Machine Learning

Conjuntos de clasificadores

Christian Oliva Moya
Luis Fernando Lago Fernández

Introducción

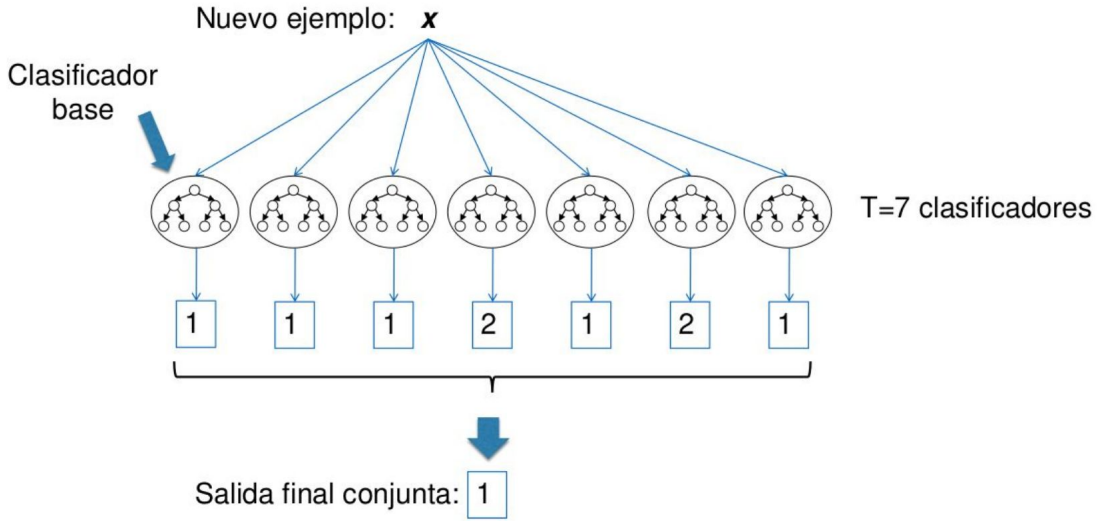
- Teorema del jurado
- ¿Qué es un conjunto de clasificadores?
  - Bagging
  - Boosting
  - Random Forest
  - Class Switching
  - Combinación de modelos

Introducción

- Teorema del jurado de Condorcet:
  - Dado un jurado (votantes) y suponiendo:
    - Errores independientes
    - La probabilidad de cada miembro del jurado de acertar es superior al 50%
  - La probabilidad del jurado en su conjunto tiende al 100% al aumentar el número de jueces

Introducción

- ¿Qué es un conjunto de clasificadores?
  - Una combinación de clasificadores que dan una salida final conjunta.
  - La predicción del conjunto es una combinación de las predicciones individuales.



Introducción

- Ventajas
  - Combinar la opinión de diferentes expertos puede mejorar el conjunto
  - Divide y vencerás: cada clasificador puede enfocarse en una parte del problema
  - No hay modelos universales: cada modelo es bueno para algunos problemas y malo para otros
  - Un conjunto es más robusto, ya que en general funciona mejor que cualquiera de los individuos que lo componen

Introducción

- Requisitos
  - Diversidad: los clasificadores deben ser distintos o no cometer los mismos errores
  - Precisión: los clasificadores deben ser suficientemente precisos individualmente

Introducción

- Ejemplo

	Clasificador 1	**Clasificador 2**	**Clasificador 3**	**Conjunto**
X1	OK	OK	FAIL	⅔ OK → OK
X2	OK	FAIL	OK	⅔ OK → OK
X3	FAIL	OK	OK	⅔ OK → OK
X4	OK	OK	OK	OK
Accuracy	75%	75%	75%	100%

Introducción

- ¿Por qué funcionan?
- Si hay muchas hipótesis compatibles con los datos…
  - Combinar varios clasificadores reduce el riesgo de elegir uno malo
- Si hay muchos mínimos locales y el algoritmo no puede alcanzar la solución óptima…
  - La combinación de varios clasificadores puede aliviar este problema
- Si la solución está fuera del espacio de hipótesis…
  - La combinación puede generar soluciones fuera del espacio de hipótesis de los clasificadores

