

# Aprendizado de Máquina

## Aula 13: Comitês

André C. P. L. F. de Carvalho  
ICMC/USP

[andre@icmc.usp.br](mailto:andre@icmc.usp.br)



# Tópicos a serem abordados

- Comitês
  - Ensembles
- Combinação de preditores
  - Sequencial
  - Paralela
  - Hierárquica
- Agrupamentos

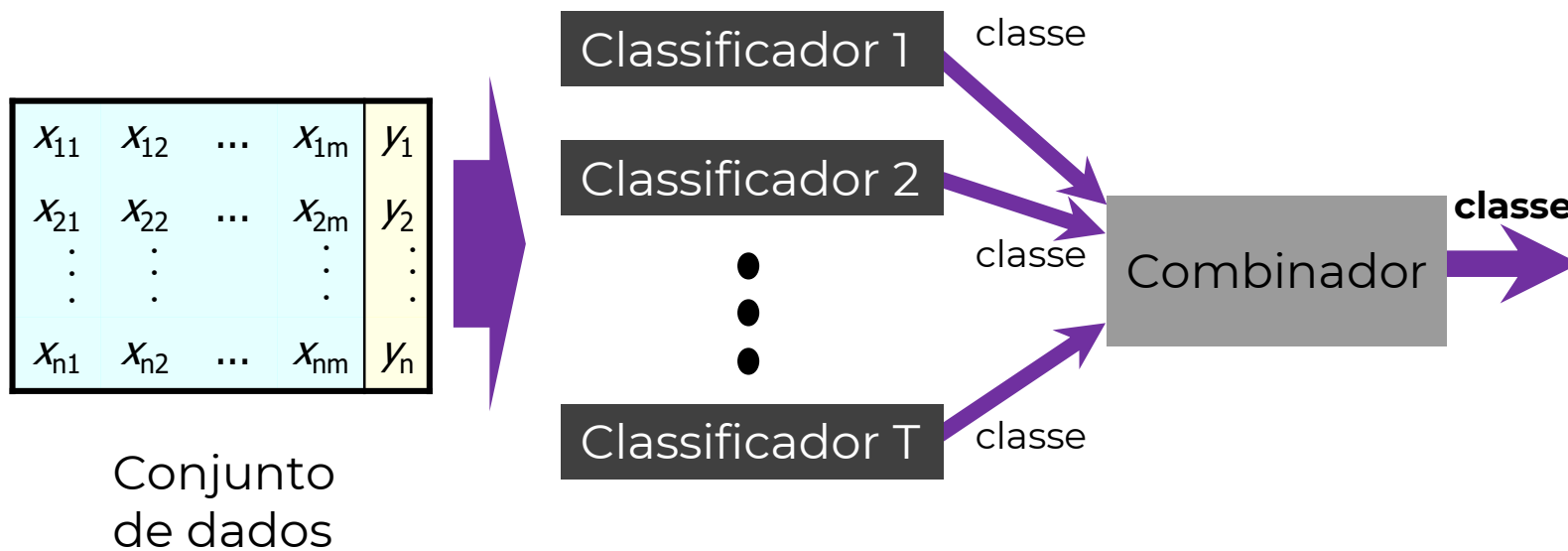
# Comitês para tarefas preditivas

- Procuram melhorar acurácia preditiva combinando predições de múltiplos estimadores
  - Classificação
    - Constroem conjunto de classificadores a partir de dados de treinamento
      - Classificadores base
    - Classe do novo exemplo é definida pela agregação da predição dos múltiplos classificadores base
  - Também podem ser usados em tarefas de regressão

