Machine Learning
Aula 5

Prof. Thiago A. N. De Andrade

Universidade Federal de Santa Maria
Departamento de Estatística

2025-09-15

Aviso aos estudantes

> - Este é um material novo e atualizado, elaborado especialmente para nosso curso **Machine Learning - UFSM 2025.2**. Entretanto, **não se configura em conteúdo original**. É apenas uma compilação resumida de conteúdos presentes nas referências citadas. Em resumo: é indispensável consultar as referências indicadas.
>
> - As imagens não são autorais e os respectivos créditos são reservados aos autores.
>
> - Este material foi integralmente produzido em `R Markdown`, utilizando o pacote `xaringan`, que possibilita a criação de apresentações **ninja**.

Supervised Learning

Floresta Aleatória (Random Forest)

Por que Random Forest?

- **Árvores isoladas** são **fáceis de interpretar**, mas têm **alta variância** e tendem a sobreajustar (*overfitting*).
- **Bagging (todos os p preditores avaliados em cada split)** reduz a variância ao **agregar** (média para regressão, voto para classificação) várias árvores treinadas em **amostras bootstrap** do treino.
- **Random Forest (RF)** = Bagging **+** aleatoriedade extra (escolher apenas m_{try} preditores candidatos a cada split) ⇒ **reduz a correlação** entre as árvores e **diminui ainda mais a variância** do *ensemble*. Em prática, as árvores são crescidas profundamente (baixo viés) e a variância é controlada pela agregação + aleatoriedade.

Notação e Convenções

- **Dados:** $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, $x_i \in \mathbb{R}^p$.
  Classificação: $y_i \in \{1, \ldots, C\}$; **Regressão:** $y_i \in \mathbb{R}$.
- **Ensemble:** T = número de árvores; índice $t = 1, \ldots, T$.
- **Árvore base (treinada com aleatoriedade Θ_t):** $h_t(x) = h(x; \Theta_t)$.
  Θ_t engloba a **amostra bootstrap** $\mathcal{B}_t$ e as **escolhas aleatórias de preditores** em cada nó.

- **Bootstrap da árvore t:** $\mathcal{B}_t$ é um multiconjunto de tamanho n formado por amostragem **com reposição** de $\{1, \ldots, n\}$.
  Out-of-bag (OOB) da árvore t: $\mathcal{O}_t = \{1, \ldots, n\} \setminus \text{suporte}(\mathcal{B}_t)$ (em média, ≈36{,}8% dos casos).

- **Parâmetro-chave:** $m_{\text{try}} \in \{1, \ldots, p\}$ = nº de preditores **amostrados aleatoriamente** e **considerados** em cada split.
  (Regra prática comum: $m_{\text{try}} \approx \sqrt{p}$ em classificação; $m_{\text{try}} \approx p/3$ em regressão.)

Predição do Ensemble

- **Regressão (média):**

$$\hat{f}_{\mathrm{RF}}(x) = \frac{1}{T}\sum_{t=1}^{T} h_t(x).$$

- **Classificação (votos/probabilidade):**

$$\hat{p}_k^{\mathrm{RF}}(x) = \frac{1}{T}\sum_{t=1}^{T} 1\{h_t(x) = k\}, \qquad \hat{y}_{\mathrm{RF}}(x) = \arg\max_{k\in\{1,\ldots,C\}} \hat{p}_k^{\mathrm{RF}}(x).$$

Por que reduzir a correlação entre árvores ajuda?

Se $\sigma^2(x) = \mathrm{Var}[h_t(x)]$ e $\rho(x)$ é a correlação entre duas árvores distintas em x, então (bagging):

$$\mathrm{Var}\left[\frac{1}{T}\sum_t h_t(x)\right] = \sigma^2(x)\left(\rho(x) + \frac{1-\rho(x)}{T}\right).$$

- **Bagging puro:** reduz o termo $\frac{1-\rho}{T}$ ao crescer T, mas **não mexe em** ρ.
- **Random Forest:** ao restringir para m_{try} preditores por split, **diminui** ρ, potencialmente **reduzindo muito mais** a variância.