Introducción

- ¿Por qué funcionan?
- Si hay muchas hipótesis compatibles con los datos…
  - Combinar varios clasificadores reduce el riesgo de elegir uno malo
- Si hay muchos mínimos locales y el algoritmo … la solución óptima…
  - …
- Si l…
  - … clasificadores

Tengo 3 modelos distintos, que a veces funcionan bien y a veces no:

Naive Bayes
Árbol de decisión
KNN

Combinar los tres hará que, si uno se equivoca, se ignore el error

Introducción

- ¿Por qué func

Entrenar con los mismos conjuntos de datos puede llevar a sesgos.
Si cada modelo se especializa en un subconjunto del dataset, la combinación puede alcanzar una mejor solución.

- Si hay mucha

  - Combinar varios clasi… …ce el riesgo de elegir uno malo

- Si hay muchos mínimos locales y el algoritmo no puede alcanzar la solución óptima…

  - La combinación de varios clasificadores puede aliviar este problema

- Si la so

  - L… cl…

Un árbol A1 con profundidad 1 se entrena con el dataset. Se equivoca en X.

Otro árbol A2 con profundidad 1 se entrena con un dataset forzando a que aprenda correctamente X. Este falla en Y.

Otro árbol A3 con profundidad 1 se entrena con el dataset forzando a que aprenda correctamente Y.

Introducción

- ¿Por qué f
- Si hay m
  - Co
- Si hay m óptima…
  - La c

Quiero clasificar perros y gatos

- El modelo M1 identifica perros y gatos y se confunde con los gatos naranjas, cree que son perros
- Un segundo modelo M2 se entrena para decir que los gatos naranjas son gatos, no perros
- El conjunto sabe identificar perros, gatos y gatos naranjas

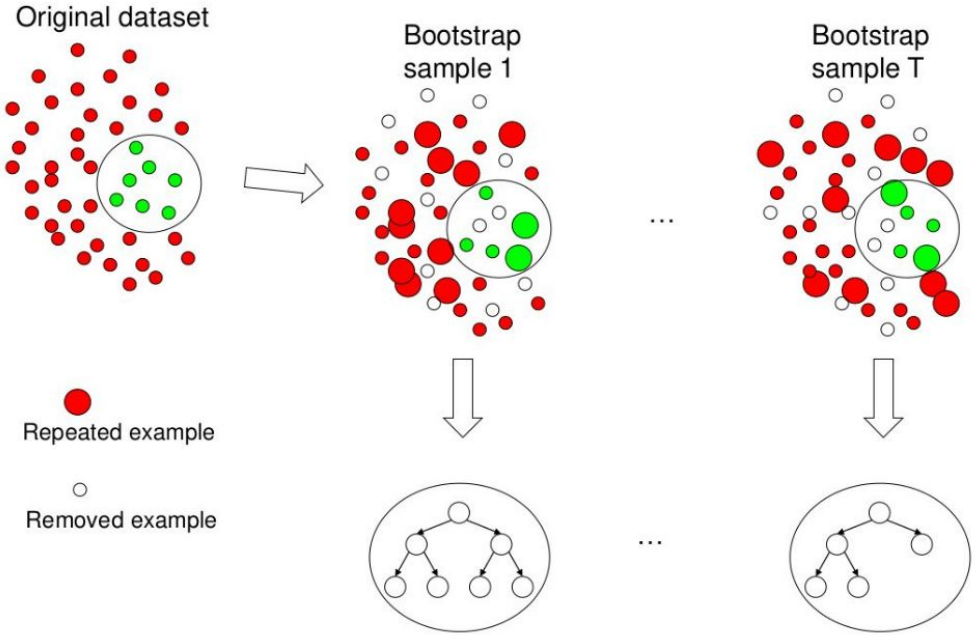
- Si la solución está fuera del espacio de hipótesis…
  - La combinación puede generar soluciones fuera del espacio de hipótesis de los clasificadores

Bagging

Muestreo con reemplazo

Bagging

- Cada clasificador se entrena con un muestreo aleatorio con reemplazo del mismo tamaño que el original.

