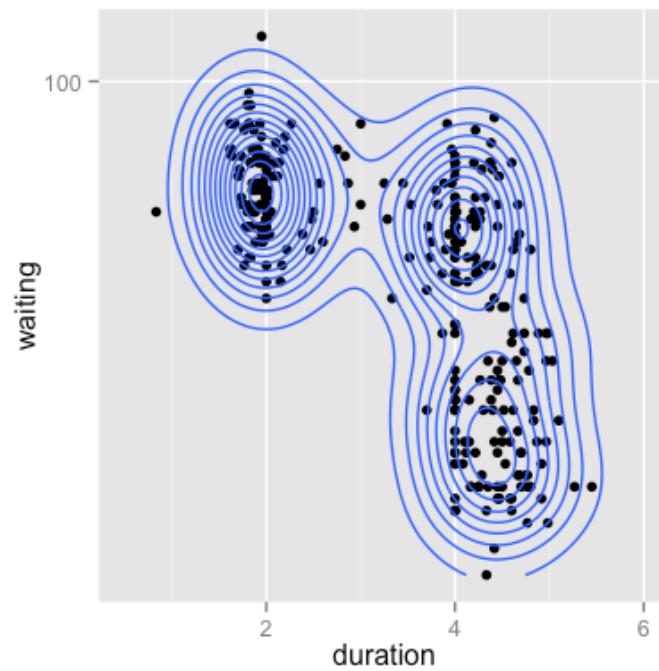
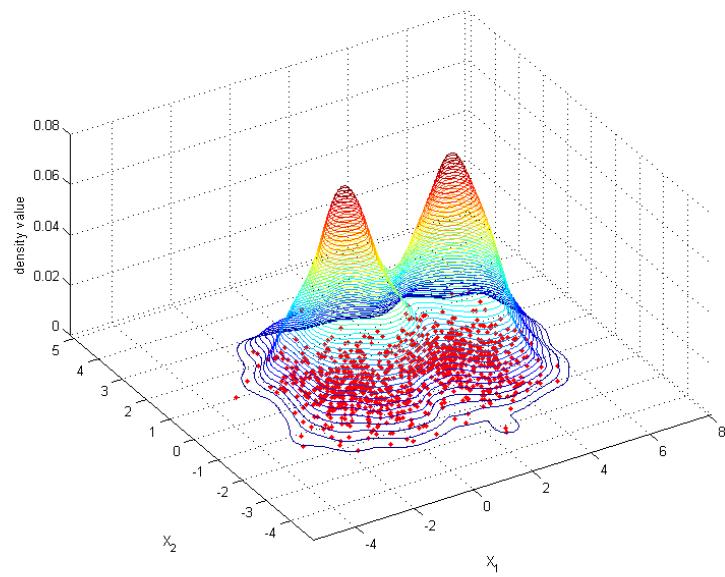
$$f : \mathcal{X} \longrightarrow [0, 1]$$

$$x \mapsto \hat{y} = q(x)$$

hipótesis implementada
por el programa

Otras Tareas

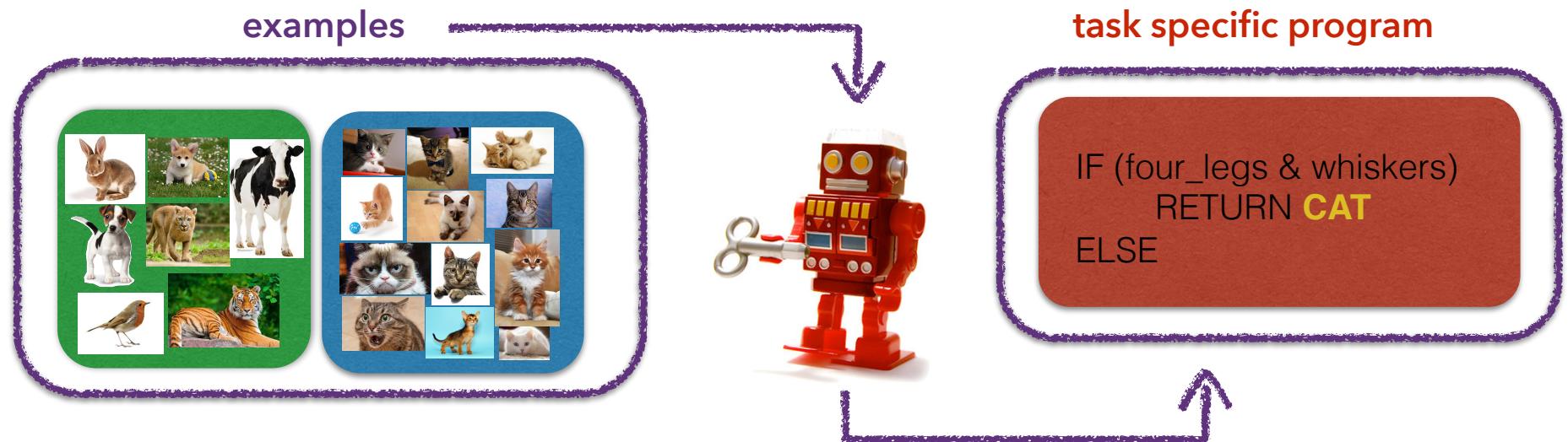
- **Estimación de densidad de probabilidad:** se desea que el programa aproxime la densidad de probabilidad asociada a un input.



Definiciones Básicas

Experiencia (E)

Información que se le proporciona al programa durante la fase de entrenamiento para que construya, mejore o adapte su solución al problema. Típicamente un conjunto de datos que representan ejemplos de la solución deseada. Este conjunto se llama **conjunto de entrenamiento o conjunto de ejemplos S**.



Definiciones Básicas

Aprendizaje Supervisado

Se dispone de un conjunto de **n** inputs con la respectiva salida o respuesta deseada.

$$S = \{x_i, y_i\}_{i=1}^n \equiv \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

Supuesto típico en regresión y clasificación.

x_1



x_2



x_3



$y_1 = \text{'perro'}$

Prof. Ricardo Ñanculef - Departamento de Informática UTEM 2021

$y_2 = \text{'gato'}$

$y_3 = \text{'gato'}$

Definiciones Básicas

Aprendizaje NO supervisado

Se dispone de un conjunto de **n** inputs sin la respuesta óptima / correcta / deseada.

$$S = \{x_i\}_{i=1}^n \equiv \{(x_1, x_2, \dots, x_n)\}$$

Escenario típico en detección de anomalías, denoising, estimación de densidades de probabilidad e imputación de valores faltantes.

x_1



x_2



x_3



Definiciones Básicas

Aprendizaje SEMI supervisado

Se dispone de un conjunto de **n** inputs algunos con la respuesta óptima / correcta / deseada y otros (gran mayoría) no.

$$S = \{\{x_i\}_{i=1}^{n_1}, \{y_i\}_{i=1}^{n_2}\}$$

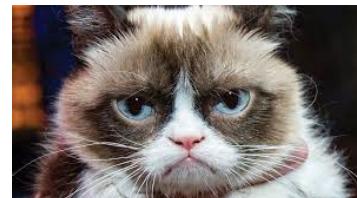
Escenario típico en la práctica.

x_1



$y_1 = \text{'perro'}$

x_2

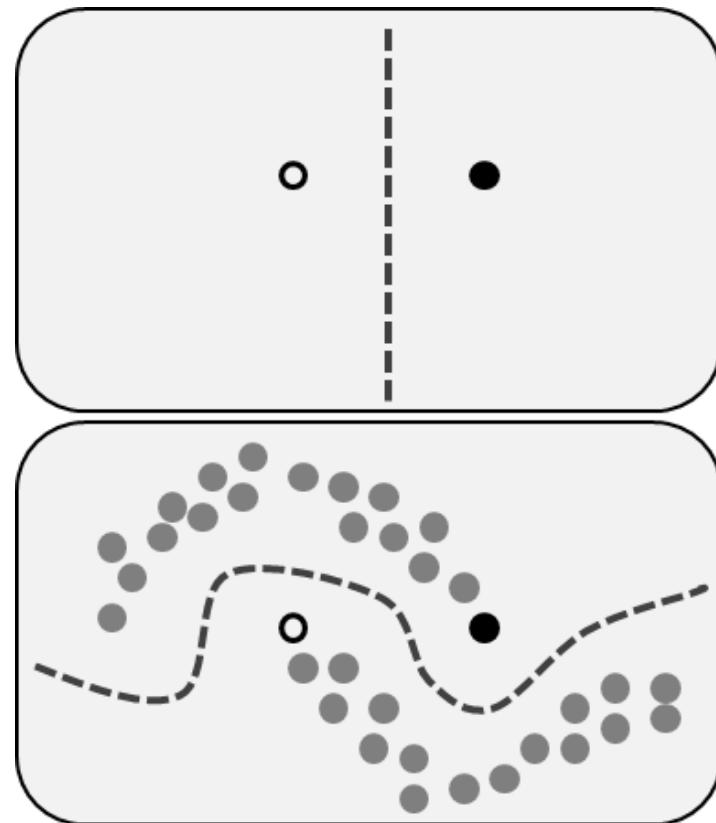


$y_2 = \text{'gato'}$

x_3



Definiciones Básicas



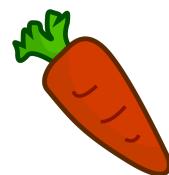
Definiciones Básicas

Aprendizaje Reforzado

La máquina/programa está inserta en un ambiente (matemáticamente definido). En vez de observar la respuesta deseada la máquina decide/predice y luego observa un valor (reward) que le permite evaluar que tan buena o mala fue su predicción/decisión.