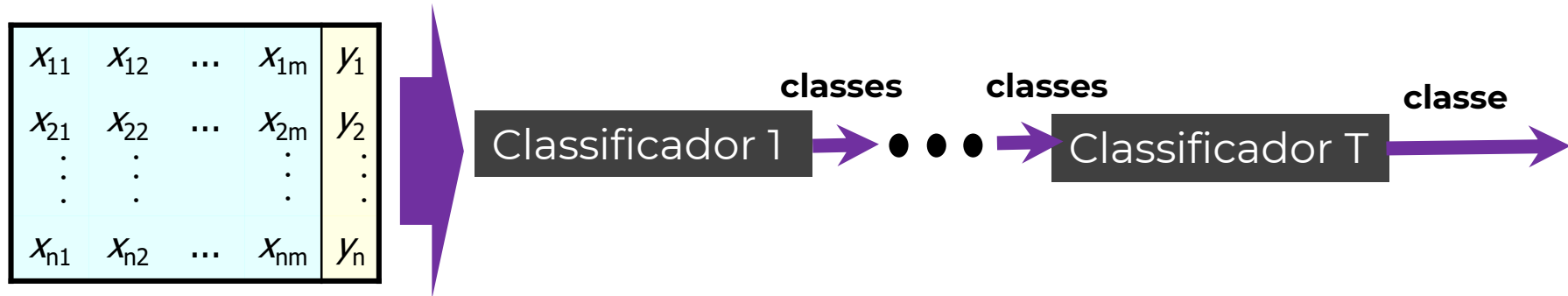
# Comitês para tarefas preditivas

- Principais abordagens de combinação
  - Paralela
  - Sequencial
  - Híbrida

# Combinação paralela

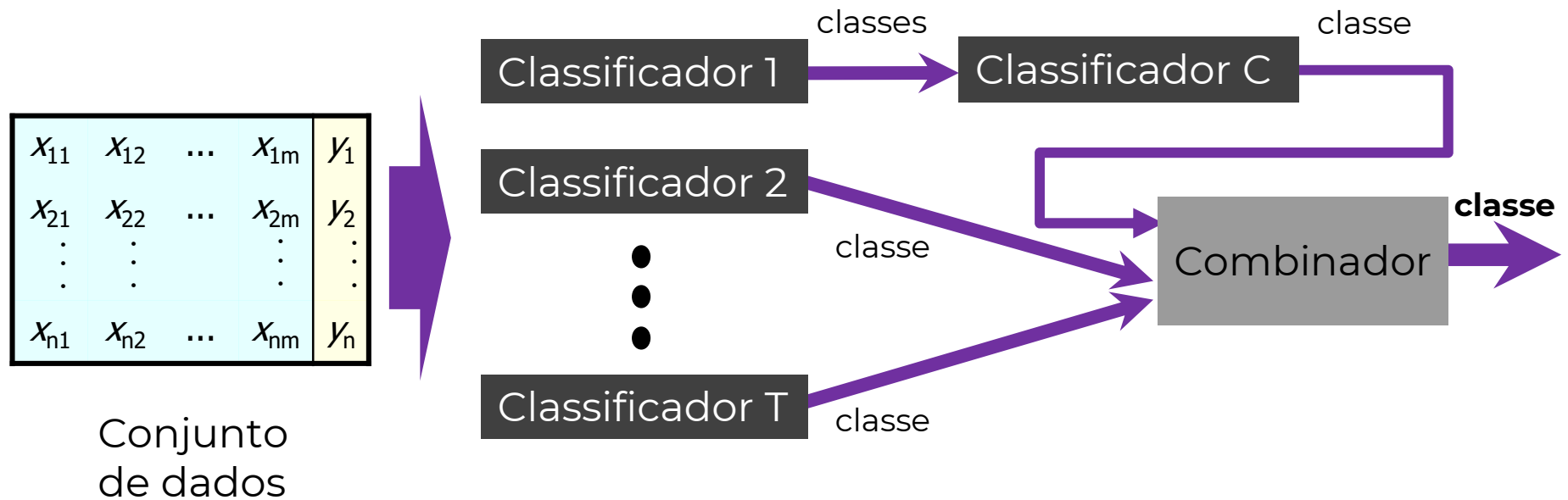


# Combinação sequencial (em cascata, pipeline)





# Combinação híbrida (hierárquica)

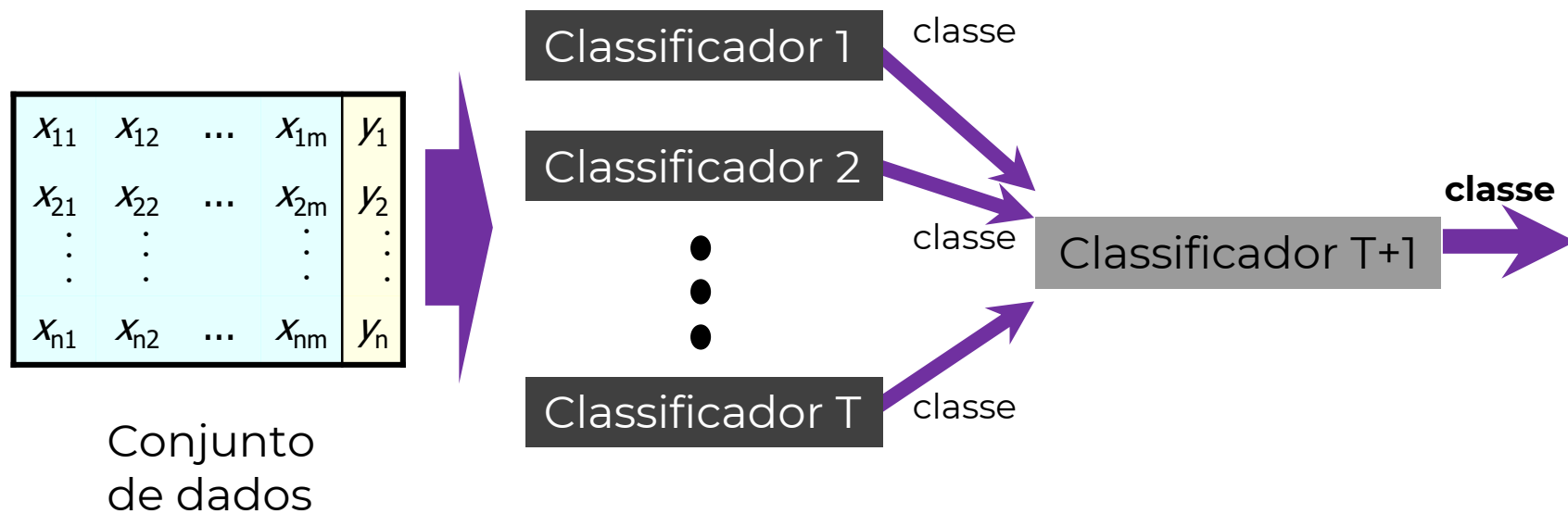


# Stacking

- Combinação hierárquica
- Induz  $T + 1$  modelos preditivos
- Um meta-modelo preditivo combina predições de  $T$  modelos base
  - Meta-modelo (meta-aprendiz): induzido por um algoritmo de AM
    - Frequentemente regressão logística
    - Aprende a fazer a combinação
  - Modelo base : induzido por um ou mais algoritmos de AM
    - Homogêneos
    - Heterógenos



# Stacking



# Bagging (Bootstrap Aggregating)

- Combinação paralela
- Induz T classificadores
  - Cada classificador é induzido por uma amostra diferente do conjunto de treinamento
    - Mesmo tamanho do conjunto original
    - Amostra definida usando bootstrap
- Classe definida por votação
- Tende a reduzir variância associada com os classificadores base
  - Reduzindo overfitting
  - Menos sensível a overfitting quando dados têm ruído
- Indicado quando o algoritmo de AM usado para gerar os classificadores (regressores) base é instável

# Bagging

- Seja um conjunto de dados de treinamento formado por 10 exemplos:  $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$

