Fundamentos de Mineração de Dados e Ciência de Dados

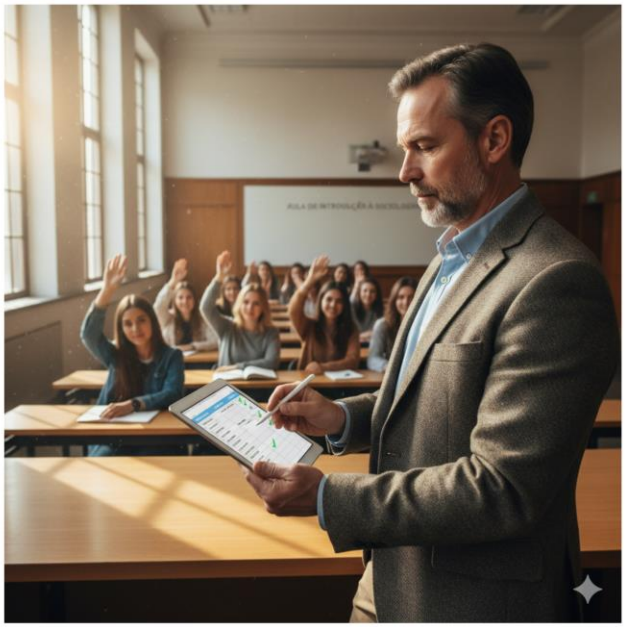
Florestas Aleatórias

Marcelo de Souza Lauretto

22 de outubro de 2025

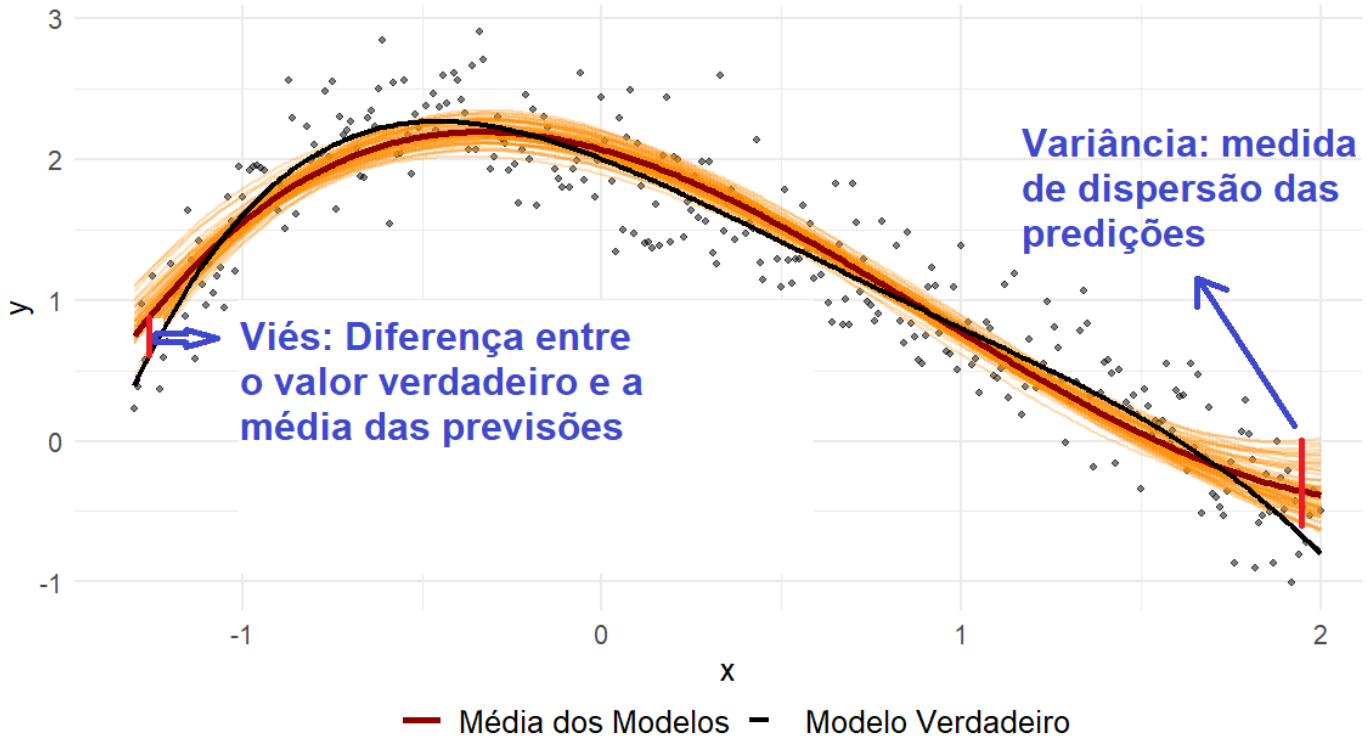
Lembrete:

Marque presença!



Viés e Variância

- Conceitos principais:
  - Viés (Bias): É o **erro sistemático** – quando nosso método de estimativa "erra o alvo" consistentemente para o mesmo lado.
  - Variância (Variance): É a **dispersão** ou **instabilidade** – quando nossas estimativas são muito espalhadas, mesmo sem um padrão alto de erro

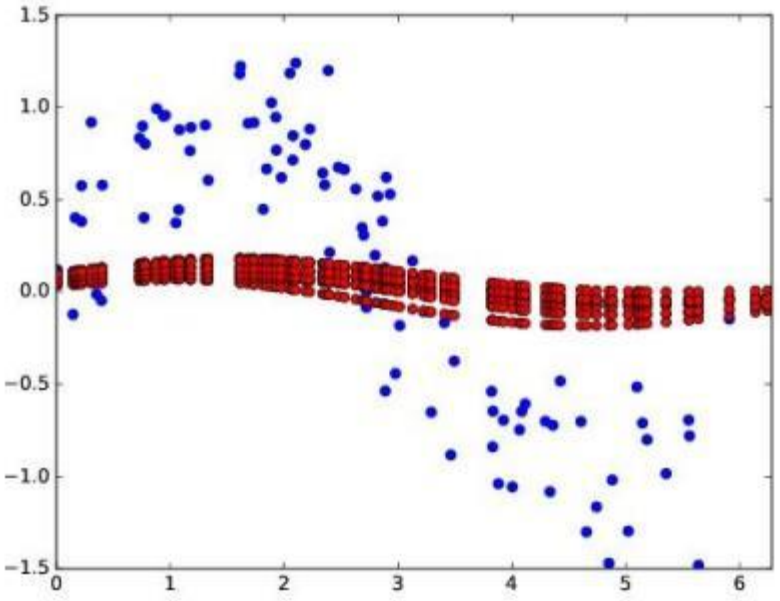


Viés e Variância

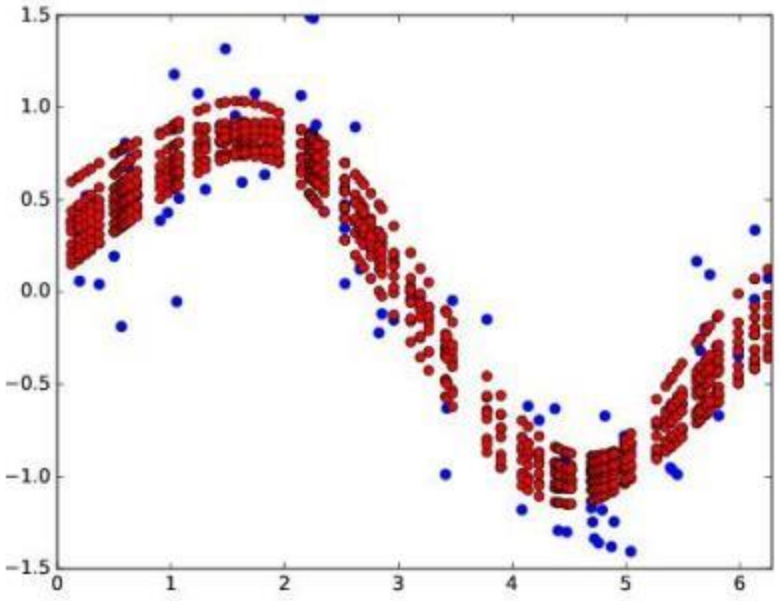
- Viés alto:
  - O modelo é **muito simples** para capturar a complexidade dos dados.
- Viés baixo:
  - O modelo é **flexível** o suficiente para se aproximar da realidade.
- Variância alta (instabilidade):
  - O modelo é **muito sensível** a pequenas mudanças nos dados de treinamento.
- Variância baixa:
  - O modelo **é estável** e produz previsões consistentes (mesmo que ele erre sistematicamente nas previsões).
- Intuitivamente:
  - Modelos “pouco complexos” possuem alto viés e baixa variância;
  - Modelos “muito complexos” possuem baixo viés e alta variância

