Aprendizaje Automático y Data Mining

Bloque II

APRENDIZAJE INDUCTIVO

Índice

- Conceptos básicos.
  - Concepto, instancia, atributo, clase.
- Árboles de decisión.
  - Estructura.
  - Generación automática.
- Otros modelos.
- Criterios de selección de modelos.
  - Selección de modelos.
  - Selección de algoritmos.
  - Resumen.

CONCEPTOS BÁSICOS

Objetivo

- El objetivo es general un modelo (general) a partir de ejemplos (específicos).
- El conjunto de ejemplos usado se llama conjunto de entrenamiento.
- Cuatro elementos fundamentales: conceptos, instancias, atributos y clases.

Definiciones

- Concepto: el modelo a inferir a partir de los ejemplos (también llamado hipótesis).
- Instancia: cada uno de los ejemplos.
- Atributo: cada una de las medidas de un ejemplo.
- Clase: el atributo que debe ser deducido a partir de los demás.

Ejemplo

Ejemplo: modelado de la probabilidad de fallo de una máquina.

- Clases: la máquina fallará / la máquina no fallará.
- Atributos: conjunto de medidas:
  - Temperatura.
  - Nivel de vibraciones.
  - Horas de funcionamiento.
  - Meses desde la última revisión.
- Instancias: ejemplos pasados (situaciones conocidas).
- Concepto: relación entre las medidas y la probabilidad de fallo:
  - *SI nivel_vibraciones = alto Y temperatura = alta ENTONCES fallará.*

Atributos

- Múltiples tipos de atributos:
- Real: puede tomar cualquier valor dentro de un cierto rango.
  - ej. temperatura como un número real (grados).
- Discreto: toma valores discretos ordenados.
  - ej. temperatura como {alta, media, baja}.
- Categórico: toma valores discretos no ordenados.
  - ej. color como {azul, rojo, amarillo}.

