Aprendizado de Máquina para Identificação de Espécie de Árvore

Marcos Paulo Diniz
Universidade de Brasília
Departamento de Ciência da Computação
Brasília, Brasil
marcosdiniz@aluno.unb.br

***Resumo*—O objetivo do experimento é medir a capacidade de um algoritmo aprender a classificar qual espécie de árvore se trata com base em suas folhas. Para isso foi-se usado a ideia de Floresta Randômica para fazer o treinamento do classificador.**

***Index Terms*—Árvoes,Floresta Randômica, aprender, classificador.**

I. Introdução

Esse experimento tem como objetivo usar o aprendizado de maquina supervisionado, por meio do algoritmo de Floresta Randômica, para a identificação de espécies de árvores a partir de folhas. No *dataset* utilizado haviam 340 dados de folhas de 36 espécies diferentes com 14 *features* diferentes.

Para a implementção do algoritmo foi-se usado a linguagem de programaçãao Python (versão 3.6.3), com auxílio das bibliotecas *Sklearn, Pandas e Numpy*.

O algoritmo de Florestas Randômicas é uma técnica de aprendizado por *ensemble*, usada aqui para classificação que consiste na criação de multiplas árvores de decisão durante o processo de treino, usadas para a decisão da classe de um dado durante o teste. É baseado na aplicação do algoritmo de agregacão por *bootstraping* no aprendizado por arvores, nesse, também, conhecido como *bagging*.

Assim, são gerados novos conjuntos de treino a partir de um original pelo *sampling* deste de forma uniforma e com substitutos. Esses novos dados de treino são então usados para o voto de um dado a ser classificado.

A tecnica de Florestas Randomicas adiciona a criaçãao de arvores pelo algoritmo de *bootstraping* um novo processo, no qual as árvores passam por um processo de averaging, visando descorrelaciona-las. Isso reduz o custo computacional do algoritmo e reduz a variância presente no set de treino, evitando a ocorrencia de *overfitting*.

Isso também faz com que 2 arvores criadas no mesmo conjunto de treino tenham variáveis diferentes escolhidas ao acaso em cada divisão e ainda torna possível adicionar novas árvores sem o aumento de variancia esperado, uma vez que estas vão passar pelo processo de média que acaba por impedir este aumento.

A seguir tem-se uma imagem ilustrando como se funciona a Floresta Randômica, suas árvores e o como o voto majoritário resulta na classificação de um element0, de acordo com as *features* desse elemento.

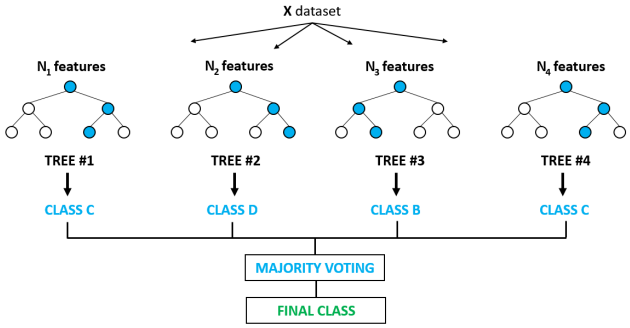

Figura 1. Floresta Randômica

Para mensurar o bom funcionamento do algoritmo proposto, foi-se usado a técnica da *Cross Validation*, que consiste dividir o mesmo *dataset* em diferentes amostras para treino e para teste (validação). O *dataset* foi dividido primeiramente em 80% para treino e 20% para testes e em seguida em 90% para treino do algoritmo e os 10% restantes para testes. O procedimento foi repitido 10 vezes para se aplicar a validação curzada. Além disso, também foram usados diferentes tamanhos de ávores (100, 200, 500 e 1000).

II. Analise do Experimento

Foi feita a aplicação da *Cross Validation* de duas formas: a primeira foi gerando grupos aleatórios de treino e teste e a segunda foi separando o conjunto de dados de 34 em 34 de forma sequencial.

No primeiro caso, foram usadas florestas de 100, 200, 500, 1000 e 1500 árvores, separando o conjunto em 80% de teste e 20% de treinamento, em seguida alterou-se o valor para 90% de teste e 10% de treino.

Ao analisar os resultados obtidos para a divisão 80/20, foi observado uma crescente positiva nos resultados com o aumento do número de árvores até quando usamos 500 árvores. Ao passar de 500 para 1000 árvores, foi notável o decaimento da acurácia, porém em seguida, com o aumento para 1500, foi percebido novamente um aumento. Por fim, foi usado 2000 árvores, para ver como que a floresta se comportaria num valor acima de 1500, e foi percebido que a

crescente notada anteriormente, teve fim e o valore de acurácia começou a ser reduzido.

Com isso, supõe-se que o valor ideal de árvores, provavelmente, se encontra entre 500 e 1500 árvores, já que com o aumento além disso, foi percebida a queda significativa entre o valor de acurácia medido, enquanto um valor entre esses teve uma queda, porém não tão significativa quando comparado ao valor medido em valores abaixo de 500 ou acima de 1500.