Decomposição viés-variância (relembrando)

Alto viés, baixa variância



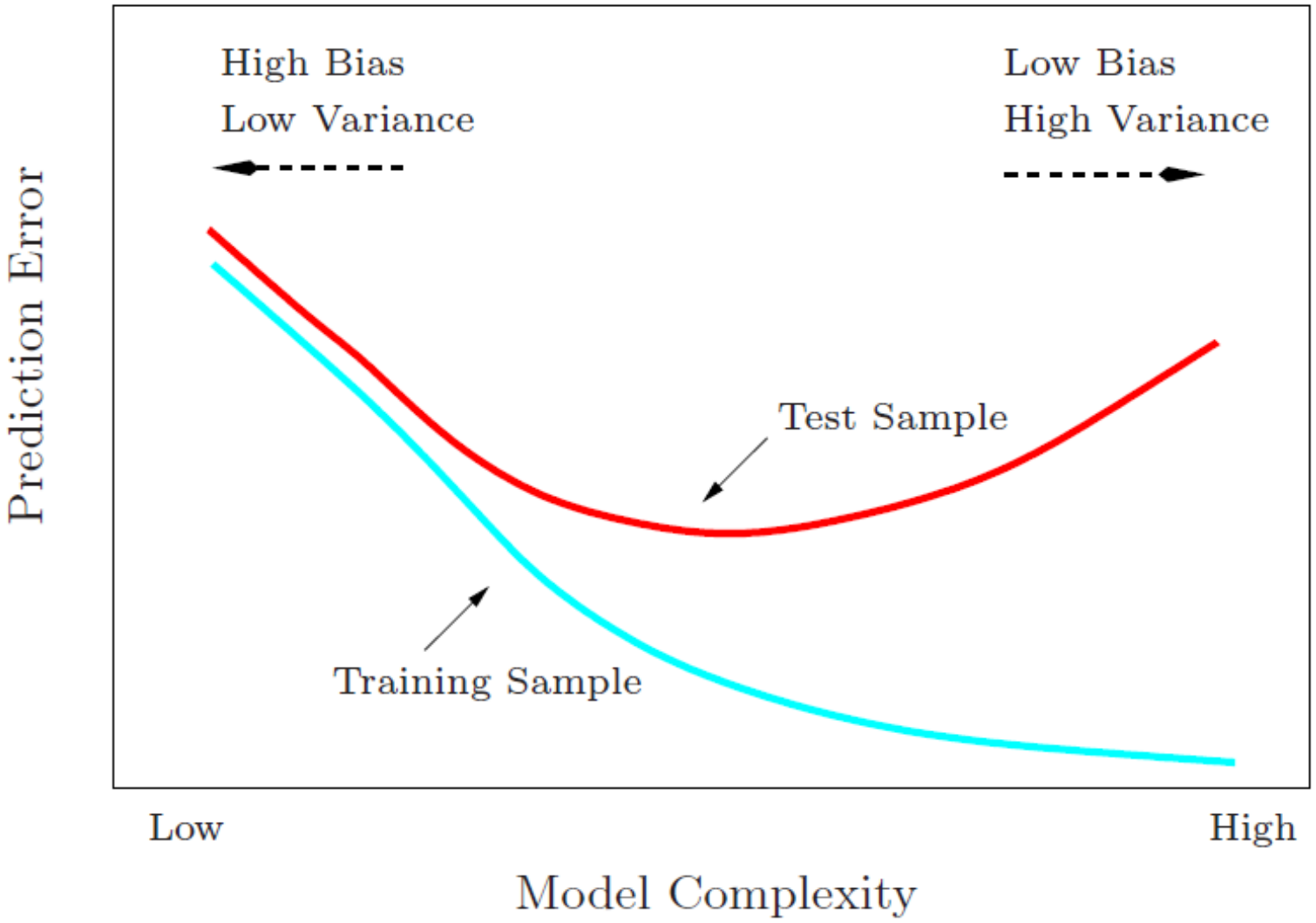
Baixo viés, alta variância



Pontos azuis: dados reais; pontos vermelhos: valores preditos

Fonte:
http://prorum.com/index.php/2526/voce-pode-exemplo-evidencia-empirica-decomposicao-variancia

Decomposição viés-variância (relembrando)



