Modelos Estadísticos para Clasificación y Regresión

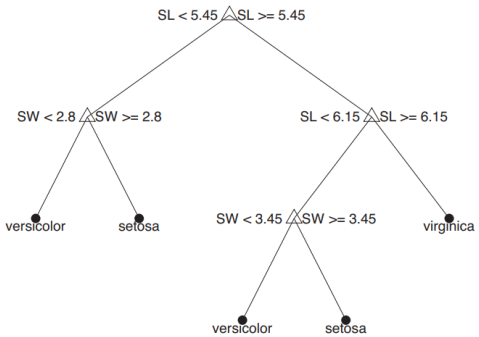
Random Forest y AdaBoost

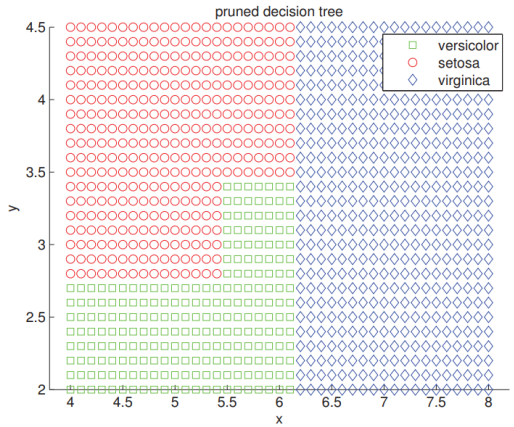
IMERL - FIng

1 de noviembre de 2023

Árboles de Decisión

Los árboles de decisión permiten particionar el espacio en regiones simples. El nombre proviene de que las reglas que se usan para segmentar el espacio se pueden resumir en un árbol.





Una vez entrenado el árbol, se predice la clase de un nuevo dato x_{nuevo} a partir de la clase mayoritaria de los datos de entrenamiento que caen en la misma hoja que x_{nuevo}.[1]

[1] Imagen extraída de [3].

Problemas de los árboles

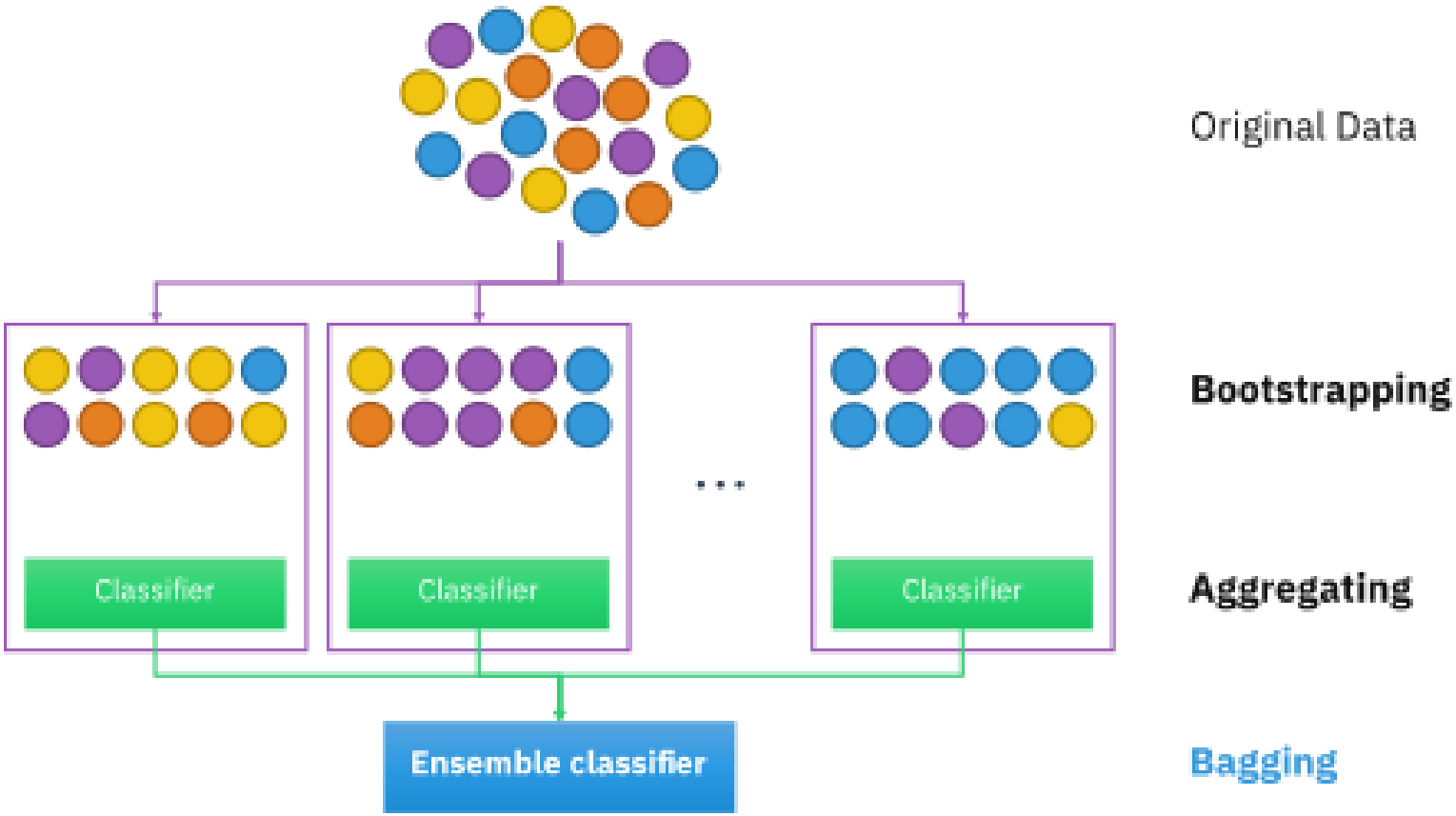
Un problema de los árboles es que son inestables: pequeños cambios en los datos de entrenamiento pueden tener grandes cambios en la estructura del árbol.
Decimos que los árboles son estimadores de alta varianza.
[3]