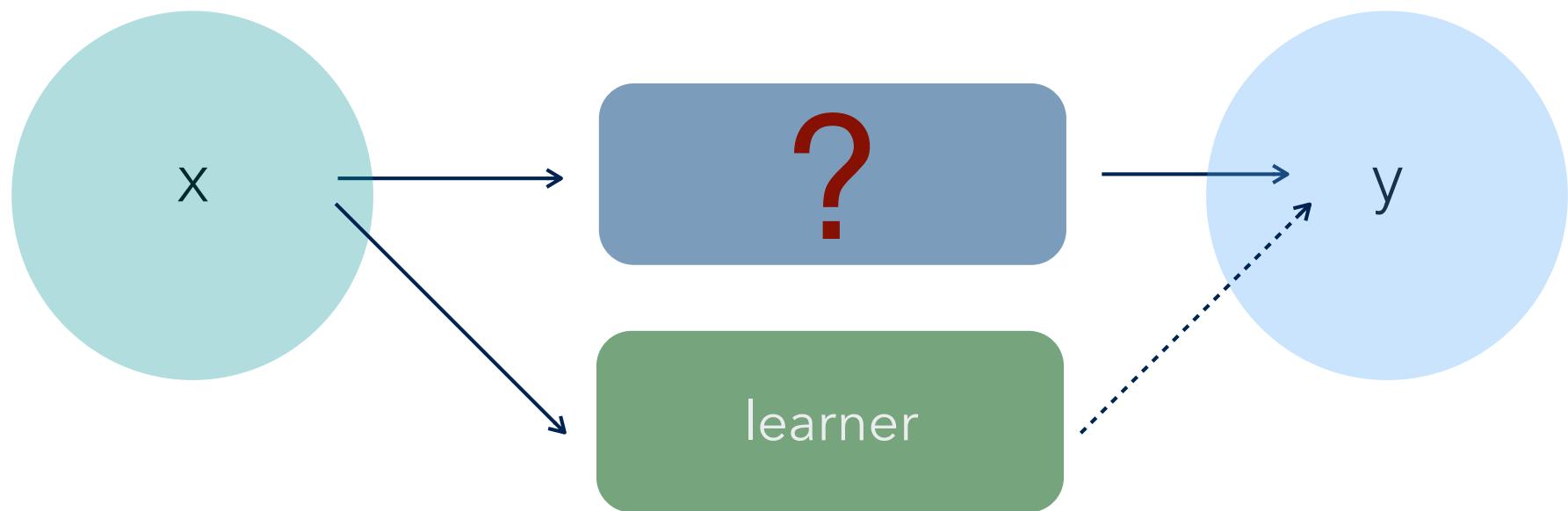
$$\hat{y} = \text{'perro'}$$



Definiciones Básicas

Experiencia (E)

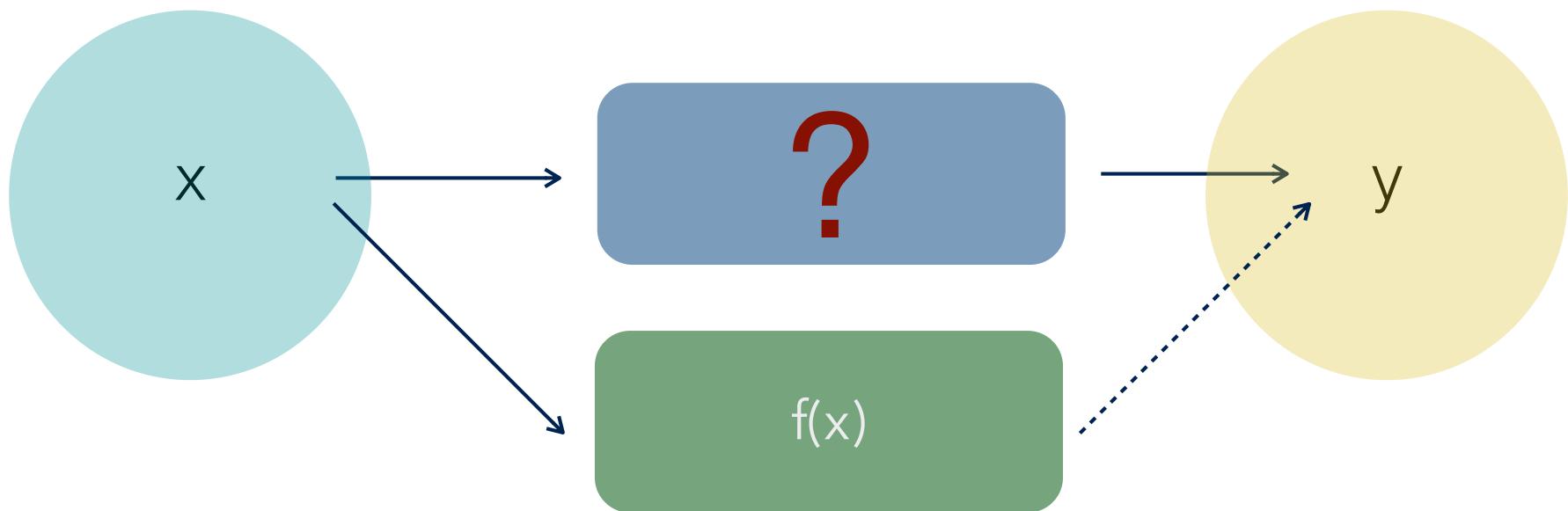
- Para aproximar la función desconocida, la máquina observa ejemplos, es decir una serie de casos input-output. Dado un input, la máquina tratará de producir el output “correcto”.



Definiciones Básicas

Hipótesis

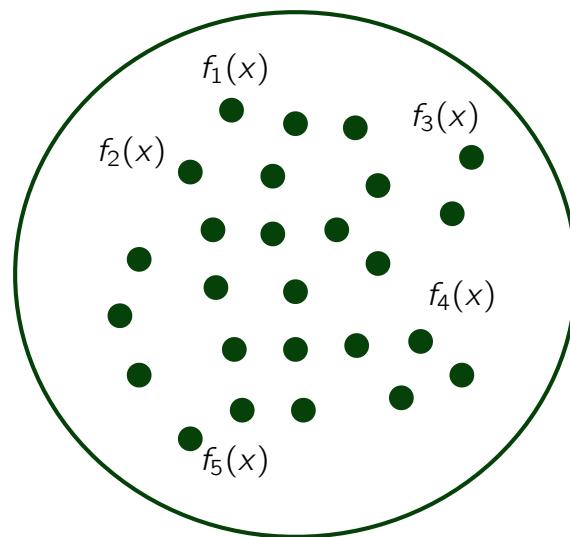
- Con este objetivo, la máquina implementa una función que permite transformar x en y . Esta función se denomina la *hipótesis* de la máquina.



Definiciones Básicas

Espacio de Hipótesis

El conjunto de todas las funciones que la máquina puede implementar para implementar la tarea.

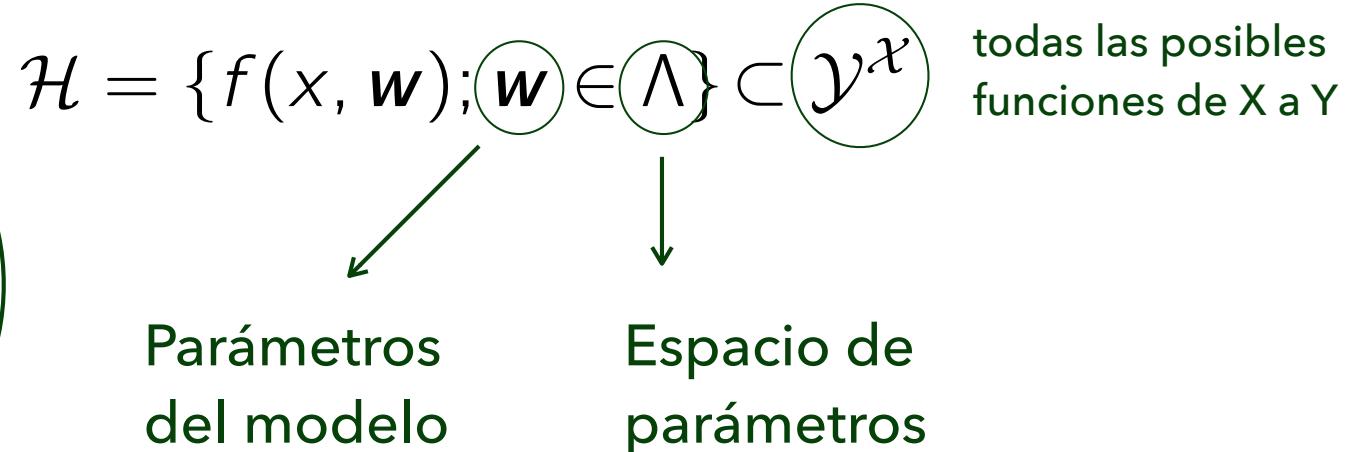
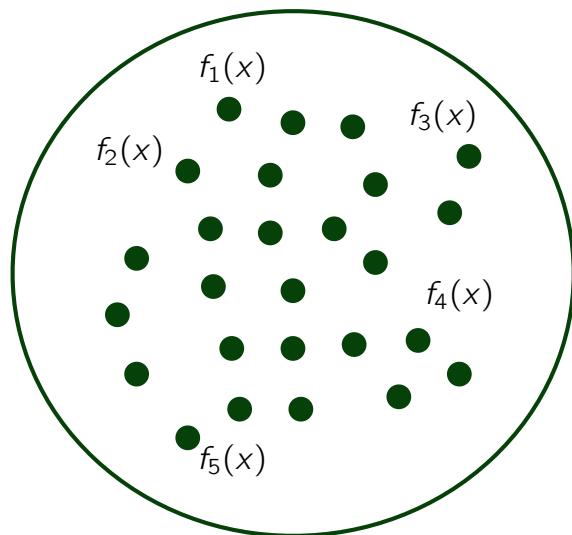