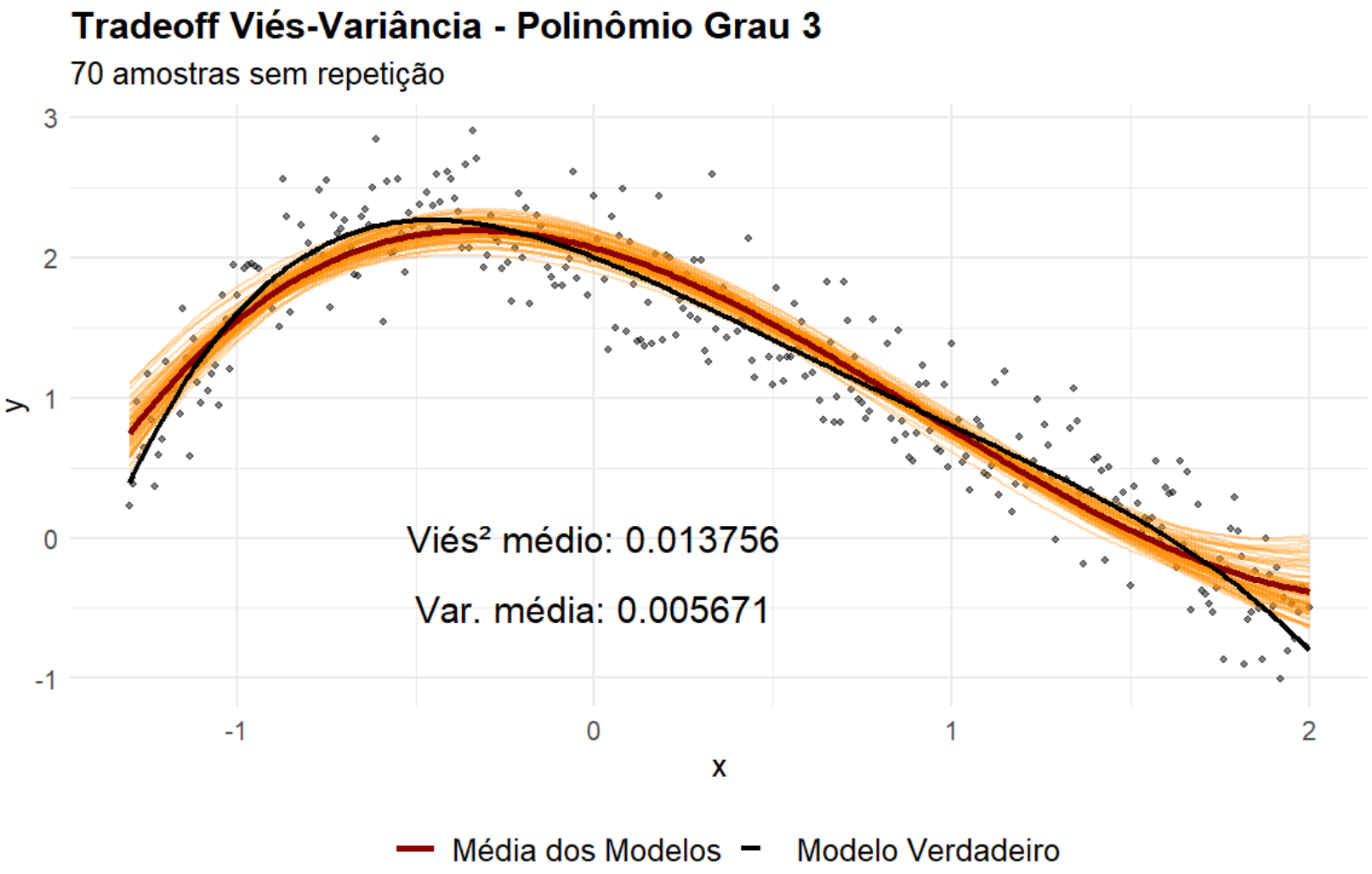
Exemplo

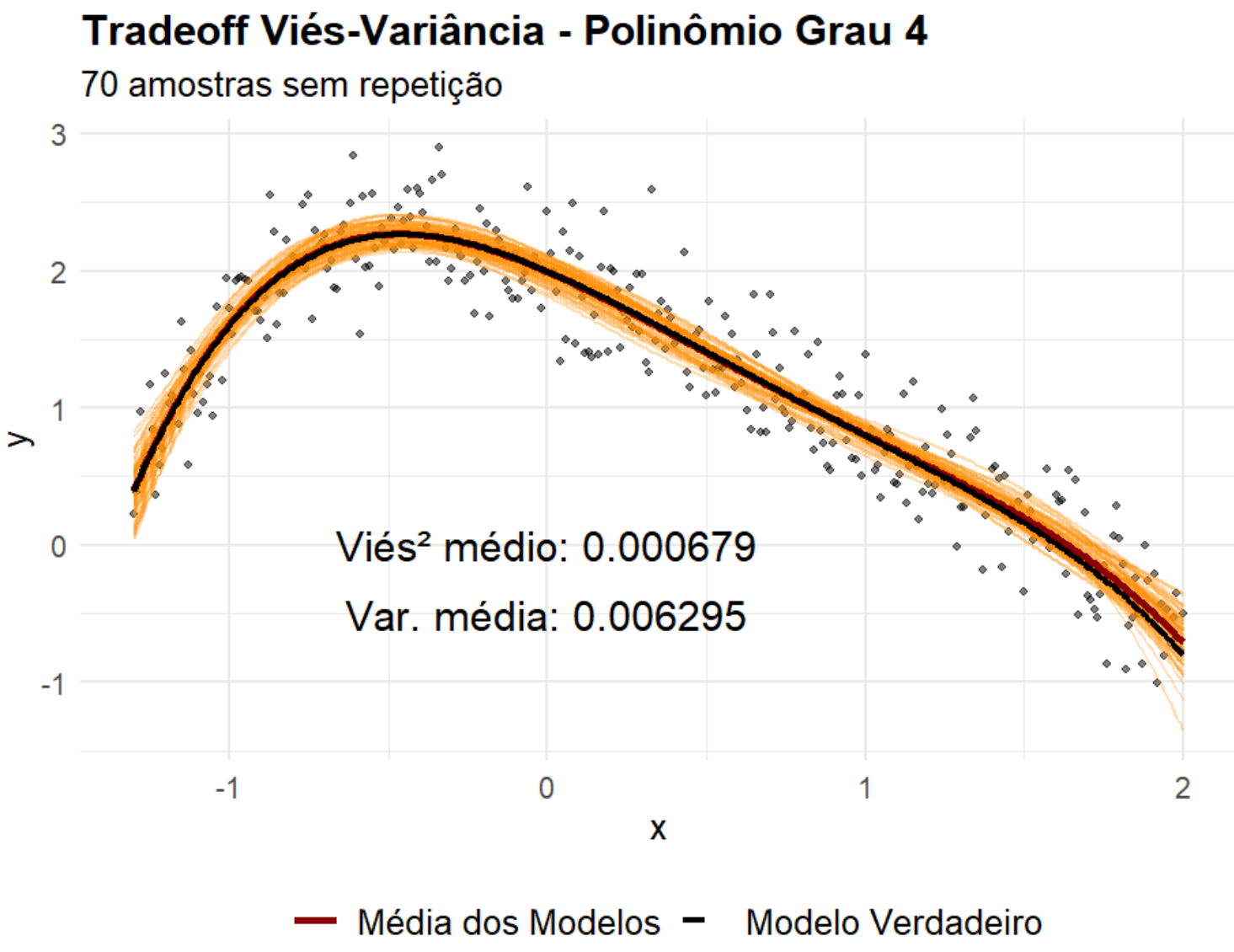
- Suponha um modelo “verdadeiro” descrito pela equação

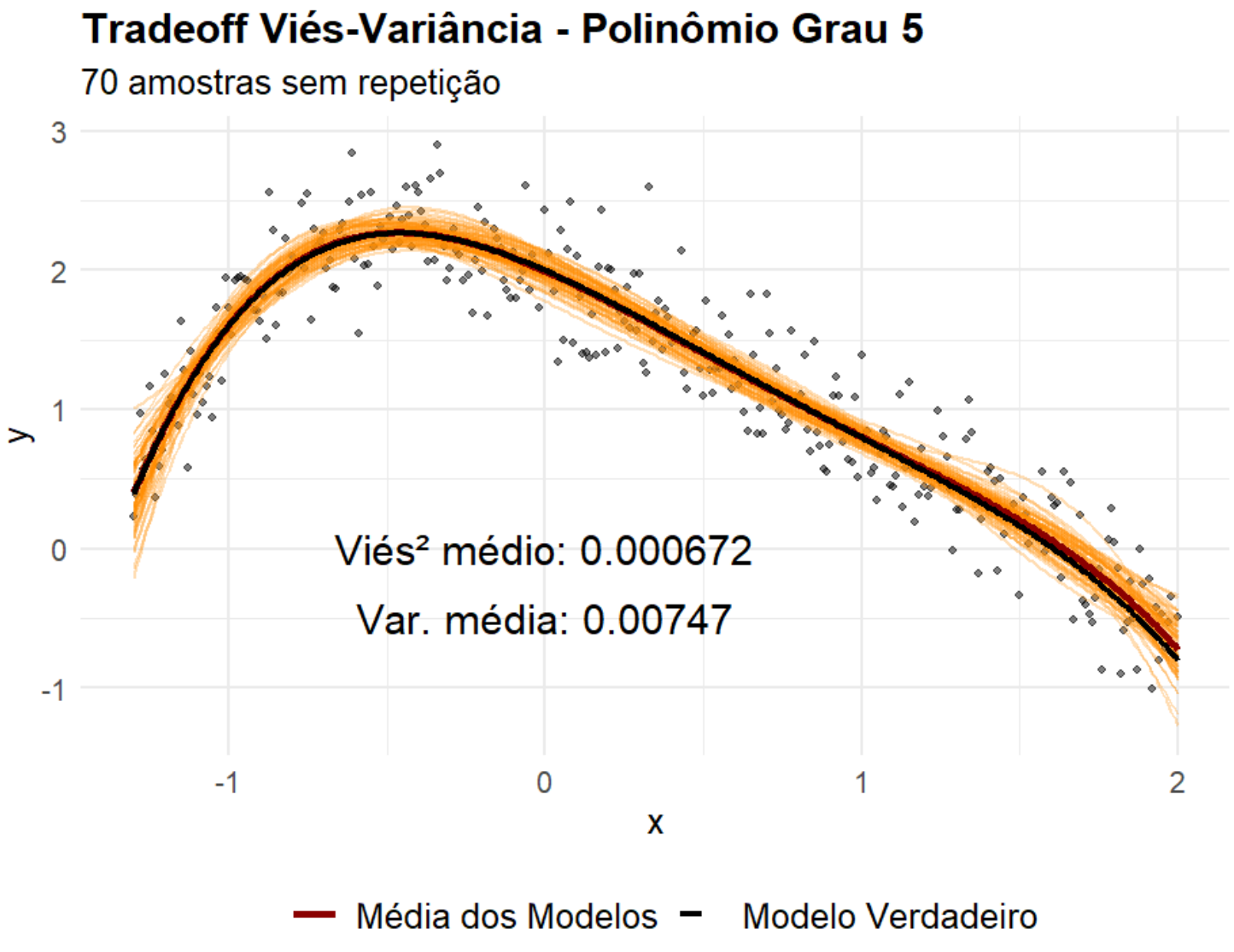
$$y = \beta_0 + \beta_1\, x + \beta_2\, x^2 + \beta_3\, x^3 + \beta_4\, x^4 + \varepsilon$$

em que $\varepsilon \sim N(0, \sigma^2)$.