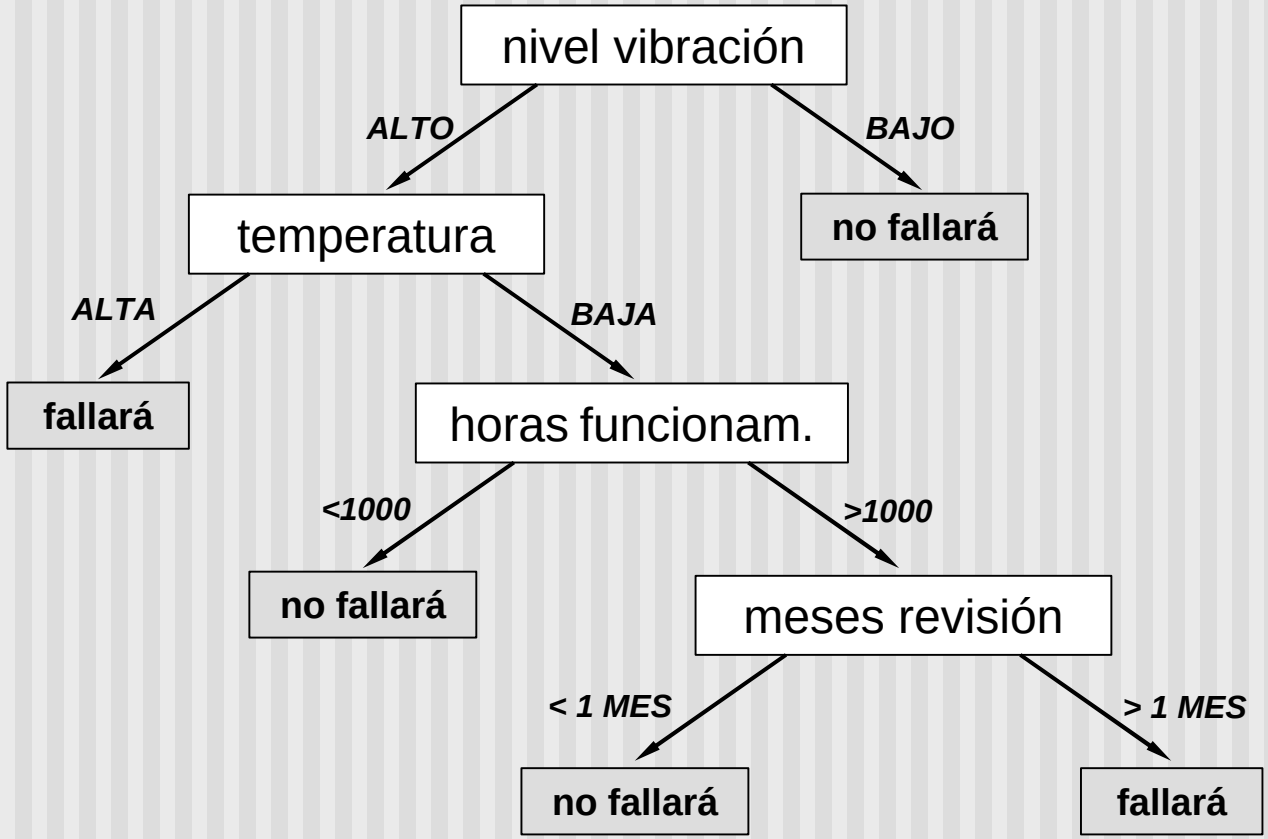
Conceptos

- Los conceptos se pueden expresar de diversas formas:
  - Árboles de decisión
  - Listas de reglas
  - Redes neuronales
  - Modelos bayesianos o probabilísticos
  - Etc.
- Los árboles de decisión son uno de los modelos más usados en aprendizaje automático.

ÁRBOLES DE DECISIÓN

Árboles de decisión (I)

- Ejemplo: modelado de la probabilidad de fallo de una máquina.



Árboles de decisión (II)

- Representan funciones lógicas (if-then).
- Compuestos de nodos y ramas.
- Nodos internos = atributos (medidas).
- Nodos hoja = clases.
- Nodo raíz = nodo superior del árbol.

- Objetivo en aprendizaje automático: inferir un árbol de decisión a partir de un conjunto de instancias o ejemplos.

Árboles de decisión (III)

- Ejemplo de conjunto de entrenamiento:

Temperatura	Nivel de vibraciones	Horas de funcionamiento	Meses desde revisión	Probabilidad de fallo
ALTA	ALTO	< 1000	> 1 MES	fallará
BAJA	BAJO	< 1000	< 1 MES	no fallará
ALTA	BAJO	>1000	> 1 MES	no fallará
ALTA	BAJO	< 1000	> 1 MES	no fallará
BAJA	ALTO	< 1000	> 1 MES	no fallará
BAJA	ALTO	>1000	> 1 MES	fallará
ALTA	ALTO	< 1000	< 1 MES	fallará

Árboles de decisión (IV)

- Múltiples formas de inferir el árbol:
  - Trivial: se crea una ruta del árbol por cada instancia de entrenamiento.
    - Árboles excesivamente grandes.
    - No funcionan bien con instancias nuevas.
  - Optimo: el árbol más pequeño posible compatible con todas las instancias.
    - Inviable computacionalmente.
  - Pseudo-optimo (heurístico): selección del atributo en cada nivel del árbol en función de la calidad de la división que produce.
    - Los principales programas de generación de árboles utilizan procedimientos similares (C4.5, CART, etc).

Árboles de decisión (V)

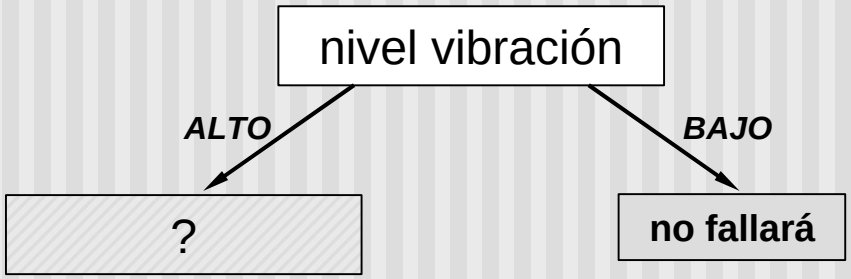
Crearemos un árbol a partir de los ejemplos de entrenamiento anteriores.

