
CRITERIOS DE SELECCIÓN DE MODELOS

1

Criterios para elegir un modelo

- Dos decisiones fundamentales:
 - El tipo de **modelo** (árboles de decisión, redes neuronales, modelos probabilísticos, etc).
 - El **algoritmo** usado para construir o ajustar el modelo a partir de las instancias de entrenamiento.
 - Por ejemplo, hay múltiples formas de construir un árbol de decisión a partir de ejemplos; algo similar sucede con las redes neuronales o el resto de modelos.

2

1: Selección del modelo

3

Selección del modelo (I)

1. Capacidad de representación

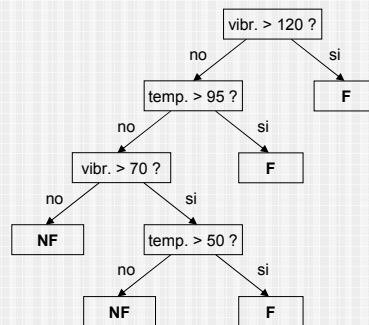
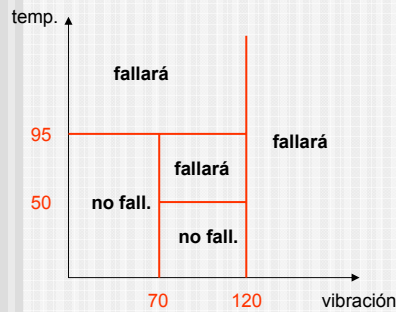
- Capacidad de expresar múltiples conceptos diferentes.
- Relacionado con el tipo de **fronteras de decisión** que se pueden crear.
- **Frontera de decisión**: frontera entre clases distintas de acuerdo con el modelo.
- Las fronteras de decisión que crea cada modelo (árboles de decisión, redes neuronales, etc.) son diferentes.

4

Selección del modelo (II)

■ Ejemplo con sólo dos atributos:

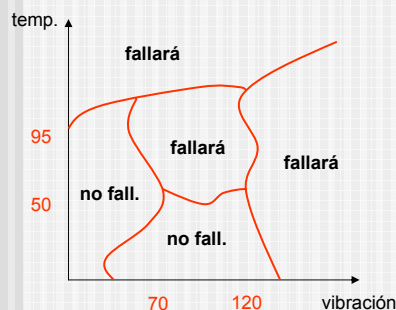
Árboles de decisión: fronteras perpendiculares a los ejes.



5

Selección del modelo (III)

■ Ejemplo con sólo dos atributos :



Redes neuronales (NN): fronteras no lineales:

- Mayor capacidad de representación.
- Permiten representar conceptos más complejos que los árboles de decisión.
- Se estudiarán más adelante.

6

Selección del modelo (IV)

2. Legibilidad:

- Capacidad de ser leído e interpretado por un humano.
- **Árboles de decisión**: fáciles de entender e interpretar: los niveles altos del árbol indican los atributos más importantes.
- **Redes neuronales**: difíciles (o imposibles) de interpretar: múltiples conexiones entre neuronas con pesos diferentes.

- Un modelo legible puede **ofrecer información** sobre el problema que se estudia (ej. indicar qué atributos afectan a la probabilidad de fallo de una máquina, y cómo).
- Un modelo no legible sólo puede ser usado como un **clasificador** (ej. Permite predecir si una máquina fallará o no aplicando el modelo).

7

Selección del modelo (V)

3. Tiempo de cómputo on-line:

- Tiempo necesario para clasificar una nueva instancia:
 - **Árboles de decisión**: tiempo necesario para recorrer el árbol, evaluando las funciones lógicas de cada nodo.
 - **Métodos probabilísticos**: tiempo necesario para calcular probabilidades o funciones de densidad de probabilidad.
 - **Redes neuronales**: tiempo necesario para realizar las operaciones (sumas, productos, sigmoides) incluidas en la red.
 - Etc.

8

Selección del modelo (VI)

Importancia del tiempo de cómputo **on-line**:

- Este tiempo se consume cada vez que se debe clasificar una nueva instancia.
- Algunas aplicaciones requieren clasificar miles de instancias.
 - Ejemplo: clasificación de cada uno de los pixels de una imagen aerea como tierra de cultivo, río, carretera, edificios, etc.
 - Es necesario clasificar millones de pixels.
 - El tiempo de cómputo es muy importante.