- Analisaremos três modelos de regressão:
  - Um formulado por um polinômio de grau 3 (modelo mais simples)
  - Um formulado por um polinômio de grau 4 (modelo ideal)
  - Um formulado por um polinômio de grau 5 (modelo mais “complexo”)


Tradeoff Viés-Variância - Polinômio Grau 3
70 amostras sem repetição
3
2
1
0
-1
y
-1
0
1
2
x
Viés² médio: 0.013756
Var. média: 0.005671
Média dos Modelos
Modelo Verdadeiro


Tradeoff Viés-Variância - Polinômio Grau 4
70 amostras sem repetição
3
2
1
0
-1
y
-1
0
1
2
x
Viés² médio: 0.000679
Var. média: 0.006295
Média dos Modelos
Modelo Verdadeiro


Tradeoff Viés-Variância - Polinômio Grau 5
70 amostras sem repetição
3
2
1
0
-1
y
-1
0
1
2
x
Viés² médio: 0.000672
Var. média: 0.00747
Média dos Modelos
Modelo Verdadeiro

Comitês de classificadores

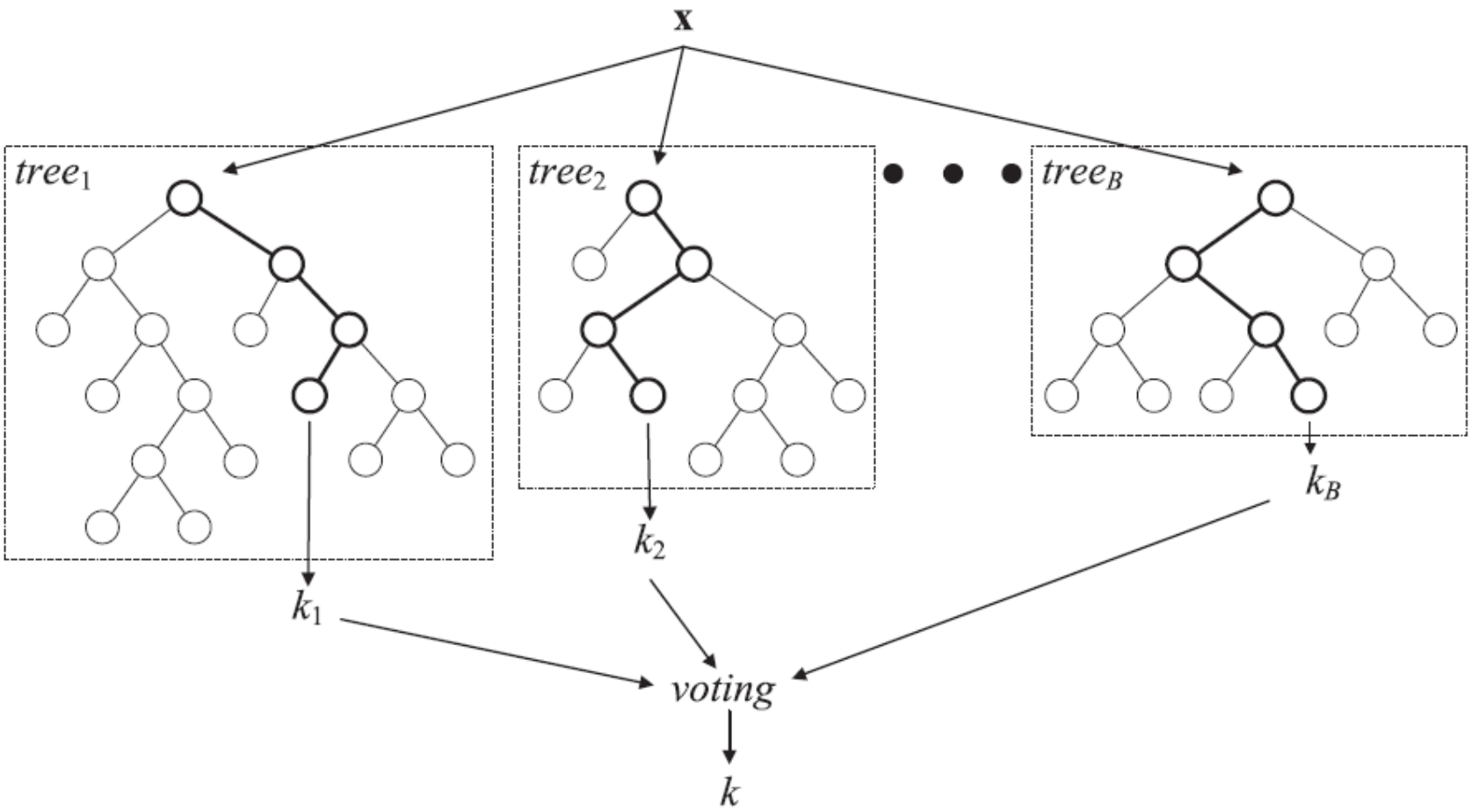
- *Classificador fraco*: preditor $\hat{f}(x)$ com baixo viés e alta variância.
- Um *Ensemble* (comitê) de classificadores é um classificador agregado, composto por uma coleção de classificadores fracos.
- Idéia: reduzir o erro esperado de classificação através da redução da variância total do classificador agregado.
- Requisitos dos classificadores individuais:
  - Possuir desempenho individual razoável
  - Possuir comportamentos diversos entre si (idealmente: baixa correlação entre classificadores).

Bagging

- Bagging [Breiman, 1996]: **B**ootstrap **Agg**regat**ing**
  - Geram-se reamostras *bootstrap* - amostras sorteadas do conjunto original, de mesmo tamanho e via sorteio simples com reposição (ideias apresentadas adiante)
  - Cada subconjunto amostrado é utilizado para a construção de um novo classificador
  - A classificação final é realizada por um sistema de votação
    - Nova instância é submetida a cada classificador isolado, atribuindo-se a classe final com base na contagem de "votos" nas classes
  - Alta quantidade de classificadores tenderia a reduzir a variância na classificação do conjunto, especialmente na presença de dados com ruído

