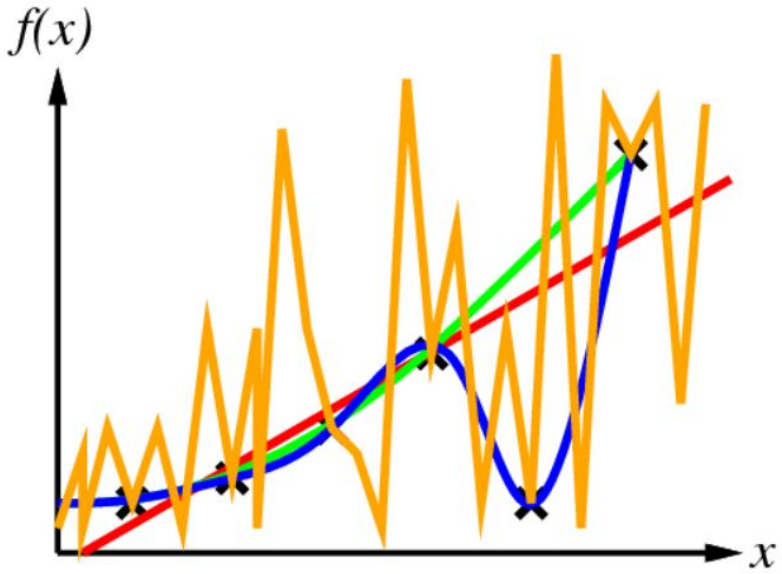
Aprendizado de Máquina - INF01017
Prof. Bruno Castro da Silva
Profa. Mariana R. Mendoza

Aula 07

Uso de múltiplos modelos em Aprendizado Supervisionado (I)

Relembrando:

Aprendizado como uma busca por hipóteses



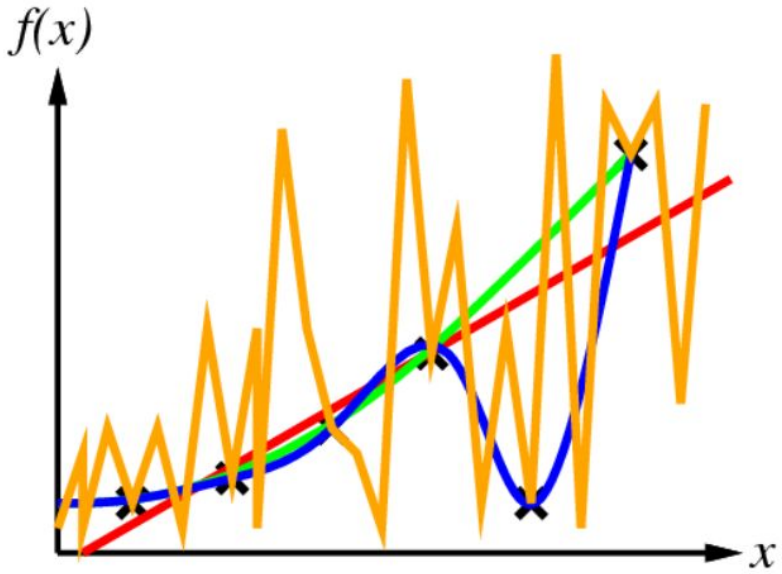
Procurar no espaço de hipóteses pela hipótese (modelo) capaz de descrever as relações entre os objetos e que melhor se ajuste aos dados

Podem haver **múltiplas hipóteses consistentes** com os dados analisados

Como determinar o melhor modelo?

Relembrando:

Aprendizado como uma busca por hipóteses



Múltiplos algoritmos disponíveis para solução de uma mesma tarefa.

Alguns algoritmos requerem otimização de parâmetros, gerando múltiplos modelos para os mesmos dados.

Como determinar o melhor modelo?

Exemplo adaptado do material didático de John Kelleher and Brian Mac Namee

Relembrando:

No Free Lunch Theorem (*Machine Learning*)

Não existe um algoritmo com melhor desempenho universal

Não é possível estabelecer *a priori* que método de ML em particular se sairá melhor na resolução de um problema específico, pois não há um algoritmo que se saia melhor para todos os problemas de decisão

Na última aula: é preciso **experimentação** e uso de métricas e estratégias de avaliação de modelos para selecionar o melhor

Relembrando:

No Free Lunch Theorem (*Machine Learning*)

Diferentes algoritmos de ML exploram...

Diferentes linguagens de representação

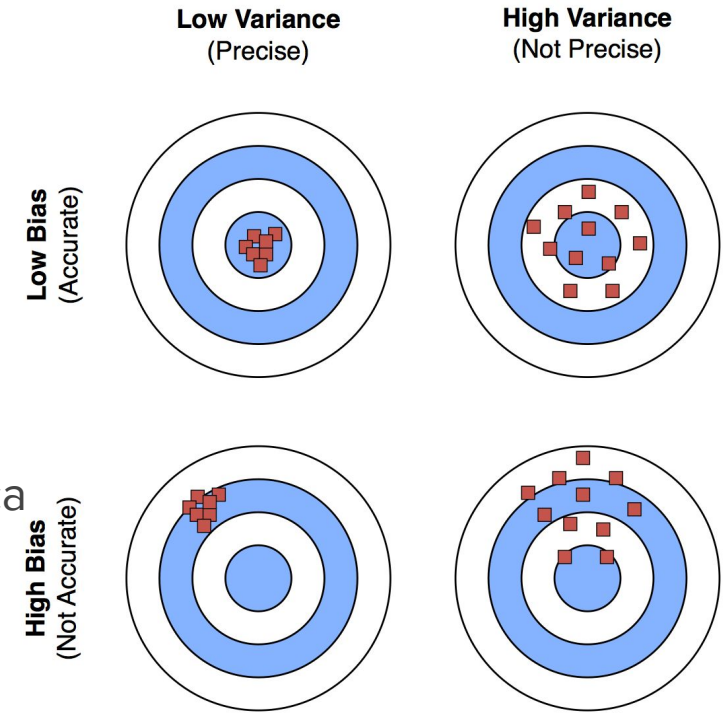
Diferentes espaços de busca por hipóteses

Diferentes funções de avaliação de hipóteses

Relembrando:

Viés-Variância da Taxa de erro

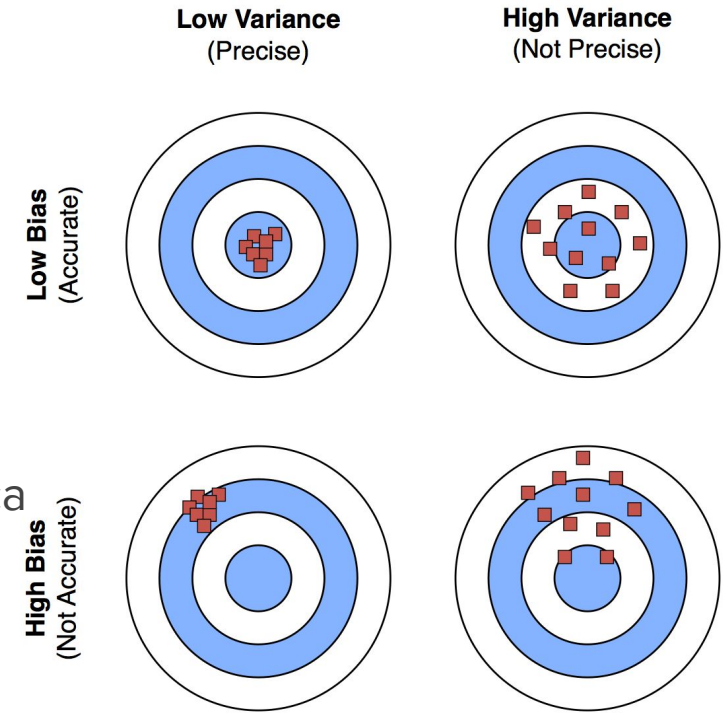
- Erros cometidos por algoritmos supervisionados podem ser decompostos em viés e variância
- Objetivo: equilíbrio entre viés e variância
- Na prática:
  - Baixo viés e alta variância
    - Ex: KNN, árvores de decisão
  - Alto viés e baixa variância
    - Ex: regressão linear, regressão logística



Relembrando:

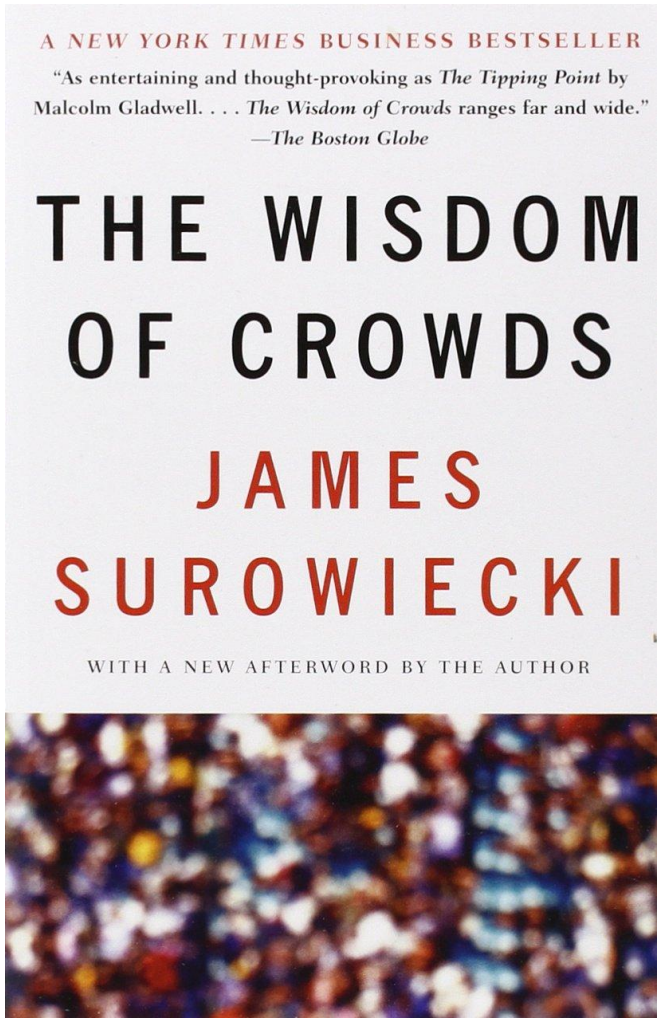
Viés-Variância da Taxa de erro

- Erros cometidos por algoritmos supervisionados podem ser decompostos em viés e variância
- Objetivo: equilíbrio entre viés e variância
- Na prática:
  - Baixo viés e alta variância
    - Ex: KNN, árvores de decisão
  - Alto viés e baixa variância
    - Ex: regressão linear, regressão logística



Existem benefícios em se combinar **múltiplos modelos**, com diferentes vieses?

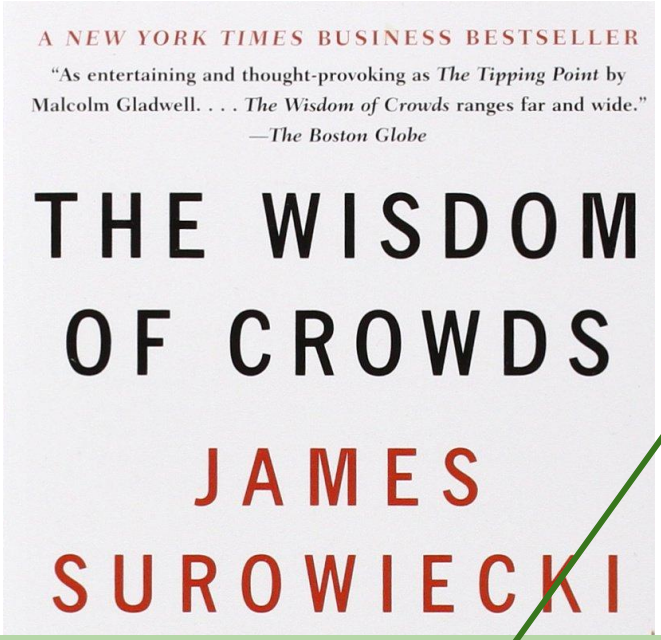
Wisdom of Crowds



Sabedoria das multidões:

O conhecimento que emerge de uma decisão coletiva frequentemente tem desempenho melhor do que aquele obtido por qualquer um dos membros individualmente, mesmo pelos especialistas

Wisdom of Crowds



Sabedoria das multidões:

O conhecimento que emerge de uma decisão coletiva frequentemente tem **desempenho melhor** do que aquele obtido por qualquer um dos membros individualmente, mesmo pelos especialistas

4 Condições que caracterizam multidões sábias:

- Diversidade de opiniões
- Independência
- Descentralização
- Agregação

Wisdom of Crowds

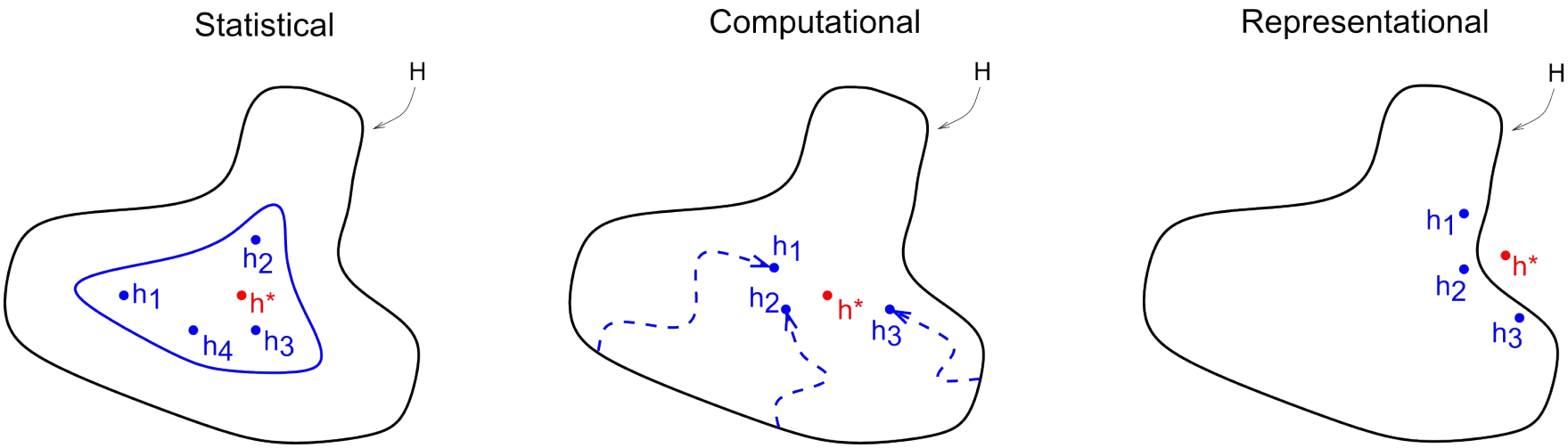
- Experimento com o pote de Jelly Beans
  - Michael Mauboussin, em Janeiro de 2007, fez um experimento com 73 estudantes da Columbia Business School
  - A partir da análise visual de um pote cheio de Jelly Beans, quantos balas existem no pote?
  - O pote tinha **1.116** jelly beans
  - Palpites variaram entre 250 - 4.100
    - Erro médio de 700 (62%)

A média aritmética entre todos os palpites dos 73 estudantes foi de **1151**: erro de 3%

Para os curiosos, um exemplo deste experimento está disponível em: https://www.youtube.com/watch?v=iOucwX7Z1HU

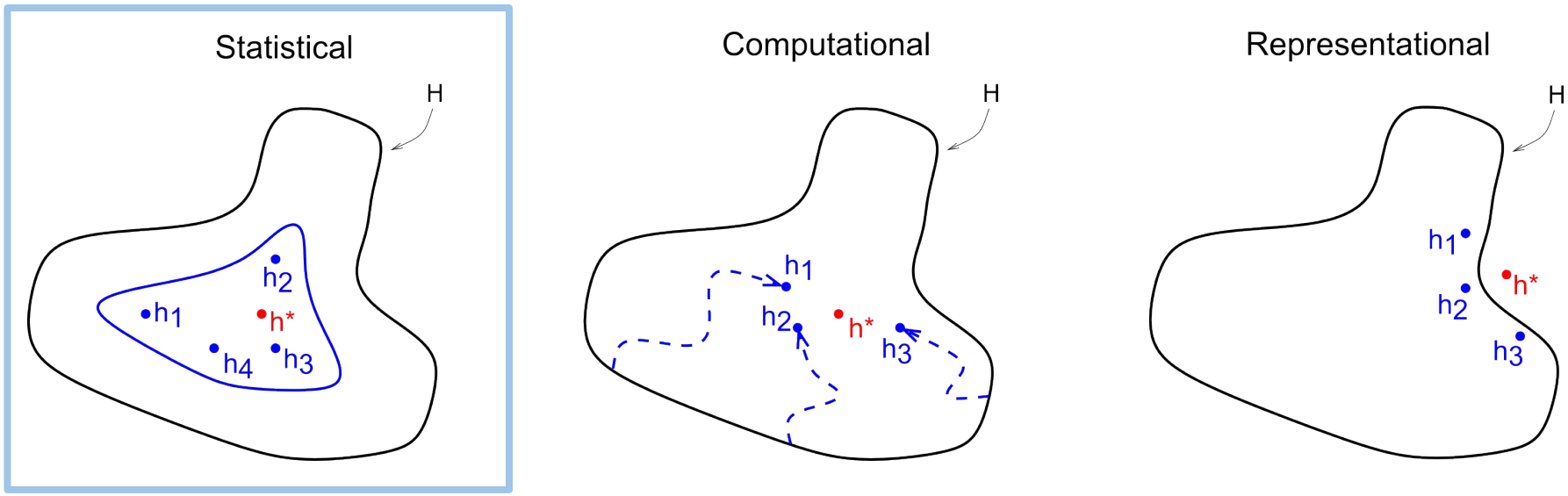
No contexto de ML:

Por que múltiplos modelos?



No contexto de ML:

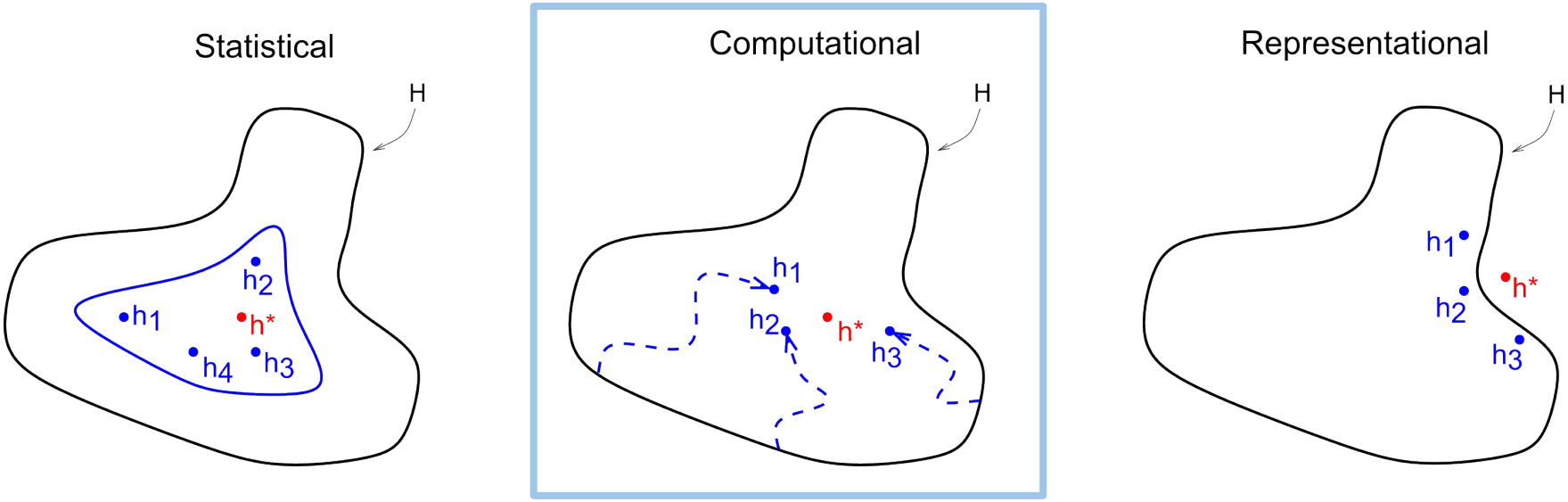
Por que múltiplos modelos?