9

Criterios de selección de modelos

2: Selección del algoritmo

10

Selección del algoritmo (I)

1. Tiempo de cómputo off-line.

- Tiempo necesario para construir o ajustar el modelo a partir de los ejemplos de entrenamiento.
 - **Árboles de decisión**: tiempo necesario para elegir la estructura del árbol y los atributos a situar en cada uno de los nodos.
 - **Redes neuronales**: tiempo necesario para ajustar los pesos de las conexiones (se estudiará más adelante).
 - Etc.
- Ejemplo: un árbol de decisión se puede generar utilizando diferentes algoritmos. El tiempo empleado por cada algoritmo puede ser diferente.

11

Selección del algoritmo (II)

Importancia del tiempo de cómputo **off-line**.

- Sólo se consume una vez, cuando se han recopilado todos los ejemplos de entrenamiento y se genera el modelo con ellos.
- Dependiendo de la aplicación, no es un problema que el tiempo de cómputo on-line sea elevado (es aceptable tener un ordenador procesando durante un día entero para obtener el resultado).

12

Selección del algoritmo (III)

2. Dificultad de ajuste de parámetros.

- **Algoritmo ideal:** no dispone de parámetros para ajustar o es muy poco sensible a la modificación de los parámetros: es fácil generar el modelo (ejemplo: algoritmos de generación de árboles de decisión).
- **Mal algoritmo:** muchos parámetros para ajustar y gran sensibilidad a sus modificaciones: es difícil ajustar el modelo para obtener resultados óptimos (ejemplo: entrenamiento de redes neuronales).

13

Selección del algoritmo (IV)

3. Robustez ante instancias de entrenamiento ruidosas.

- Instancia de entrenamiento ruidosa: **etiquetada incorrectamente** (ejemplo: una máquina que no falló etiquetada incorrectamente como máquina que sí falló).
- Algunos algoritmos pueden funcionar adecuadamente aunque haya instancias ruidosas en el conjunto de entrenamiento (ejemplo: **árboles de decisión, redes neuronales**).
- Otros algoritmos no ofrecen buenos resultados (ejemplo: **vecino más cercano**).

14

Selección del algoritmo (V)

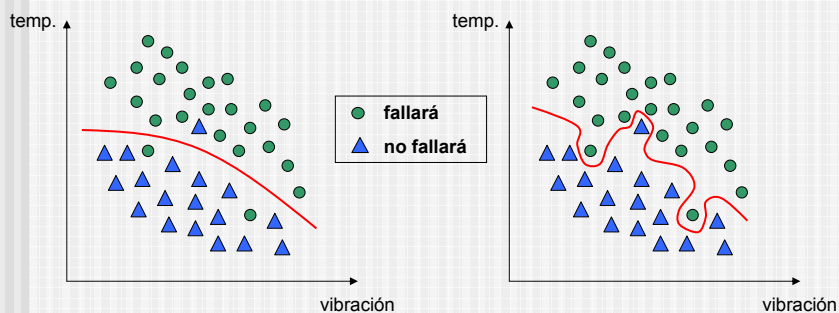
4. Sobreajuste (overfitting).

- Problema muy común.
- El modelo está demasiado ajustado a las instancias de entrenamiento, y no funciona adecuadamente con nuevas instancias.
- El modelo no es capaz de **generalizar**.
- Normalmente, fronteras de decisión muy complejas producen sobreajuste.

15

Selección del algoritmo (VI)

Ejemplo con dos atributos:



Frontera de decisión correcta:

- No consigue el 100% de clasificaciones correctas con los ejemplos de entrenamiento.
- Pero las clases están correctamente separadas.

Frontera de decisión sobreajustada:

- Consigue el 100% de clasificaciones correctas con los ejemplos de entrenamiento.
- Pero la frontera es artificial.

16

Resumen

Selección del modelo:

1. Capacidad de representación.
2. Legibilidad.
3. Tiempo de cómputo on-line.

Selección del algoritmo:

1. Tiempo de cómputo off-line.
2. Dificultad de ajuste de parámetros.
3. Robustez ante ejemplos de entrenamiento ruidosos.
4. Sobreajuste.

Algunos de los criterios anteriores están relacionados (ej. sobreajuste, robustez ante ejemplos de entrenamiento ruidosos).

17

CRITERIOS DE SELECCIÓN DE MODELOS

18