Tabela I
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 80% PARA TREINO E 20% PARA TESTE

Número de Árvores	100
Acurácia	0.7058823529411765

Tabela II
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 80% PARA TREINO E 20% PARA TESTE

Número de Árvores	200
Acurácia	0.7352941176470589

Tabela III
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 80% PARA TREINO E 20% PARA TESTE

Número de Árvores	500
Acurácia	0.8235294117647058

Tabela IV
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 80% PARA TREINO E 20% PARA TESTE

Número de Árvores	1000
Acurácia	0.7941176470588235

Tabela V
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 80% PARA TREINO E 20% PARA TESTE

Número de Árvores	1500
Acurácia	0.6764705882352942

Tabela VI
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 80% PARA TREINO E 20% PARA TESTE

Número de Árvores	2000
Acurácia	0.7205882352941176

Em seguida, foi executado o mesmo procedimento, porém dessa vez mudando o tamanho das amostras de teste e treino, passando para 10% e 90%, respectivamente. Nesse segundo teste, foram aplicadas precições para florestas com 100, 200, 500, 1000 e 1500 árvores.

Dessa vez, foi notável que com o aumento do número de árvores até 500, foi aumentando o número daa acurácia média. Quando o número de árvores passou a ser 1000, houve uma pequena queda e em seguida, quando aumentou para 1500 houve novamente uma subida. Levando a crer que o valor ideal de árvores deve estar em torno de 500. Vale ressaltar, também, que com a amostra de apenas de 10% para testes corre-se o risco do classificador ficar distorcido, em vista do tamanho do *dataset*. Com isso, é entendível a diferença entre os resultados das amostras de 80/20 e 90/10.

Tabela VII
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Número de Árvores	100
Acurácia	0.6806393298059963

Tabela VIII
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Número de Árvores	200
Acurácia	0.6971825396825396

Tabela IX
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Número de Árvores	500
Acurácia	0.7141816578483244

Tabela X
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Número de Árvores	1000
Acurácia	0.7055158730158729

Tabela XI
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Número de Árvores	1500
Acurácia	0.7127380952380952

Por fim, foi executado o programa separando o dataset, dessa vez não de forma aleatória, mas sim seguindo a seguencia dos dados, separando de 34 em 34 os dados do dataset.

Dessa vez foi notado uma ocilação grande do resultado, se comparado ao resultado obtido com amostras mais aleatórias. Vale ressaltar que para esse caso, o dataset foi modificado, para não se usar os dados na seguência que se encontravam, se não os dados seriam avaliados apenas para uma classe, e essa não faria parte do treinamento, na disposição inicial do dataset. A seguir os resultados obtidos nessa última simulação.

Tabela XII
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	1:34
Acurácia	0.7127380952380952

Tabela XIII
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	34:68
Acurácia	0.7647058823529411

Tabela XIV
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	68:102
Acurácia	0.7352941176470589

Tabela XV
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	102:136
Acurácia	0.6176470588235294

Tabela XVI
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	136:170
Acurácia	0.7352941176470589

Tabela XVII
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	170:204
Acurácia	0.6764705882352942

Tabela XVIII
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	204:238
Acurácia	0.5294117647058824

Tabela XIX
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	238:272
Acurácia	0.7352941176470589

Tabela XX
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	272:306
Acurácia	0.6764705882352942

Tabela XXI
DSEMPENHO DA FLORESTA RANDÔMICA AMOSTRADA COM 90% PARA TREINO E 10% PARA TESTE

Amostra de Teste	306:340
Acurácia	0.5588235294117647

III. CONCLUSÕES

Após analise dos dados, é notável que a Floresta Randômica se comporta de melhor maneira quando suas amostras são aleatórias, tal fato se deve pela garantia melhor (em relação a não aleatória) de não ocorrer o vício do classificador, umas vez que a amostra não será separada com algum critério, diminuindo a chance de influenciar os resultados do classificador.

Com isso, ao tomar como base os resultados obtidos com a separação aleatória, foi notável, também que paradiferentes separações de quantidades de treino e teste, necessita-se de diferente quantidade de ávores para achar o melhor resultado.

Com o algoritmo sendo trinado com 90% da amostra, é necessário de 1000 árvores para se obter um resultado satisfatório, isso é, acima de 75%.

Já com o algoritmo sendo treinado com 80% da amostra foi se percebido que um bom número de árvores está entorno de 500 e 1500, vale ressaltar, também, que o valor intermediário desse intervalo (1000 árvores), teve resultado inferior ao dos extremos.

AGRADECIMENTOS

Ao professor Alexandre Zaghetto que me apresentou, em um primeiro momento, matérias relativas à aprendizado de máquina e, com isso, criei interesse pela área.

REFERÊNCIAS

[1] Bishop, C. Pattern Recognition and Machine Learning. Springer, 2006

[2] Mitchell, T. Machine Learning. McGraw Hill, 1997.

[3] Lichman, M. (2013). UCI Machine Learning Repository. Irvine, CA: the University of California, School of Information and Computer Science.

[4] Bird, S., Klein, E., and Loper, E. (2009). Natural language processing with Python: Analyzing text with the natural language toolkit. Sebastopol, CA: O'Reilly Media.