Razão estatística: especialmente quando temos uma quantidade limitada de dados de treinamento, um algoritmo de ML pode encontrar múltiplas hipóteses com desempenho comparável para estes dados. Combinar múltiplos modelos reduz o custo de selecionar um classificador com menor poder de generalização para novos dados.

No contexto de ML:

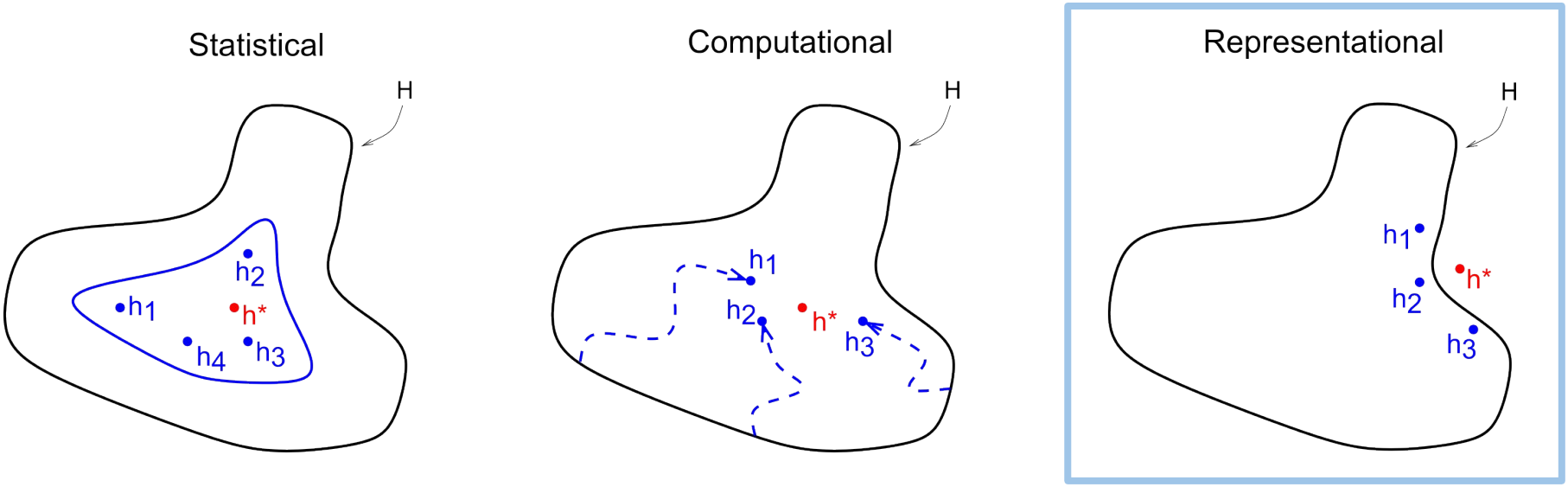
Por que múltiplos modelos?



Razão computacional: muitos algoritmos de ML são baseados em algum tipo de busca local, e assim estão suscetíveis a ficarem presos em ótimos locais. Combinar múltiplos modelos pode fornecer uma melhor aproximação do ótimo global.

No contexto de ML:

Por que múltiplos modelos?



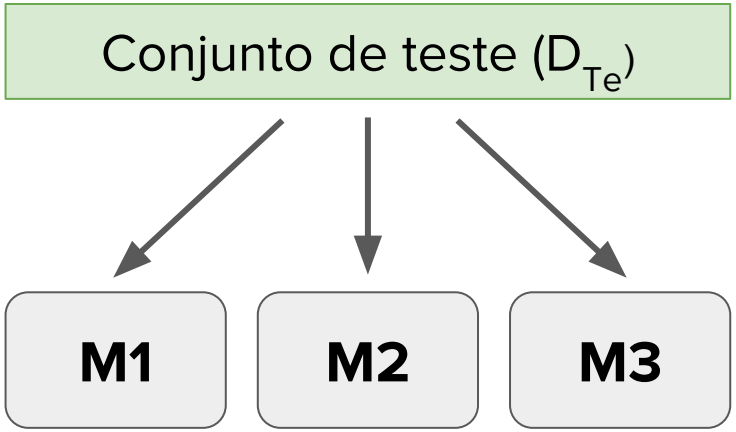
Razão representacional: em muitas aplicações, a melhor hipótese (ou hipótese verdadeira) simplesmente é muito complexa e não pode ser representada por nenhuma das possibilidades do espaço de hipóteses do algoritmo. Combinar múltiplos modelos pode permitir expandir a fronteira do espaço de hipóteses.

Intuição básica:

Múltiplos modelos em ML

D_{Te}

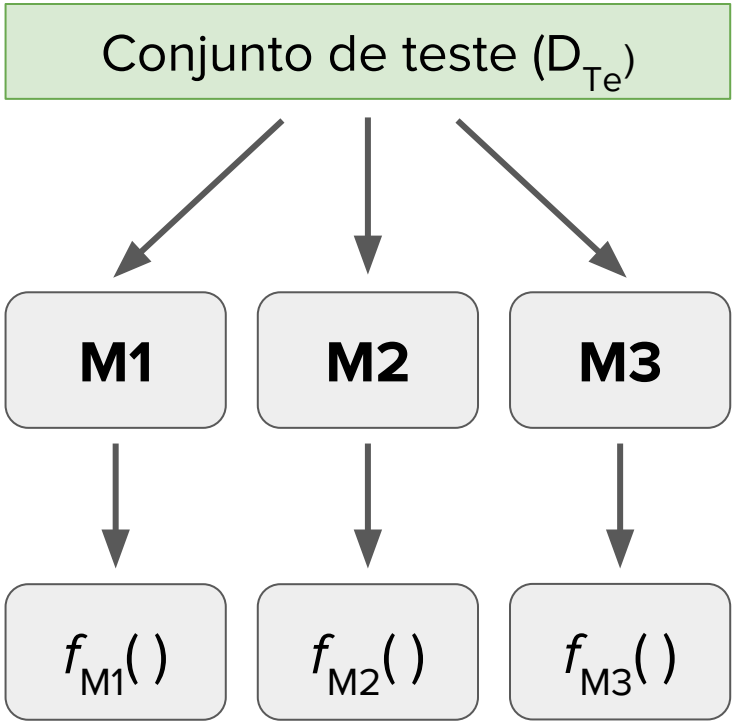
x_k	y_k
x_1	1
x_2	1
x_3	1
x_4	1
x_5	1
x_6	1
x_7	1
x_8	1
x_9	1
x_{10}	1
Acurácia	



Intuição básica:

Múltiplos modelos em ML

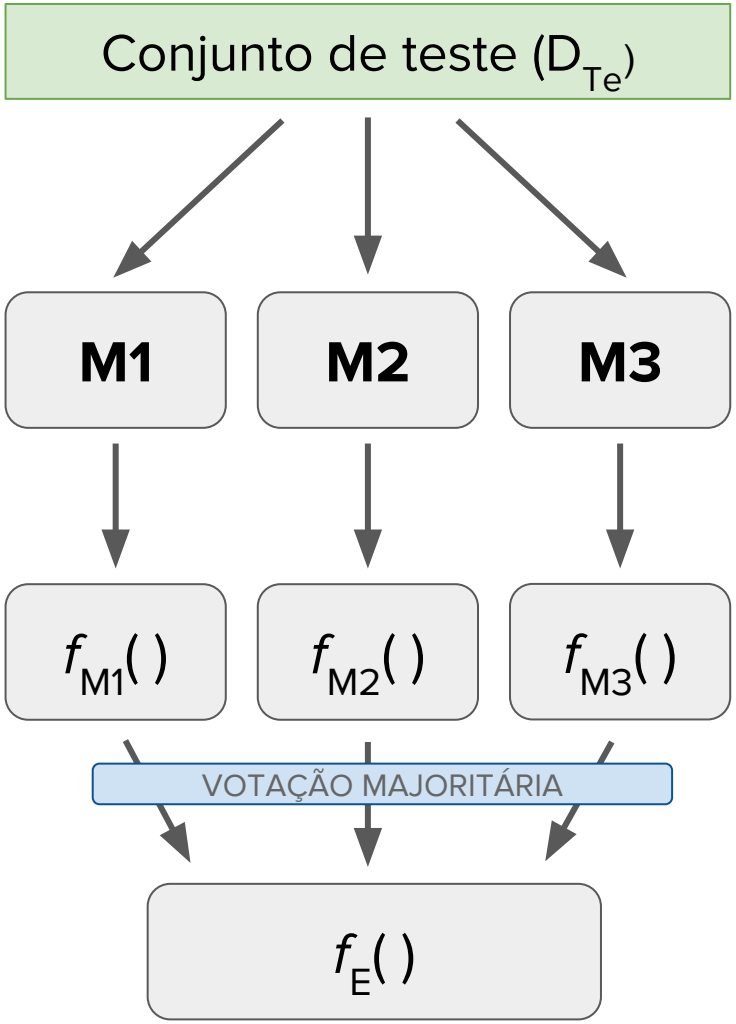
$\mathbf{x}_k$	$\mathbf{y}_k$	$f_{M1}(\,)$	$f_{M2}(\,)$	$f_{M3}(\,)$
$\mathbf{x}_1$	1	1	1	1
$\mathbf{x}_2$	1	1	0	1
$\mathbf{x}_3$	1	0	1	1
$\mathbf{x}_4$	1	1	1	1
$\mathbf{x}_5$	1	1	1	1
$\mathbf{x}_6$	1	1	1	0
$\mathbf{x}_7$	1	0	0	0
$\mathbf{x}_8$	1	1	1	0
$\mathbf{x}_9$	1	1	0	1
$\mathbf{x}_{10}$	1	0	1	1
Acurácia		70%	70%	70%



Intuição básica:

Múltiplos modelos em ML

$\mathbf{x_k}$	$\mathbf{y_k}$	$f_{M1}(\,)$	$f_{M2}(\,)$	$f_{M3}(\,)$	$f_E(\,)$
$\mathbf{x}_1$	1	1	1	1	1
$\mathbf{x}_2$	1	1	0	1	1
$\mathbf{x}_3$	1	0	1	1	1
$\mathbf{x}_4$	1	1	1	1	1
$\mathbf{x}_5$	1	1	1	1	1
$\mathbf{x}_6$	1	1	1	0	1
$\mathbf{x}_7$	1	0	0	0	0
$\mathbf{x}_8$	1	1	1	0	1
$\mathbf{x}_9$	1	1	0	1	1
$\mathbf{x}_{10}$	1	0	1	1	1
Acurácia		70%	70%	70%	**90%**

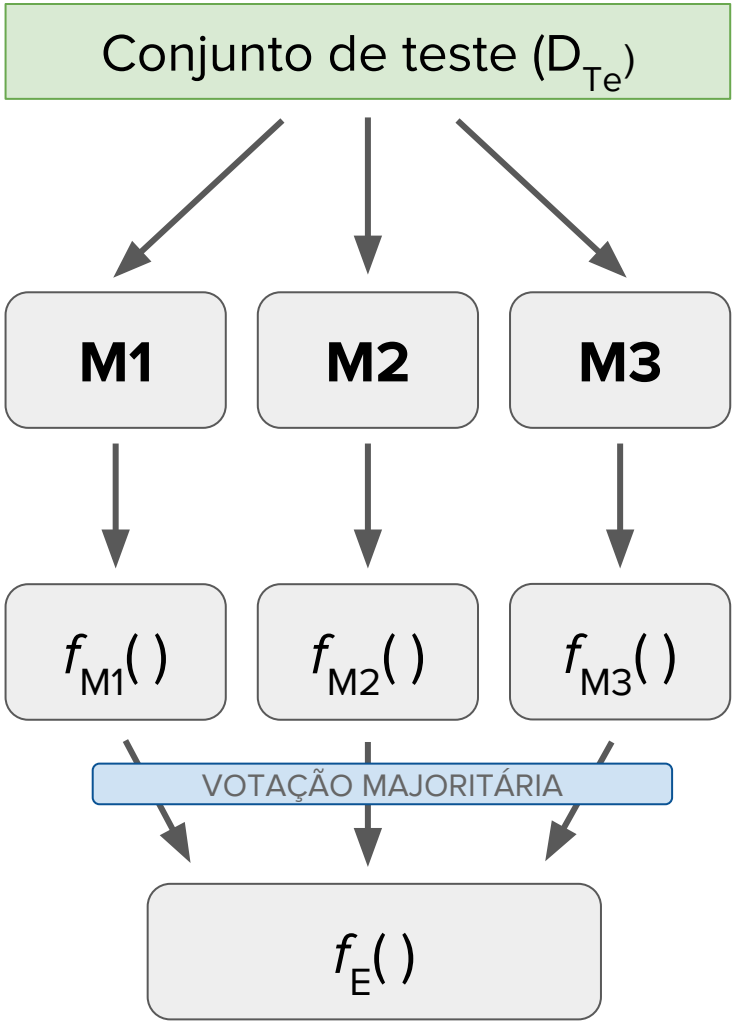


Intuição básica:

Múltiplos modelos em ML

x_k	y_k	$f_{M1}(\,)$	$f_{M2}(\,)$	$f_{M3}(\,)$	$f_E(\,)$
x_1	1	1	1	1	1
x_2	1	1	0	1	1
x_3	1	0	1	1	1
x_8	1	1	1	0	1
x_9	1	1	0	1	1
x_{10}	1	0	1	1	1
Acurácia		70%	70%	70%	**90%**

O consenso obtido com a combinação de modelos tende a ter **maior acurácia** que os classificadores bases que o compõem



Intuição básica:

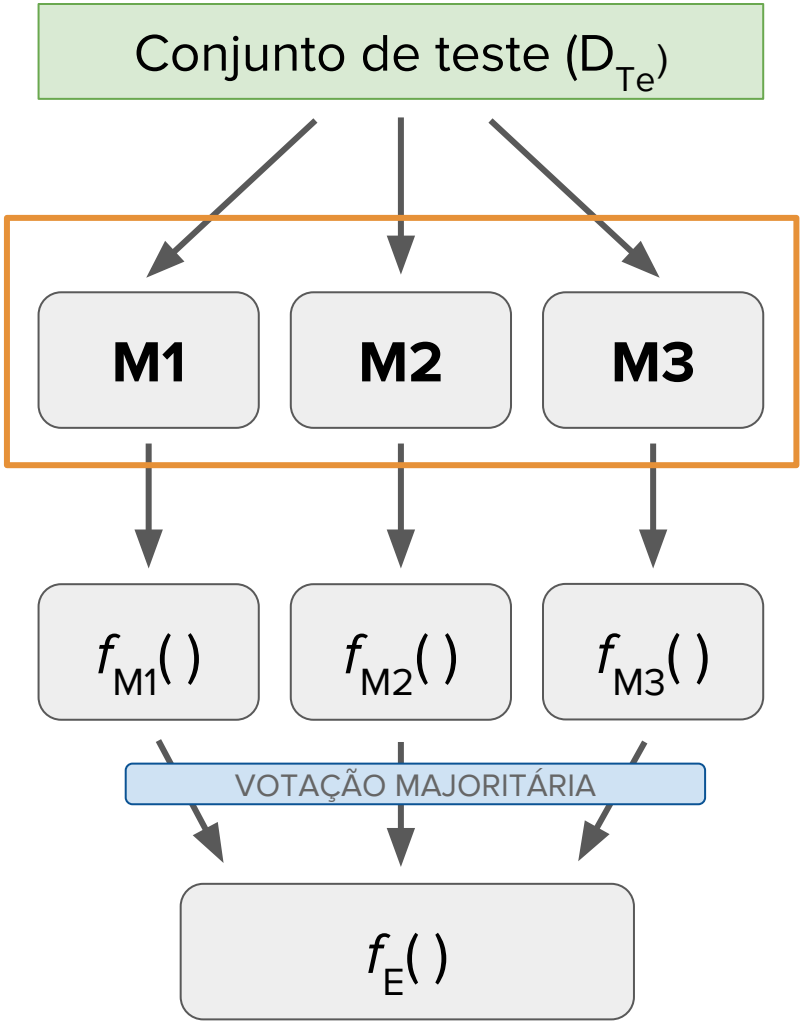
Múltiplos modelos em ML

"O consenso obtido do *ensemble* tende a ter maior acurácia que os classificadores que o compõem"

Condições para utilidade de *ensembles*:

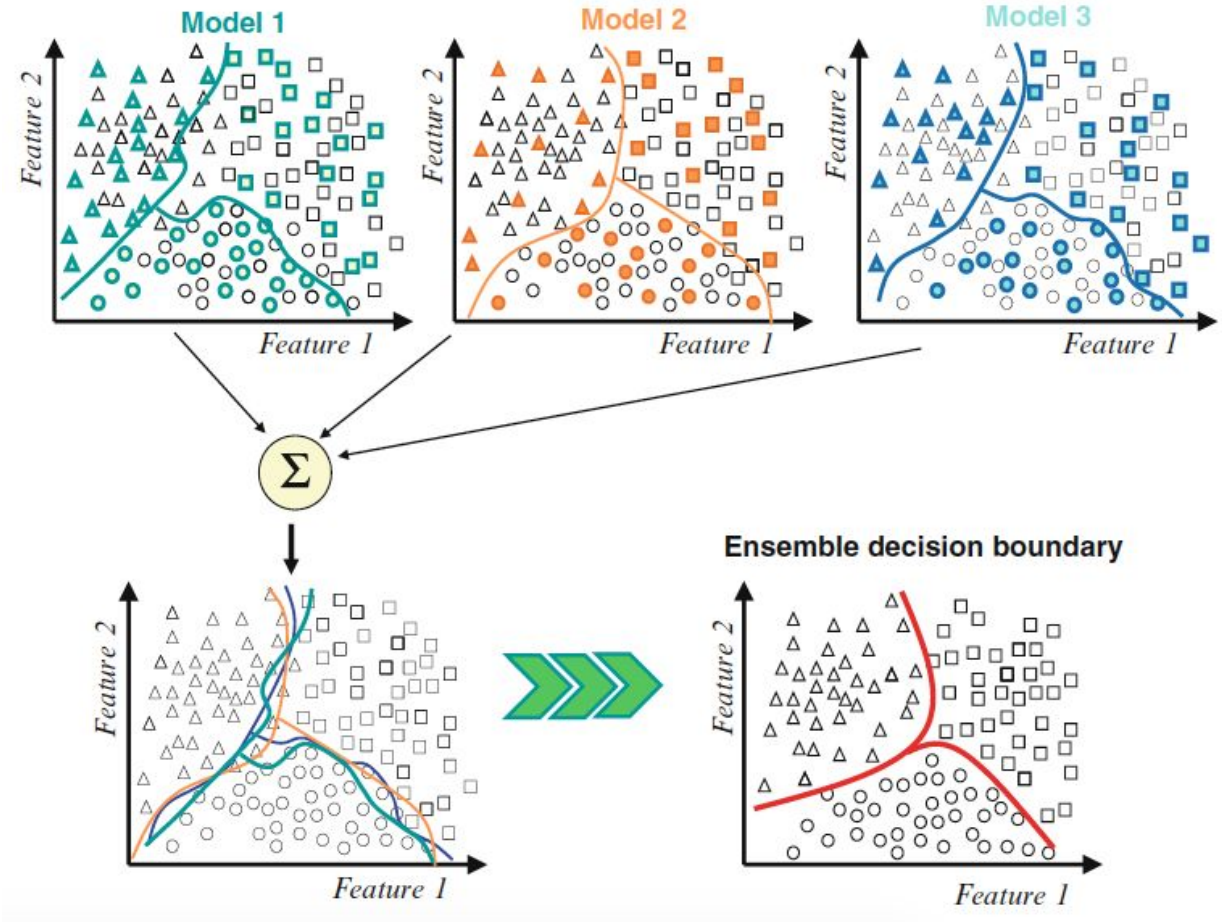
"**acurácia e diversidade**"