$$\mathcal{H} = \{f(x, \mathbf{w}); \mathbf{w} \in \Lambda\}$$

Definiciones Básicas

Espacio de Hipótesis

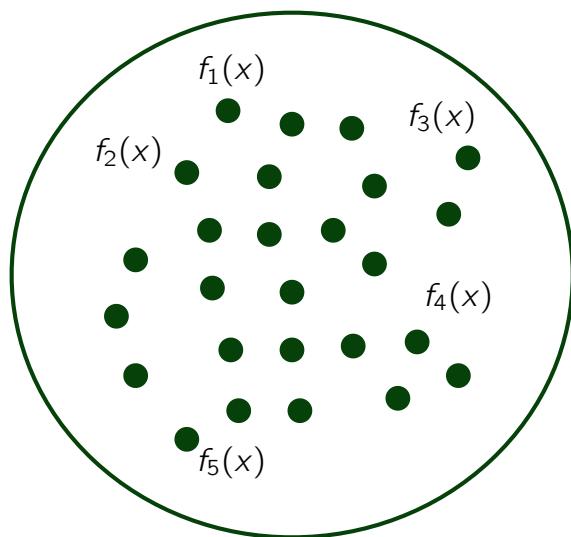
Normalmente, este espacio está *parametrizado*, es decir, cada función queda identificada por un conjunto finito de parámetros.



Definiciones Básicas

Medida de Desempeño (P)

Funcional **R** que permite medir cuantitativamente la calidad de la función **f(x)** implementada por el programa.



$$\mathcal{H} = \{f(x, \mathbf{w}); \mathbf{w} \in \Lambda\}$$

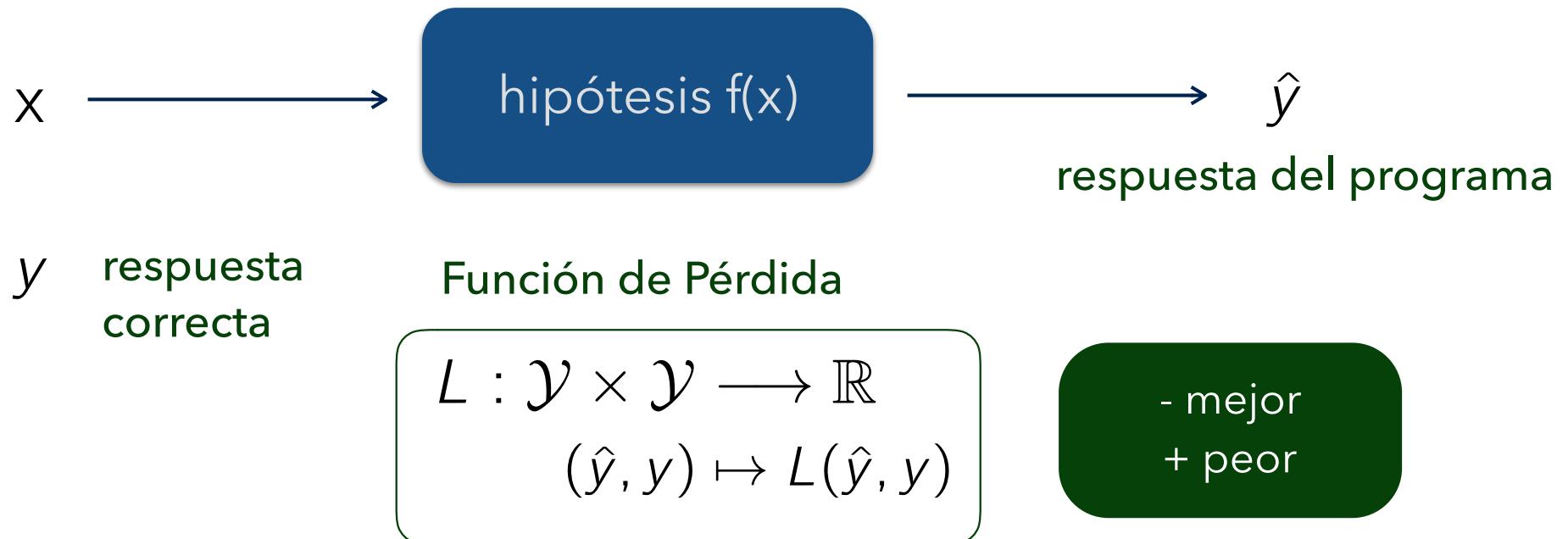
Medida de desempeño

$$R : \mathcal{H} \rightarrow \mathbb{R}$$

Definiciones Básicas

Función de Pérdida (**Loss function**)

Función **L** que permite medir la calidad de la hipótesis **f(x)** implementada por el programa sobre un input específico **x** y posiblemente la respuesta correcta/deseada/óptima para ese input.



Función de Costo (Loss)

- **Clasificación**



- **Misclassification Loss**

$$L(\hat{y}, y) = I(\hat{y} \neq y) = \begin{cases} 1 & \text{si } \hat{y} \neq y \\ 0 & \text{si } \hat{y} = y \end{cases}$$

Función de Costo (Loss)

- **Regresión**



- **Squared Loss** $L(\hat{y}, y) = (\hat{y} - y)^2$
- **Epsilon Insensitive Loss**

$$L(\hat{y}, y) = (|\hat{y} - y| - \epsilon)_+ = \begin{cases} |\hat{y} - y| & \text{si } |\hat{y} - y| \geq \epsilon \\ 0 & \text{en otro caso} \end{cases}$$

Función de Costo (Loss)

- **Estimación de densidad de probabilidad**

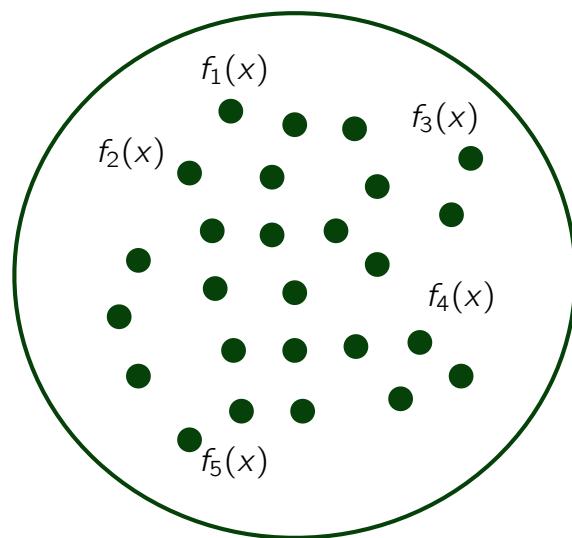


- **Negative Log Loss** $L(\hat{y}) = -\ln \hat{y} = -\ln q(x)$

Definiciones Básicas

Medida de Desempeño (P)

Funcional **R** que permite medir cuantitativamente la calidad de la función **f(x)** implementada por el programa.



$$\mathcal{H} = \{f(x, w); w \in \Lambda\}$$

Medida de Desempeño Canónica
Error de Predicción o Riesgo

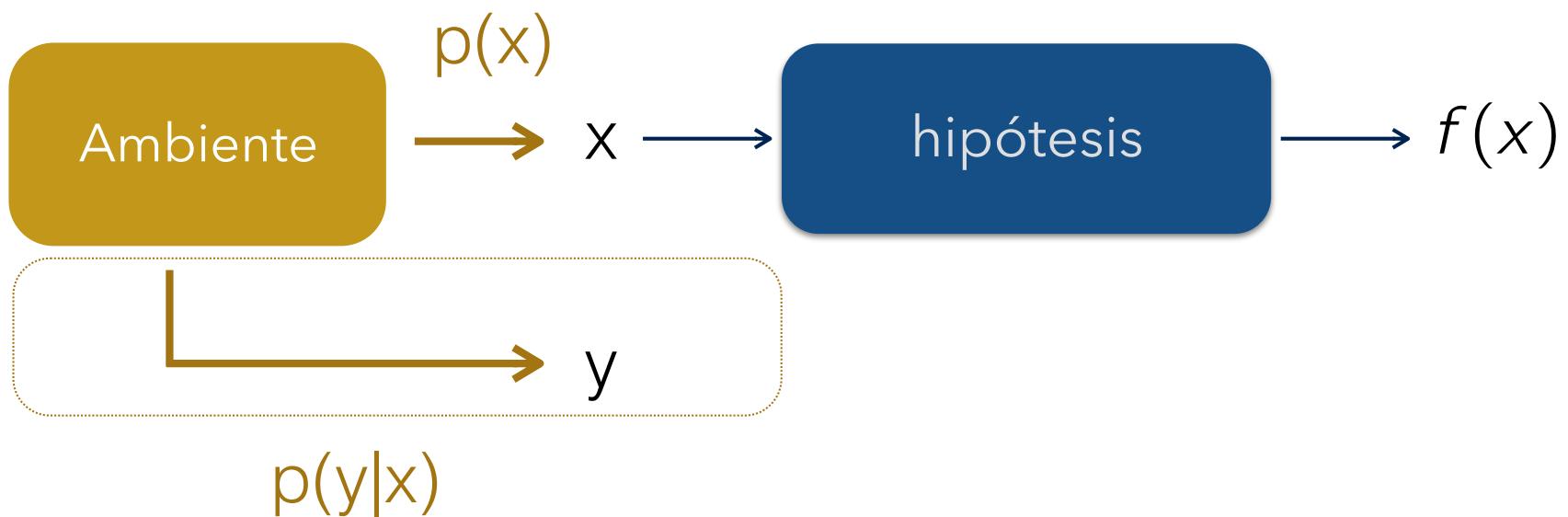
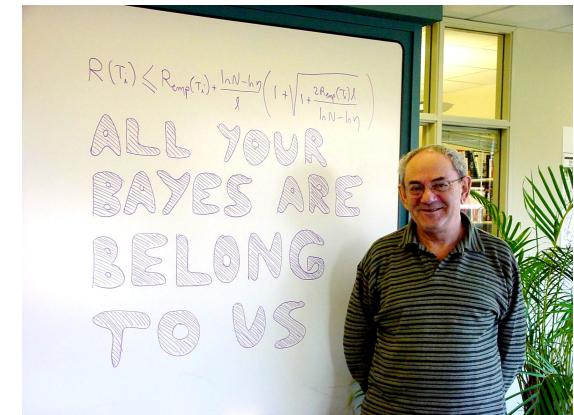
$$R(f) = E(L(f(x), y))$$