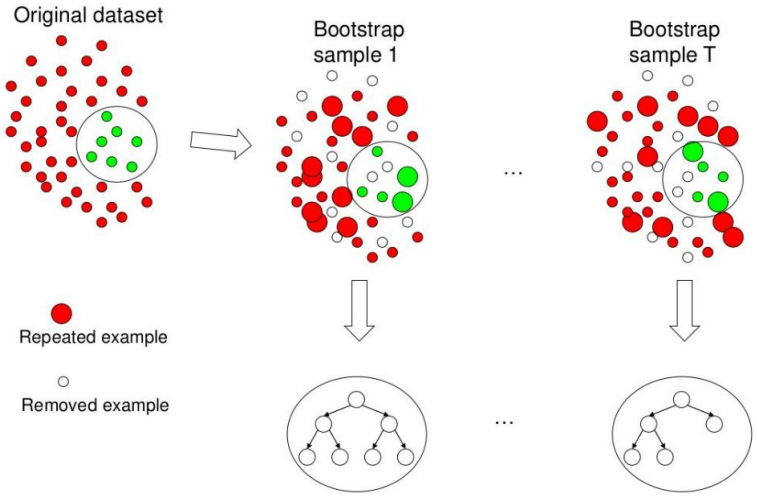


Bagging

- Construcción del dataset y entrenamiento:
  - Repetir tantas veces como número de clasificadores:
    - Muestreo aleatorio D_i con reemplazo (random choice con replace=True)
    - Entrenar el modelo M_i con el dataset D_i

Bagging

- Ventajas
  - Muy robusto frente a errores de etiquetado
  - Se puede paralelizar con facilidad
  - En principio, es robusto frente al overfitting

