ML Under Modern Optimization Lens - Random Forest - Exercícios

Giovanni Amorim

Junho, 2023

1 Random Forest

O exercício consiste na aplicação do método de validação cruzada para tuning de hiper-parâmetros do algoritmo Random Forest implementado na biblioteca "DecisionTree.jl". A avaliação foi feita no dataset "adult", com a tarefa de previsão de classe de renda de indíviduos civis baseado em características individuais. A base original possui 32.561 amostras, porém apenas 5.000 amostras foram consideradas no exercício, para reduzir a complexidade computacional.

Alguns hiperparâmetros do algoritmo foram mantidos fixos para a avaliação:

1. partial_sampling: Porção dos dados que serão sorteados e utilizados para treinar cada árvore individual. (0.7)
2. min_samples_split: Número mínimo necessário de amostras para uma divisão de galho. (2)
3. min_purity_increase: Pureza (Gini index) mínima necessária para uma divisão de galho. (0.0)

Outros hiperparâmetros foram avaliados em múltiplas opções:

1. n_subfeatures: Número de features a serem sorteadas e consideradas para o cálculo de cada split. $[2, 6, 12, 14]$
2. n_trees: Número de árvores a serem treinadas para formar a floresta. $[10, 15, 20]$
3. max_depth: Profundidade máxima de uma árvore individual. $[5, 10, 20]$
4. min_samples_leaf: Número mínimo necessário de amostras em uma folha. $[5, 10, 15]$

Foi feita uma iteração por todas as (108) possíveis combinações dos hiperparâmetros variáveis, em que cada iteração representa uma validação cruzada em 3 divisões do conjunto de dados. Em cada etapa da validação cruzada, o algoritmo é treinado, ou seja, as árvores são geradas calculando melhores splits iterativamente, considerando 2 partes em 3 da base e a métrica de acurácia é calculada levando em conta a 3ª parte (de validação). Ao fim de cada iteração, se calcula a acurácia média dos 3 "folds" para avaliação daquela combinação de hiperparâmetros.

Os resultados finais mostraram uma equivalência de performance entre algumas combinações, como era de se esperar dado a quantidade de combinações avaliadas, porém a combinação de hiperparâmetros de melhor performance teve uma performance diferente de todos (mesmo que próxima do segundo lugar). Os melhores e piores 5 conjuntos avaliados são mostrados na tabela abaixo:

n_subfeatures	n_trees	max_depth	min_samples_leaf	avg_acc
14	20	10	5	85.014
6	20	10	10	84.974
14	15	10	5	84.954
6	20	10	5	84.733
12	20	10	5	84.673
2	10	5	15	81.952
2	20	5	10	81.932
2	15	5	5	81.892
2	20	5	5	81.812
2	10	5	5	81.192

Analisando os resultados, podemos tirar algumas conclusões, por mais que seja necessário avaliações mais profundas para afirmações mais generalizáveis:

1. Um número muito baixo de features no cálculo de cada split parece afetar negativamente a performance do modelo;
2. A profundidade de árvore 10 parece ser a mais adequada entre as avaliadas para a tarefa;

Finalmente, o melhor conjunto de hiperparâmetros foi utilizado para avaliar a parformance do algoritmo de árvore de decisão única na mesma tarefa de classificação. Vale notar que nem todos os hiperparâmetros do algoritmo random forest são aplicáveis na árvore de decisão, então foram aproveitados os seguintes:

1. max_depth
2. min_samples_leaf
3. min_samples_split
4. min_purity_increase

O resultado final foi de uma acurácia média igual a 82.953, indicando que o algoritmo random forest tem uma performance significativamente melhor na tarefa de classificação avaliada.