- **Taxa de erro** dos classificadores **< 50%**
- Classificadores com **erros de predição independentes** (erros não correlacionados ou, correlacionados negativamente)
- Nestas condições, se classificadores possuem taxa de erro semelhante (ex: 45%), o erro esperado do *ensemble* decresce linearmente com o número de modelos



Efeitos de Ensemble Learning

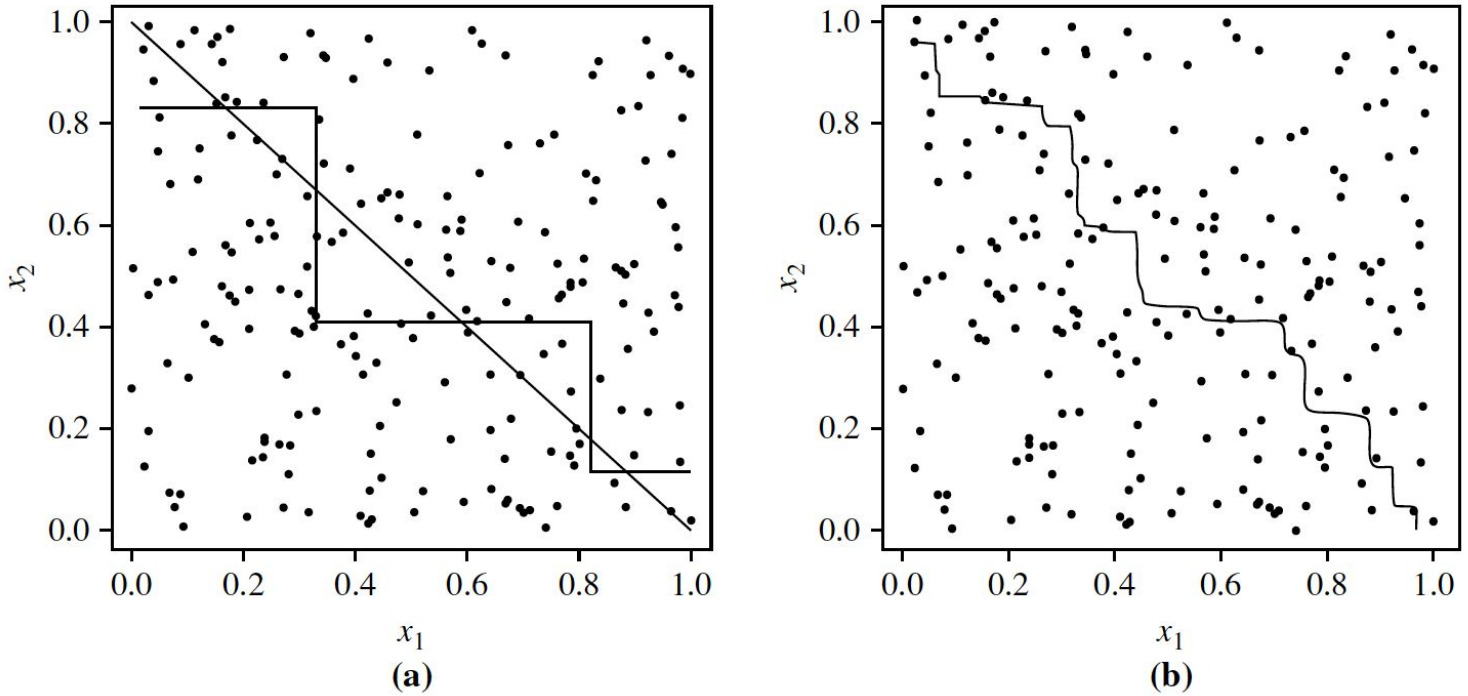
Exemplos do efeito de múltiplos classificadores na fronteira de decisão



Fonte: Polikar, 2006

Efeitos de Ensemble Learning

Exemplos do efeito de múltiplos classificadores na fronteira de decisão

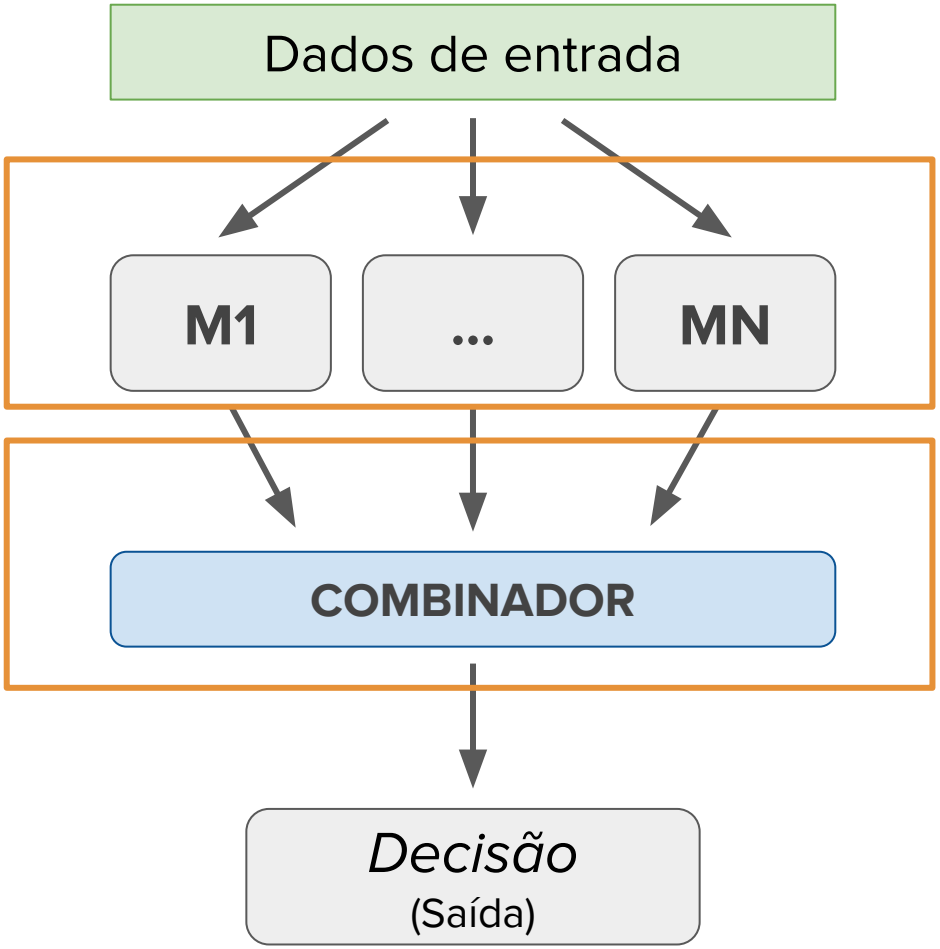


(a) Fronteira de decisão de uma única árvore de decisão

(b) Fronteira de decisão de um ensemble de árvores de decisão

Ensemble learning:

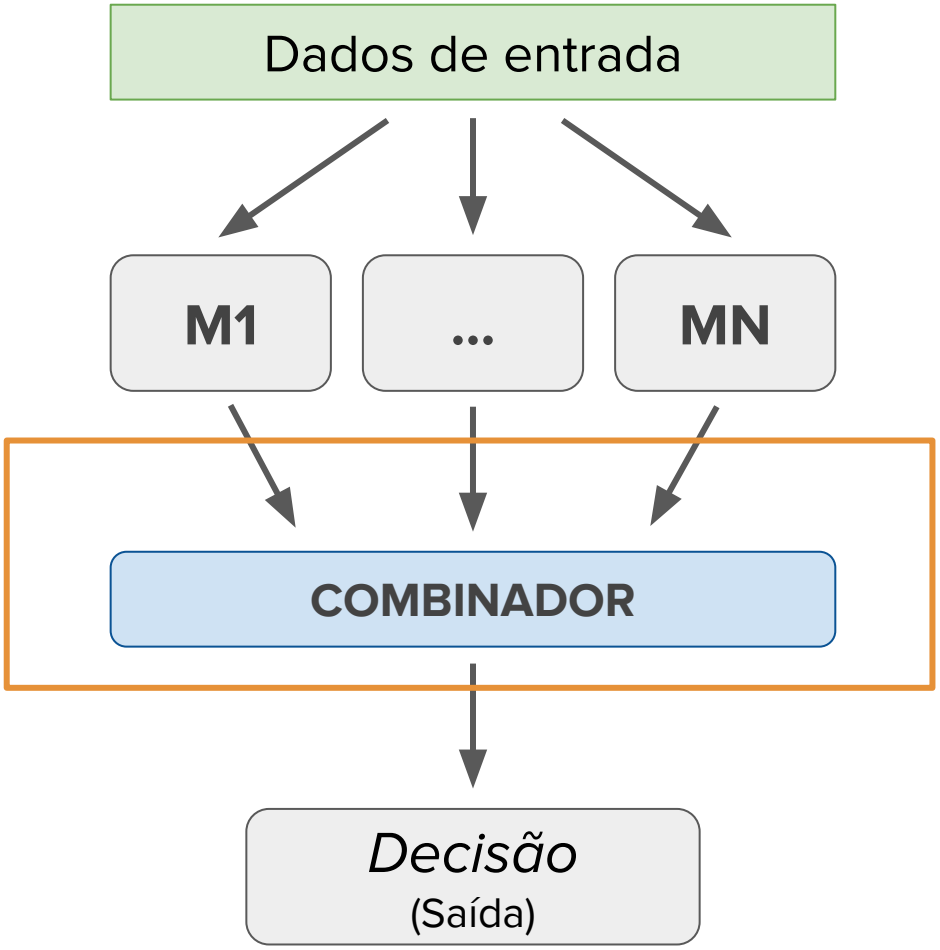
Múltiplos modelos em ML



Como gerar diferentes modelos?

Como combinar modelos?

Combinando previsões de modelos



Combinando previsões de modelos

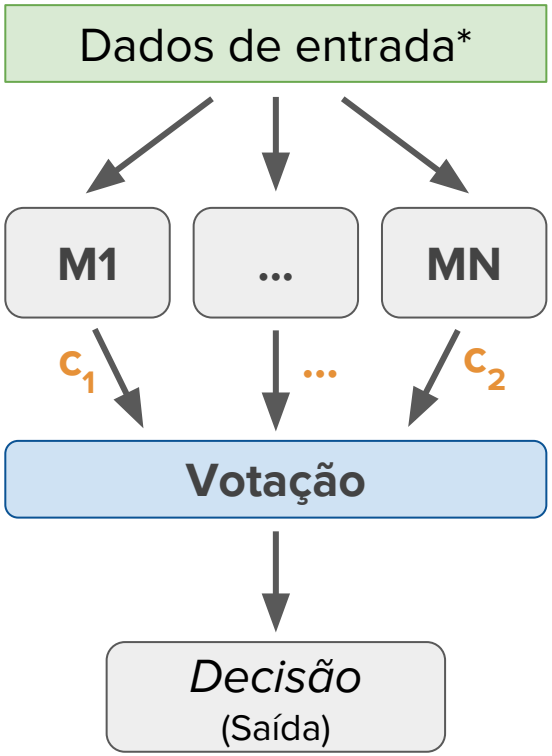
Métodos de votação vs. Métodos de Seriação

Diferem quanto ao tipo de dados ao qual se aplicam:
valores discretos vs. valores continuos

Combinando previsões de modelos

Métodos de votação vs. Métodos de Seriação

Diferem quanto ao tipo de dados ao qual se aplicam:
valores discretos vs. valores continuos



*Supondo classificação binária

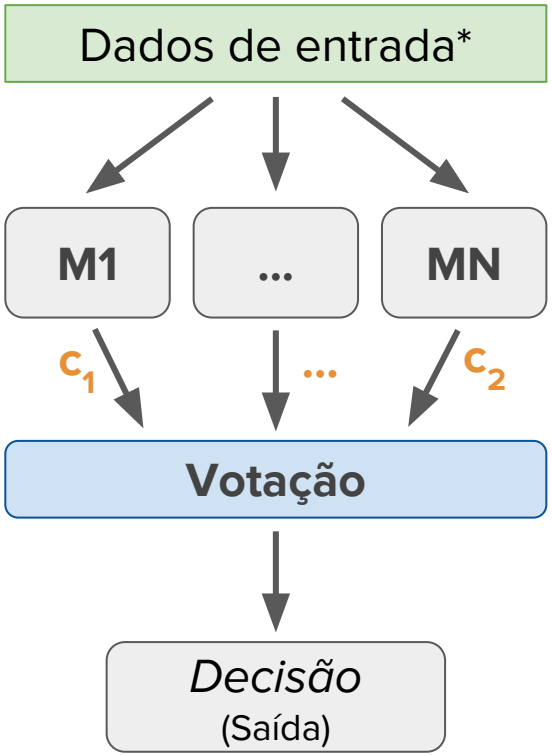
- Classificadores que geram um rótulo de classe (c_1, c_2, ... , c_k) como saída
- **Votação uniforme**: opinião de todos os classificadores contribui igualmente para a decisão final. Classe mais votada (votação majoritária) é dada como saída.
  - *d* é vetor binário, $d_{t,j}$=1 indica que classificador *t* retornou como saída a classe *j*.

$$\sum_{t=1}^{T} d_{t,J} = \max_{j=1}^{C} \sum_{t=1}^{T} d_{t,j}$$

Combinando previsões de modelos

Métodos de votação vs. Métodos de Seriação

Diferem quanto ao tipo de dados ao qual se aplicam:
valores discretos vs. valores continuos



*Supondo classificação binária

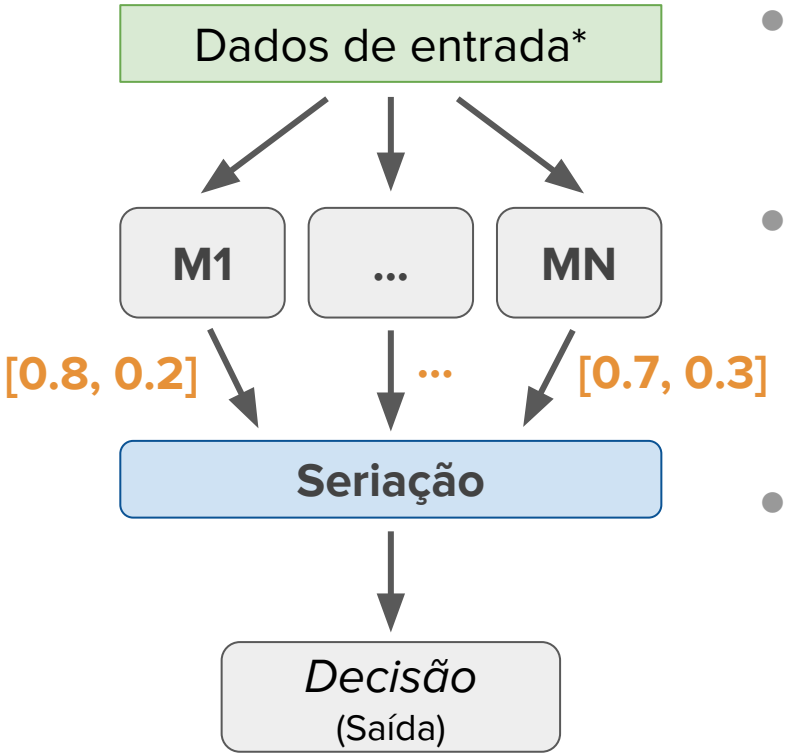
- Classificadores que geram um rótulo de classe (c_1, c_2, ... , c_k) como saída
- **Votação ponderada**: se é razoável que alguns classificadores sejam mais qualificados que outros, votos podem ser ponderados para refletir essa condição. Pesos (w_t) derivados da avaliação do modelo sobre dados de treinamento ou de validação

$$\sum_{t=1}^{T} w_t d_{t,J} = \max_{j=1}^{C} \sum_{t=1}^{T} w_t d_{t,j}$$

Combinando previsões de modelos

Métodos de votação vs. Métodos de Seriação

Diferem quanto ao tipo de dados ao qual se aplicam:
valores discretos vs. valores continuos



- Classificadores que geram como saída uma estimativa de probabilidade de uma entrada pertencer a cada uma das classes possíveis
- A fusão de classificadores probabilísticos é feita com base em regras de combinação algébrica
  - Soma, produto, máximo, mínimo, mediana..
- A classe selecionada como saída é aquela que maximiza o valor de fusão calculado de acordo com a regra escolhida

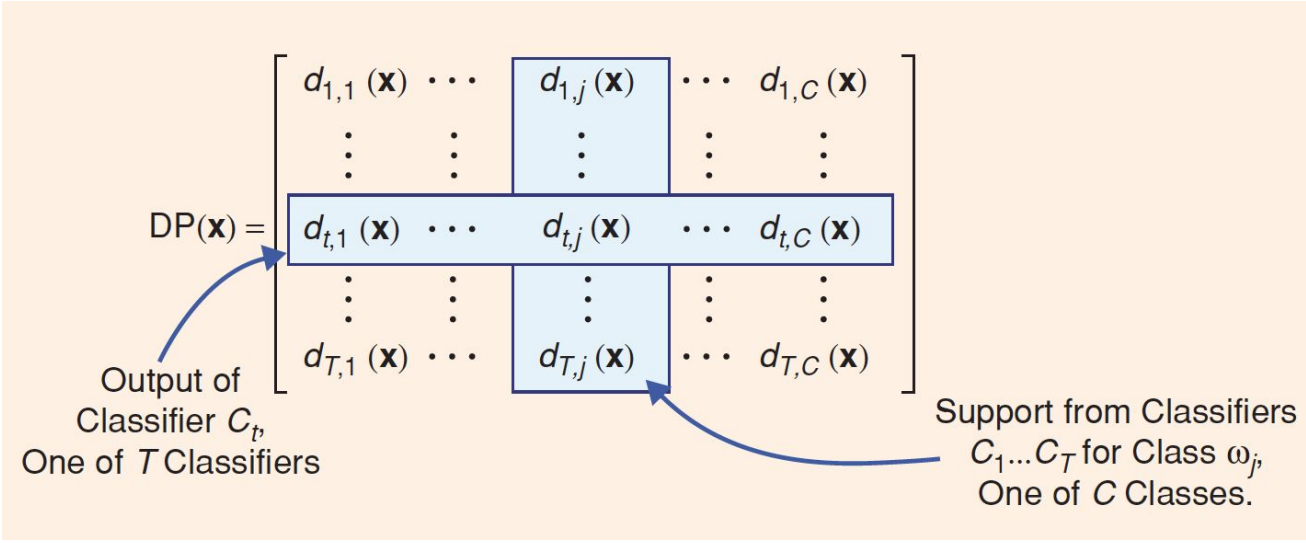
*Supondo classificação binária

Combinando previsões de modelos

Métodos de votação vs. Métodos de Seriação

Diferem quanto ao tipo de dados ao qual se aplicam: valores discretos vs. valores continuos