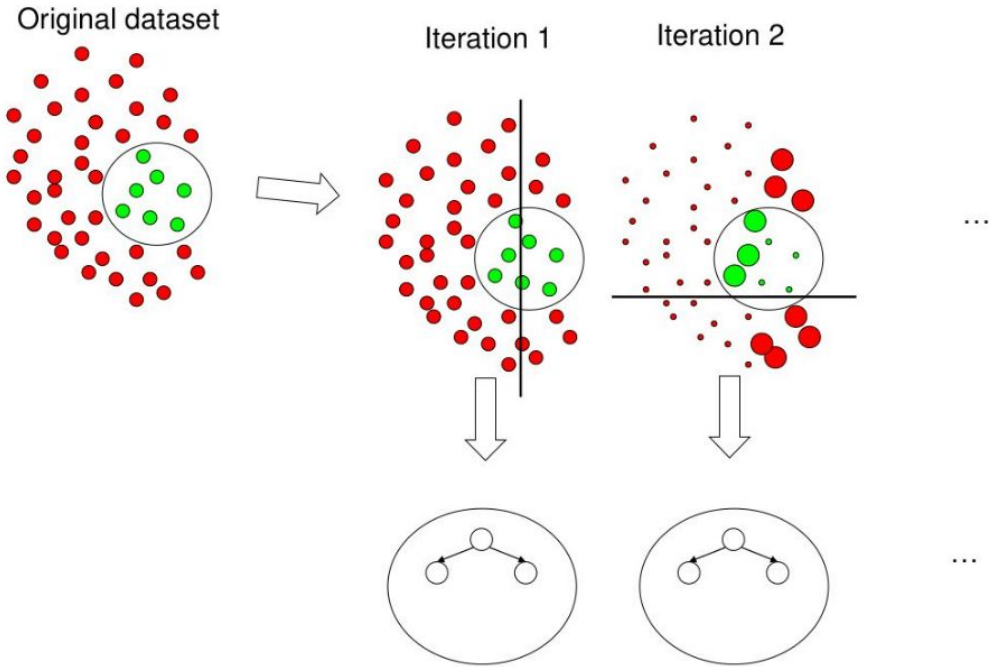


Boosting

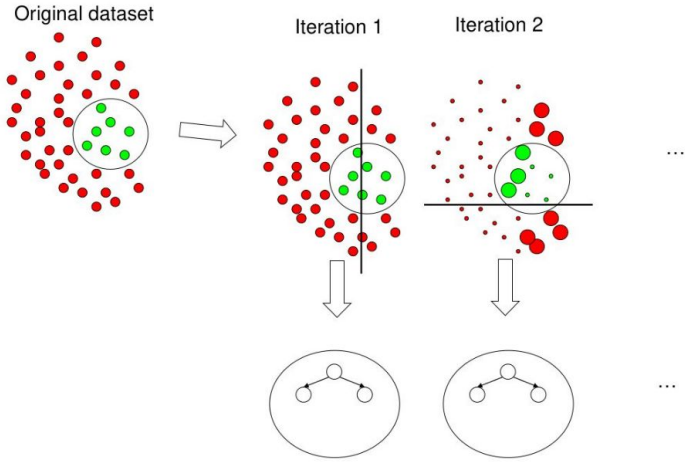
Muestreo con pesos

Boosting

- Cada clasificador añadido al conjunto intenta mejorar los errores de los anteriores



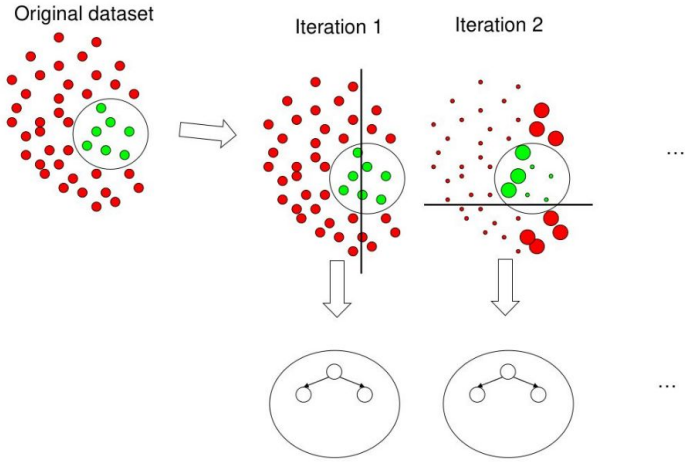
Boosting



- Construcción del dataset y entrenamiento:
  - Inicializar un vector de pesos **w** equiponderado determina la probabilidad de ser elegido
  - Para cada iteración hasta tener N modelos:
    - Muestreo aleatorio ponderado D_i con reemplazo (random choice con replace=True y p=**w**)
    - Entrenar el modelo M_i con el dataset D_i
    - El peso **w** aumenta para los datos mal clasificados
    - El peso **w** disminuye para los datos bien clasificados

Boosting

- Ventajas
  - Muy buen rendimiento frente al overfitting
  - Es muy sensible a outliers
  - No es fácil paralelizar, ya que es secuencial

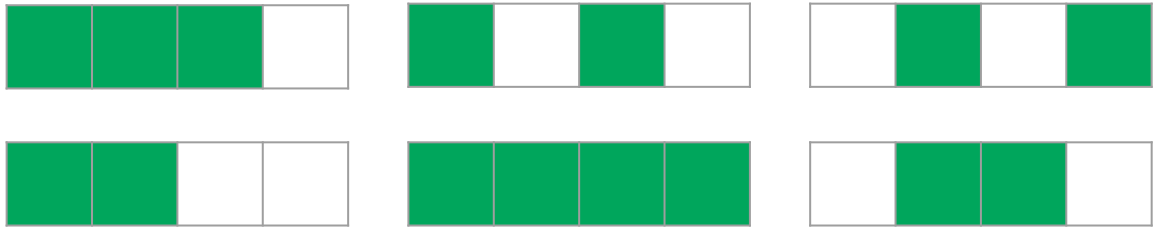


Random Forest

Bagging y selección de atributos

Random Forest

- Cada clasificador se entrena con un subconjunto de los atributos usando bagging
- Se compone de árboles que realizan de forma aleatoria una selección de atributos
- Normalmente no se podan