- ¿Qué atributo elegir para el primer nodo?

ATRIBUTO	VALORES	CLASE	
		fallará	*no fallará*
Temperatura	Alto	2	2
	Bajo	1	2
Nivel de vibraciones	Alto	3	1
	Bajo	0	3
Horas defuncionamiento	< 1000	2	3
	>1000	1	1
Meses desde revisión	> 1 mes	2	3
	< 1 mes	1	1

Árboles de decisión (VI)

- Árbol construido hasta el momento:



No fallará (1 instancia)

fallará (3 instancias)

no fallará (3 instancias)

fallará (0 instancias)

- ¿Qué atributo se debe usar en el siguiente nivel del árbol (rama izquierda)?

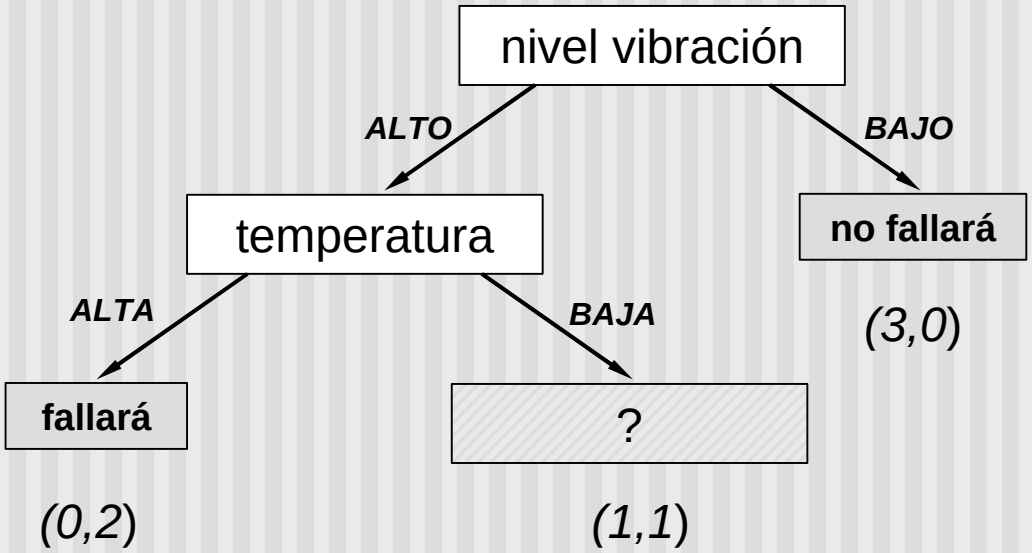
Árboles de decisión (VII)

Sólo aquellos ejemplos de entrenamiento que llegan al nodo se utilizan para elegir el nuevo atributo:

ATRIBUTO	VALORES	CLASE	
		fallará	*No fallará*
Temperatura	Alta	2	0
	BAja	1	1
Horas de funcionamiento	< 1000	2	1
	>1000	1	0
Meses desde revisión	> 1 mes	2	1
	< 1 mes	1	0

Árboles de decisión (VIII)

- Árbol construido hasta el momento:



- ¿Qué atributo se debe usar en el siguiente nivel del árbol (rama dereecha)?