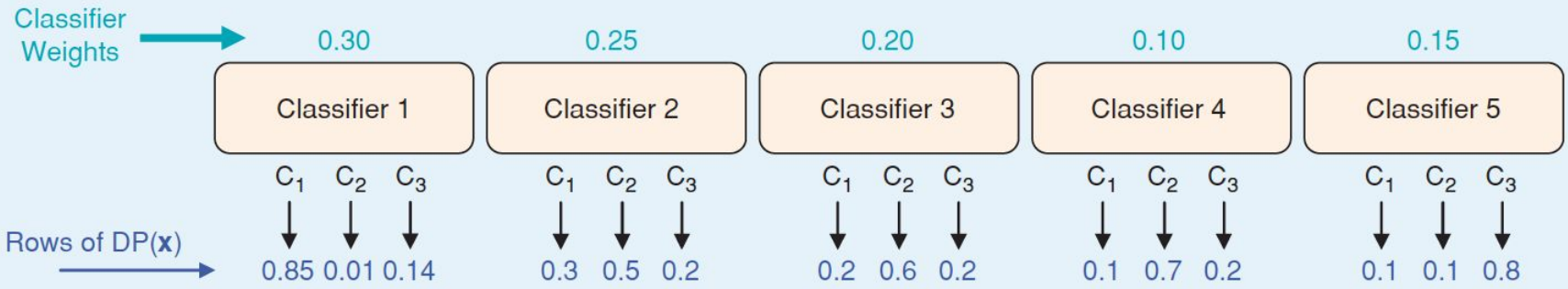
Decision profile: sumariza a matriz de suporte para cada combinação classificador *t* e classe *j* em um *ensemble*



Fonte: Polikar, 2006

Combinando previsões de modelos

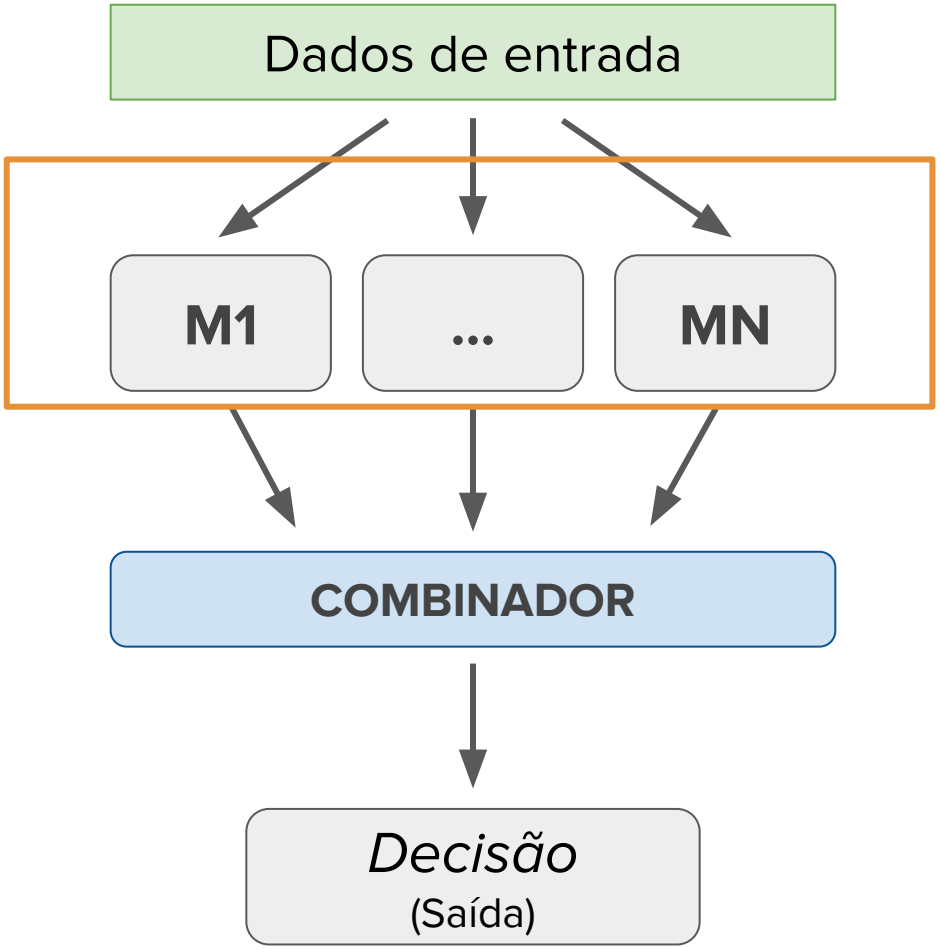
Métodos de votação vs. Métodos de Seriação



Product Rule:	**C_1: 0.0179,**	C_2: 0.00021,	C_3: 0.0009
Sum Rule:	C_1: 1.55,	**C_2: 1.91,**	C_3: 1.54
Mean Rule:	C_1: 0.310,	**C_2: 0.382,**	C_3: 0.308
Max Rule:	**C_1: 0.85,**	C_2: 0.7,	C_3: 0.8
Median Rule:	C_1: 0.2,	**C_2: 0.5,**	C_3: 0.2
Minimum Rule:	C_1: 0.1,	C_2: 0.01,	**C_3: 0.14**
Weighted Average	**C_1: 0.395,**	C_2: 0.333,	C_3: 0.272
Majority Voting:	C_1: 1,	**C_2: 3,**	C_3: 1 Vote
Weighted Majority Voting:	C_1: 0.3,	**C_2: 0.55,**	C_3: 0.15 Votes

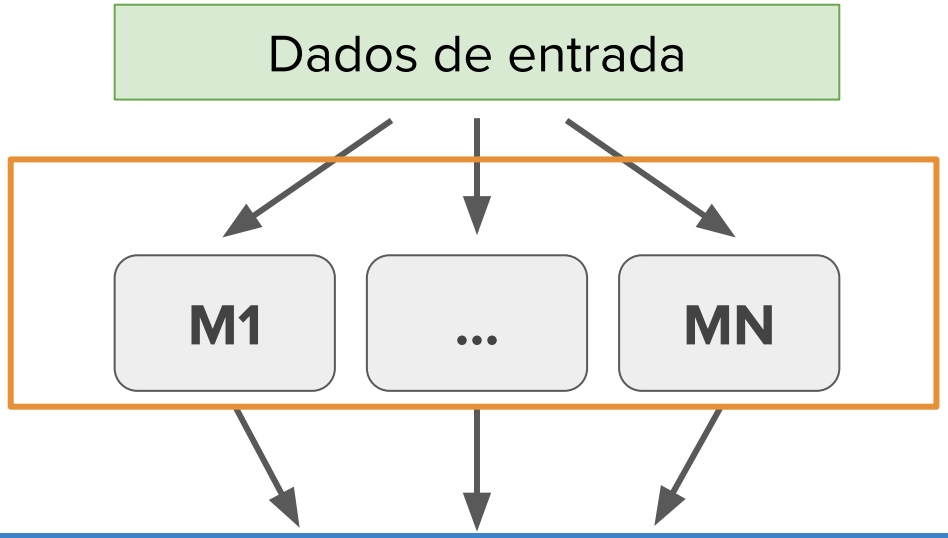
Fonte: Polikar, 2006

Construindo modelos múltiplos de classificação



Como gerar diferentes modelos?

Construindo modelos múltiplos de classificação



Como gerar diferentes modelos?

DIVERSIDADE E ACURÁCIA

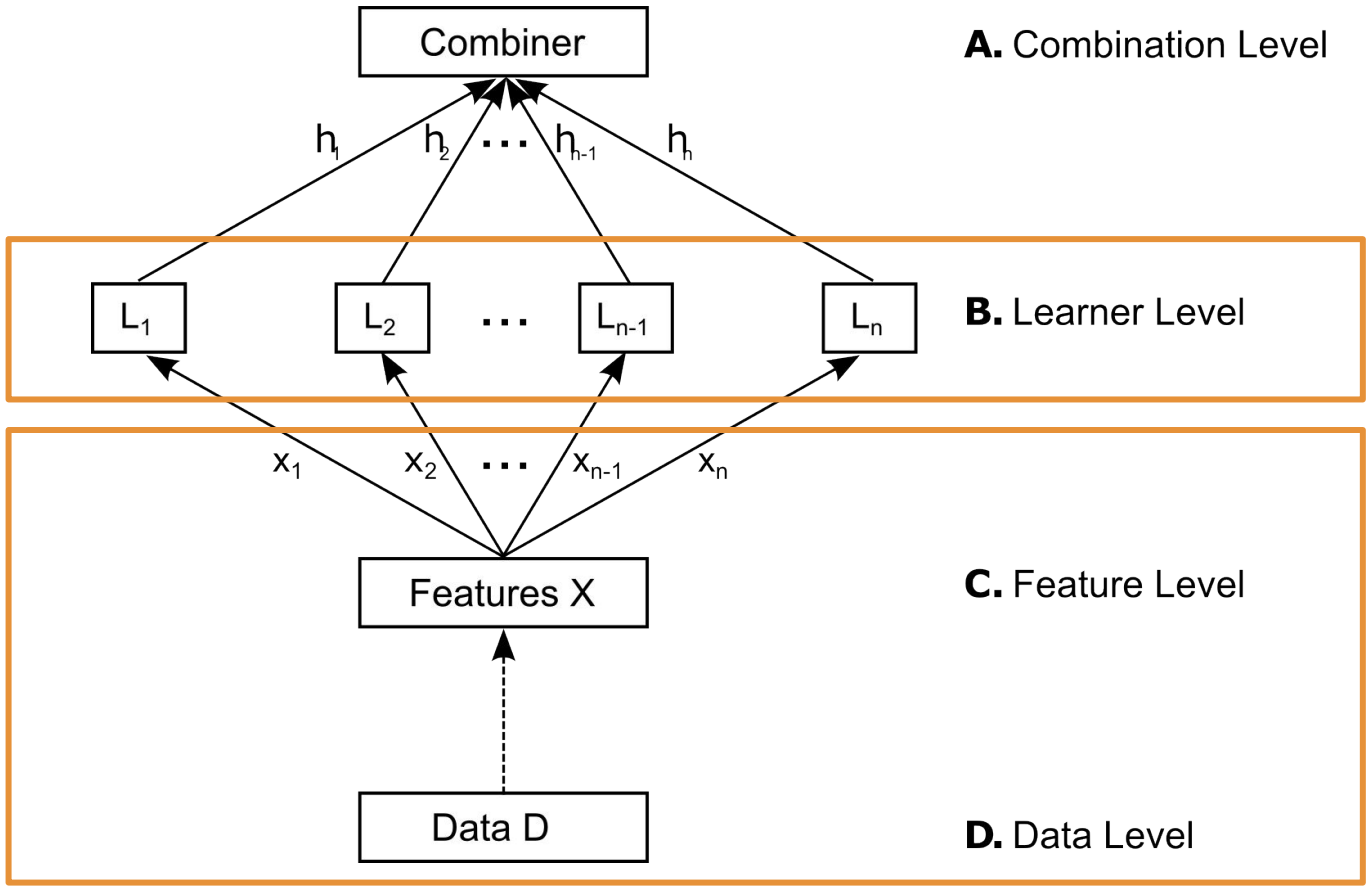
Como os modelos individuais (classificadores bases) serão gerados? Como serão diferentes entre eles?

Objetivo é melhorar diversidade do *ensemble*, mas a maioria dos métodos não maximiza explicitamente esta métrica

Construindo modelos múltiplos de classificação

Taxonomia proposta por Kuncheva, 2004

Diversidade em um ensemble pode ser introduzida em diferentes níveis



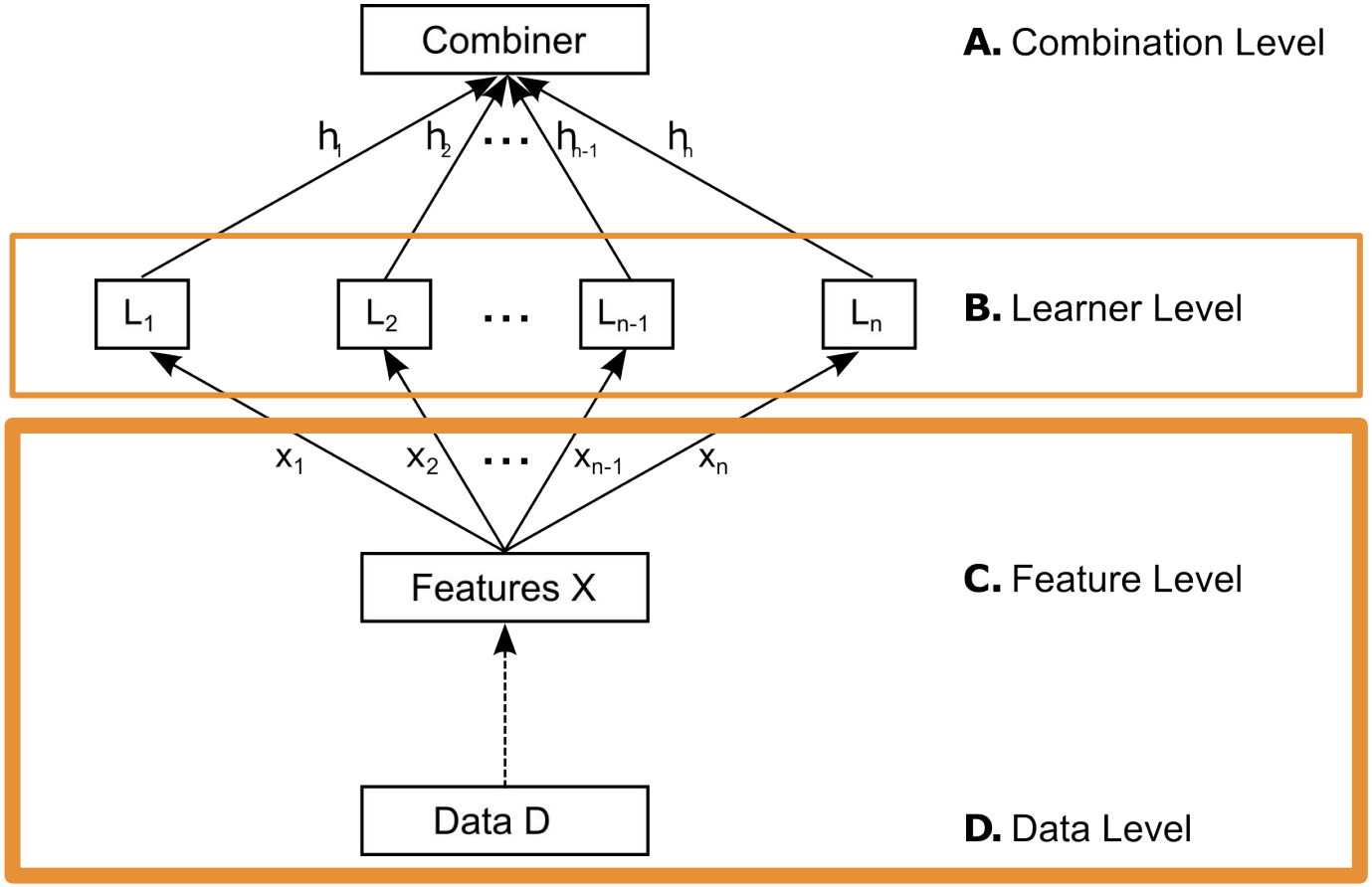
Ensemble Heterogêneo/ Homogêneo

Ensemble Homogêneo

Construindo modelos múltiplos de classificação

Taxonomia proposta por Kuncheva, 2004

Métodos baseado em amostragem dos dados de treinamento



Ensemble Heterogêneo/ Homogêneo

Ensemble Homogêneo

Bagging

Bootstrap aggregating

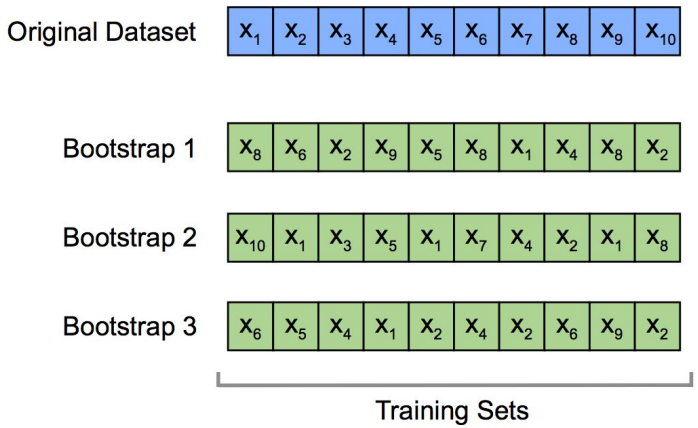
- **Bagging**: diversidade é obtida a partir de múltiplas réplicas do conjunto de treinamento obtidas por amostragem com reposição
  - Para um *n* grande, em média 63.2% das instâncias serão selecionadas para as réplicas dos conjuntos de treinamento

Exemplos **out-of-bag** (~ ⅓)

Bagging

Bootstrap aggregating

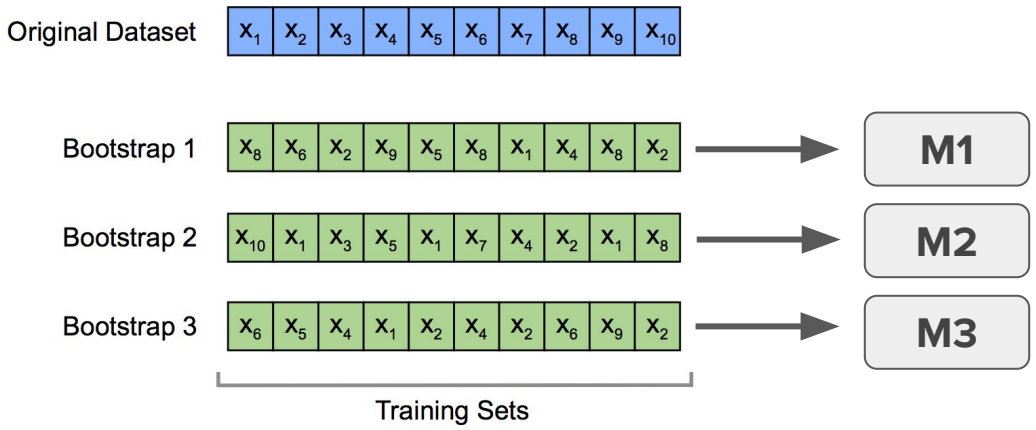
- **Bagging**: diversidade é obtida a partir de múltiplas réplicas do conjunto de treinamento obtidas por amostragem com reposição
  - Diferentes subconjuntos (*bootstrap replicados*) são aleatoriamente criados a partir dos dados originais



Bagging

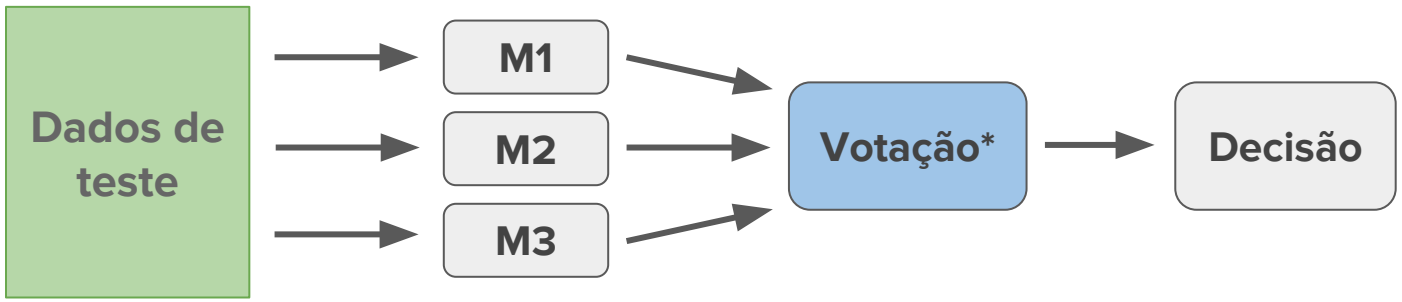
Bootstrap aggregating

- **Bagging**: diversidade é obtida a partir de múltiplas réplicas do conjunto de treinamento obtidas por amostragem com reposição
  - Diferentes subconjuntos (*bootstrap replicados*) são aleatoriamente criados a partir dos dados originais
  - Para cada bootstrap, um classificador é treinado sem variação no algoritmo (tipo e parâmetros)



Bagging
Bootstrap aggregating

- **Bagging**: diversidade é obtida a partir de múltiplas réplicas do conjunto de treinamento obtidas por amostragem com reposição
  - Diferentes subconjuntos (*bootstrap replicados*) são aleatoriamente criados a partir dos dados originais
  - Para cada bootstrap, um classificador é treinado sem variação no algoritmo (tipo e parâmetros)
  - Para classificação de novas instâncias, votação majoritária é aplicada sobre a saída de todos os modelos no *ensemble*
  - Proposto por Breiman (1996)



* alternativamente: média de probabilidades preditas, ou de saídas contínuas

Bagging

Algoritmo

Input:

- Training data S with correct labels $\omega_i \in \Omega = \{\omega_1, \ldots, \omega_C\}$ representing C classes
- Weak learning algorithm **WeakLearn,**
- Integer T specifying number of iterations.
- Percent (or fraction) F to create bootstrapped training data

Do $t = 1, \ldots, T$

1. Take a bootstrapped replica S_t by randomly drawing F percent of S.
2. Call **WeakLearn** with S_t and receive the hypothesis (classifier) h_t.
3. Add h_t to the ensemble, $\mathbf{E}$.