Riesgo

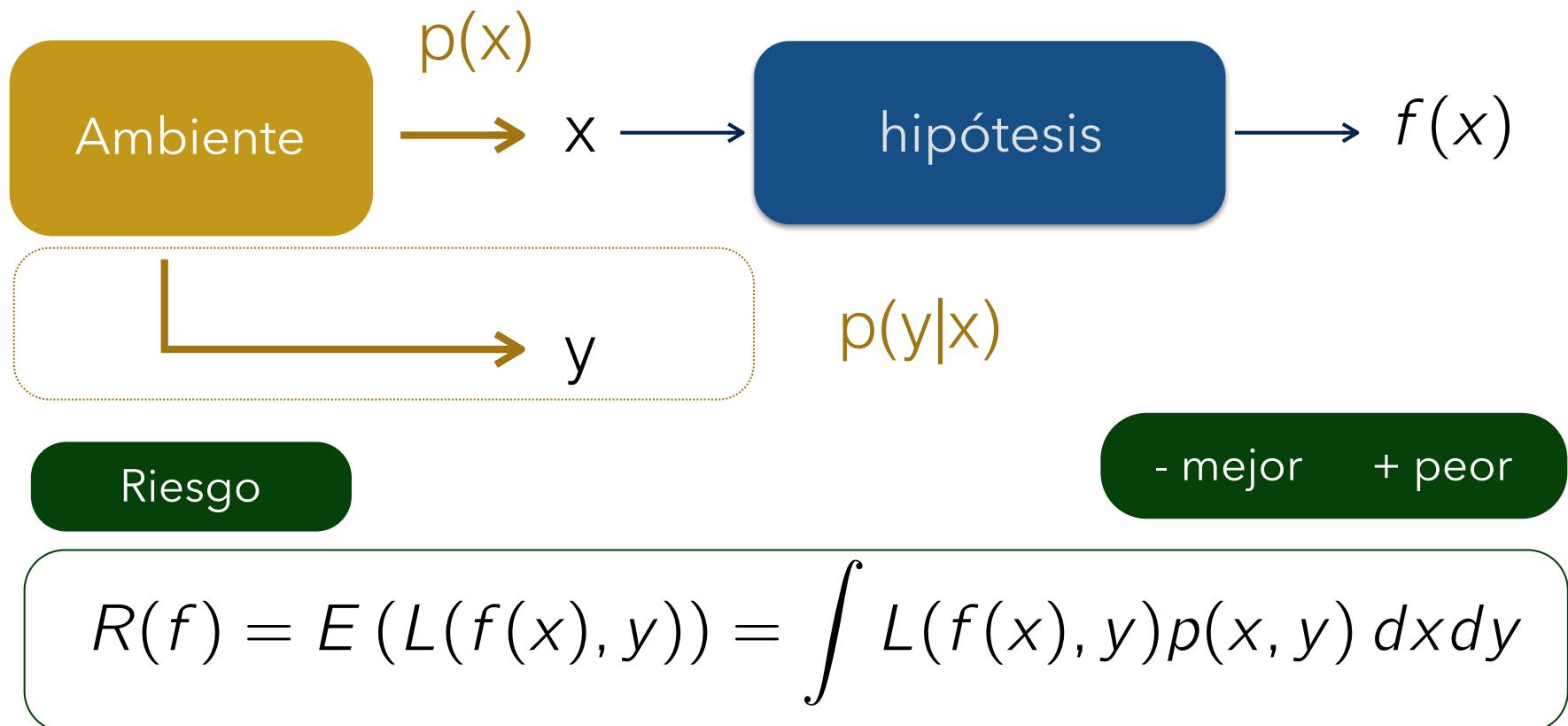
Error de Predicción (Riesgo)

Medida de Desempeño Canónica
Error de Predicción o Riesgo

$$R(f) = E(L(f(x), y))$$



Error de Predicción (Riesgo)



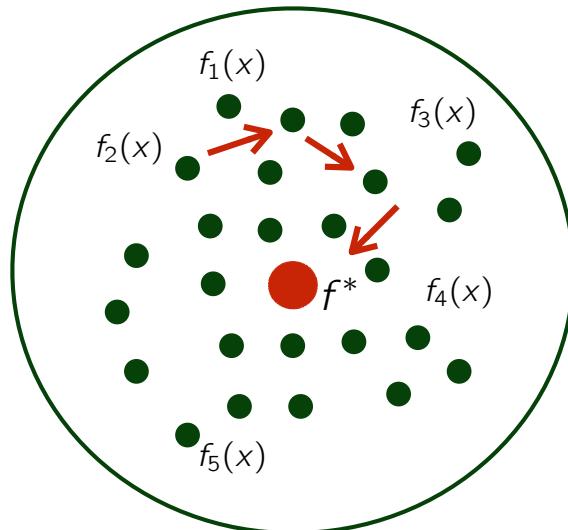
Error de Predicción (Riesgo)

Objetivo del Aprendizaje

=

Minimizar el Riesgo

$$\min R(f) = E(L(f(x), y)) \text{ s.t. } f \in \mathcal{H}$$



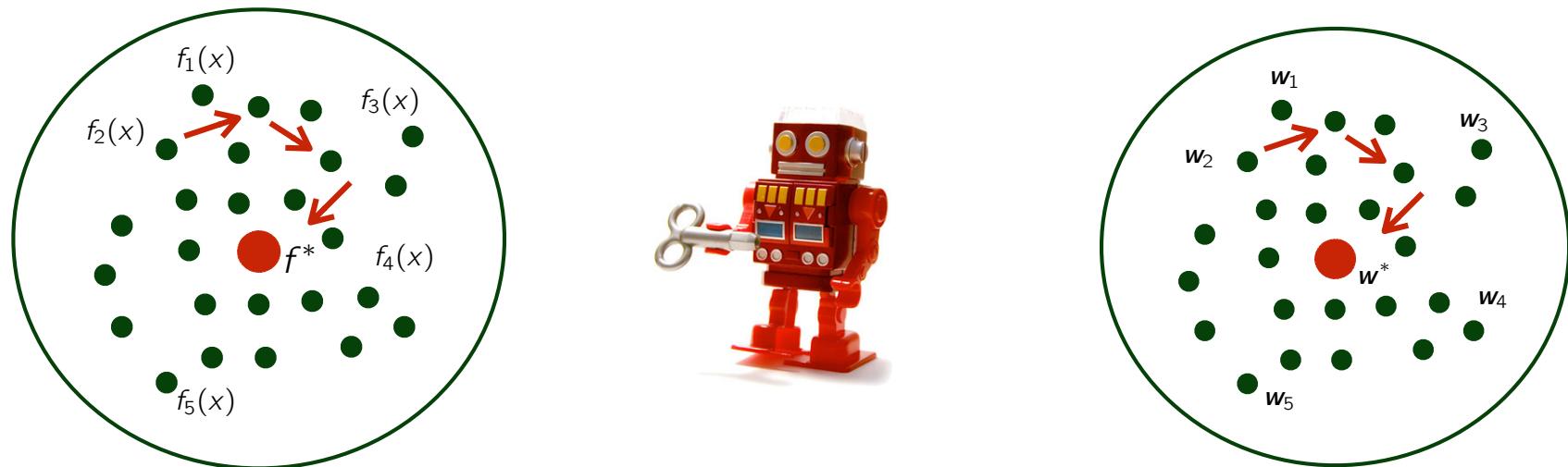
Learning as Search

```
IF (four_legs & whiskers)
    RETURN CAT
ELSE
    RETURN NO CAT
```

Error de Predicción (Riesgo)

Learning as Search

$$\min R(\mathbf{w}) = E(L(f(x, \mathbf{w}), y)) \text{ s.t. } \mathbf{w} \in \Lambda$$



$$\mathcal{H} = \{f(x, \mathbf{w}); \mathbf{w} \in \Lambda\} \subset \mathcal{Y}^{\mathcal{X}}$$

