Mauricio Luque Jiménez maulujim@correo.ugr.es 78004003D

Comparación entre XGBoost y LightGBM

Tanto XGBoost como LightGBM están basados en Gradient Boosting, están diseñados para generar árboles de decisión y son muy usados actualmente por su eficacia y eficiencia. Pero, ¿en qué se parecen realmente?¿En qué se diferencian?

Empezando por la primera de las características, es interesante analizar cómo está estructurado cada algoritmo para ser lo más eficaz posible, partiendo de que ambos son muy parametrizables. En primer lugar, XGBoost es un algoritmo basado en obtener la mayor ganancia de información en cada iteración, como menciona Esri[1], empresa que aplica XGBoost en uno de sus productos. Además, tiene una estrategia de crecimiento de árboles en profundidad, lo cual puede llevar a árboles más interpretables, equilibrados y estables. Por otra parte, para evitar sobreajuste, incluye uso de regularización para reducir la sensibilidad a datos individuales y aumentar la generalización del modelo. Mientras tanto, la documentación oficial de LighGBM[2] destaca que éste es un algoritmo basado en histogramas: agrupa valores de atributos continuos en subconjunto de características discretos para agilizar el entrenamiento y el uso de memoria. Agregado a lo anterior, a diferencia de XGBoost, tiene una estrategia de crecimiento de árboles en hojas, que, si bien puede suponer modelos menos estables, pueden ser más profundos y precisos.

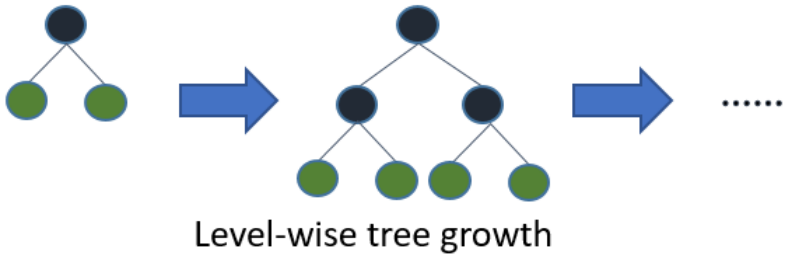

Figura 1: Ejemplo de estructura de árbol de decisión aplicando XGBoost, con crecimiento en profundidad

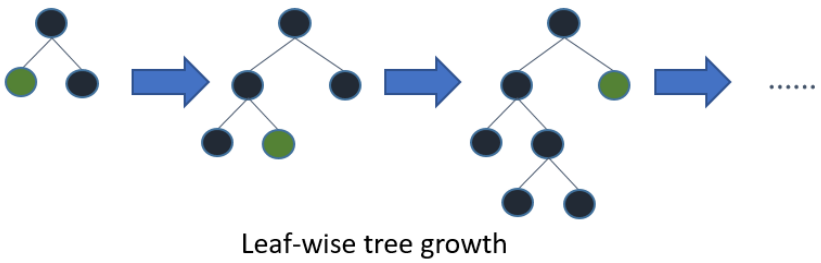

Figura 2: Ejemplo de estructura de árbol de decisión aplicando LightGBM, con crecimiento en hoja

Pasando a la segunda de las características, podemos destacar que ambos algoritmos resultan bastante eficientes (ambos permiten computación paralela tanto en CPU como en GPU). Por un lado, según un artículo[3] del portal *Medium*, "XGBoost destaca por un uso eficiente de la memoria aprovechando toda la potencia de procesamiento de los computadores multinúcleos de la actualidad, pero sobre todo por la computación paralela en GPU y en clúster de computadoras, lo que hace factible el entrenar modelos en conjuntos de datos muy grandes del orden de millones de muestras de entrenamiento." Además, según IBM[4], "cuenta con un algoritmo de precarga con reconocimiento de caché que ayuda a reducir el tiempo de ejecución para conjuntos de datos grandes". Por otro lado, LightGBM, según su propia documentación, lleva a cabo una fragmentación que consigue que cada trabajador se centre en un subconjunto de características sin necesidad de intercambiar información con otros trabajadores, ahorrando costes de comunicación. De esta manera, es un algoritmo más rápido en entrenamiento, pero más lento en tests debido a la estructura generada en el árbol de decisión. No obstante, si bien ambos algoritmos destacan por su eficiencia, cabe destacar que, en la fase de entrenamiento, LightGBM suele rendir notoriamente por encima de XGBboost, especialmente para conjuntos de datos sensiblemente grandes.