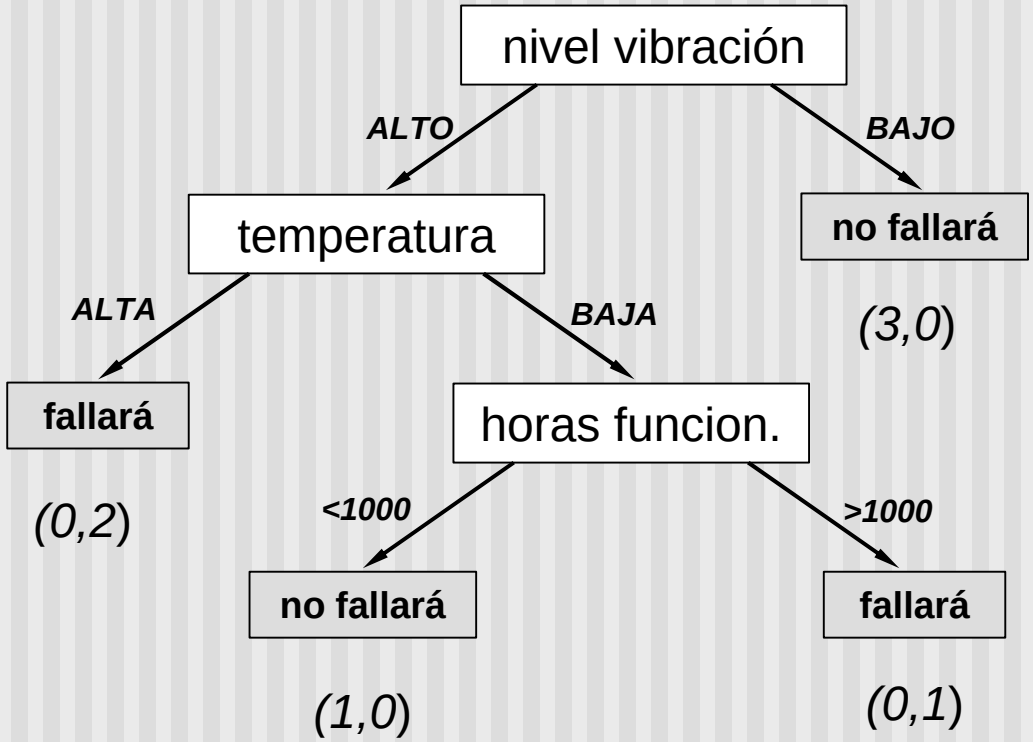
Árboles de decisión (IX)

De nuevo, sólo aquellos ejemplos de entrenamiento que llegan al nodo se utilizan para elegir el nuevo atributo:

ATRIBUTO	VALORES	CLASE	
		fails	*works*
Horas de funcionamiento	< 1000	0	1
	>1000	1	0
Meses desde revisión	> 1 mes	1	1
	< 1 mes	0	0

Árboles de decisión (X)

- Árbol obtenido finalmente:



... muy similar al árbol original, utilizando sólo 7 ejemplos de entrenamiento!

OTROS MODELOS

Otros modelos

- Los árboles de decisión son sólo uno de los posibles modelos.
- En los próximos apartados se explican otras posibilidades.
- Dependiendo de la aplicación, se deberá elegir un modelo u otro.
- A continuación se indican algunos criterios para elegir modelos.

CRITERIOS DE SELECCIÓN

Criterios para elegir un modelo

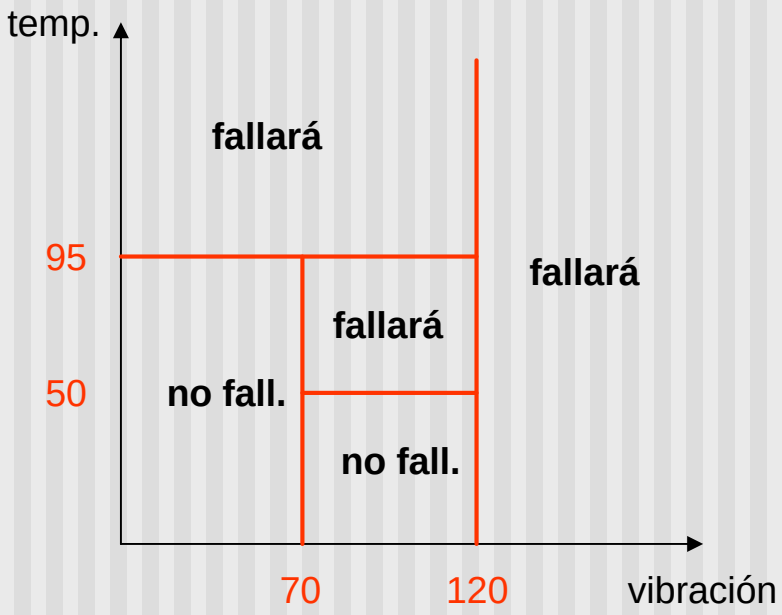
- Dos decisiones fundamentales:
  - El tipo de modelo (árboles de decisión, redes neuronales, modelos probabilísticos, etc).
  - El algoritmo usado para construir o ajustar el modelo a partir de las instancias de entrenamiento.
    - Por ejemplo, hay múltiples formas de construir un árbol de decisión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