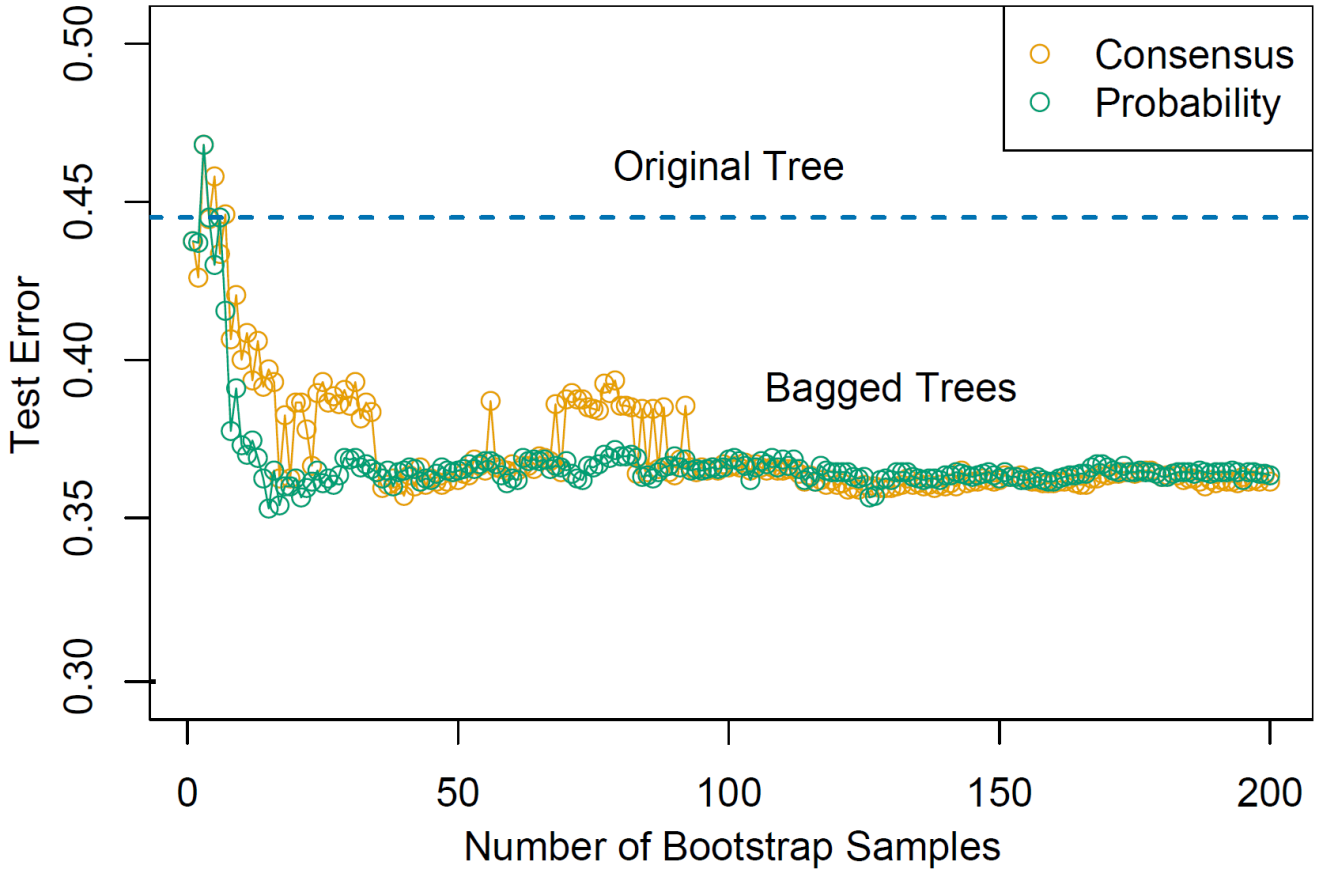
Bagging

- Exemplo: bagging de árvores de classificação



Bagging

- Erro de classificação tende a um limite com o aumento do número de amostras bootstrap (e, consequentemente, de árvores)

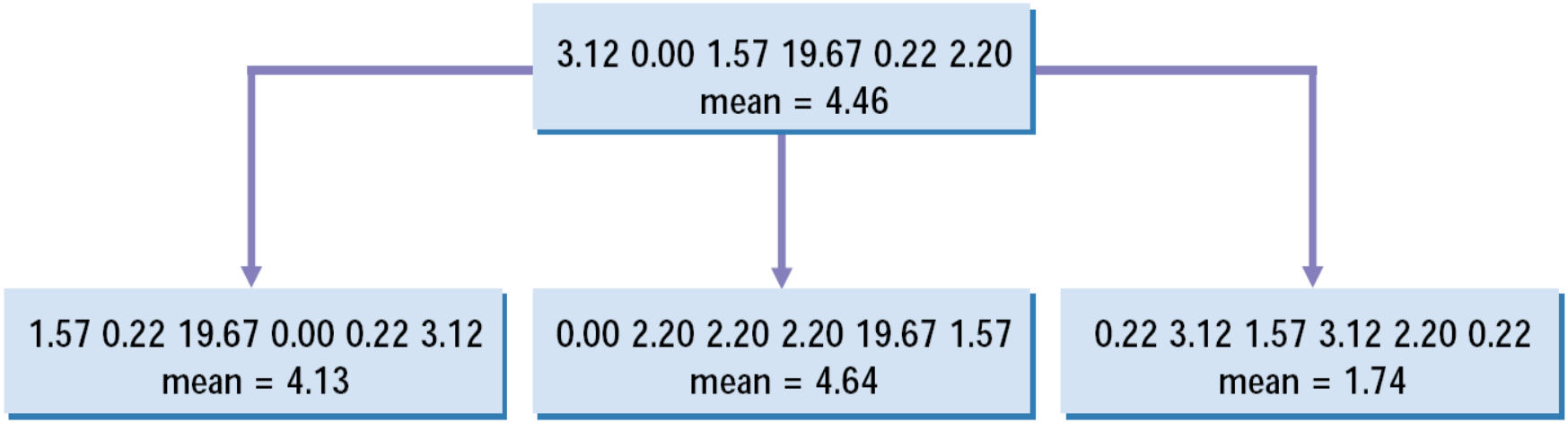


Adaptado de Hastie, Tibshirani, Friedman (2009, Cap.8)

Bootstrap – ideias gerais

THE BOOTSTRAP IDEA

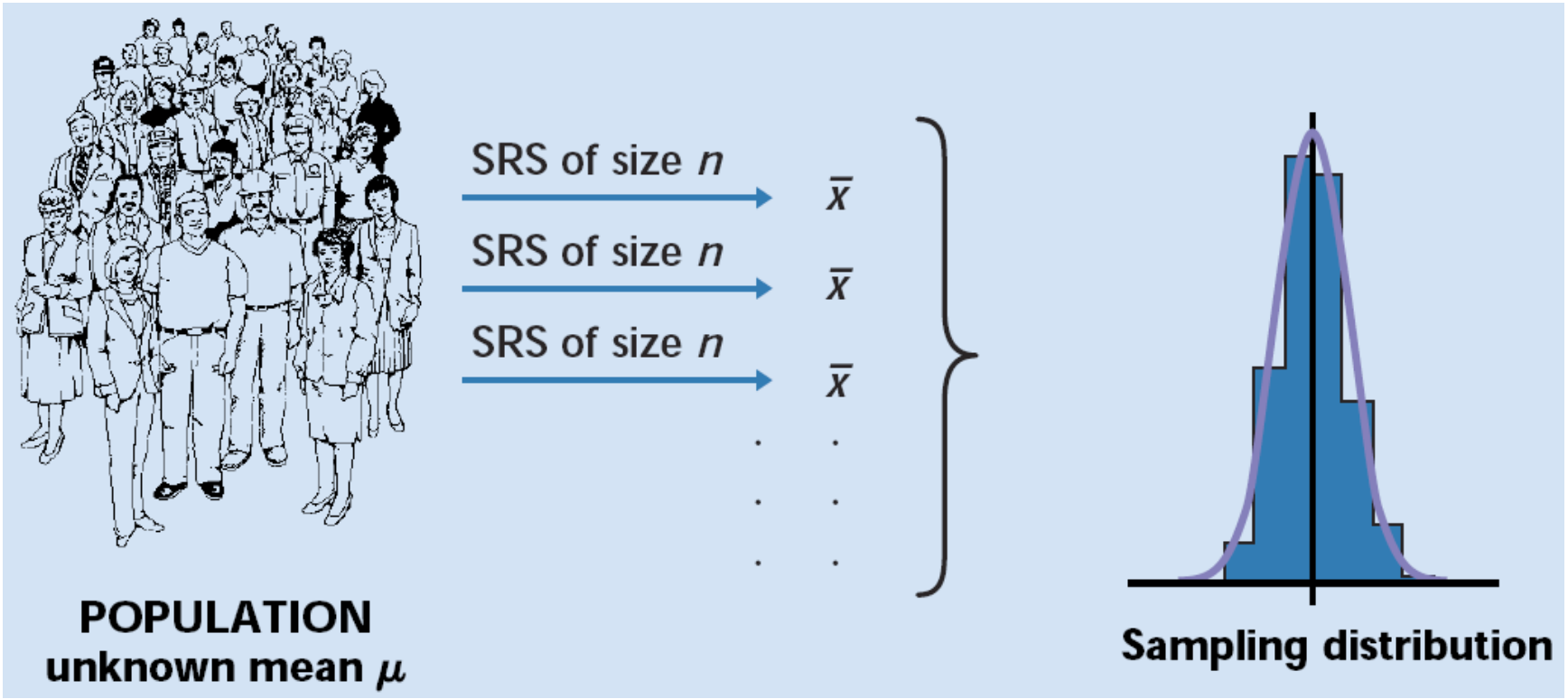
The original sample represents the population from which it was drawn. So resamples from this sample represent what we would get if we took many samples from the population. The bootstrap distribution of a statistic, based on many resamples, represents the sampling distribution of the statistic, based on many samples.



