Comparação de Variações da Floresta Randômica

1st Alberto Neto
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
albertotdneto@hotmail.com

***Abstract*—Neste relatório é utilizado o algoritmo de aprendizagem supervisionada de florestas randômicas para a resolução do problema de classificação de espécies de plantas, tendo como *features* atributos de forma e textura de imagens de suas folhas.**

Foram comparadas variações no uso do Bootstrap, na função de ganho, na quantidade de features por árvore e na quantidade de árvores por floresta para encontrar a que possuisse maior acurácia, utilizando o *cross validation* como algoritmo de treinamento e teste devido à baixa quantidade de dados.

No final a variação utilizando Bootstrap, função de ganho Gini e P/2 = 7 *features* e 200 árvores foi decidida como a melhor.

***Index Terms*—Random Forests, Machine Learning, Leaf.**

I. Introdução

Na utilização de um algoritmo de aprendizagem de máquina são várias as variações possíveis para que se consiga um resultado satisfatório.

Perceba que um resultado satisfatório depende dos objetivos e do problema em específico. Pode-se querer, por exemplo alta acurácia, baixa complexidade computacional ou até alta confiabilidade na resposta com o algoritmo classificando somente aqueles com alta probabilidade.

Neste trabalho será estudado como podemos fazer variações no algorítmo de florestas randômicas, com o objetivo de obter-se a melhor acurácia possível. Para realizarmos a comparação dos algoritmos, utilizaremos o problema descrito na seção seguinte.

II. Problema

O problema a ser solucionado será o de classificação de espécies de plantas através de fotos de suas folhas. O conjunto de dados a ser utilizado, o conjunto *leaf*, pode ser encontrado em https://archive.ics.uci.edu/ml/machine-learning-databases/00288/.

São encontradas no conjunto de dados 14 *features* já extraídas das imagens. Elas são:

1) Excentricidade
2) Proporção
3) Alongamento
4) Solidez
5) Convexidade estocástica
6) Fator isoperimétrico
7) Profundidade máxima de indentação
8) *Lobedness*
9) Intensidade média
10) Contraste médio
11) Suavidade
12) Terceiro momento
13) Uniformidade
14) Entropia

TABLE I
Espécies de folhas e quantidade de exemplos

Nº	Nome	Quant.	Nº	Nome	Quant.
1	Quercus suber	12	21	Fraxinus sp.	10
2	Salix atrocinera	10	22	Primula vulgaris	12
3	Populus nigra	10	23	Erodium sp.	11
4	Alnus sp.	8	24	Bougainvillea sp.	13
5	Quercus robur	12	25	Arisarum vulgare	9
6	Crataegus monogyna	8	26	Euonymus japonicus	12
7	Ilex aquifolium	10	27	Ilex perado ssp. azorica	11
8	Nerium oleander	11	28	Magnolia soulangeana	12
9	Betula pubescens	14	29	Buxus sempervirens	12
10	Tilia tomentosa	13	30	Urtica dioica	12
11	Acer palmatum	16	31	Podocarpus sp.	11
12	Celtis sp.	12	32	Acca sellowiana	11
13	Corylus avellana	13	33	Hydrangea sp.	11
14	Castanea sativa	12	34	Pseudosasa japonica	11
15	Populus alba	10	35	Magnolia grandiflora	11
16	Acer negundo	10	36	Geranium sp.	10
17	Taxus bacatta	5	37	Aesculus californica	10
18	Papaver sp.	12	38	Chelidonium majus	10
19	Polypolium vulgare	13	39	Schinus terebinthifolius	10
20	Pinus sp.	12	40	Fragaria vesca	11

As *features* de 1 a 7 são atributos de forma e as de 8 a 14, atributos de textura.

Importante notar que nem todas as imagens de folhas tiveram suas *features* extraídas e disponibilizadas nesse banco de dados. Das 40 espécies, apenas 30 possuem disponibilização das características, que são justamente as consideradas folhas simples pelo autor do conjunto de dados.

Na tabela I é possível ver o nome das espécies das plantas assim como a quantidade de exemplos existentes de cada. As espécies analisadas serão as de número 1 até 15 e de número 22 até 36 (as simples que possuem as *features* extraidas). Perceba que a quantidade de exemplos de cada é baixo, sendo a planta com maior quantidade a N° 11, com 16 dados, e a com menor quantidade a N° 17, com 5 dados. Ou seja, as classes não estão balanceadas.