Bagging

Una manera de reducir la varianza de una estimación es promediar muchas estimaciones. Por ejemplo, podemos entrenar M árboles diferentes en subconjuntos de datos, elegidos aleatoriamente con reemplazo, y luego elegir como predicción la clase más votada.

Problema: suele ocurrir que las predicciones de los diferentes árboles están fuertemente correlacionadas, lo cual resulta en una varianza no mucho menor a la de un único árbol.
[2] [3]

Bagging



2

[2] Imagen extraída de *Wikipedia*

Random Forest

Para evitar esto, en *Random Forest* se realiza *feature sampling*: el atributo con el que se va a hacer el split en un nodo de un árbol se elige de un subconjunto de todos los atributos.

Es decir, los árboles son entrenados independientemente en muestras aleatorias de los datos, pero cada split en cada árbol se realiza usando un subconjunto aleatorio de atributos. Con esto se logra decorrelacionar los árboles, y se realiza una exploración más exhaustiva de los datos con respecto a *bagging*, ya que permite que se consideren atributos que no serían elegidos originalmente para los splits.
[2] [3]

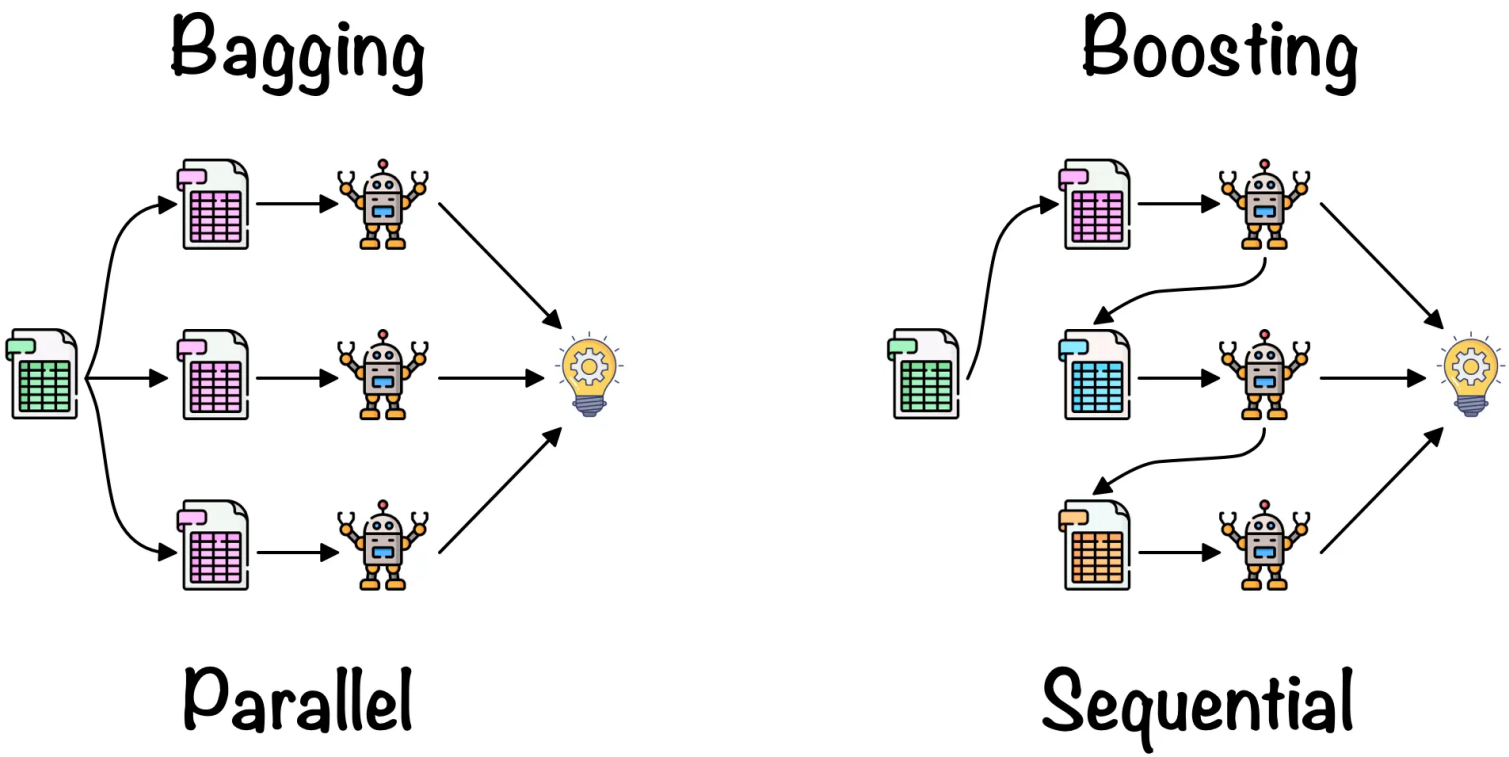
Boosting

Se entrena un gran número de árboles, pero en vez de entrenarlos en paralelo como en *bagging*, se los entrena **secuencialmente**: cada árbol se construye utilizando información de los árboles entrenados previamente.

Cada árbol se entrena utilizando una versión modificada del dataset original, que tiene en cuenta los datos que fueron "difíciles"para los árboles anteriores. Así, el i-ésimo árbol intentará clasificar bien a las muestras en las que los árboles $1, \ldots, i-1$ fallaron.

La predicción final se realiza mediante un voto ponderado de las predicciones individuales de cada clasificador.

Boosting



3

[3] Imagen extraída de *Towards Data Science*

AdaBoost

AdaBoost[1] fue la primera implementación práctica exitosa de la idea de boosting.

En cada iteración del entrenamiento, se asigna un peso a cada dato, que tiene en cuenta qué tan mal lo han clasificado los árboles anteriores.
Puede encontrarse una implementación del clasificador en sklearn.ensemble.AdaBoostClassifier.

Preguntas?

Referencias

[1] Yoav Freund y Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". En: *Computational Learning Theory*. Ed. por Paul Vitányi. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, págs. 23-37. ISBN: 978-3-540-49195-8.

[2] Gareth James et al. *An Introduction to Statistical Learning*. Springer International Publishing, 2023. DOI: 10.1007/978-3-031-38747-0. URL: https://doi.org/10.1007/978-3-031-38747-0.

[3] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013. ISBN: 9780262018029 0262018020.