Hesterberg et al. (2003)

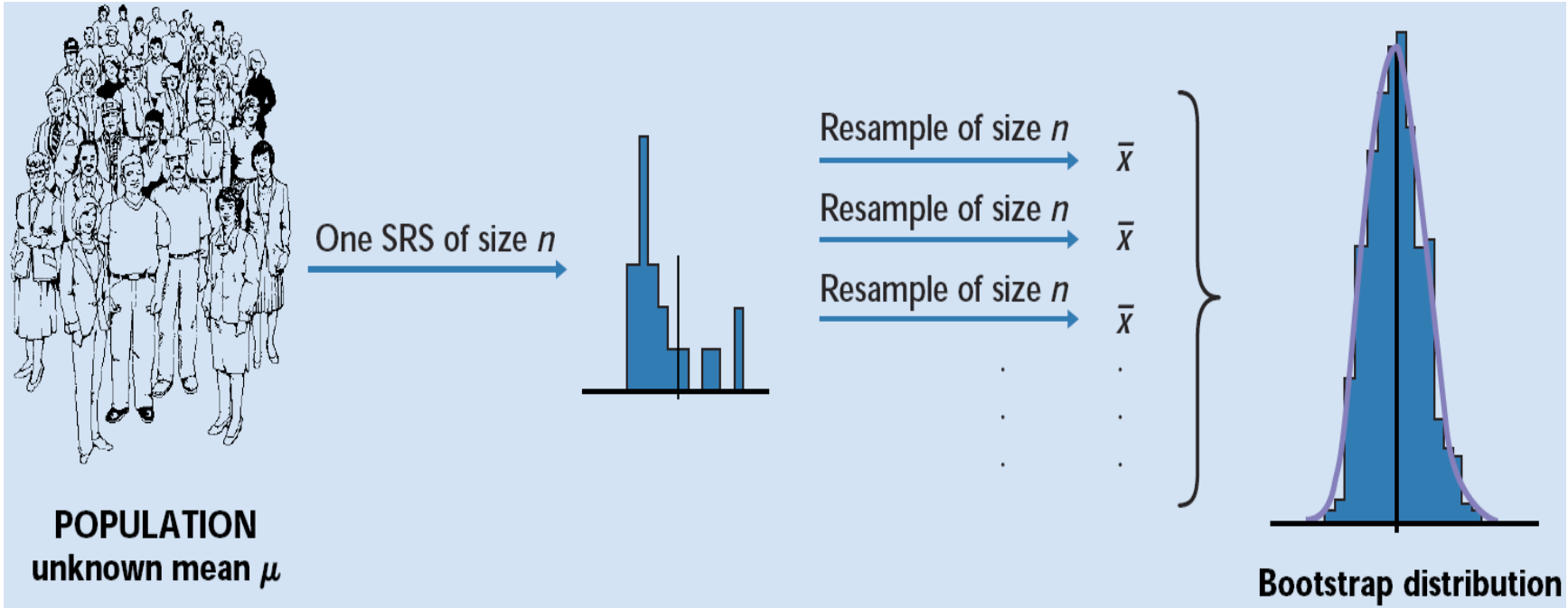
Bootstrap – ideias gerais

- Ideia da distribuição amostral de uma média $\bar{x}$:
  Obtida a partir de um grande número de amostras tomadas da população:



Bootstrap – ideias gerais

- Ideia da distribuição via bootstrap: quando só conseguimos obter uma amostra da população (caso típico), usamos reamostragem bootstrap como um substituto para outras possíveis amostras da população



Relembrando a aula anterior: Algoritmo geral para indução de árvores binárias

Construir-Árvore(t, $\mathcal{L}$, Stp, Label, score)

Se o conjunto de treinamento $\mathcal{L}$ satisfaz a regra de parada Stp(t),

- **então** rotule t de acordo com a regra Label(t)
- **caso contrário**
  - A) Para cada atributo a_j, $j = 1 \ldots M$:
    - para cada divisão válida possível $s_{j,i} \in \{s_{j,1}, s_{j,2}, \ldots s_{j,q}\}$ do atributo a_j, avalie score($\mathcal{L}$, a_j, $s_{j,i}$)
    - escolha a partição s_j^* com pontuação máxima
  - B) Escolha o atributo a^* que produz a pontuação máxima e sua divisão ótima s^*
  - C) Rotule t com o atributo a^*
  - D) Divida o conjunto de treinamento $\mathcal{L}$ nos subconjuntos $\mathcal{L}_L$, $\mathcal{L}_R$ induzidos por s^*
  - E) Crie novos nós filhos t_L, t_R correspondentes aos subconjuntos $\mathcal{L}_L$, $\mathcal{L}_R$
  - F) Chame recursivamente o algoritmo **Construir-Árvore** em t_L, $\mathcal{L}_L$
    Chame recursivamente o algoritmo **Construir-Árvore** em t_R, $\mathcal{L}_R$

Random Forests

- Comitê do tipo *bagging* baseado em árvores de classificação/regressão
- Ideia central: melhorar a redução da variância no bagging através da redução da correlação entre as árvores.
- Há várias formulações para *random forests*; aqui descrevemos a mais difundida, proposta por Breiman (2001).
  - CART como algoritmo base de construção das árvores, com uma diferença importante:
    - Sorteio aleatório de um subconjunto de atributos candidatos para particionar cada nó
- Notação usada nos próximos slides:
  - B: número de amostras bootstrap a serem sorteadas (parâmetro de entrada)
  - m: número de atributos a serem sorteados para a divisão de cada nó (ver slide "Alguns Parâmetros Relevantes" adiante)
  - $\mathcal{L}^*$: amostra bootstrap obtida do conjunto de treinamento $\mathcal{L}$
  - T_b: árvore induzida sobre a b-ésima amostra bootstrap
  - $\{T_b\}_1^B$: floresta completa

Construção das Random Forests

1. Para $b = 1$ até B:
   a) Gere uma amostra bootstrap $\mathcal{L}^*$ a partir do conjunto de treinamento $\mathcal{L}$. Atribua um nó raiz t à amostra $\mathcal{L}^*$
   b) Construa uma árvore de decisão T_b a partir da amostra Z^*:
      - Se a condição de parada for atendida,
        - Rotule t de acordo com a regra $\mathrm{Label}(t)$
      - Senão,
        - Selecione aleatoriamente m variáveis do total de p variáveis preditoras
        - Escolha a melhor divisão para $\mathcal{L}^*$ entre as m variáveis sorteadas
        - Divida o conjunto $\mathcal{L}^*$ nos sub-conjuntos $\mathcal{L}_L^*$ e $\mathcal{L}_R^*$, criando os dois nós filhos t_L, t_R
        - Aplique recursivamente o passo (b) a t_L, $\mathcal{L}_L^*$
          Aplique recursivamente o passo (b) a t_R, $\mathcal{L}_R^*$.
2. Retorne a floresta completa $\{T_b\}_1^B$.