III. Métodos

Nas subseções a seguir serão comparadas variações do algoritmo de florestas randomicas, com o intuito de encontrar

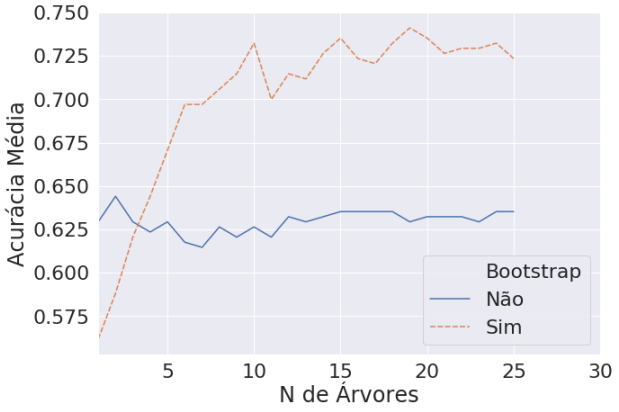

Fig. 1. *Cross validation* 10 para floresta. A função de ganho utilizada foi a entropia, com todas as features em cada árvore.

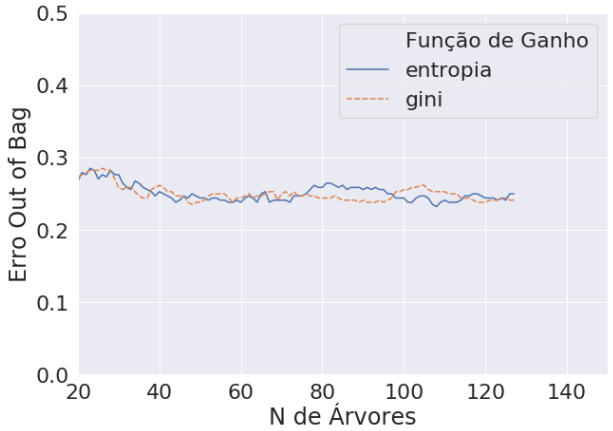

Fig. 2. Erro *out of bag*. Em ambas as variações todas as features foram utilizadas em cada árvore.

a variação que, no problema analisado, tenha a maior taxa de acurácia de predição possível.

É valido notar que, para resolver o problema do desbalanceamento dos dados, foram adicionados pesos às classes em todos os algoritmos a seguir.

As variações discutidas serão nos seguintes pontos:

- Treinamento e Teste
- *Bootstrap*
- Função de Ganho
- Quantidade de features por árvore
- Número de Árvores

A. Treinamento e Teste

Como o conjunto de dados é bem limitado, fazer uma divisão de, por exemplo, 30% de dados para teste e 70% para treinamento muito provavelmente não traria uma estimativa boa da capacidade de predição do algorítmo para dados não vistos. Na espécie N° 4 temos 8 observações, logo 6 observações seriam de treinamento e 2 de teste. Dependendo de quais 2 serão escolhidas aleatoriamente para compor o teste, teríamos uma variação muito grande no treinamento e na validação.

Portanto, será utilizado um *cross validation* 10 para o treinamento e teste de cada um dos algorítmos, com todos os dados incluidos para realizar o *cross validation*.

B. Bootstrap

A primeira variação a ser considerada é a utilização ou não do *Bootstrap*. Este consiste em, ao escolher o subconjunto de dados para cada árvore, "devolver" cada observação escolhida para que haja a possibilidade dela ser escolhida novamente para o mesmo subconjunto. Note o tamanho destes subconjuntos são iguais ao tamanho do conjunto original.

O intuito desta variação é diminuir a variância do algorítmo. De fato, para n observações Z, cada uma com variância σ^2, a variância da média Z' é dada por

$$\sigma^2/n \tag{1}$$

que diminui conforme aumentamos n.

Note que, como temos poucos dados, uma observação com valores muito díspares do resto causaria um impacto muito maior na variância do que se tivéssemos muitos dados.

Pelo gráfico da figura 1, como esperado, a acurácia média não muda para a floresta sem *bootstrap* pois as árvores serão extremamente similares, tendo em vista que o conjunto de dados e os *features* são os mesmos em cada uma.

Utilizando o *Bootstrap*, temos que a acurácia com poucas árvores é menor do que não utilizando. Isso se deve ao fato de que, com poucas árvores, é possível não termos alguns dados de treinamento sendo escolhidos nos subconjuntos, o que gera perda de informação importante ao treinamento.

Porém, conforme o número de árvores aumenta, a acurácia da floresta com *Bootstrap* aumenta significantemente, com uma aparente estabilização após 10 árvores, com acurácia de aproximadamente 0.725.

C. Função de Ganho

As duas funções mais utilizadas para mensurar o ganho quando utiliza-se florestas randômicas são a Gini Index e a Entropia.

A Gini é dada pela fórmula

$$Gini = 1 - \sum_{i=1}^{C} (pi)^2 \tag{2}$$

e a Entropia

$$E = \sum_{i=1}^{C} -pi * \log_2(pi) \tag{3}$$

Onde C é quantidade total de classes e pi é a proporção da classe i pelo total.