End

Test: Simple Majority Voting – Given unlabeled instance **x**

1. Evaluate the ensemble $\mathbf{E} = \{h_1, \ldots, h_T\}$ on **x**.
2. Let $v_{t,j} = \begin{cases} 1, & \text{if } h_t \text{ picks class } \omega_j \\ 0, & \text{otherwise} \end{cases}$ (8)

   be the vote given to class ω_j by classifier h_t.
3. Obtain total vote received by each class

$$V_j = \sum_{t=1}^{T} v_{t,j}, \quad j = 1, \ldots, C \tag{9}$$

4. Choose the class that receives the highest total vote as the final classification.

Fonte: Polikar, 2006

Bagging

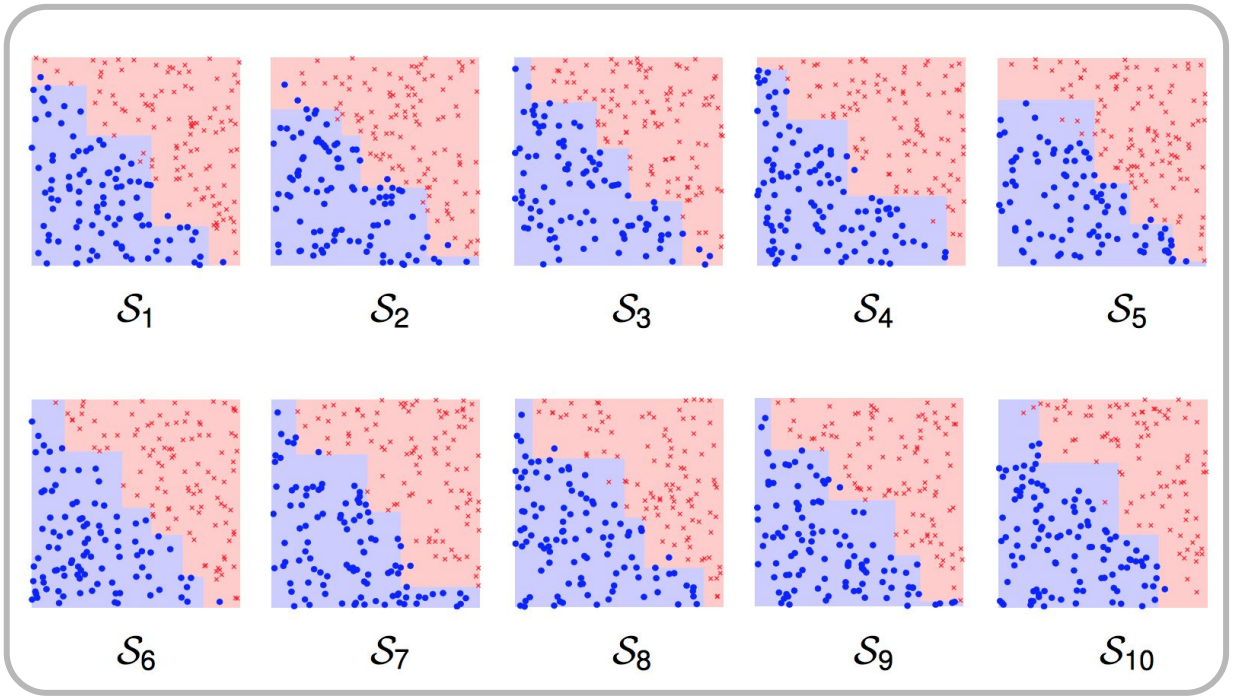
Bootstrap aggregating

- Bagging é especialmente indicado quando os dados rotulados são limitados, e para uso com algoritmos instáveis
  - Árvores de decisão, K-NN, Redes Neurais
- Procedimento que reduz a variância do classificador ao tomar a "média" entre múltiplos modelos.
- Possui pouco efeito sobre o viés
- O viés do classificar ensemble (*bagged*) pode ser marginalmente inferior que o de seus classificadores base, mas o foco da metodologia é a redução significativa da variância

Bagging

Bootstrap aggregating

- Com árvores de decisão:
  - Mudanças nos dados de treinamento afetam o processo de divisão de nós, alterando a escolha do melhor atributo ou o ponto de corte para atributos contínuos
  - Alterações refletidas na fronteira de decisão (S_1 a S_{10} são boostraps)



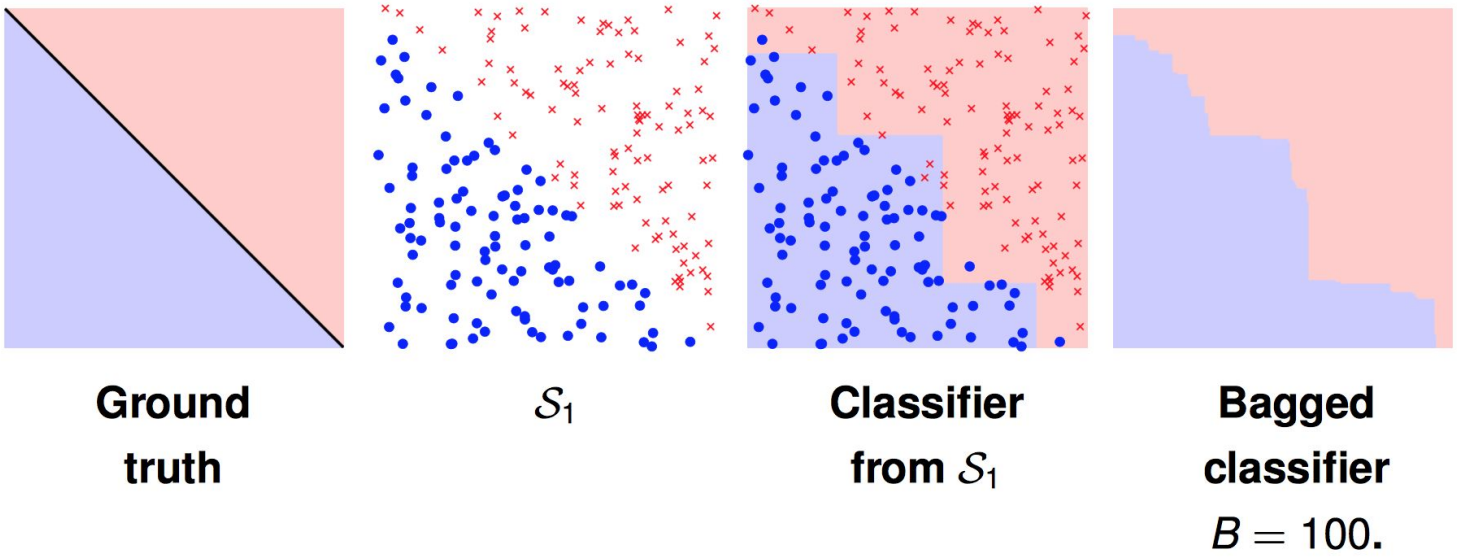
Fronteira de decisão esperada

Fonte: Sullivan, Maki, 2013

Bagging

Bootstrap aggregating

- Indicado para algoritmos de aprendizado com baixo viés, e alta variância

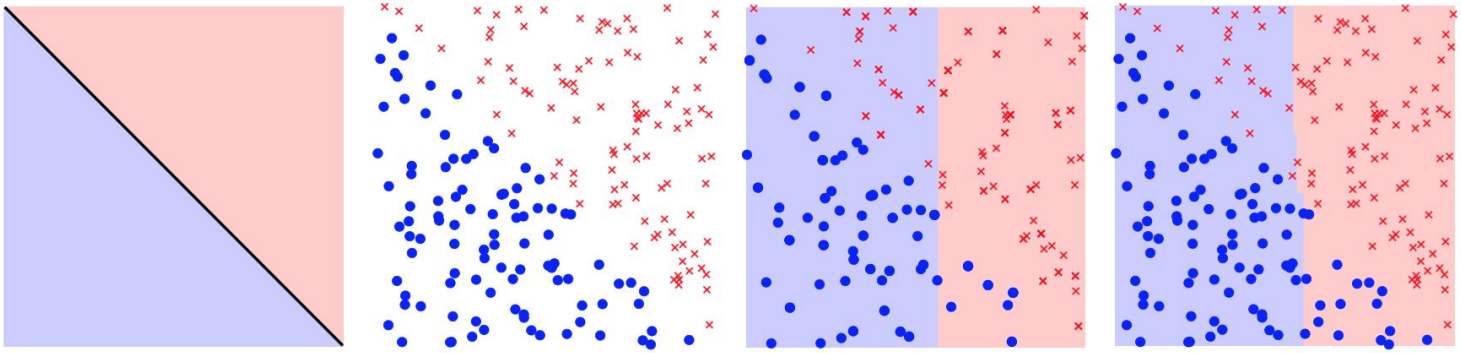


Fonte: Sullivan, Maki, 2013

Bagging

Bootstrap aggregating

- Em casos de algoritmos com alto viés e baixa variância, não há benefícios claros no uso de *bagging*
  - Lembrando: *Bagging reduz variância, tem pouco impacto no viés*

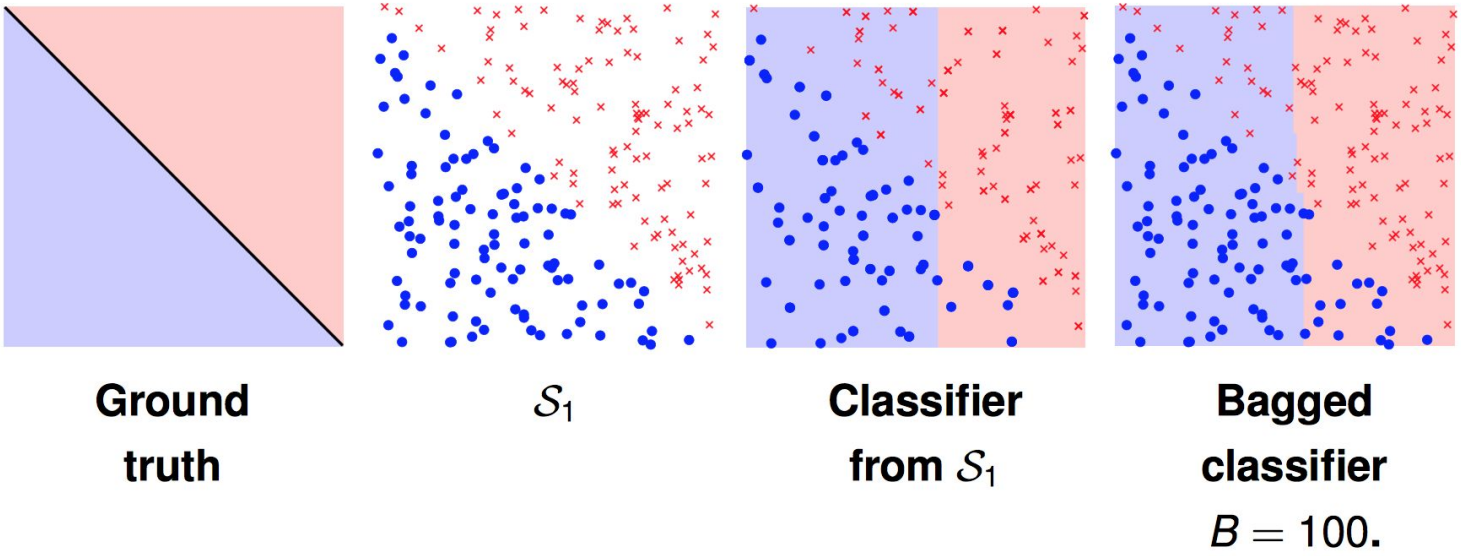


Fonte: Sullivan, Maki, 2013

Bagging

Bootstrap aggregating

- Em casos de algoritmos com alto viés e baixa variância, não há benefícios claros no uso de *bagging*
  - Lembrando: *Bagging reduz variância, tem pouco impacto no viés*



Como proceder quando temos classificadores fracos (*weak learners*), com alto viés e baixa variância?

Boosting

- *É possível criar um classificador forte a partir de várias classificadores fracos (i.e., alto viés)?*

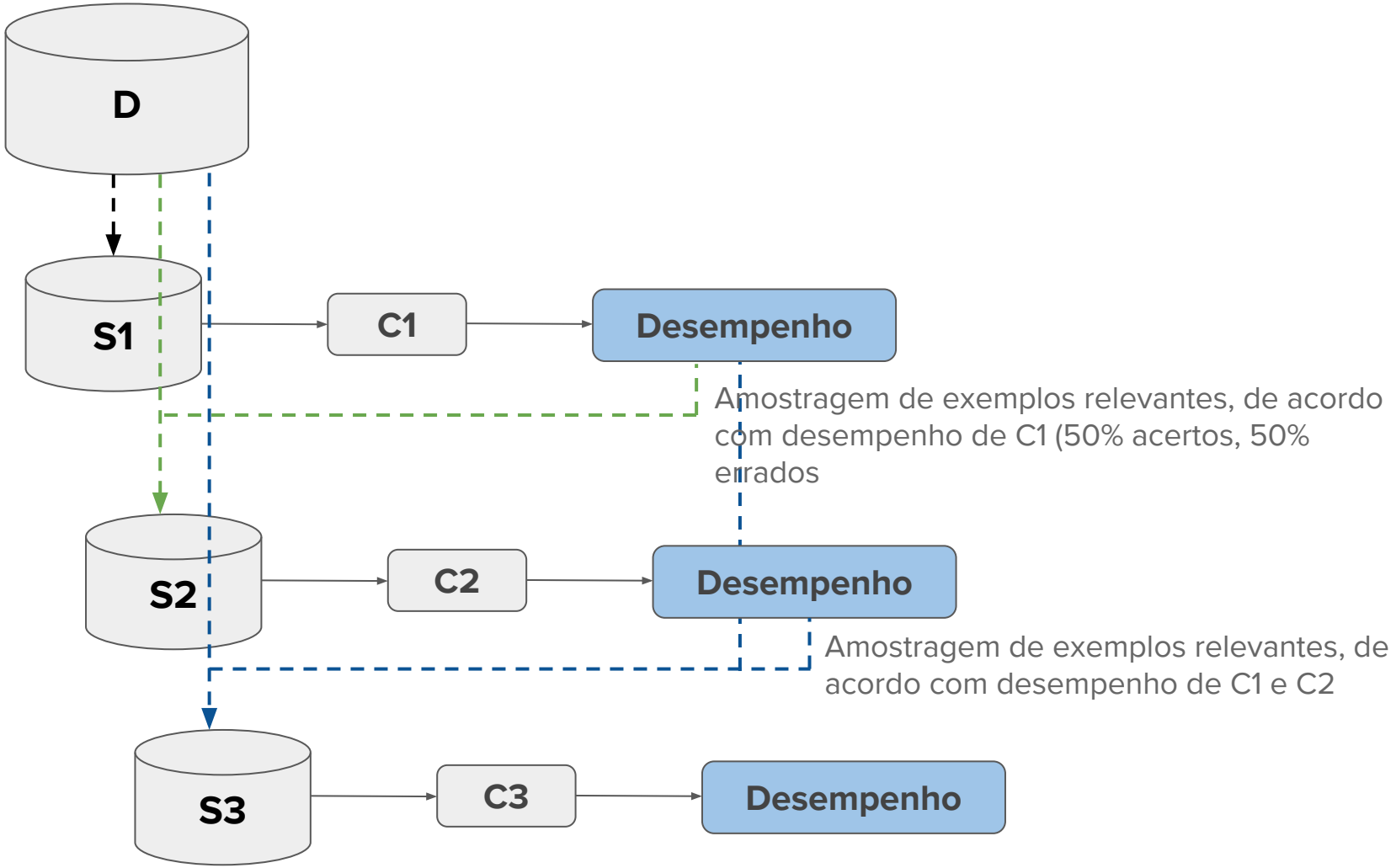
- **Boosting:** Utiliza amostragem sem reposição para criar subconjuntos de dados de treinamento, usados para treinar classificadores sequencialmente (Schapire, 1990)
  - Escolha de subconjuntos é baseada em estratégia que visa priorizar a seleção de instâncias mais relevantes
  - Novos classificadores são treinados focando sempre no **erro** dos classificadores anteriores

- **Definição da saída do ensemble** (abordagem genérica): votação ponderada entre todos os classificadores treinados.