Amostra:  $x_8, x_6, x_3, x_5, x_3, x_{10}$

Amostra 1

Amostra:  $x_3, x_7, x_1, x_5, x_5, x_1$

Amostra 2

...

...

Amostra:  $x_6, x_2, x_4, x_9, x_6, x_4$

Amostra T

# Bagging

Amostra:  $x_8, x_6, x_3, x_5, x_3, x_{10}$

Amostra:  $x_3, x_7, x_1, x_5, x_5, x_1$

⋮

Amostra:  $x_6, x_2, x_4, x_9, x_6, x_4$

Classificador 1

Classificador 2

⋮

Classificador T

classe

classe

classe

Combinador

classe

# Boosting

- Combinação sequencial
- Induz T modelos preditivos
  - Classificadores base
- Família de algoritmos
  - AdaBoost é um dos mais conhecidos
    - Classificador base é uma árvore de decisão Stump
      - Um nó interno e duas folhas (algoritmo fraco)

# AdaBoost

- A cada iteração induz um classificador base
  - Pondera cada exemplo do conjunto de dados de treinamento pelo desempenho do classificador base quando aplicado a ele
    - Quanto mais difícil de ser aprendido, maior o peso associado ao exemplo (e maior a chance de ser selecionado na próxima iteração)
    - E vice-versa
- Classe definida por votação ponderada (menor o erro, maior o peso)
- Indicado quando algoritmo que gera modelo base é fraco
  - Classificador induzido tem desempenho pouco melhor que o de classificadores que classificam exemplos de forma aleatória
  - Classificador base (classificador fraco) e o classificador final (classificador forte)