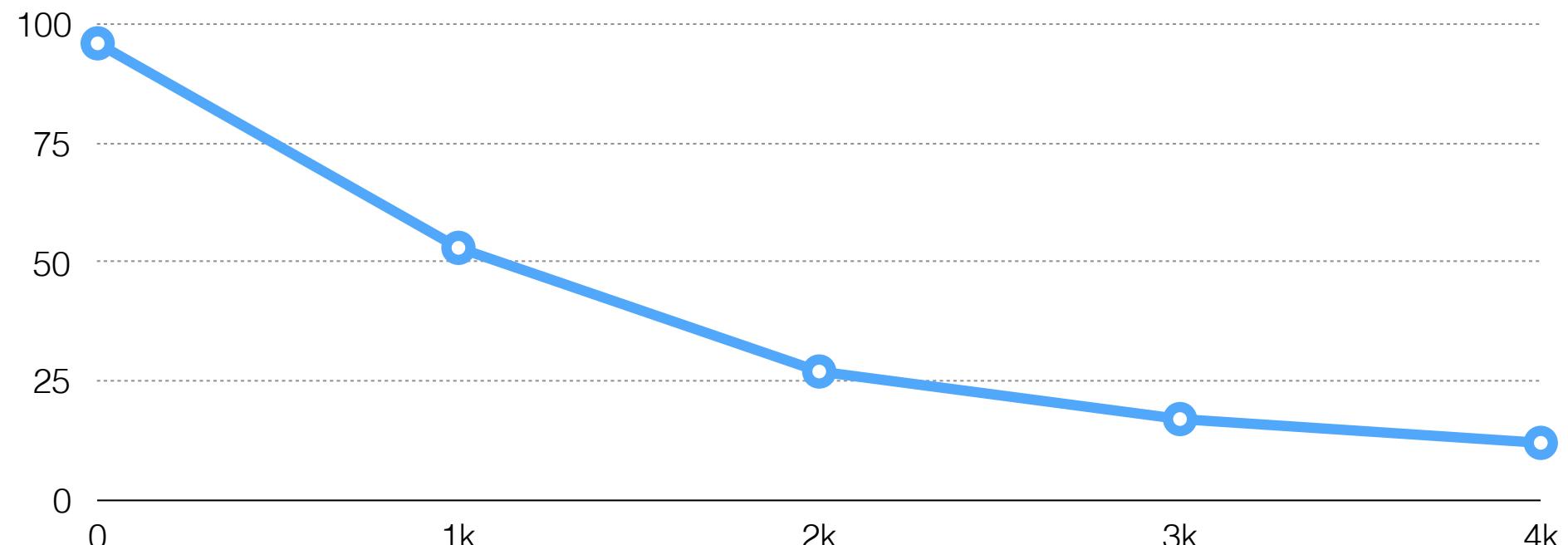
Parámetros
del modelo

Espacio de
parámetros



Error de Predicción (Riesgo)

Riesgo



Experience

Definiciones Básicas



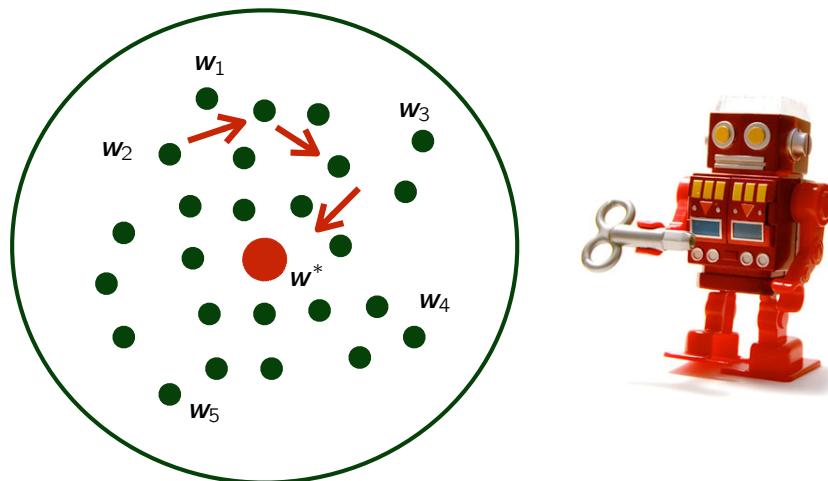
Prof. Tom Mitchell (1997)

“A program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E**”

Learning no es (sólo) Optimización!

Objetivo del Aprendizaje = Minimizar el Riesgo

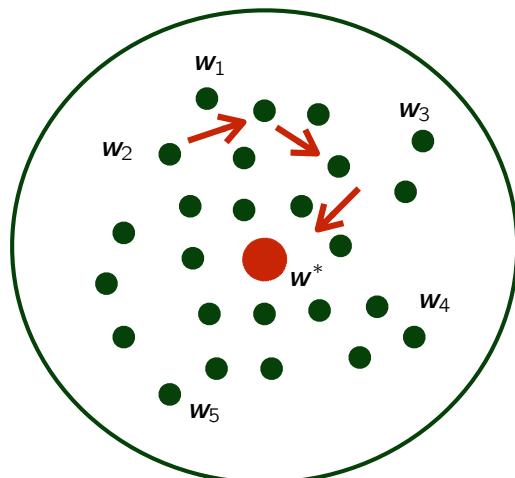
$$\min_{f \in \mathcal{H}} R(f) = \int L(f(x), y) p(x, y) dx dy$$



Learning no es (sólo) Optimización!

Objetivo del Aprendizaje = Minimizar el Riesgo

$$\min_{f \in \mathcal{H}} R(f) = \int L(f(x), y) p(x, y) dx dy$$



Gran Problema

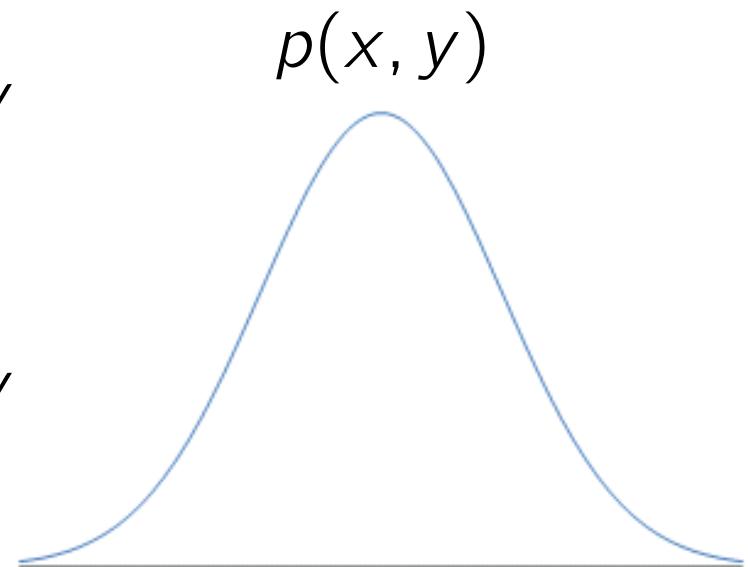
Evaluar el riesgo requiere conocer la distribución de probabilidad $p(x,y)$, i.e., saber exactamente cómo se generan los datos.

Principio de Inducción

Medida de desempeño **dependiente de los datos** que permite medir aproximar el riesgo **R**.

$$R(f) = \int L(f(x), y) p(x, y) dx dy$$

$$\hat{R}(f) = \int L(f(x), y) \hat{p}(x, y) dx dy$$



Principio de Inducción

Error de Entrenamiento (Riesgo Empírico)

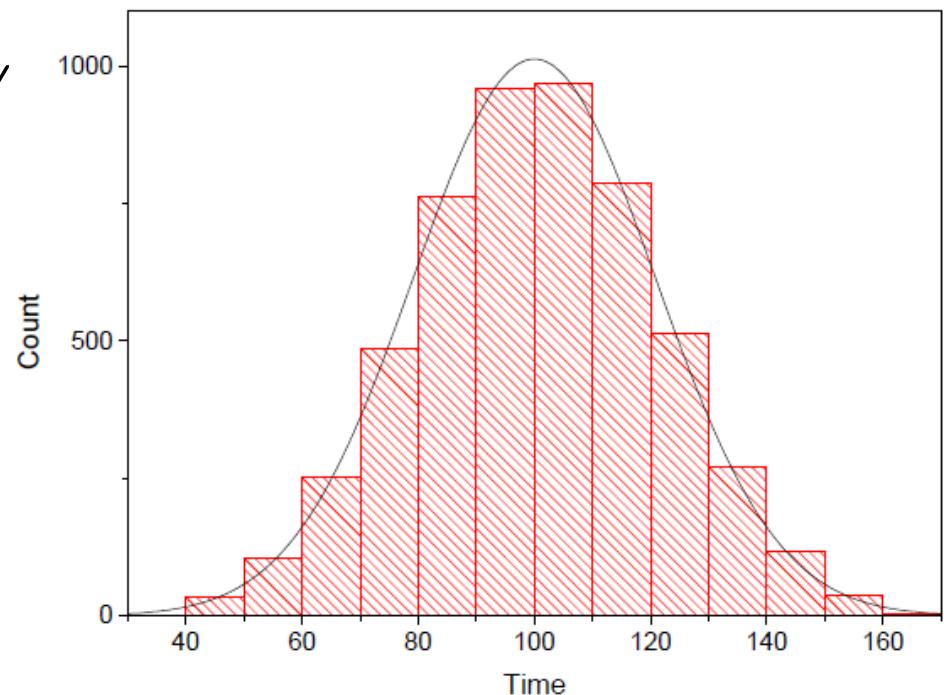
Criterio obtenido al reemplazar el riesgo teórico **R por su versión muestral o empírica.**