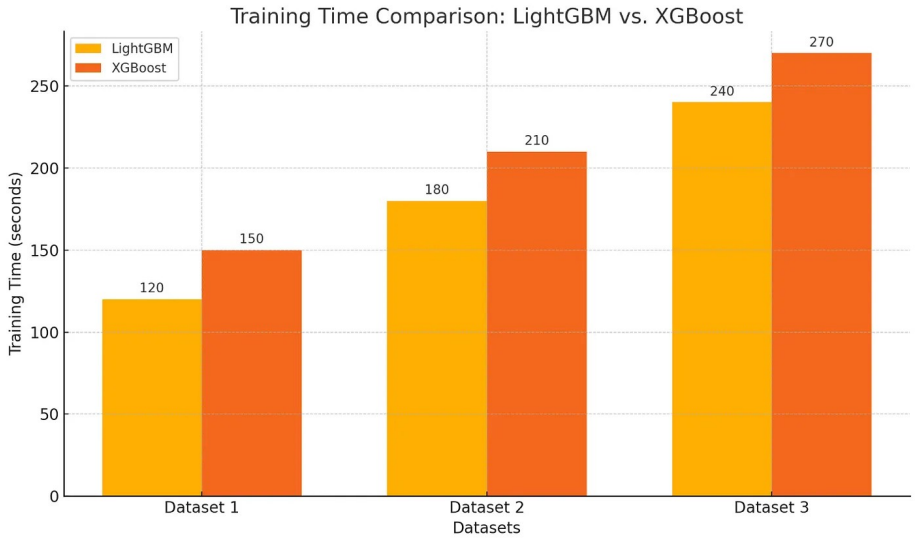

Figura 3[5]: Comparación de tiempos de entrenamiento de ambos algoritmos para distintos tamaños de conjuntos de entrenamiento

En conclusión, ambos algoritmos destacan por generar modelos precisos y eficientes, si bien XGBoost está más centrado en la ganancia de información, la interpretabilidad y estabilidad del modelo generado y el procesamiento multinúcleo, mientras que LightGBM está orientado a histogramas, la precisión del modelo y la eficiencia en memoria. Algoritmos compatibles a la vez que distintos, de manera que la preferencia por uno u otro debe depender del problema a abordar y de sus características.

Bibliografía

- [1] Funcionamiento de XGBoost, página oficial de Esri: https://pro.arcgis.com/es/pro-app/latest/tool-reference/geoai/how-xgboost-works.htm
- [2] Documentación oficial de LighGBM: https://lightgbm.readthedocs.io/en/latest/Features.html
- [3] Computación paralela en XGBoost, Medium: https://medium.com/@oscars.cortezmo/introducci%C3%B3n-a-los-m%C3%A9todos-de-ensamble-y-al-algoritmo-de-xgboost-caso-pr%C3%A1ctico-e8cb0d58394b#26cb
- [4] Características de XGBoost, IBM: https://www.ibm.com/es-es/topics/xgboost#f2
- [5] Comparativa de rendimiento en entrenamiento, Medium: https://medium.com/@srijan.ramavat/lightgbm-vs-xgboost-a-comparative-analysis-688e9b2684ed

Otros artículos consultados

- https://xgboost.readthedocs.io/en/stable/parameter.html
- https://xgboost.readthedocs.io/en/stable/tutorials/input_format.html
- https://www.themachinelearners.com/xgboost-python/
- https://medium.com/@jboscomendoza/tutorial-xgboost-en-python-53e48fc58f73
- https://lightgbm.readthedocs.io/en/latest/Parameters.html
- https://medium.com/@data-overload/comparing-xgboost-and-lightgbm-a-comprehensive-analysis-9b80b7b0079b
- https://neptune.ai/blog/xgboost-vs-lightgbm
- https://dataheadhunters.com/academy/xgboost-vs-lightgbm-gradient-boosting-in-the-spotlight/