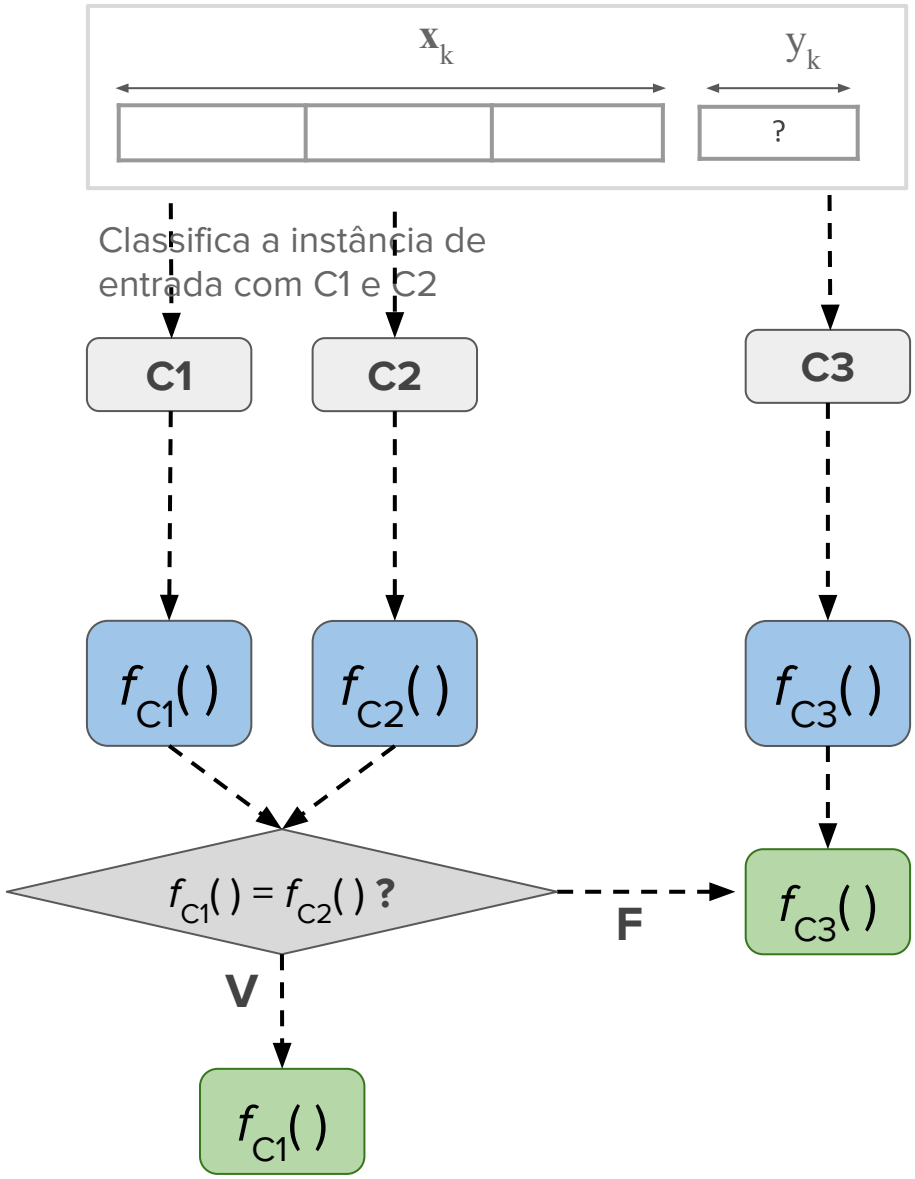
Boosting

- Intuição básica de Boosting para um *ensemble* com 3 classificadores
- Treinamento:
  - Seleciona um subconjunto S1 aleatório de n_1 exemplos ($n_1 < n$), e treina um classificador "fraco" C1
  - Cria um novo subconjunto S2 com dados mais relevantes de acordo com o desempenho de C1: aproximadamente metade de instâncias preditas corretamente por C1, e metade predita incorretamente
  - Treina um classificador fraco C2 usando S2
  - Cria um novo subconjunto S3 com todas as instâncias para as quais C1 e C2 discordam quanto à classe predita
  - Treina um classificador fraco C3 usando S3

Boosting: Treinamento



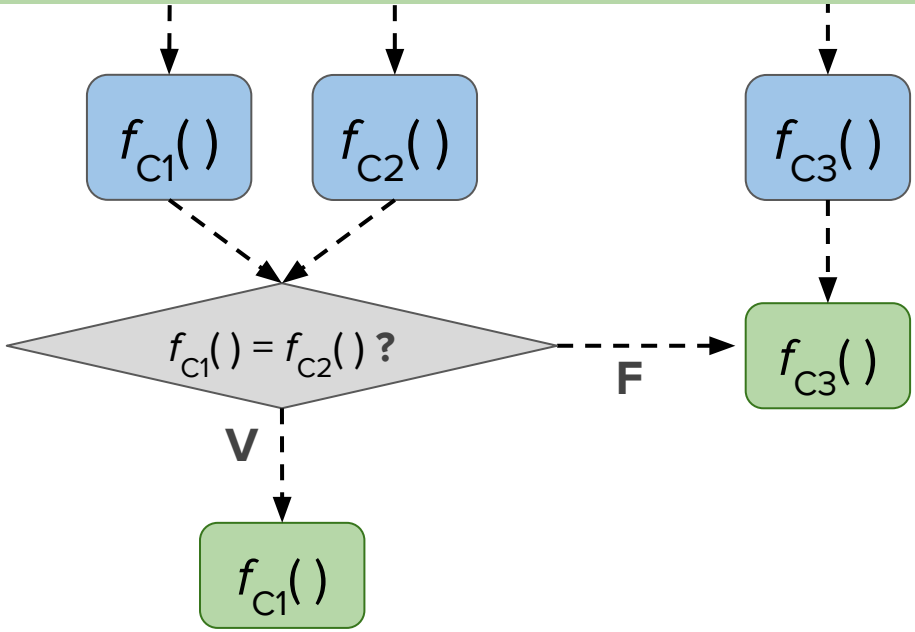
Boosting: Teste



Boosting: Teste

Intuição básica do teste com Boosting:

- **Para uma nova instância de teste, classifique com C1 e C2**
- **Se C1 e C2 concordam quanto à classe predita, adote esta saída como a classificação final**
- **Se C1 e C2 discordam, utilize C3 para a classificação final da instância**



Boosting

Algoritmo

Input:

- Training data S of size N with correct labels $\omega_i \in \Omega = \{\omega_1, \omega_2\}$;
- Weak learning algorithm **WeakLearn**.

Training

1. Select $N_1 < N$ patterns without replacement from S to create data subset S_1.
2. Call **WeakLearn** and train with S_1 to create classifier C_1.
3. Create dataset S_2 as the most informative dataset, given C_1, such that half of S_2 is correctly classified by C_2, and the other half is misclassified. To do so:
   a. Flip a fair coin. If Head, select samples from S, and present them to C_1 until the first instance is misclassified. Add this instance to S_2.
   b. If Tail, select samples from S, and present them to C_1 until the first one is correctly classified. Add this instance to S_2.
   c. Continue flipping coins until no more patterns can be added to S_2.
4. Train the second classifier C_2 with S_2.
5. Create S_3 by selecting those instances for which C_1 and C_2 disagree. Train the third classifier C_3 with S_3.

Test – Given a test instance **x**

1. Classify **x** by C_1 and C_2. If they agree on the class, this class is the final classification.
2. If they disagree, choose the class predicted by C_3 as the final classification.

Fonte: Polikar, 2006

AdaBoost

- **Adaptive Boosting:**
  - Utiliza amostragem sem reposição para criar subconjuntos de dados de treinamento usados para treinar classificadores sequencialmente.
  - Define uma distribuição sobre os dados: associa um **peso ajustável** a cada exemplo do conjunto de treinamento que reflete a sua relevância
    - permite identificar os exemplos mais "difíceis" de serem aprendidos (Freund e Schapire, 1996) - estes são os mais relevantes e devem ser priorizados no treinamento de novos classificadores

AdaBoost

- **<u>Adap</u>tive <u>Boost</u>ing:**
  - Inicialmente todos os exemplos possuem o mesmo peso 1/*n*, onde *n* é o número de exemplos no conjunto de dados.
  - O algoritmo treina um classificador usando um subconjunto S_t aleatório obtido da distribuição definida sobre os exemplos:
    - Intuição básica: exemplos com maior peso são mais relevantes e devem ter maior "chance" de escolha para S_t
      - Na primeira iteração, todos os exemplos possuem a mesma chance de escolha (mesmo peso!)
    - De acordo com o desempenho do Classificador C_t treinado sobre S_t, os pesos dos exemplos são ajustados:
      - peso de exemplos preditos incorretamente aumenta
      - peso de exemplos preditos corretamente diminui
  - Ensemble é criado a partir de uma votação ponderada dos **T** classificadores bases treinados, onde o peso de cada classificador é estimado durante o treinamento com base em seu desempenho

Processo iterativo:
treina T classificadores sequencialmente

AdaBoost

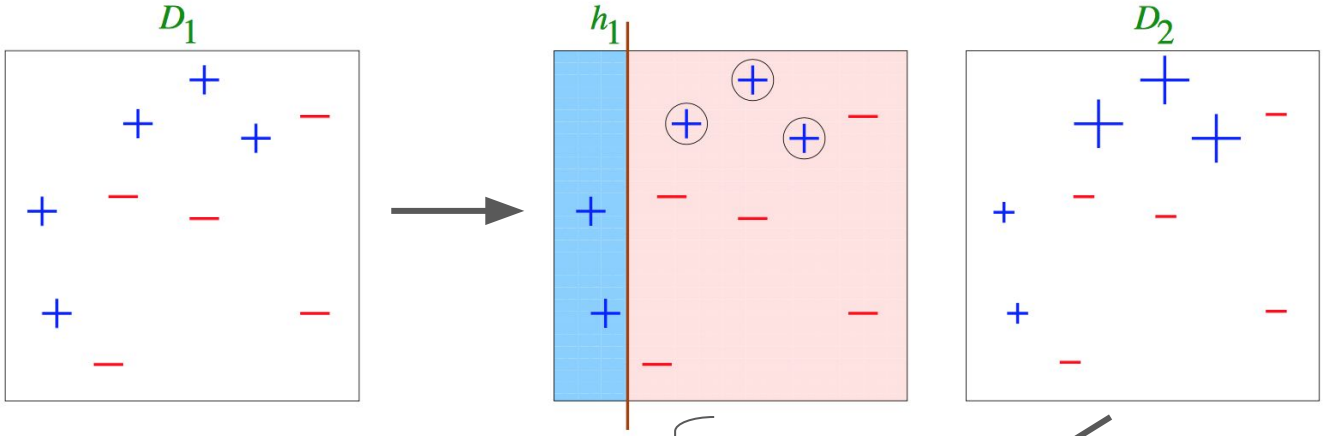
- **<u>Adap</u>tive <u>Boost</u>ing:**
  - Inicialmente todos os exemplos possuem o mesmo peso 1/*n*, onde *n* é o número de exemplos no conjunto de dados.
  - O algoritmo treina um classificador usando um subconjunto S_t aleatório obtido da distribuição definida sobre os exemplos:
    - Intuição básica: exemplos com maior peso são mais relevantes e devem ter maior "chance" de escolha para S_t
      - Na primeira iteração, todos os exemplos possuem a mesma chance de escolha (mesmo peso!)
    - De acordo com o desempenho do Classificador C_t treinado sobre S_t, os pesos dos exemplos são ajustados:
      - peso de exemplos preditos incorretamente aumenta
      - peso de exemplos preditos corretamente diminui

Processo iterativo:
treina T classificadores sequencialmente

Efeito do ajuste de pesos: cada novo classificador treinado foca seu aprendizado nos exemplos mais "difíceis" (aqueles não resolvidos pelos classificadores anteriores)

AdaBoost

Round 1



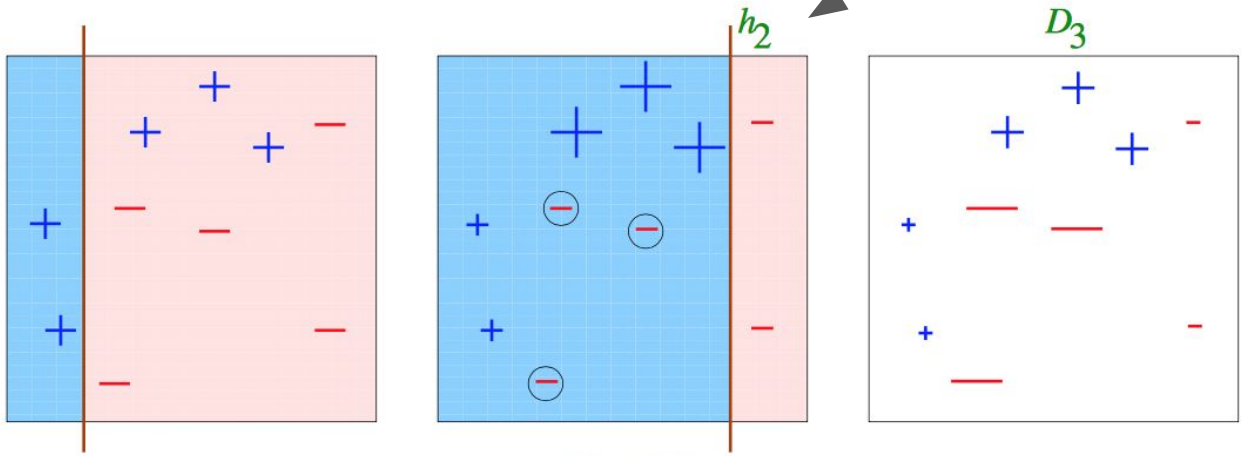
Instâncias classificadas incorretamente da Classe + têm o peso incrementado

Erro e peso do classificador h_1

$\varepsilon_1 = 0.30$

$\alpha_1 = 0.42$

Round 2



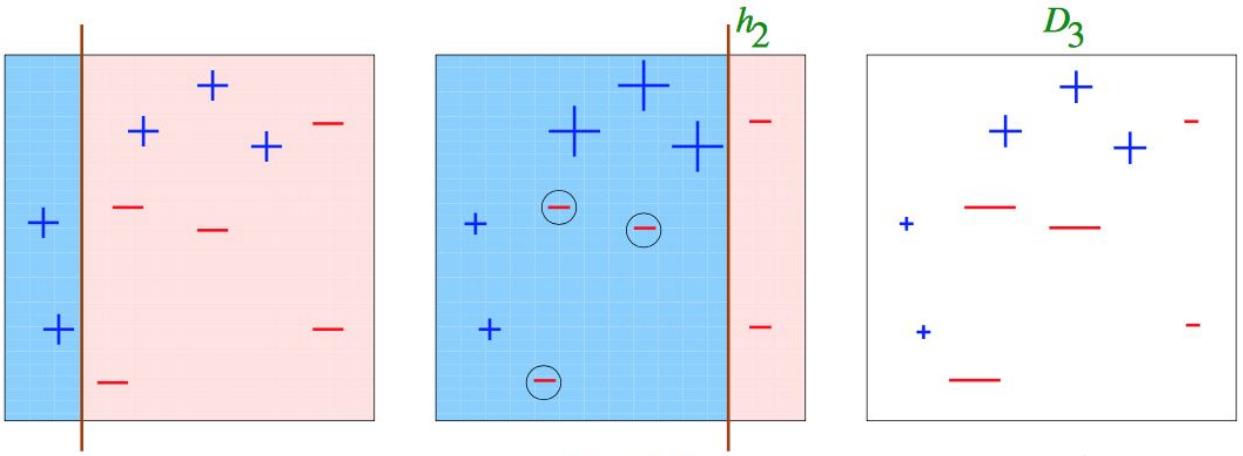
$\varepsilon_2 = 0.21$

$\alpha_2 = 0.65$

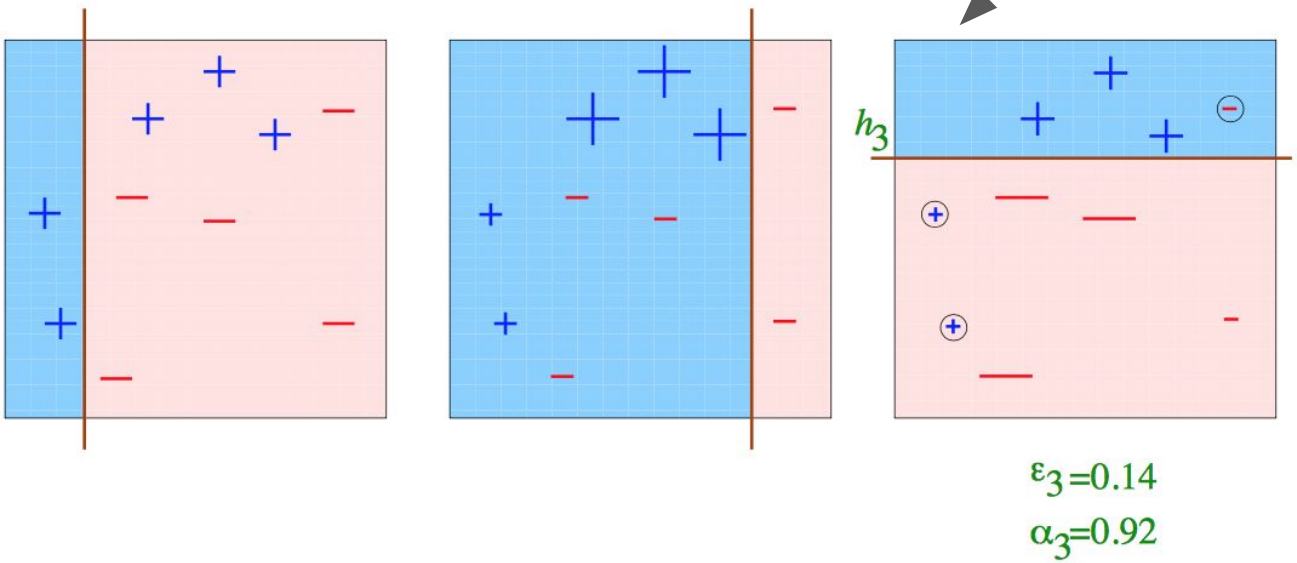
Instâncias classificadas corretamente da Classe + têm o peso decrementado, enquanto instâncias classificadas incorretamente da Classe - têm o peso incrementado

AdaBoost

Round 2

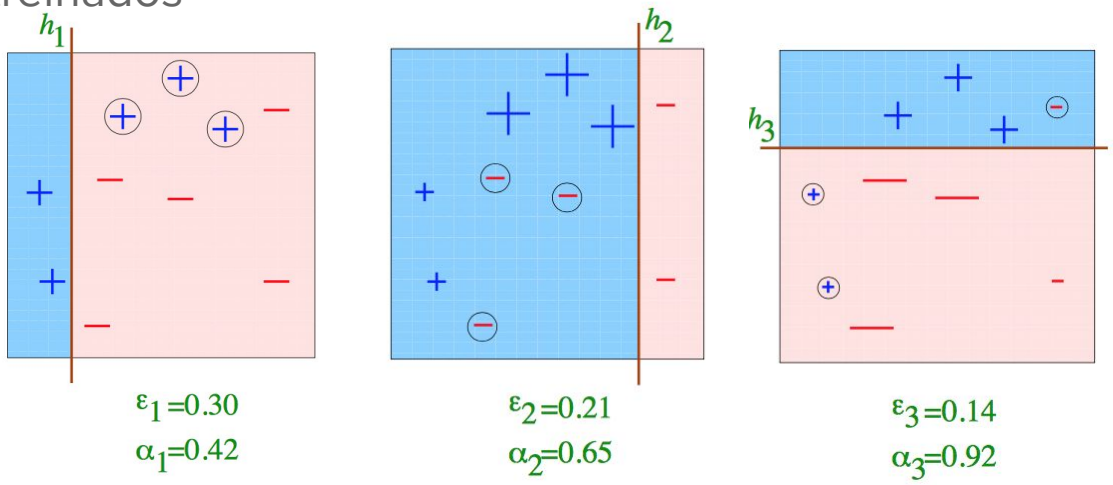


Round 3 (último)

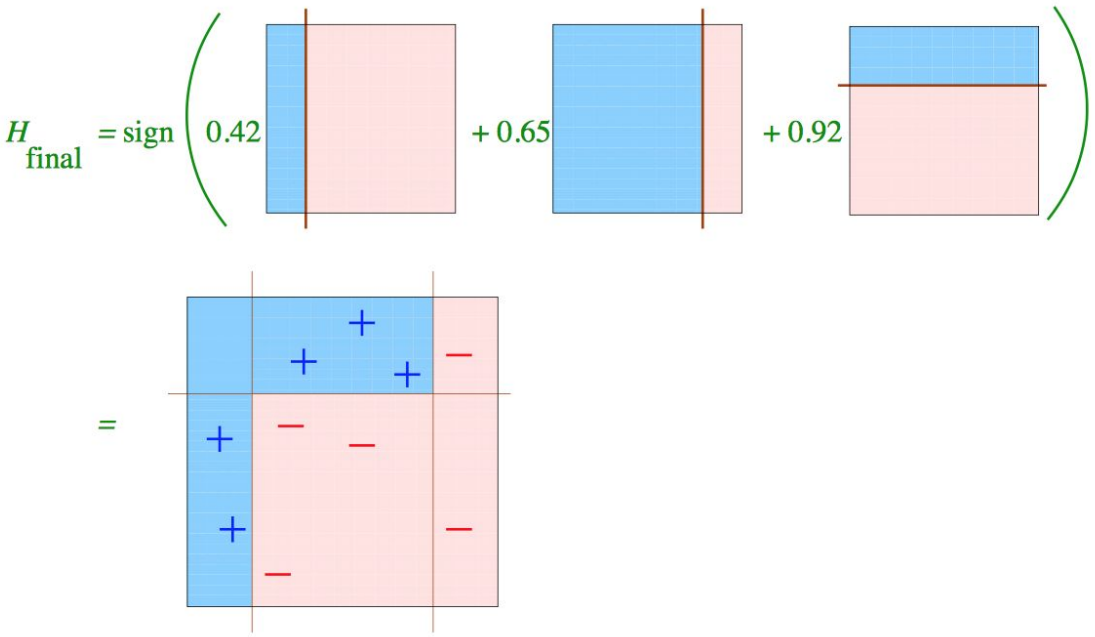


AdaBoost

Classificadores treinados



Classificação de novas instâncias:



AdaBoost

Algoritmo

Input:

- Sequence of N examples $S = [(\mathbf{x}_i, y_i)], i = 1, \cdots, N$ with labels $y_i \in \Omega, \Omega = \{\omega_1, \ldots, \omega_C\}$;
- Weak learning algorithm **WeakLearn**;
- Integer T specifying number of iterations.