$$\hat{R}(f) = \int L(f(x), y) \hat{p}(x, y) dx dy$$

$$\hat{p}_{\text{emp}}(x, y) = \frac{1}{n} \sum_i \delta_i(x, y)$$

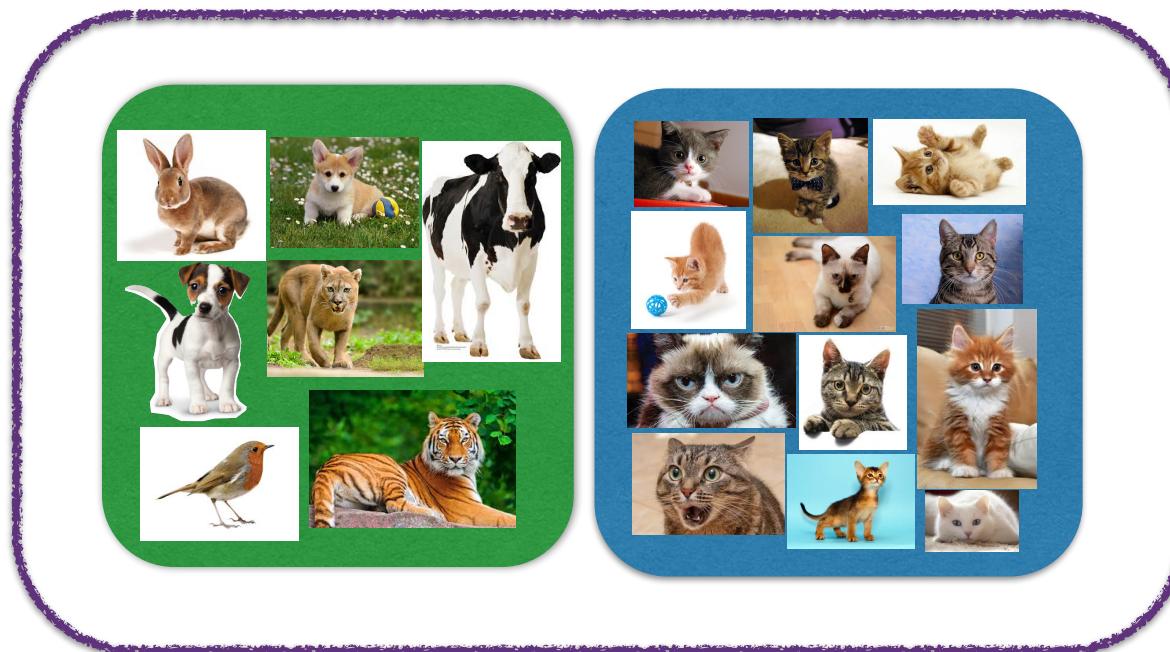
Error de Entrenamiento

$$\hat{R}_{\text{emp}}(f) = \sum_i L(f(x_i), y_i)$$



Generalización vs Overfitting

Ejemplos de entrenamiento



Error de Entrenamiento

$$\hat{R}_{\text{emp}}(f) = \sum_i L(f(x_i), y_i)$$

Generalización vs Overfitting

Casos nuevos / desconocidos



Error de Predicción (Riesgo)

$$R(f) = \int L(f(x), y)p(x, y) dx dy$$

Generalización vs Overfitting

- Es fácil demostrar (tarea, asumir datos IID) que

Error de Entrenamiento

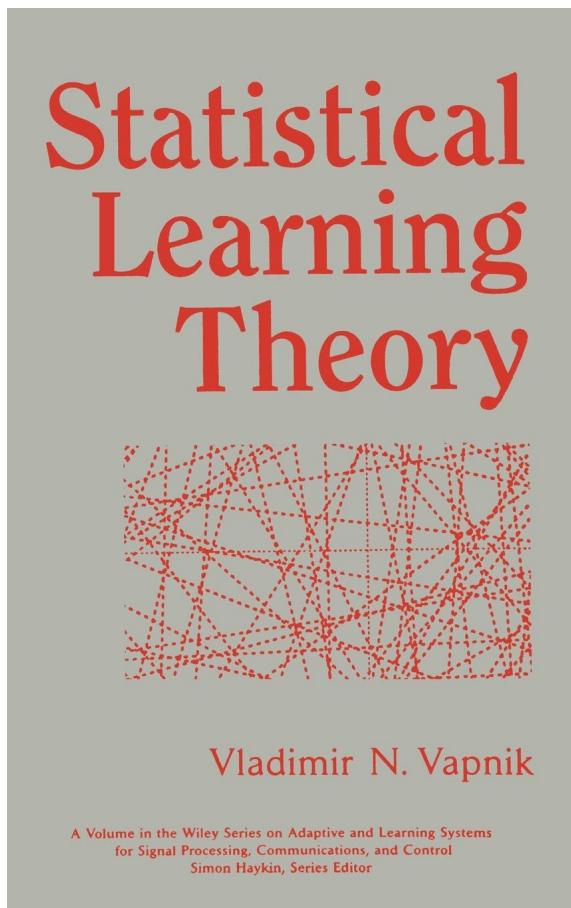
$$\hat{R}_{\text{emp}}(f) = \sum_i L(f(x_i), y_i)$$

Error de Predicción (Riesgo)

$$R(f) = \int L(f(x), y) p(x, y) dx dy$$

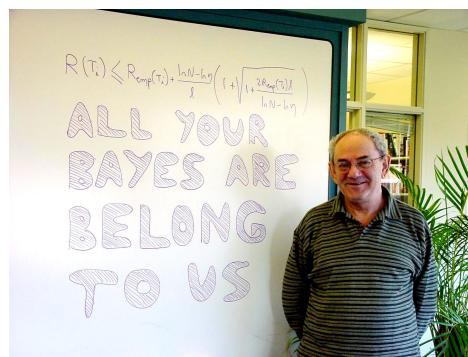
- ¿Qué tan grande puede ser la diferencia?

Generalización vs Overfitting

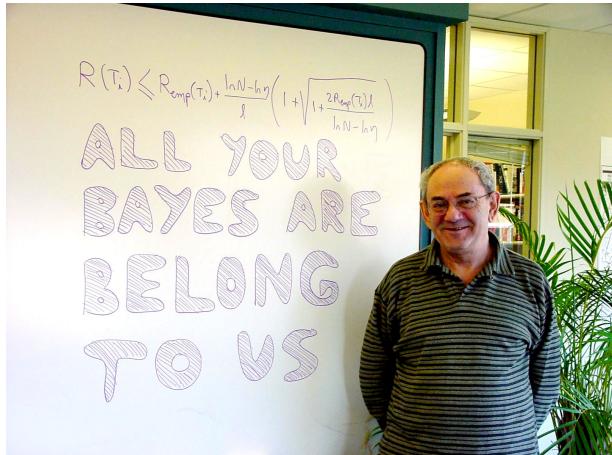


Prof. W. Vapnik (STL)

La diferencia depende de la **complejidad/capacidad** del espacio de hipótesis H sobre el cuál aprende la máquina. Una forma de medir esa complejidad de es la **dimensión VC** (para modelos lineales proporcional al número de parámetros entrenables).



Generalización vs Overfitting



Prof. W. Vapnik (STL)

Sea \mathbf{c} la dimensión VC del espacio de hipótesis. Entonces, con probabilidad $1 - \eta$ se verifica que

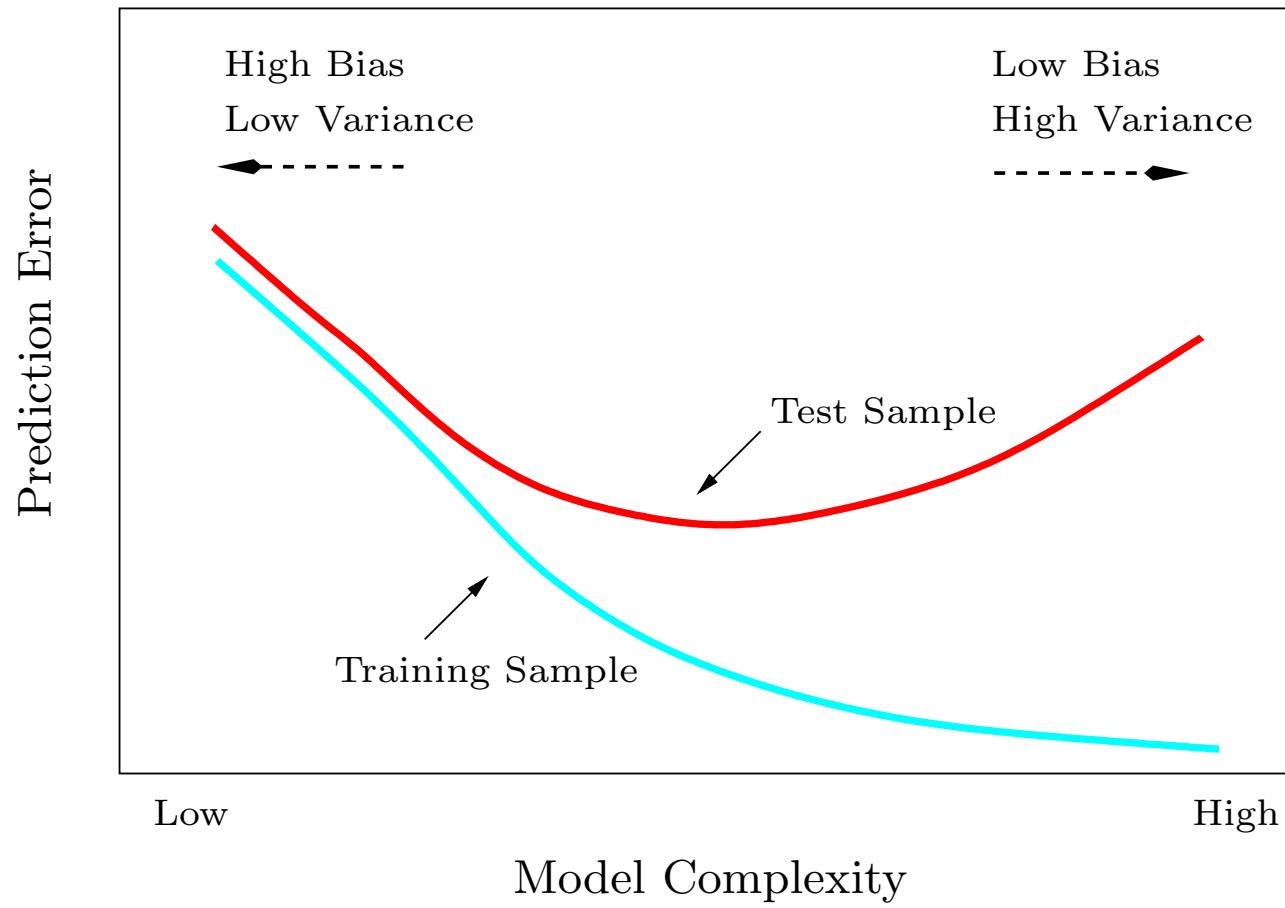
$$R(f) \leq \hat{R}_{\text{emp}}(f) + \sqrt{\frac{c \log(\frac{2n}{c} + 1) - \log(\frac{\eta}{4})}{n}}$$