Veja que, no gráfico da figura 2, o erro duas variações se aproxima bastante (+-0.25) e aparentemente estão estabilizados.

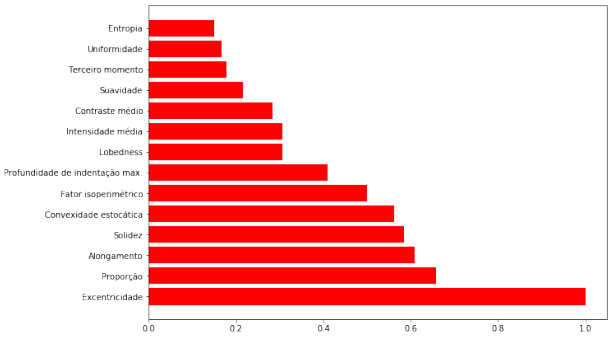

Fig. 3. Importância relativa das *features*

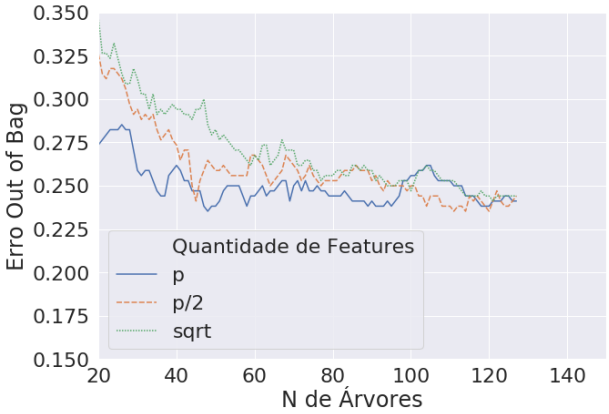

Fig. 4. Erro OOB Por Quantidade de Features

É de se esperar que os valores das duas funções de ganho sejam parecidos pois suas fórmulas são bem similares, porém ainda devemos escolher uma. Como o erro *out of bag* não auxiliou muito na decisão, faremos um *cross validation* 10:

TABLE II
ACURÁCIA DAS VARIAÇÕES NA FUNÇÃO DE GANHO

N árvores	Acurácia Gini	Acurácia Entropia
25	0.741 +-0.043	0.724 +-0.035
50	0.747 +-0.030	0.729 +-0.047
75	0.753 +-0.046	0.738 +-0.040
100	0.744 +-0.044	0.747 +-0.051
125	0.765 +-0.032	0.753 +-0.055

Para as duas variações da floresta, foram utilizados, além do bootstrap, todas as features em todas as árvores.

Veja que, das 5 quantidades de árvores, o Gini se mantém melhor em questão de acurácia média em 4. Com 125 árvores, melhor acurácia pra ambas as funções, a diferença do Gini pra Entropia é de 0.012 (1.2%). O Gini também é mais consistente em 125 árvores, seu desvio padrão é de 0.032 contra 0.055 da Entropia.

Por ter consistência e acurácia um pouco maiores no melhor caso, utilizaremos a função de ganho Gini.

Note que, apesar dos argumentos apresentados, caso utilizado em dados não vistos esta diferença, que é muito pequena, pode sumir. Logo essa escolha não representa um ganho em acurácia e sim um possível ganho caso nossa estimativa do erro de teste seja válida.

D. Importância das features

Para considerarmos a escolha de quantos features do total cada árvore da floresta terá, é interessante ver a importância de cada.

Para o teste de importância das *features* será utilizada uma floresta com uma única árvore e a função de ganho gini.

Nota-se do gráfico da figura 3 que a excentricidade é a característica de maior importância, sendo aproximadamente 1.4 vezes maior que a segunda, a proporção, e 5 vezes maior que os 3 menores, entropia, uniformidade e terceiro momento.

Como há uma diferença considerável da importância relativa dos parâmetros utilizados, é bem provável que a maioria das árvores das florestas serão parecidas, ou seja, altamente correlacionadas.

Para evitar esse problema, utilizaremos uma quantidade menor (que a total) de features por árvore.

No caso do problema de classificação analisado, temos que o número de features p = 14, logo p/2 = 7 e $\sqrt{p} = 3$

Do gráfico da figura 4 vemos que o erro OOB das florestas a partir de aproximadamente 80 árvores se estabilizam e tomam valores bem parecidos, próximos de 0.250. Novamente, como o erro OOB não nos da informação suficiente para a tomada de uma decisão, necessitaremos utilizar o *cross validation* 10. Podemos encontrar os resultados na tabela IV.