Initialize $D_1(i) = \frac{1}{N}., i = 1, \cdots, N$ (11)

Do for $t = 1, 2, \ldots, T$:

1. Select a training data subset S_t, drawn from the distribution D_t.
2. Train **WeakLearn** with S_t, receive hypothesis h_t.
3. Calculate the error of h_t: $\varepsilon_t = \sum_{i:h_t(\mathbf{x}_i) \neq y_i} D_t(i)$. (12)

   If $\varepsilon_t > 1/2$, **abort**.
4. Set $\beta_t = \varepsilon_t/(1 - \varepsilon_t)$. (13)
5. Update distribution

$$D_t : D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(\mathbf{x}_i) = y_i \\ 1, & \text{otherwise} \end{cases} \quad (14)$$

   where $Z_t = \sum_i D_t(i)$ is a normalization constant chosen so that D_{t+1} becomes a proper distribution function.

Test – Weighted Majority Voting: Given an unlabeled instance $\mathbf{x}$,

1. Obtain total vote received by each class

$$V_j = \sum_{t:h_t(\mathbf{x})=\omega_j} \log \frac{1}{\beta_t}, j = 1, \ldots, C. \quad (15)$$

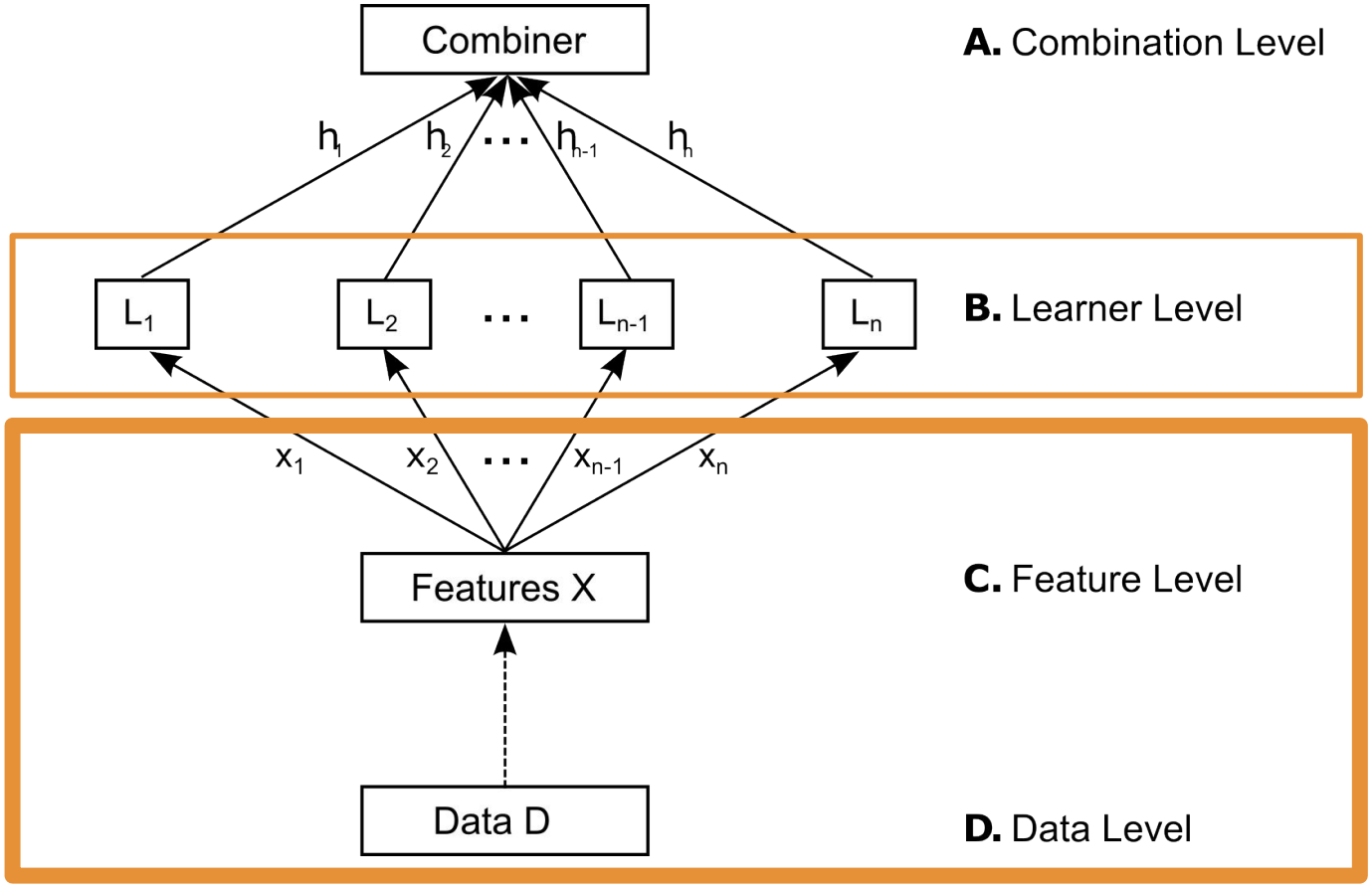
2. Choose the class that receives the highest total vote as the final classification.

Fonte: Polikar, 2006

Construindo modelos múltiplos de classificação

Taxonomia proposta por Kuncheva, 2004

Métodos baseados em amostragem dos dados de treinamento e injeção de aleatoriedade

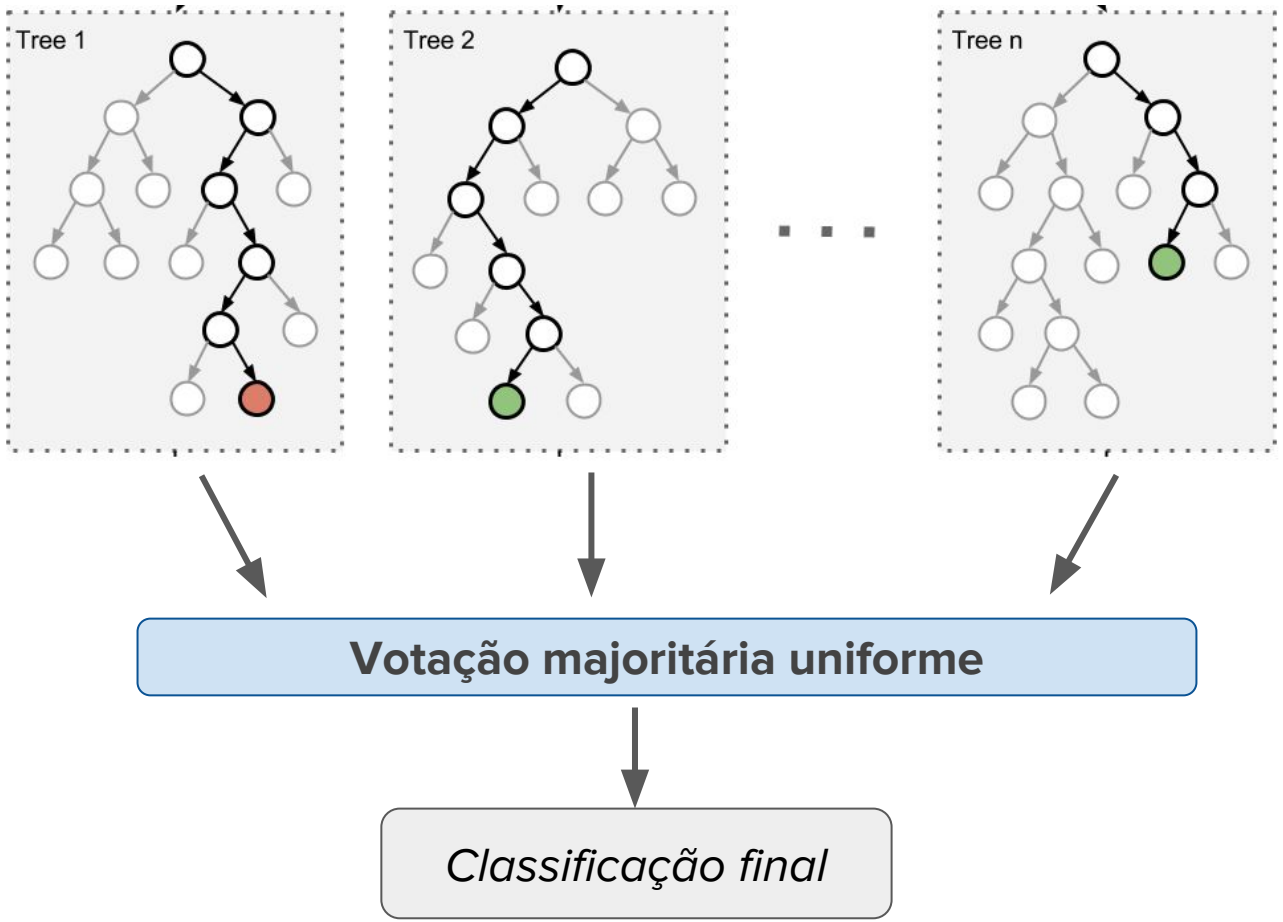


Ensemble Heterogêneo/ Homogêneo

Ensemble Homogêneo

Florestas aleatórias

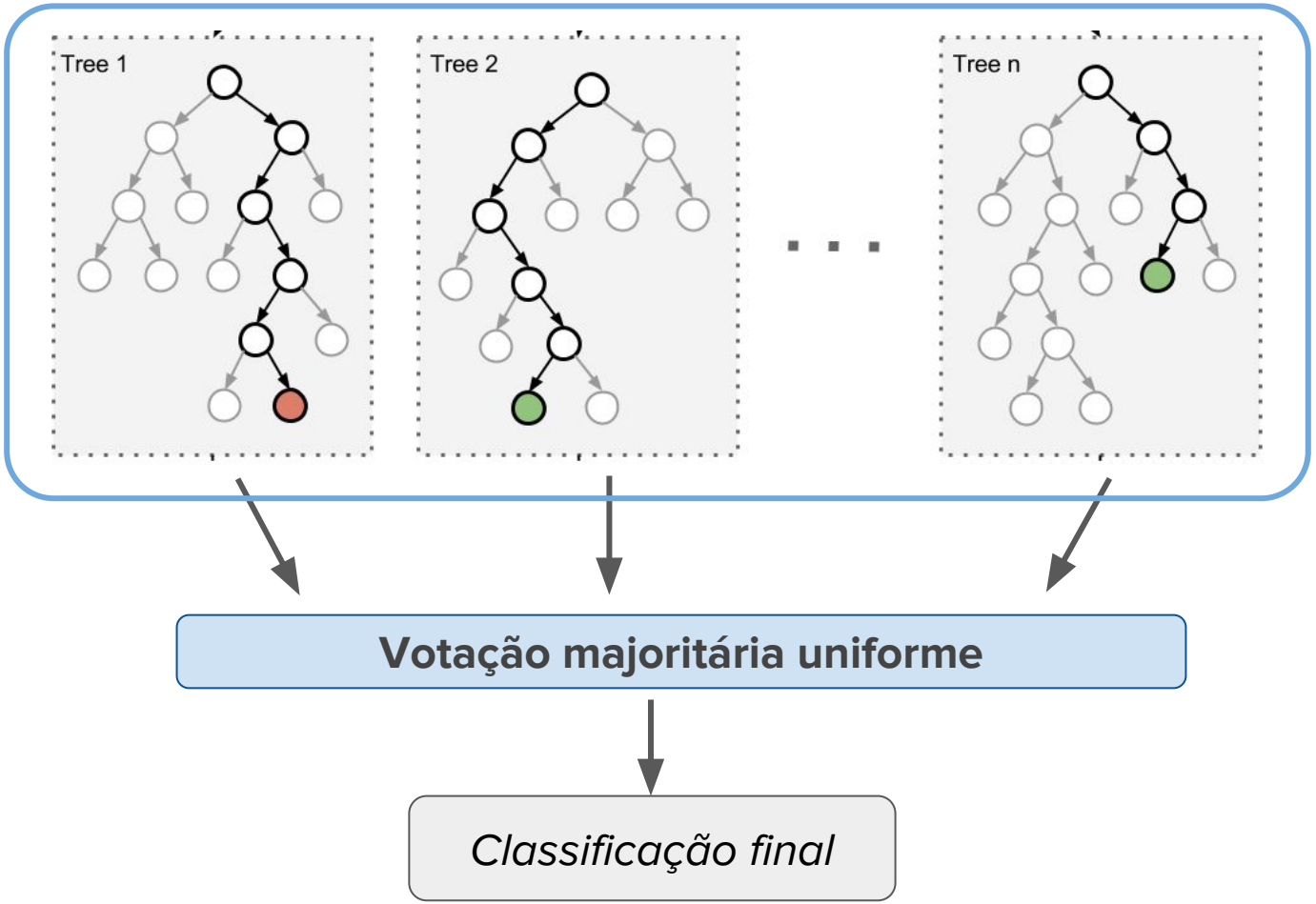
- Breiman (2001): Treinamento de várias árvores de decisão, cujas previsões são combinados por votação majoritária uniforme



Florestas aleatórias

- Breiman (2001): Treinamento de várias árvores de decisão, cujas previsões são combinados por votação majoritária uniforme

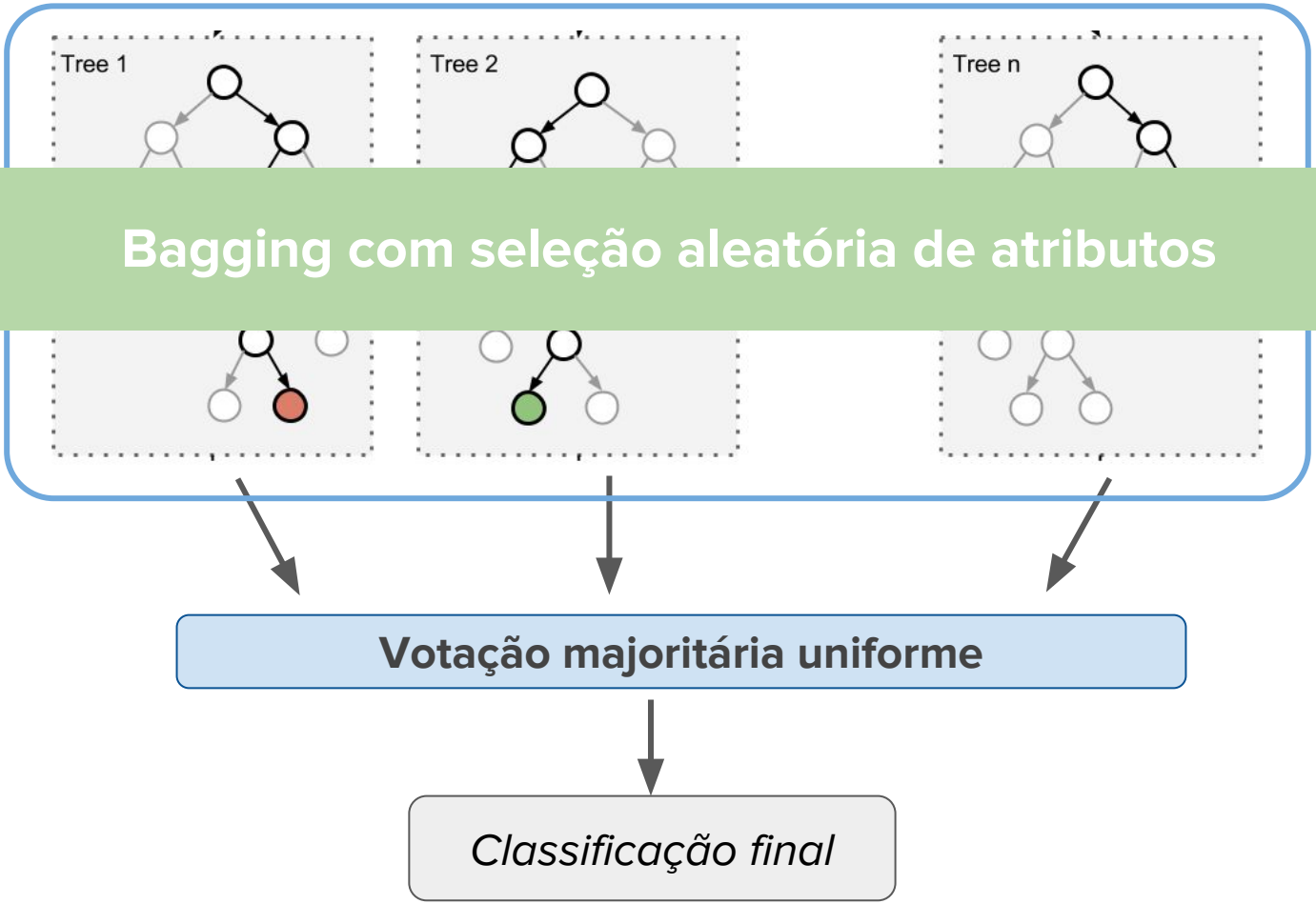
DIVERSIDADE?



Florestas aleatórias

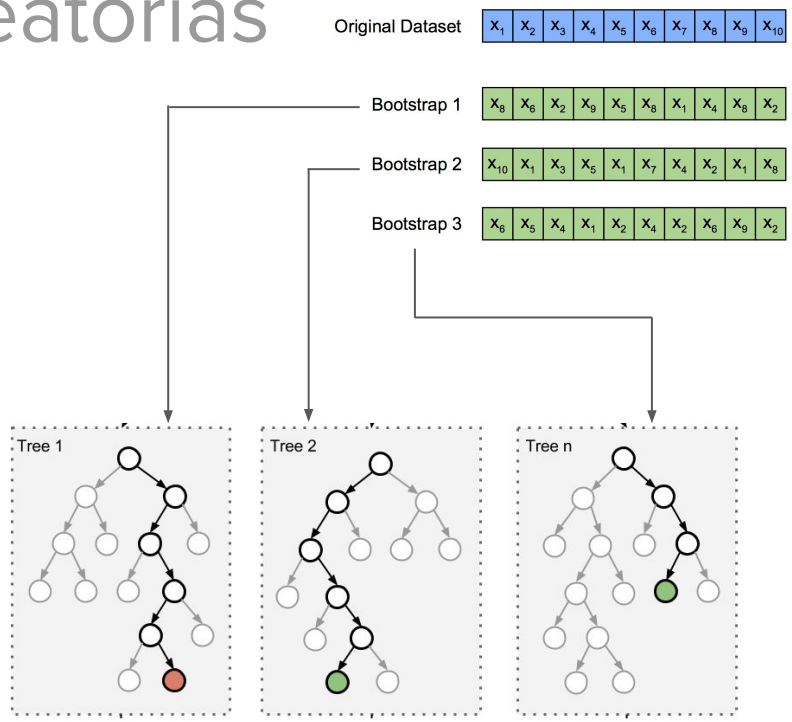
- Breiman (2001): Treinamento de várias árvores de decisão, cujas previsões são combinados por votação majoritária uniforme

DIVERSIDADE?