Error "de pruebas"
(riesgo)

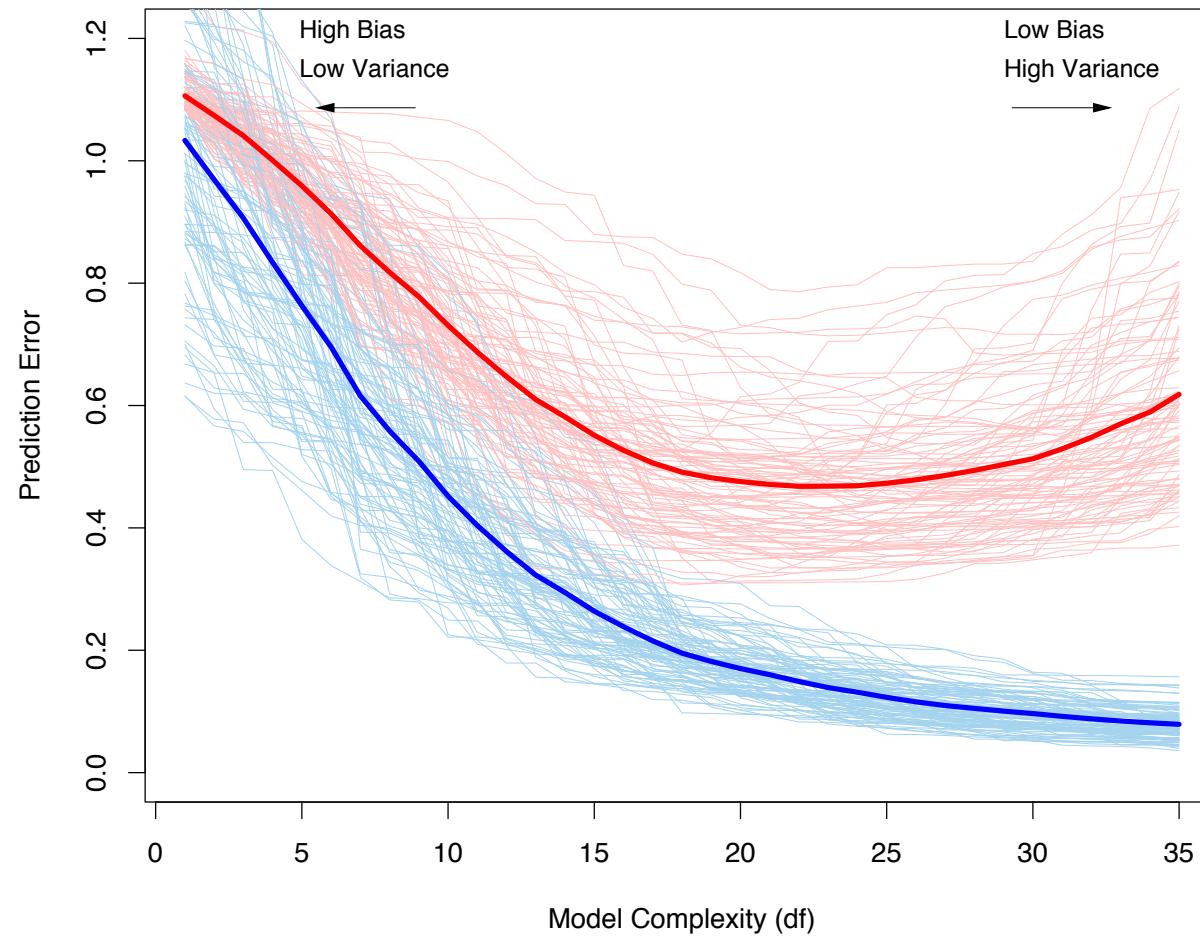
Error de
entrenamiento

Complejidad

Generalización vs Overfitting

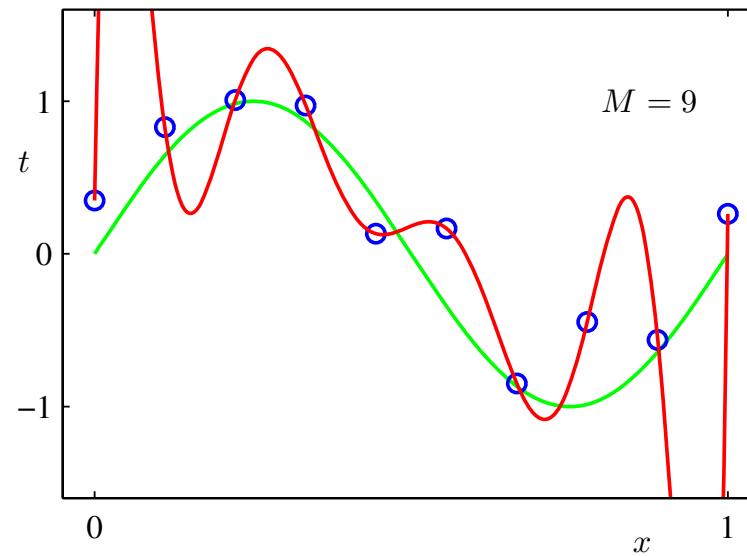
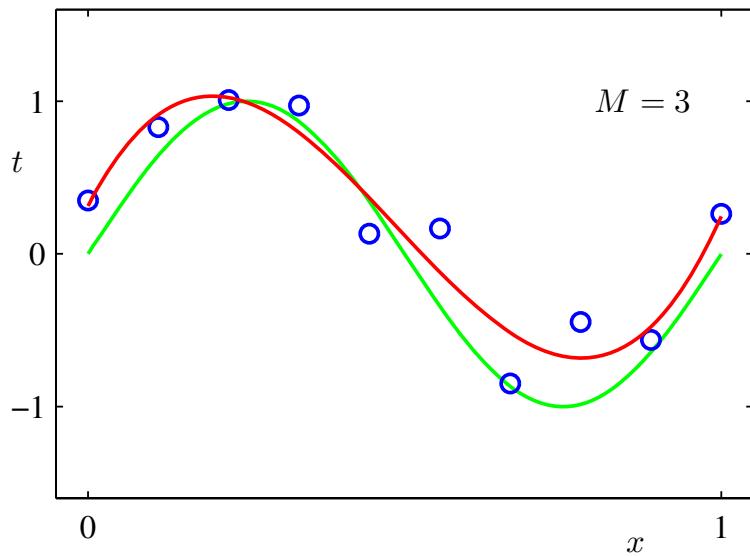


Generalización vs Overfitting



Experimento

- Demostraremos el problema del overfitting en un ejemplo sencillo que involucra regresión polinomial.



Generalización vs Overfitting

- Si el espacio de hipótesis es muy "complejo" en términos de dimensión VC para el número de datos de entrenamiento disponibles,