# Boosting

- Seja um conjunto de dados treinamento formado por 5 exemplos:  $\{x_1, x_2, x_3, x_4, x_5\}$

Exemplos	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
Pesos atuais	0,20	0,20	0,20	0,20	0,20
Classificação	Correta	Incorreta	Correta	Correta	Incorreta
Novos pesos	0,10	0,35	0,10	0,10	0,35

Exemplos	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
Pesos atuais	0,10	0,35	0,10	0,10	0,35
Classificação	Correta	Incorreta	Correta	Incorreta	Correta
Novos pesos	0,00	0,60	0,00	0,25	0,15

# Algoritmos para comitês de árvores de decisão

- Combinam a predição de várias árvores de decisão (ADs), usando:
  - Algoritmos baseados em Bagging
    - Random forests (1995)
      - Combina as árvores no final do processo de treinamento
  - Algoritmos baseados em Boosting
    - Extreme gradient boosting (2014)
      - Começa a combinar as árvores no início do processo de treinamento
    - LightGBM (2017)
    - CatBoost (2017)

# Algoritmo random forests (RFs)

- Combina T ADs, mas pode combinar modelos gerados por qualquer algoritmo de AM
  - Baseada em Bagging
    - Cada árvore é treinada com uma amostra do conjunto de treinamento
  - Cada árvore é induzida usando um subconjunto aleatório dos atributos preditivos
    - Usado na escolha do atributo preditivo para cada nó
  - Classificação ocorre por votação
  - Hiperparâmetros definem número de ADs e número de atributos preditivos para cada AD

# Algoritmo random forests (RFs)

- Treinamento

*Para  $i = 1$  até um número  $T$  pré-definido de árvores:*

*Extrair por bootstrap uma amostra dos dados de treinamento*

*Selecionar aleatoriamente  $m$  dos  $M$  atributos preditivos*

*Enquanto um critério de parada não for atingido (número de objetos no nó)*

*Aplicar um algoritmo de indução de árvore de decisão a amostra para os  $m$  atributos*

*Resultado é um comitê de ADs*

- Teste

Uma predição para um novo objeto retorna:

Média das saídas, para regressão

Classe mais votada, para classificação

# Algoritmo random forests (RFs)

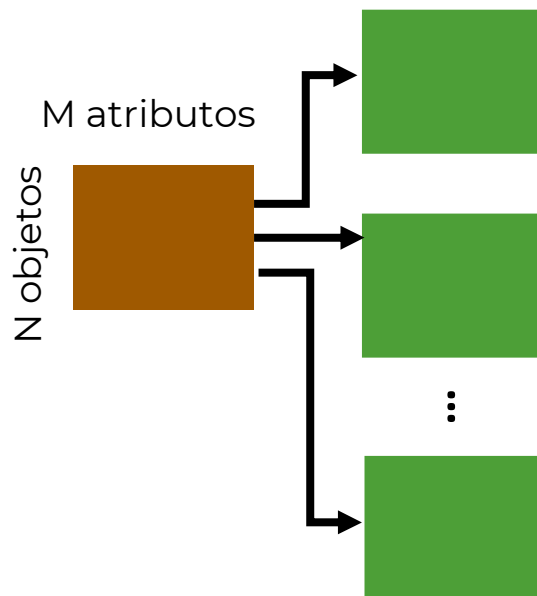
## Conjunto de dados de treinamento

M atributos

N objetos



# Algoritmo Random Forest



Cria amostras do conjunto de treinamento usando bootstrap

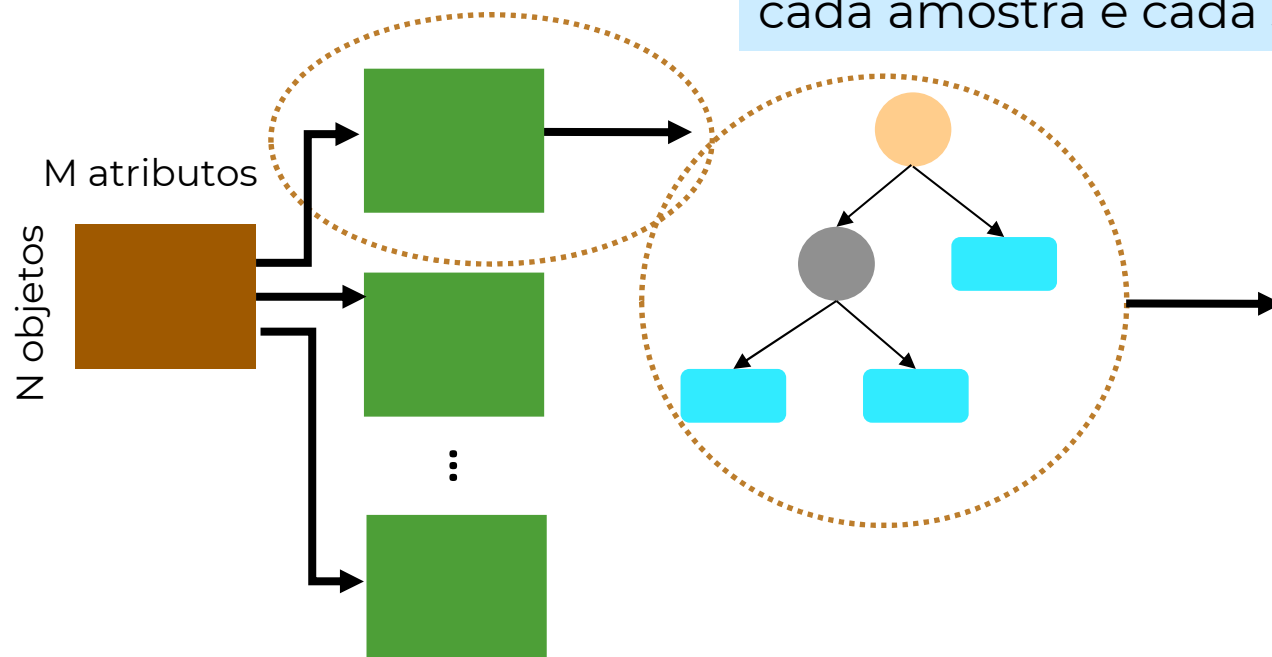
Para cada uma das amostras seleciona  $m$  dos  $M$  atributos originais,  $m < M$