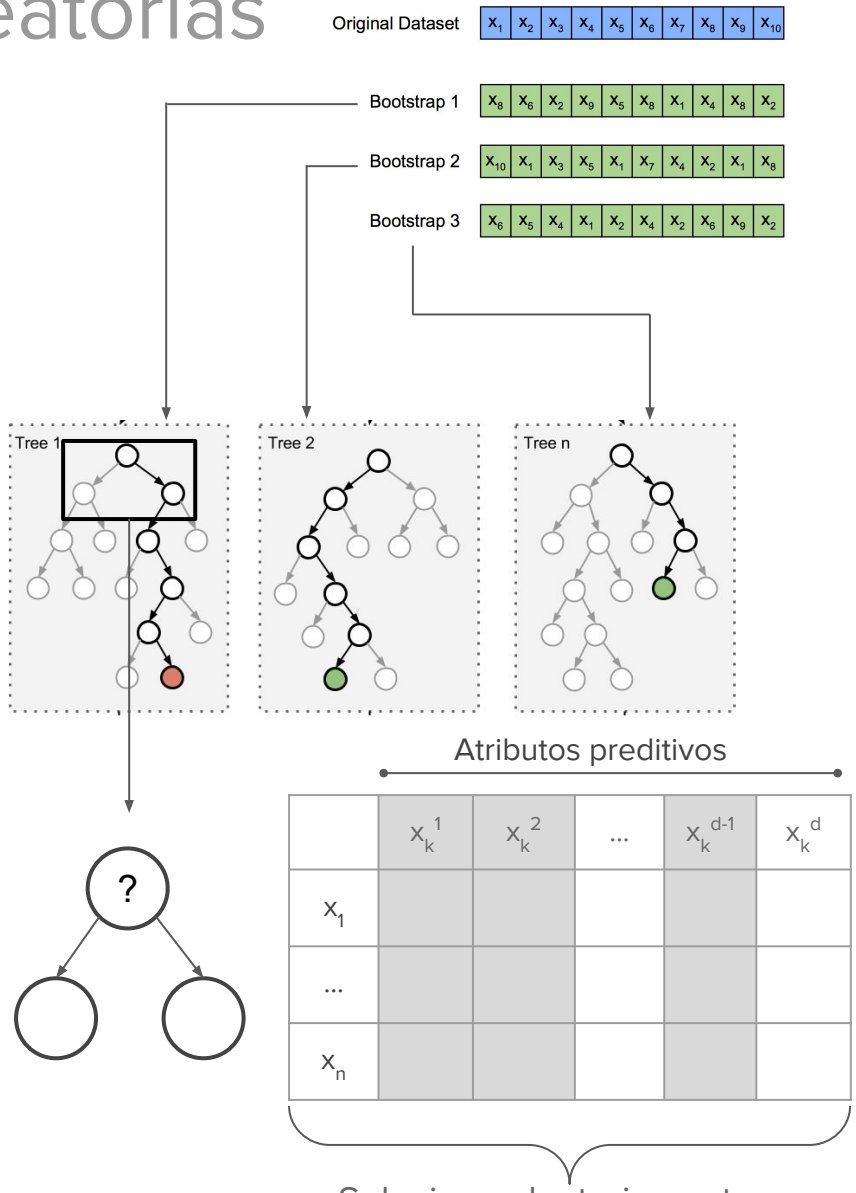
Florestas aleatórias

- **Bagging + seleção aleatória de atributos**
  - *Bagging*: cada árvore é treinada sobre um subconjunto dos dados de treinamento (bootstrap), gerado através de amostragem com reposição, contendo *n* exemplos



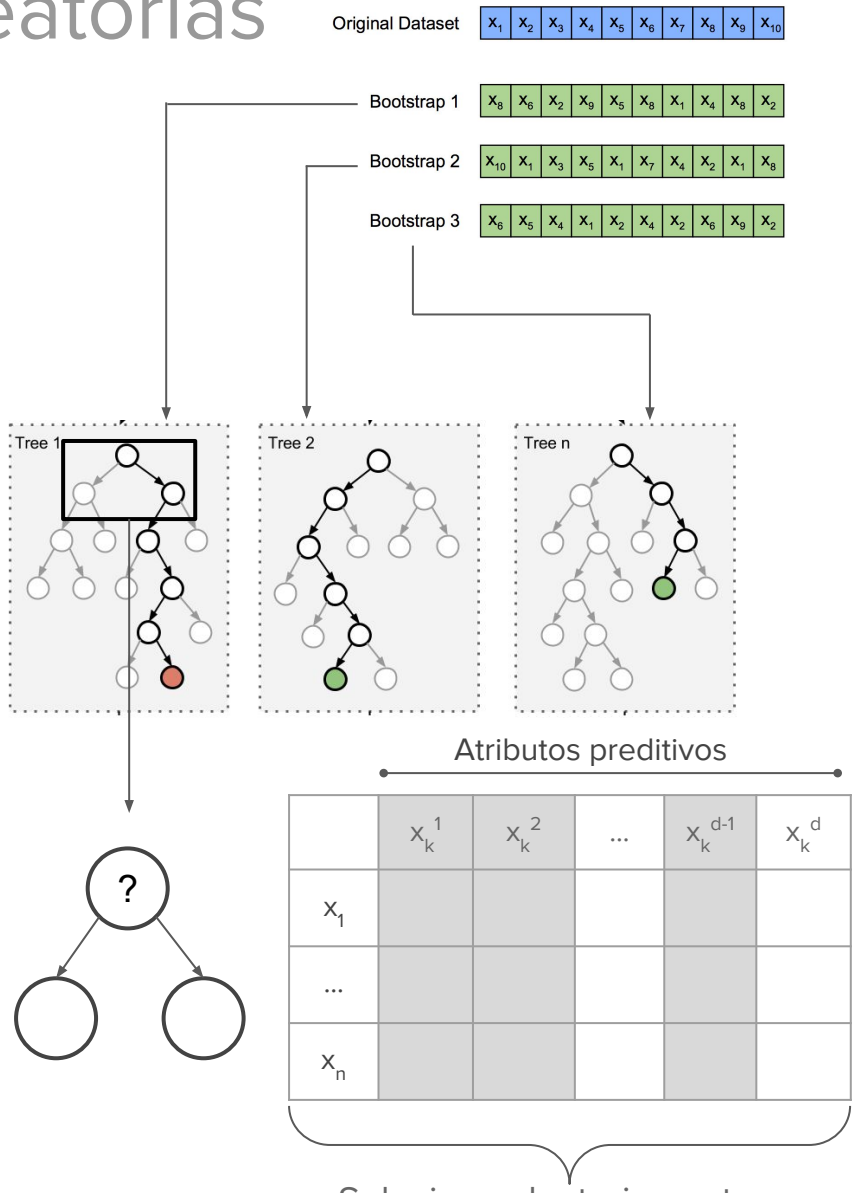
Florestas aleatórias

- **Bagging + seleção aleatória de atributos**
  - *Bagging*: cada árvore é treinada sobre um subconjunto dos dados de treinamento (bootstrap), gerado através de amostragem com reposição, contendo *n* exemplos
  - *Seleção aleatória de atributos*: Para cada divisão de nós, apenas um subconjunto aleatório de ***m*** atributos é considerado para avaliação, sendo escolhido aquele que minimiza a impureza do nó dentre estes avaliados



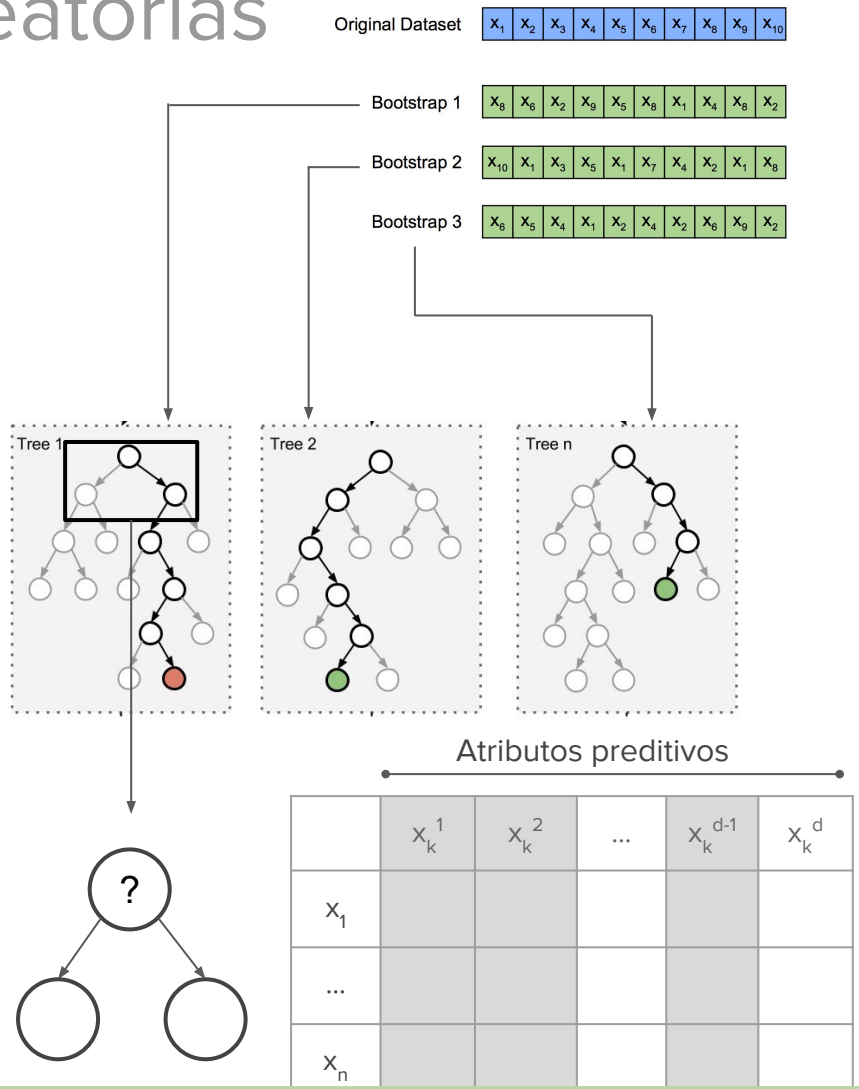
Florestas aleatórias

- **Bagging + seleção aleatória de atributos**
  - *Bagging*: cada árvore é treinada sobre um subconjunto dos dados de treinamento (bootstrap), gerado através de amostragem com reposição, contendo *n* exemplos
  - *Seleção aleatória de atributos*: Para cada divisão de nós, apenas um subconjunto aleatório de ***m*** atributos é considerado para avaliação, sendo escolhido aquele que minimiza a impureza do nó dentre estes avaliados
  - Ambas amostragens geram alta diversidade entre as árvores treinadas
  - Porção OOB (out-of-bag, ~ 3 dos dados) é utilizada para avaliar desempenho de cada modelo treinado e estimar relevância de atributos



Florestas aleatórias

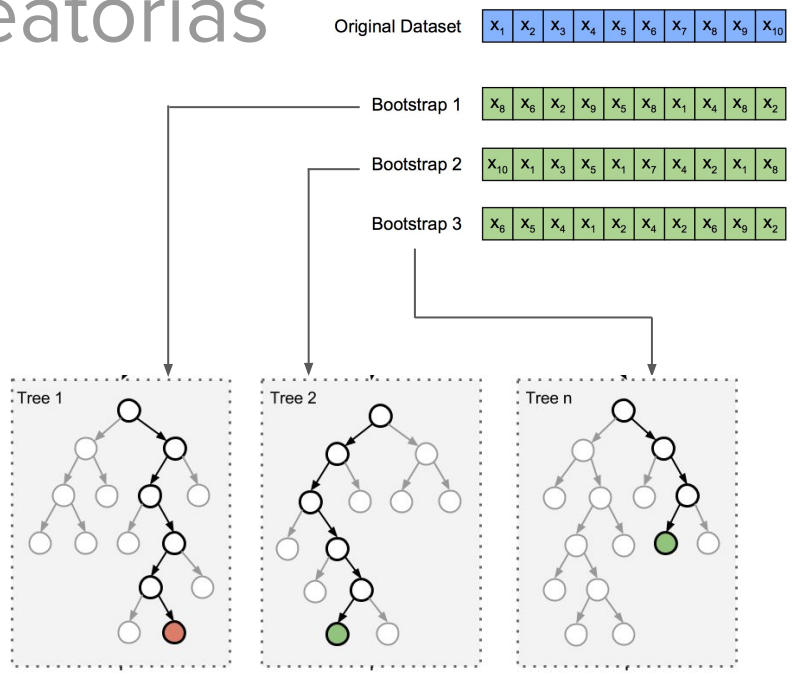
- **Bagging + seleção aleatória de atributos**
  - *Bagging*: cada árvore é treinada sobre um subconjunto dos dados de treinamento (bootstrap), gerado através de amostragem com reposição, contendo *n* exemplos
  - *Seleção aleatória de atributos*: Para cada divisão de nós, apenas um subconjunto aleatório de ***m*** atributos é considerado para avaliação, sendo escolhido aquele que minimiza a impureza do nó dentre estes avaliados
  - Ambas amostragens geram alta diversidade entre as árvores treinadas
  - Porção OOB (out-of-bag, ~ 3 dos dados)



Florestas aleatórias (*Random Forests*) é um dos modelos mais competitivos atualmente para aprendizado supervisionado

divisão para este nó

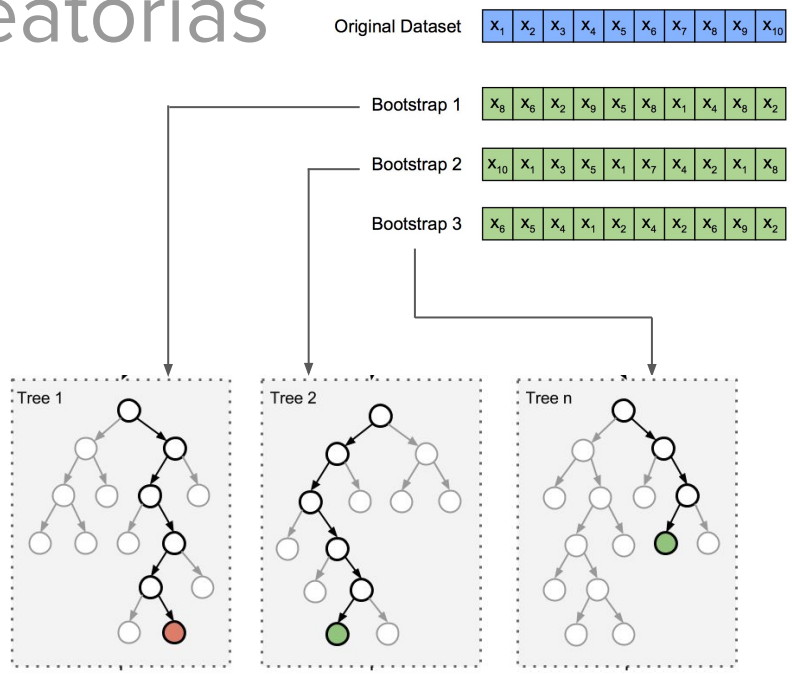
Florestas aleatórias



Algoritmo de treinamento: Florestas Aleatórias

1. Para cada boostrap *b* = 1 ... B
   2. Obtenha um boostrap de tamanho N a partir dos dados originais
   3. Construa uma árvore de decisão com base no boostrap *b*, repetindo recursivamente:
      4. Selecione ***m*** atributos aleatoriamente dentre todos os atributos disponíveis
      5. Escolha o melhor atributo (ex: índice Gini) para divisão de um novo nó dentre os ***m*** atributos aleatórios
      6. Inclua um novo nó na árvore de acordo com a seleção do passo anterior, dividindo os dados em dois subconjuntos.
7. Retorne um ensemble de árvores construídas

Florestas aleatórias



Algoritmo de treinamento: Florestas Aleatórias

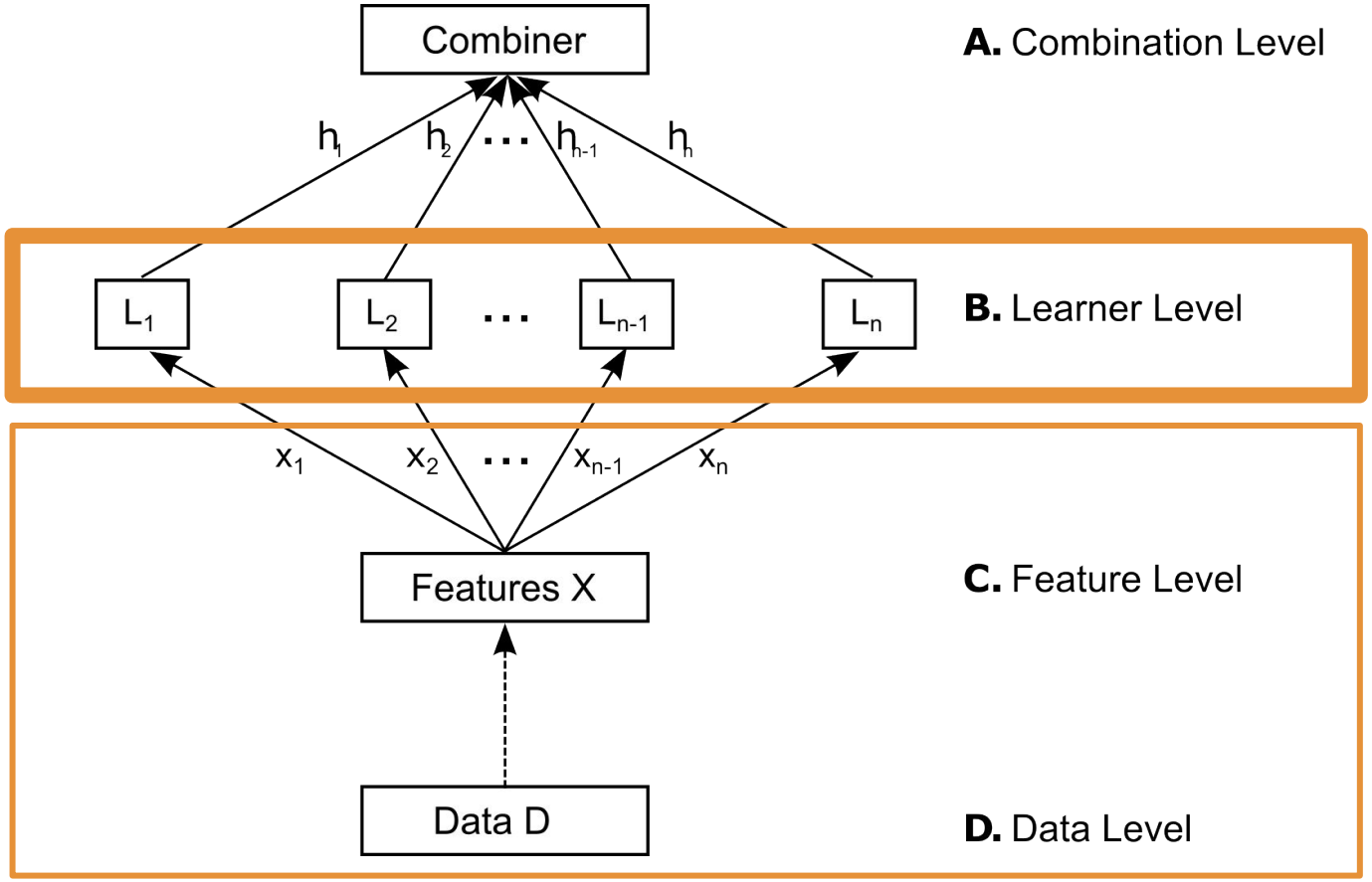
1. Para cada boostrap *b* = 1 ... B
   2. Obtenha um boostrap de tamanho N a partir dos dados originais
   3. Construa uma árvore de decisão com base no boostrap *b*, repetindo recursivamente:
      4. Selecione ***m*** atributos aleatoriamente dentre todos os atributos disponíveis
      5. Escolha o melhor atributo (ex: índice Gini) para divisão de um novo nó dentre os ***m*** atributos aleatórios
      6. Inclua um novo nó na árvore de acordo com a seleção do passo anterior, dividindo os dados em dois subconjuntos.
7. Retorne um ensemble de árvores construídas

Valor default recomendado: raiz quadrada do número total de atributos

Na próxima aula....

Construindo modelos múltiplos de classificação

Métodos baseado em combinação de classificadores heterogêneos



Ensemble Heterogêneo/ Homogêneo

Ensemble Homogêneo