Error de Entrenamiento

$$\hat{R}_{\text{emp}}(f) = \sum_i L(f(x_i), y_i) <<$$

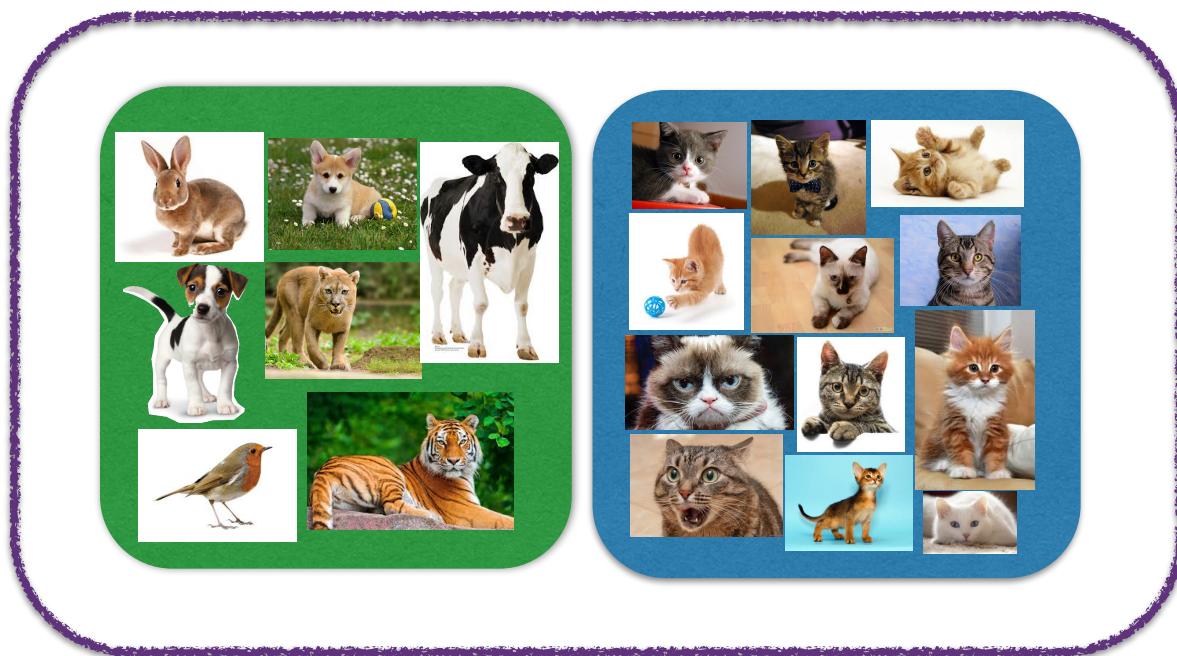
Error de Predicción (Riesgo)

$$R(f) = \int L(f(x), y) p(x, y) dx dy$$

- Cuando esto ocurre se dice el modelo/goritmo cae en **overfitting**.

Generalización vs Overfitting

Datos conocidos

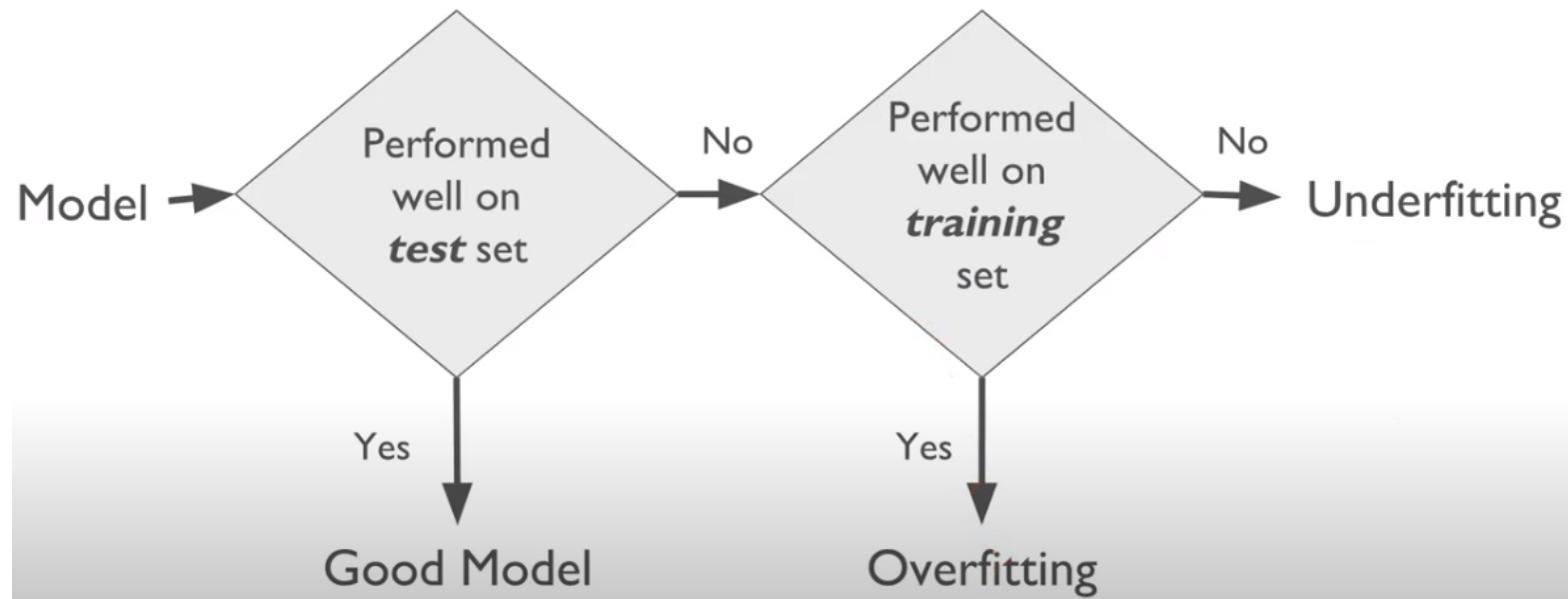


Generalización vs Overfitting

Casos nuevos / desconocidos



Generalización vs Overfitting



- En estricto rigor un algoritmo puede sub-ajustar generalizando bien.

Datos de Prueba (Test Set)

- La discusión anterior sugiere que una buena práctica después de aplicar un método de aprendizaje a un problema es evaluar el resultado en un **conjunto independiente del conjunto de entrenamiento**.

Datos disponibles

Entrenamiento

Test

Datos de Prueba (Test Set)

- La discusión anterior sugiere que una buena práctica después de aplicar un método de aprendizaje a un problema es evaluar el resultado en un **conjunto independiente del conjunto de entrenamiento**.

Entrenamiento

Test

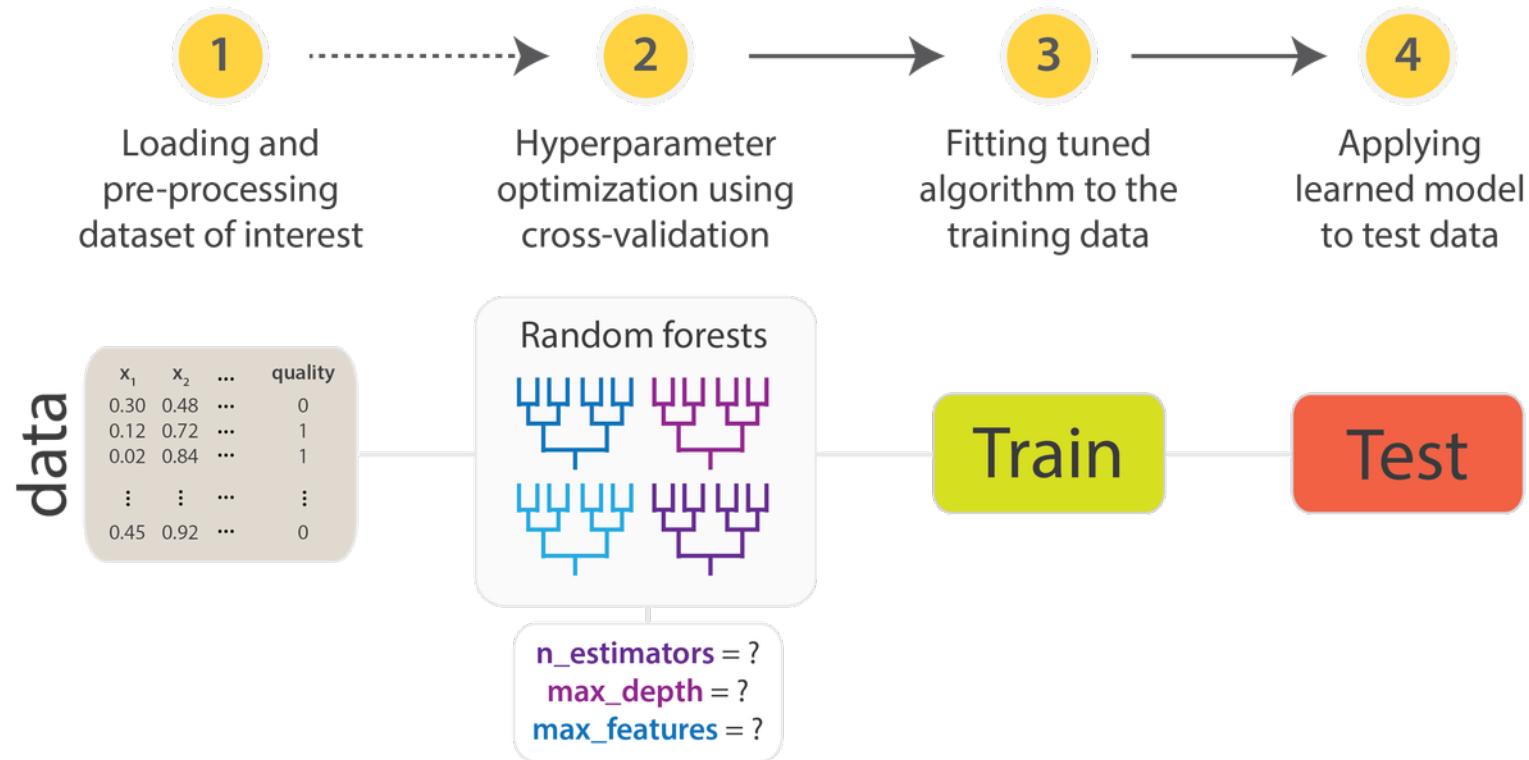
||

||

$$\hat{R}_{\text{emp}}(f) = \sum_i L(f(x_i), y_i)$$

$$R(f) = \int L(f(x), y) p(x, y) dx dy$$

Datos de Prueba (Test Set)



Conceptos Revisados

- Tipos de Tarea: Clasificación, regresión, otros.
- Tipos de "Experiencia" y Supervisión.
- Hipótesis.
- Función de Costo (Loss).
- Error de Predicción.
- Error de Entrenamiento y Overfitting.
- Error de Pruebas.