Se  $m = M$ , é o mesmo que Bagging



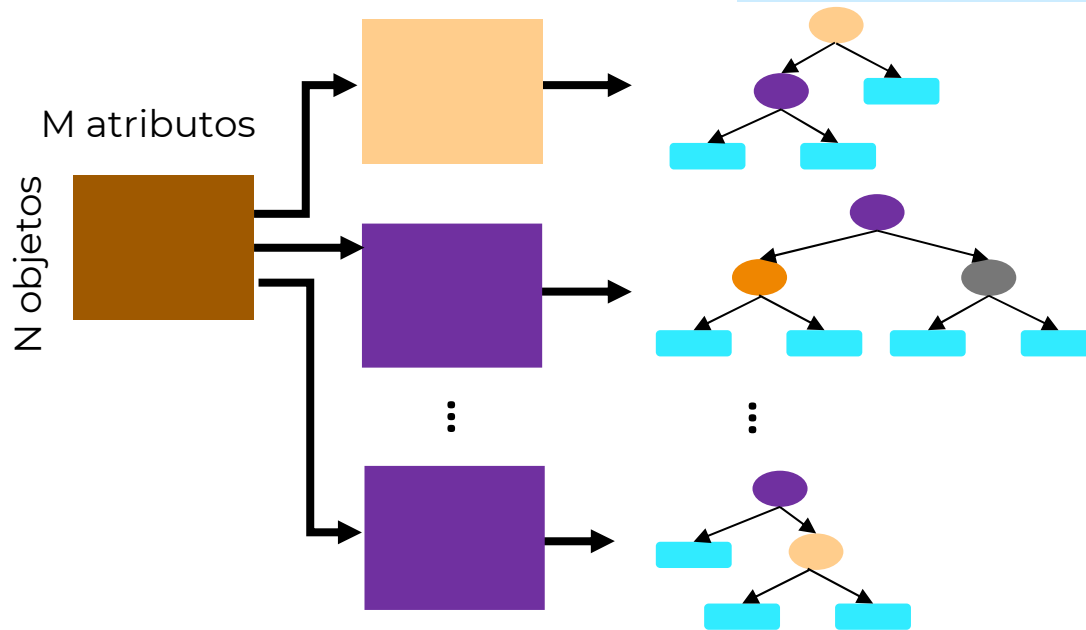
# Algoritmo Random Forest

Constrói uma árvore de decisão para cada amostra e cada subconjunto  $m$

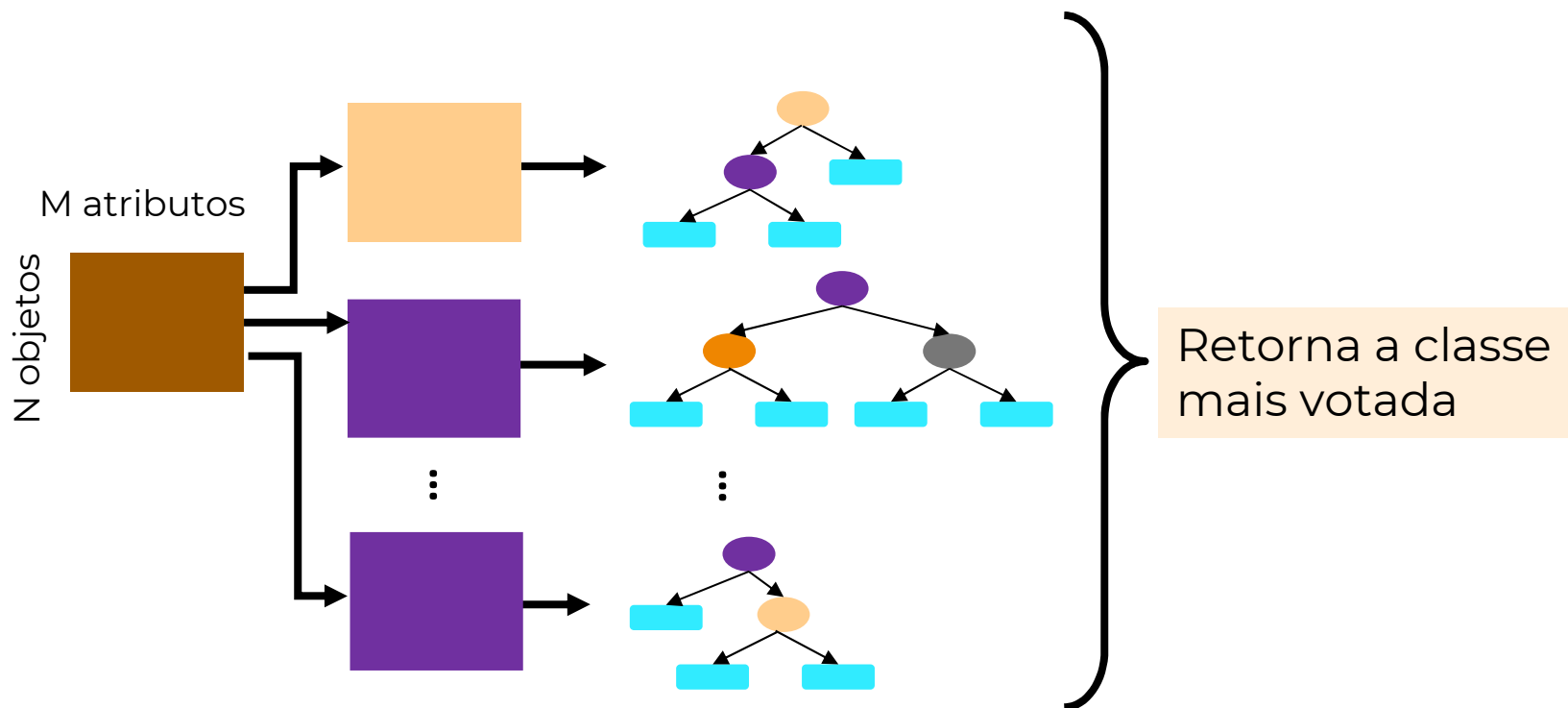


# Algoritmo Random Forest

Constrói uma árvore de decisão para cada amostra e cada subconjunto  $m$



# Algoritmo Random Forest



# Extreme Gradient Boosting

- Combina árvores induzidas por um algoritmo de AM, geralmente pelo algoritmo CART
- Treinamento aditivo
  - Induz uma árvore
    - Inclui ela no comitê
      - Induz próxima árvore
    - ...
- Pondera a resposta de cada árvore para reduzir complexidade do modelo final
  - De acordo com a acurácia preditiva da árvore

# Conclusão

- Combinação de estimadores em geral aumenta desempenho preditivo
  - E reduz variância
  - Desempenho mais estável (menor desvio padrão dos desempenhos)
- As vezes chamado de meta-aprendizado
- Regressão
  - Combinação em geral usa média simples ou ponderada

Fim do  
módulo