Adaptado de Hastie, Tibshirani, Friedman (2009, Cap.15)

Predição de novas instâncias

- Após a construção de $\{T_b\}_1^B$, a predição do valor/classe da variável resposta de uma nova instância $\boldsymbol{x}$ é como segue.
- Denotemos $\hat{y}_b(\boldsymbol{x})$ o valor/classe predito pela árvore T_b.
  - No contexto de classificação: voto majoritário
    $\hat{y}_{RF}(x) = \text{voto majoritário entre } \hat{y}_1(\boldsymbol{x}), \hat{y}_2(\boldsymbol{x}) \ldots, \hat{y}_B(\boldsymbol{x})$
  - No contexto de regressão: média

$$\hat{y}_{RF}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{y}_b(\boldsymbol{x})$$

Adaptado de Hastie, Tibshirani, Friedman (2009, Cap.15)

Erro Out-Of-Bag

- Erro Out-Of-Bag (OOB):
  - Em cada iteração b, a reamostra bootstrap Z^* usa aproximadamente 63% dos exemplos originais para construir a árvore T_b

    $$\Pr(\text{observação } i \in \text{amostra bootstrap } b) = 1 - \left(1 - \frac{1}{N}\right)^N \approx 1 - e^{-1} = 0.632$$

  - Logo, aproximadamente 37% dos exemplos originais não são usados na indução da árvore T_b;
  - Esses exemplos (comumente chamados *out-of-bag*) podem ser usados para estimar o erro de cada árvore da floresta.
  - Cálculo do erro *out-of-bag* (OOB):
    - Para cada exemplo $(\boldsymbol{x}_i, y_i)$ do conjunto original, obtenha a classe predita $\widehat{y_i}$ computando os votos apenas das árvores que não usaram $(\boldsymbol{x}_i, y_i)$ em sua construção;
    - A estimativa do erro OOB é obtida a partir da matriz de confusão obtida com o passo acima.

- A estimativa do erro OOB é, na prática, equivalente à do erro de validação cruzada, mas obtida com um custo computacional menor (não há necessidade de treinar os classificadores várias vezes).

Medida de importância de atributos

- Avaliação da importância de atributos:
  - Implementações de random forests usualmente possuem duas medidas de importância de atributos (Bastos, 2014):
    - *Importância baseada no erro*: uma vez construída a floresta, permuta-se aleatoriamente os valores do atributo entre os exemplos do conjunto de teste (OOB). O aumento percentual do erro de classificação sobre os exemplos com valores permutados em relação ao erro sobre os exemplos originais fornece a medida de importância do atributo.
      (ver próximo slide)
    - *Importância baseada no Índice de Gini*: esta medida avalia a soma dos decréscimos da impureza (Índice de Gini) de todos os nós na floresta rotulados pelo atributo avaliado
  - Esses métodos podem servir para seleção dos atributos mais relevantes (o que é importante quando o custo da coleta de dados é proporcional ao número de atributos)
  - Bastos et al (2014) discute a seleção de atributos e apresentam uma comparação empírica de desempenho desses dois índices, juntamente com dois novos indicadores propostos.

Medida de importância de atributos

- Importância baseada no erro - exemplo: importância do atributo x_3

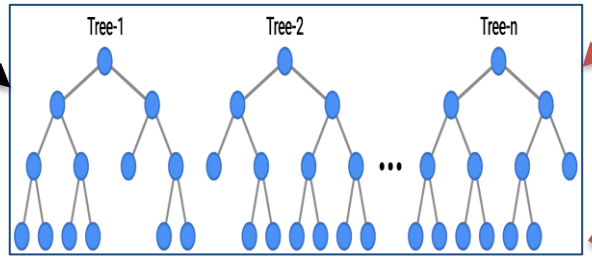
Conjunto original

x_1	x_2	x_3	x_4	x_5	x_6	y
X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	y_1
X_{21}	X_{22}	X_{23}	X24	X_{25}	X_{26}	y_2
X_{31}	X_{32}	X_{33}	X_{34}	X_{35}	X_{36}	y_3
X_{41}	X_{42}	X_{43}	X_{44}	X_{45}	X_{46}	y_4
X_{51}	X_{52}	X_{53}	X_{54}	X_{55}	X_{56}	y_5
X_{61}	X_{62}	X_{63}	X_{64}	X_{65}	X_{66}	y_6
X_{71}	X_{72}	X_{73}	X_{74}	X_{75}	X_{76}	y_7
X_{81}	X_{82}	X_{83}	X_{84}	X_{85}	X_{86}	y_8
X_{91}	X_{92}	X_{93}	X_{94}	X_{95}	X_{96}	y_9

Conjunto com permutação de x_3

x_1	x_2	x_3	x_4	x_5	x_6	y
X_{11}	X_{12}	X_{83}	X_{14}	X_{15}	X_{16}	y_1
X_{21}	X_{22}	X_{73}	X24	X_{25}	X_{26}	y_2
X_{31}	X_{32}	X_{63}	X_{34}	X_{35}	X_{36}	y_3
X_{41}	X_{42}	X_{93}	X_{44}	X_{45}	X_{46}	y_4
X_{51}	X_{52}	X_{23}	X_{54}	X_{55}	X_{56}	y_5
X_{61}	X_{62}	X_{53}	X_{64}	X_{65}	X_{66}	y_6
X_{71}	X_{72}	X_{43}	X_{74}	X_{75}	X_{76}	y_7
X_{81}	X_{82}	X_{33}	X_{84}	X_{85}	X_{86}	y_8
X_{91}	X_{92}	X_{13}	X_{94}	X_{95}	X_{96}	y_9

RF previamente construída



Predições e cálculo do erro

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_9 \Rightarrow Err_{oob}$

Predições e cálculo do erro

$\hat{y}_1^{(3)}, \hat{y}_2^{(3)}, \ldots, \hat{y}_9^{(3)} \Rightarrow Err_{oob}^{(3)}$

$$IE(x_3) = \frac{Err_{oob}^{(3)} - Err_{oob}}{Err_{oob}}$$

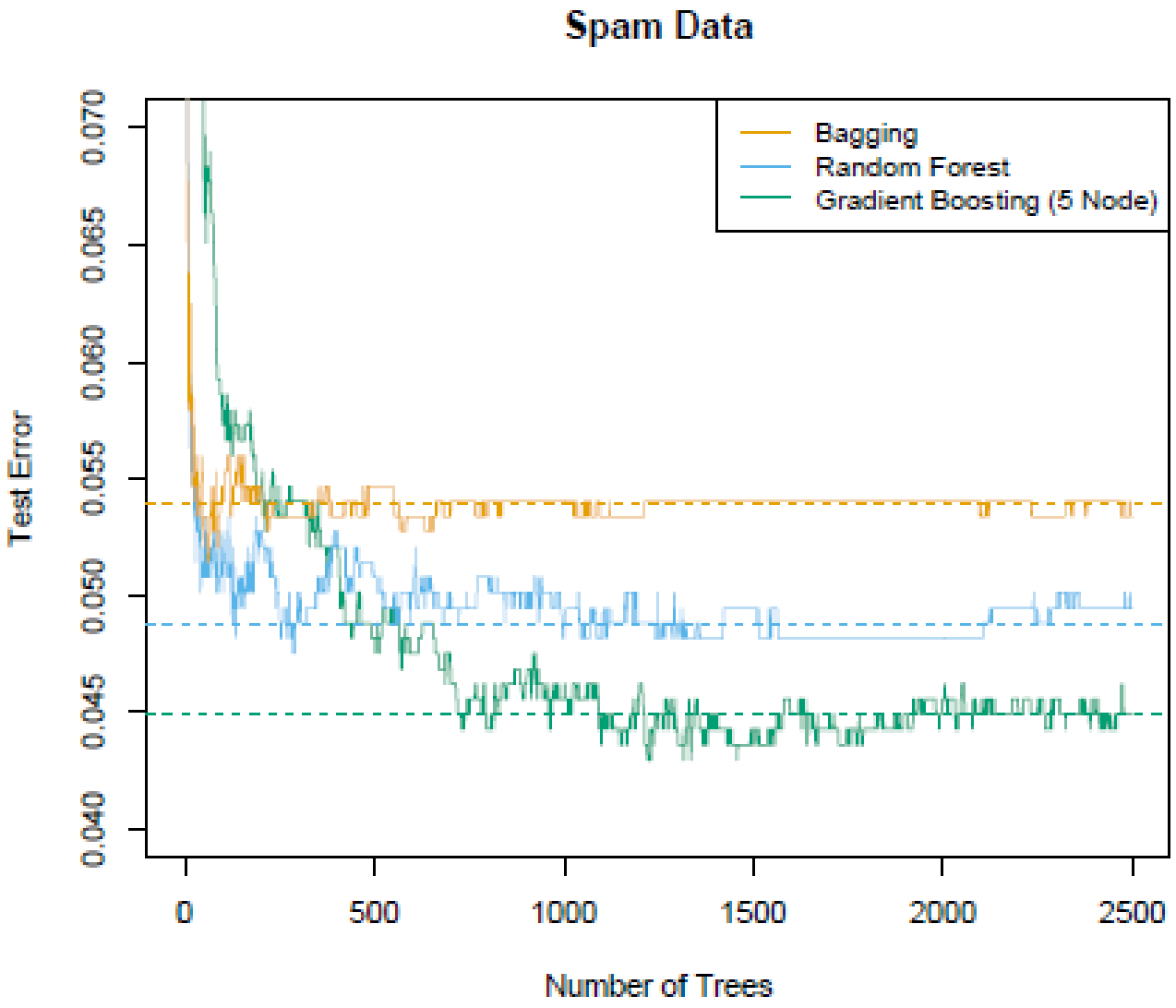
Aumento relativo no erro OOB das predições *depois* da permutação do atributo