Selección del modelo (I)

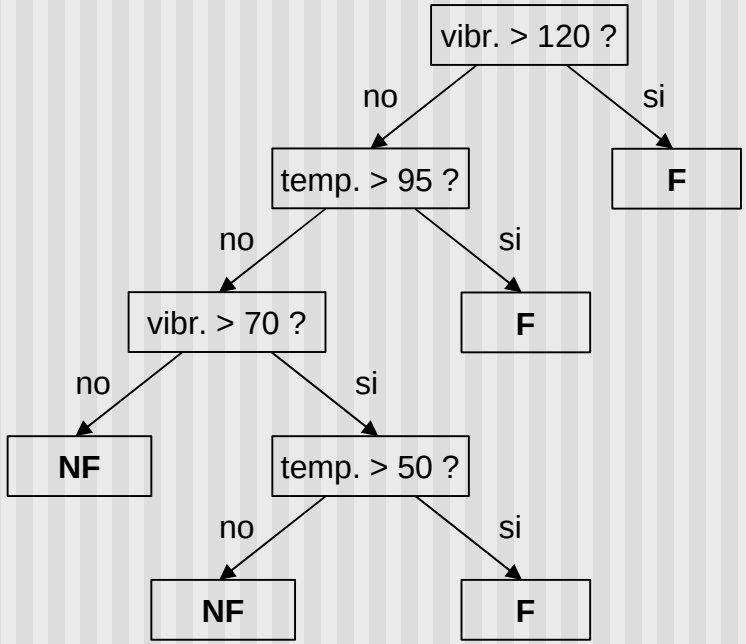
1. Capacidad de representación
   - Capacidad de expresar múltiples conceptos diferentes.
   - Relacionado con el tipo de fronteras de decisión que se pueden crear.
   - Frontera de decisión: frontera entre clases distintas de acuerdo con el modelo.
   - Las fronteras de decisión que crea cada modelo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

Selección del modelo (II)

- Ejemplo con sólo dos atributos:

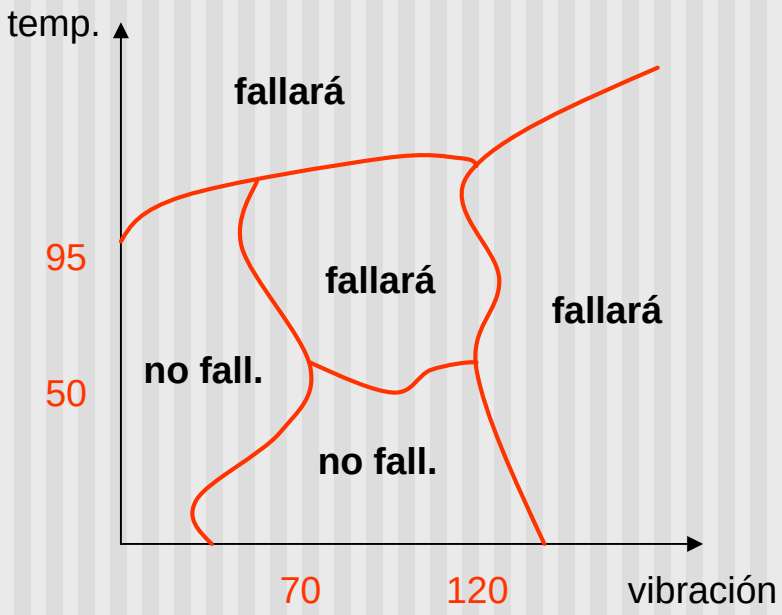


Árboles de decisión: fronteras perpendiculares a los ejes.