Random Forest

- Construcción del dataset y entrenamiento:
  - Repetir tantas veces como número de clasificadores:
    - Muestreo aleatorio D_i con reemplazo (random choice con replace=True)
    - Selección aleatoria de atributos (columnas) para generar D_i'
    - Entrenar el modelo M_i con el dataset D_i'

Es necesario guardar qué atributos utiliza cada modelo M_i para la evaluación

Random Forest

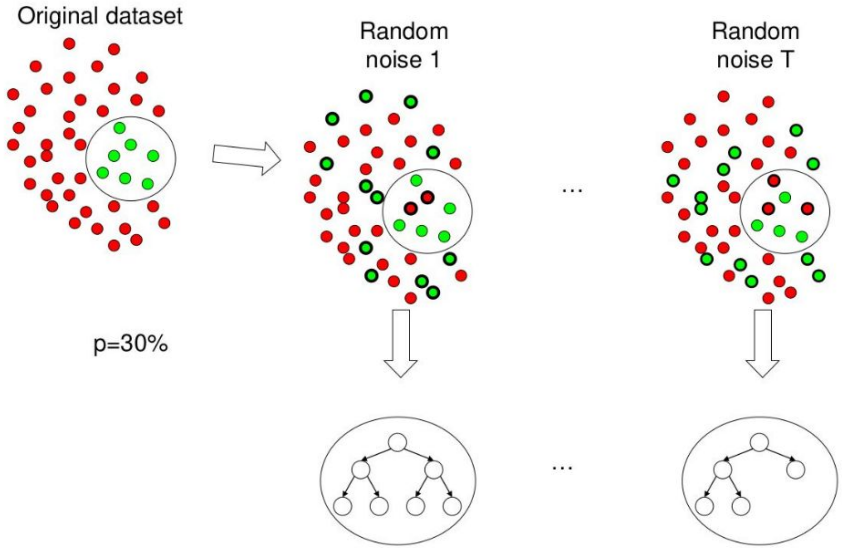
- Ventajas
  - Mayor grado de aleatoriedad que bagging al seleccionar atributos
  - Mayor rendimiento en la mayoría de los casos que bagging y boosting
  - Muy robusto frente a datos ruidosos
  - Tiene poco overfitting
  - Es fácilmente paralelizable
  - Se suele entrenar con modelos muy sencillos (árboles de decisión)

Class Switching

Manipulación de las clases

Class Switching

- Modificación aleatoria de las clases:
  - La diversidad se consigue modificando las etiquetas de los datos de entrenamiento
  - Para cada clasificador, se alteran los datos con una probabilidad P



Class Switching

- Construcción del dataset y entrenamiento:
  - Repetir tantas veces como número de clasificadores:
    - Selección aleatoria de targets Y_i a modificar con probabilidad P
    - Entrenar el modelo M_i con el dataset D para aprender la clase Y_i

Consideraciones finales

- La estrategia de decisión está basada en:
  - Voto simple (absoluto)
    - Si la predicción del modelo individual es mayor que 0.5, entonces suma 1
  - Voto ponderado (relativo)
    - Utiliza directamente la probabilidad de predicción de cada modelo

Ejemplo:	0.1	0.6	0.6
	Voto simple:	2 / 3 =	0.67
	Voto ponderado:	1.3 / 3.0 =	0.43

Consideraciones finales

- Aspectos positivos de la clasificación mediante conjuntos:
  - Buen rendimiento
  - Pocos parámetros que ajustar
  - Mejora la generalización reduciendo el overfitting
- Aspectos negativos de la clasificación mediante conjuntos:
  - Son más lentos que un clasificador simple
  - Puede no ser una buena opción para modelos más complejos
  - Se pierde la interpretabilidad de los modelos individuales