Alguns parâmetros relevantes

Notação usada na função randomForest do R:

- Mtry: número de variáveis a serem sorteadas na partição de cada nó
  - Costuma ser importante.
    - Mtry muito alto: árvores com alta similaridade nas predições (alta correlação)
    - Mtry muito baixo: árvores com pouca informação disponível
  - Default:
    - Classificação: $\sqrt{quantidade\ de\ variáveis}$
    - Regressão: $\lfloor quantidade\ de\ variáveis\ \div 3 \rfloor$
- B: número de árvores a serem construídas
  - quanto maior, melhor (ver próximo slide).
- Nodesize: número de exemplos para interromper a partição de um nó.
  - Eventualmente, calibração pode ajudar. Default: 1
- Weights, classwt: pesos das instâncias ou das classes.
  - Útil para dados desbalanceados ou situações em que um tipo de erro é mais relevante que outro

Quantidade de Árvores x Erro



Comentários finais

- Bom desempenho em comparação com métodos baseados em árvores de classificação individuais
  Ver próximo slide (Frizzarini, 2013)
- Boa robustez para dados com alta dimensionalidade
- Possibilidade de incorporação de pesos nas classes
  - Útil para aprendizado sobre dados desbalanceados
- Tratamento de dados faltantes
  - Ver Breiman & Cutler; Witten et al (2011)
- Método de seleção de atributos baseado em RFs:
  - VSURF - Variable Selection Using Random Forests (Genuer & Poggi, 2020)
- Próximo slide: Resultados comparativos entre Random Forests e algoritmos indutores de árvores individuais
  - Células verdes: melhor resultado obtido entre todos os métodos
  - Células azuis: resultado com diferença não significante em relação ao melhor resultado ($\mathrm{p{-}valor} \geq 0.1$)

Conj. Dados	Classe Min. (%)	F-Score β = 1 (%)						AUC (%)					
		Ctree	DDBT	J48	LMT	RForest	RPART	Ctree	DDBT	J48	LMT	RForest	RPART
PageBlock	6,3	88,4	86,5	89,8	90,4	91,2	89,2	95,2	96,5	94,9	95,4	94,7	94,1
Bank	11,5	48,7	40,4	45,2	41,6	43,9	43,7	69,4	77,3	67,3	64,4	66,5	66,1
Spect	20,6	15,3	52,3	49,7	48,5	49,6	52,0	56,0	72,8	68,6	68,5	69,1	70,0
SpectF	20,6	22,3	41,6	40,0	36,2	38,9	38,7	58,0	65,8	63,5	62,0	63,2	63,3
Blood	23,8	45,7	47,1	44,1	45,0	38,0	41,7	65,3	65,8	64,4	65,3	60,7	63,5
Haberman	26,5	32,5	42,6	32,5	34,6	35,9	39,9	60,2	62,5	62,6	60,0	59,4	62,3
Planning	28,6	-	36,6	-	-	5,9	21,7	50,0	55,1	50,0	50,0	47,5	46,8
St_credit	30,0	50,4	56,0	43,1	50,4	50,9	45,0	66,0	68,8	61,3	66,1	66,9	63,1
St_credit_n	30,0	48,3	56,9	49,0	52,9	53,9	52,1	65,1	69,4	65,1	68,1	68,5	67,3
Column_2C	32,3	68,2	74,4	68,4	75,5	75,7	69,4	76,6	81,8	77,1	82,6	82,5	77,4
Monks2	32,9	-	42,6	-	21,8	70,2	50,4	50,0	53,9	50,0	57,5	79,4	65,8
TicTacToe	34,7	88,1	56,4	75,9	97,2	98,3	87,6	91,2	66,9	81,9	97,4	98,5	90,5
Magic	35,2	76,6	73,3	77,2	79,1	82,1	70,7	81,6	79,4	82,1	83,4	85,7	77,4
Ionosphere	35,9	83,0	87,1	82,8	87,5	91,0	83,5	86,4	90,3	86,6	89,6	92,4	86,8
Wdbc	37,3	90,8	90,4	92,2	96,7	94,8	88,1	93,0	92,2	93,8	97,3	95,7	90,5
St_heart	44,4	75,1	72,0	75,7	81,6	81,6	80,8	79,3	74,6	78,1	84,0	83,6	83,4
Sonar	46,6	70,6	60,7	72,9	72,4	81,3	68,2	72,6	66,9	75,0	75,2	83,0	71,4
Monks3	47,2	100,0	97,4	100,0	99,7	100,0	100,0	100,0	97,2	100,0	99,8	100,0	100,0
Chess	47,8	98,4	93,7	99,4	99,7	98,7	96,8	98,5	94,0	99,5	99,7	98,7	96,9
Monks1	50,0	80,0	80,0	94,9	93,0	100,0	83,5	75,4	75,4	94,8	93,5	100,0	83,3

Referências

- Bastos, D.G.; Nascimento, P.S.; Lauretto, M.S. Análise empírica de desempenho de quatro métodos de seleção de características para random forests. Revista Brasileira de Sistemas de Informação 7(2), 2014. http://www.seer.unirio.br/index.php/isys/article/view/3309
- Breiman, L. Random forests. Machine Learning 45(1), 5-32, 2001.
- Breiman, L.; Cutler, A. Random Forests. http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm
- Frizzarini, C. Algoritmo para indução de árvores de classificação para dados desbalanceados. Dissertação de Mestrado. Programa de Pós-Graduação em Sistemas de Informação, Universidade de São Paulo, 2013. https://teses.usp.br/teses/disponiveis/100/100131/tde-19022014-101043/publico/ClaudioFrizzarini.pdf
- Genuer, R.; Poggi, J-M. Random Forests with R. Use R! Series, Cham: Springer, 2020.
- Hastie, T.; Tibshirani, R.; Friedman (2009), J. The Elements of Statistical Learning. 2nd Edition. Springer.
- Hesterberg,T. et al. Bootstrap Methods and Permutation Tests. Companion Chapter 18 to The Practice of Business Statistics. 2003 https://www.chrisbilder.com/boot/schedule/boot_intro_pbs18.pdf