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.

Selección del modelo (IV)

2. Legibilidad:
   - Capacidad de ser leido e interpretado por un humano.
   - Árboles de decisión: fáciles de entender e intepretar: 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 clasifcador (ej. Permite predecir si una máquina fallará o no aplicando el modelo).

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.

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.

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 elegr 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.

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).

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 párámetros para ajustar y gran sensibilidad a sus modificaciones: es difícil ajustar el modelo para obtener resultados óptimos (ejemplo: entrenamiento de redes neuronales).

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).

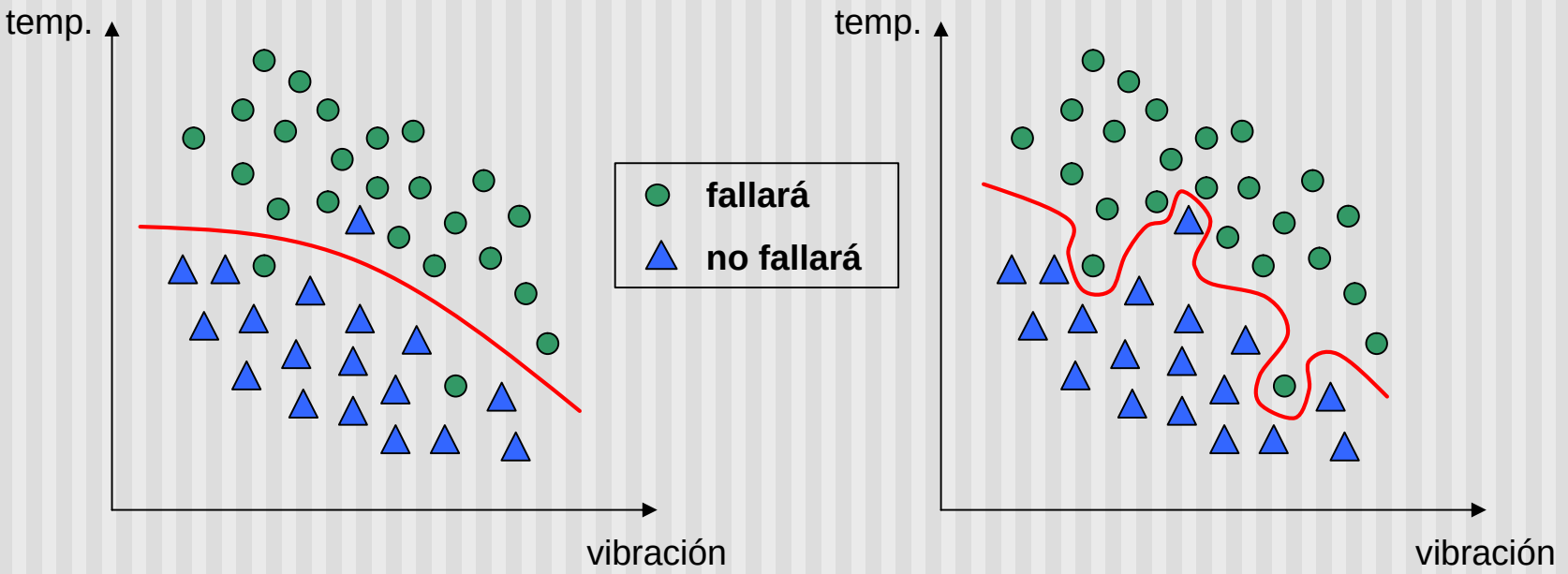
Selección del algoritmo (V)

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.

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.

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).