TABLE III
ACURÁCIA DAS VARIAÇÕES NA QUANTIDADE DE FEATURES

N árvores	Acurácia P	Acurácia P/2	Acurácia $\sqrt{P}$
25	0.741 +-0.043	0.741 +-0.034	0.744 +-0.046
50	0.747 +-0.030	0.774 +-0.030	0.744 +-0.054
75	0.753 +-0.046	0.771 +-0.026	0.744 +-0.057
100	0.744 +-0.044	0.774 +-0.030	0.747 +-0.058
125	0.765 +-0.032	0.776 +-0.030	0.756 +-0.040

Observando a tabela, temos que p/2 features teve os melhores resultados (atingindo consistentemente acurácia de 0.776), especificamente melhor que p *features*, apesar da diferença não ser tão grande, confirmando a hipótese estabelecida na análise da importância das *features*.

Em 100 árvores, por exemplo, a diferença das variações p/2 para p é de 0.03 (3%).

A acurácia de $\sqrt{p}$ *features*, que em muitos problemas traz ótimos resultados, foi be parecida com a de p. Como nossa quantidade de parâmetros é pequena (14), neste caso específico teremos $\sqrt{p} = \sqrt{14} = 3$ *features*, se aproximando bastante de uma árvore completamente aleatória (p = 1). Isso pode ser uma possível explicação para este resultado.

Pelo melhor valor das acurácias e pela consistência, será escolhido p/2 = 7 *features* para cada árvore.

1) Número de Árvores: Nos testes anteriores foram testadas várias quantidades de árvores em cada floresta para termos uma ideia do impacto das variações passadas em vários níveis.

TABLE IV
ACURÁCIA PARA VÁRIAS QUANTIDADES DE ÁRVORES POR FLORESTA

N árvores	Acurácia
100	0.774 +-0.030
125	0.776 +-0.030
150	0.779 +-0.027
175	0.782 +-0.020
200	0.785 +-0.023
225	0.779 +-0.020
250	0.782 +-0.024
275	0.782 +-0.024
300	0.774 +-0.026
325	0.774 +-0.026
350	0.779 +-0.030

Dos gráficos das figuras 2 e 4 vemos que, a partir de aproximadamente 75 árvores, há uma estabilização do erro OOB.

Para escolher a quantidade, devemos ter um número de árvores que explore várias possibilidades de resampling e de combinação de features. Com isto garantido, aumentar mais a quantidade não nos trará maiores benefícios, então é plausível pensar que não conseguiremos aumentar significativamente a acurácia com mais de 75 árvores.

Para efeitos de teste, comparemos a quantidade de árvores no nosso melhor algorítmo até então (com Bootstrap, função de ganho Gini e p/2 *features*) para várias quantidades de árvores a partir de 100.

Note que, a diferença do melhor, com 200 árvores (0.785 de acurácia), para o pior, com 100 (0.774), é de apenas 0.011 (1.1%), quantidade pouco significativa tendo em vista que o desvio padrão do melhor é de 0.020.

É preciso ter em mente que, quanto mais árvores, maior é o poder computacional necessário para treinamento. Porém isto será desconsiderado tendo em vista que o objetivo principal é comparar os algoritmos quanto às taxa de acurácia.

Assim como aconteceu na escolha da função de ganho, não temos uma escolha fácil pois todos os valores de acurácia são bem próximos. Além disso, essa acurácia é uma estimativa que pode não refletir a acurácia real quando testado com dados não vistos.

Com isto em mente, voltemos a comparação de 200 árvores contra 100 árvores. Se nossa estimativa for próxima da realidade, 200 árvores realmente se sai melhor pois possui melhor acurácia média e menor desvio padrão (maior consistência).

Uma segunda opção a ser considerada é de 225 árvores, pois possui o menor desvio padrão dentre todos e sua acurácia média é próxima da maior, porém apostaremos na com maior média.

Para o algoritmo final, será escolhido 200 árvores por floresta.

IV. CONCLUSÃO

Após diversas variações do algoritmo especificados na seção de Métodos, o algoritmo julgado como o melhor possui as seguintes características:

- Utiliza Bootstrap
- Função de ganho é o Gini
- A quantidade máxima de *features* consideradas para o *split* é P/2 = 14, onde P é a quantidade de *features* total
- Número de árvores por floresta é 200

Foi possível entender que algumas mudanças, como a quantidade de árvores por floresta e a função de ganho, podem ser difíceis de realizar.

No caso da função de ganho, a diferença, tanto na prática como na teoria, do Gini index e da Entropia (as duas funções analisadas) são pequenas.

Já na quantidade de árvores, após a estabilização, aumentar discriminadamente não causa melhoria significativa.

O fato de não haver muitos dados é um limitador pois deixa mais difícil fazer o teste dos algorítmos já que se deixarmos muitos dados de fora, o treinamento poderá ser ruim, e se fizermos teste com poucos dados, teremos um teste não muito preciso. Mesmo com o uso do *cross validation* para tentar conseguir uma boa estimativa dos resultados, o *cross validation* ainda possui um certo viés.