Algoritmo de Treinamento (RF). Para $t = 1, \ldots, T$:

1. **Bootstrap:** amostrar $\mathcal{B}_t$ (tamanho n, com reposição).
2. **Crescer uma árvore CART (Classification and Regression Trees) totalmente expandida** (geralmente sem poda), repetindo:
   - No nó $\mathcal{N}$, **sortear** m_{try} preditores dentre $\{1, \ldots, p\}$.
   - Escolher o split que **maximiza o decréscimo de impureza** $\Delta_{\mathcal{I}}$ usando **apenas** esses m_{try}.
   - Parar por critério (ex.: **min_n** por nó folha).
3. **Guardar** a árvore h_t e os índices **OOB**: $\mathcal{O}_t$.

Predição: média (regressão) ou voto majoritário (classificação) das T árvores.

Objetivo local por nó (relembrando)

O split s em um nó $\mathcal{N}$ é escolhido maximizando:

$$\Delta_{\mathcal{I}}(s;\mathcal{N}) = \mathcal{I}(\mathcal{N}) - \frac{n_L}{n_{\mathcal{N}}}\mathcal{I}(L) - \frac{n_R}{n_{\mathcal{N}}}\mathcal{I}(R),$$

com $\mathcal{I}$ = **Gini** ou **Entropia** (classif.) e **MSE no nó** (regressão). No RF, a diferença é que a busca do **melhor split** usa **apenas** m_{try} **preditores** sorteados naquele nó.

Estimativa Out-of-Bag (OOB)

- Cada observação i fica **OOB** em $\approx e^{-1} \approx 36.8\%$ das árvores.
- Defina $\mathcal{T}_i = \{t : \ i \in \mathcal{O}_t\}$ e $T_i = |\mathcal{T}_i|$.

Regressão:

$$\hat{f}_{\,\mathrm{OOB}}(x_i) = \frac{1}{T_i} \sum_{t \in \mathcal{T}_i} h_t(x_i), \quad \mathrm{MSE}_{\mathrm{OOB}} = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{f}_{\,\mathrm{OOB}}(x_i)\right)^2.$$

Classificação:

$$\hat{p}_k^{\mathrm{OOB}}(x_i) = \frac{1}{T_i} \sum_{t \in \mathcal{T}_i} 1\{h_t(x_i) = k\}, \quad \hat{y}_{\mathrm{OOB}}(x_i) = \arg\max_k \hat{p}_k^{\mathrm{OOB}}(x_i),$$

$$\mathrm{Err}_{\mathrm{OOB}} = \frac{1}{n} \sum_{i=1}^{n} 1\{\hat{y}_{\mathrm{OOB}}(x_i) \neq y_i\}.$$

→ **Dispensa CV explícito** e acompanha o **overfitting** em tempo real.

Hiperparâmetros (e efeitos práticos)

- T **(trees):** quanto maior, **menor variância** até estabilizar OOB; custo computacional cresce linearmente.
- m_{try}**:** menor $\Rightarrow$ menor **correlação** entre árvores (bom), porém **splits menos informativos** (pode aumentar viés).
  - Regras práticas comuns: $\sqrt{p}$ (classif.), $\lfloor p/3 \rfloor$ (regr.).

- **min_n (tamanho mínimo de nó folha):** controla **granularidade**; valores pequenos aumentam variância individual mas o ensemble compensa.
- **Bootstrap/subamostragem:** `replace = TRUE` (bootstrap) ou sem reposição com fração < 1.
- **Profundidade:** tipicamente **árvores totalmente crescidas** (sem poda); pode-se limitar `max_depth` se necessário.
- **Split rule:** Gini/Entropia (classif.) e Variância/MSE (regr.).

Boas Práticas

- **Curva OOB × nº de árvores:** aumente T até estabilizar.
- **Desbalanceamento:** usar **class weights** ou **amostragem estratificada** por classe.
- **Leakage:** evite preditores pós-tratamento; faça **pré-processamento dentro** do *workflow* (treino/teste).
- **Interpretação:** além da importância, use **PDPs/ICE** para efeitos marginais e **SHAP** se necessário.

Referências

- **A Recursive Partitioning Decision Rule for Nonparametric Classification**
- **Classification and Regression Trees**
- **The strength of weak learnability**

- **Additive Logistic Regression: A Statistical View of Boosting**
- **Random Forests**
- **XGBoost: A Scalable Tree Boosting System**
- **An Introduction to Statistical Learning**
- **Aprendizado de máquina: uma abordagem estatística**
- **Materiais Curso R**

Não deixe de entrar em contato comigo para tirar suas dúvidas: thiagoan.andrade@gmail.com

Estamos no  @thiagoan.andrade para networking e socializações

Obrigado!

Thanks!