

APRENDIZADO DE MÁQUINAS

Aprendizado de reglas



TÓPICOS

- 1. Aprendizado de Regras**
- 2. Regras de decisão e árvores de decisão**
- 3. Algoritmo de cobertura**
- 4. Estratégias Top-down e Bottom-up**
- 5. Exemplo CN2**
- 6. Exemplo AQ**
- 7. Aplicando regras**

REGRAS DE DECISÃO

Uma regra de decisão é uma implicação da forma:

Se A então B

Parte condicional A: conjunção de condições

Cada condição é relação entre um atributo e valores

Relações: =, ≠, <, >, <=, >=, ∈, ∉

Ex. Atributo_i = v_i, Atributo_i < v_i, Atributo_i ∈ Conjunto_valores, etc

Ex. Tempo = Ensolarado ^ Umidade <= 75 ⇒ Jogar = Sim

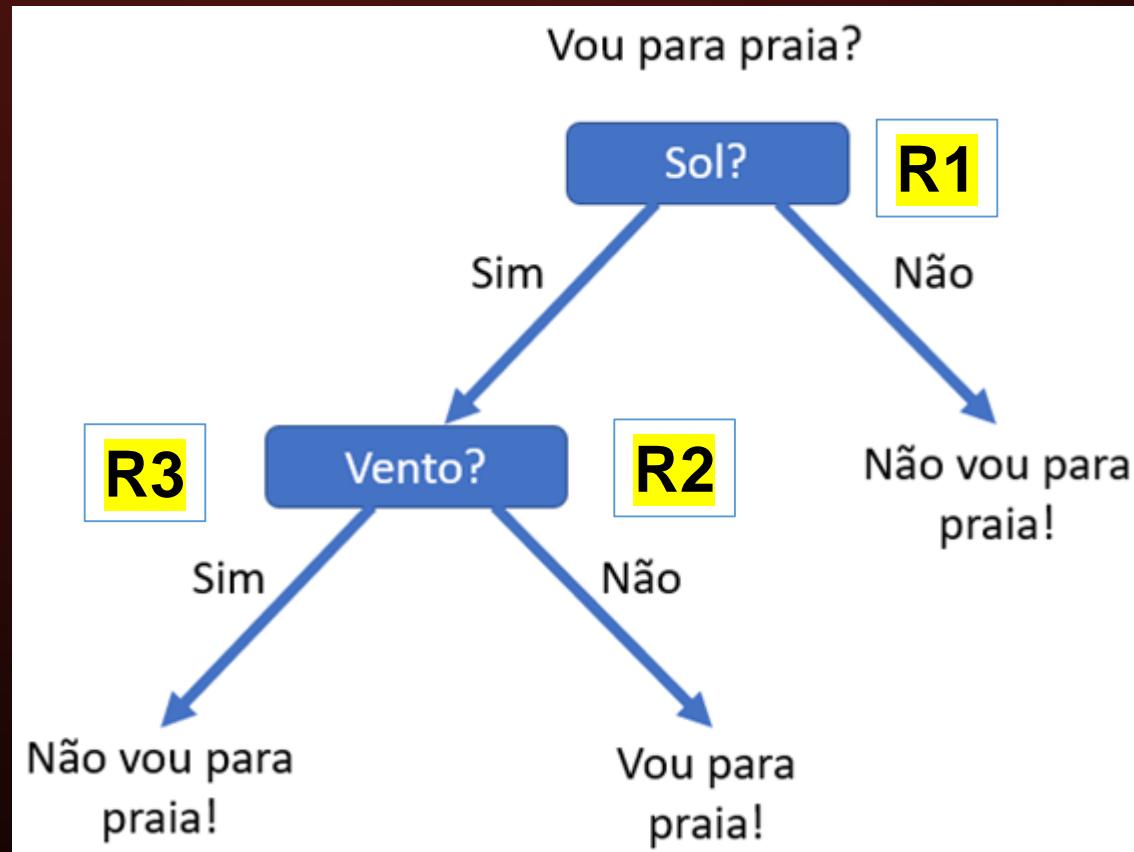
Parte Consequente B: rotula os exemplos

Tal como nas árvores de decisão, o conjunto de regras é disjunto (FND)

regra₁ ou regra₂ ou ... ou regra_n

REGRAS DE DECISÃO E ÁRVORES DE DECISÃO

Regras de Decisão e Árvores de Decisão são bastante similares em sua forma de representação para expressar generalizações dos exemplos.



**R1: Se Sol=Não
então “Não vou para a praia”**

**R2: Se Sol=Sim ^
Vento=Não então
“Vou para a praia”**

**R3: Se Sol=Sim ^
Vento=Sim então
“Não vou para a praia”**

REGRAS DE DECISÃO E ÁRVORES DE DECISÃO

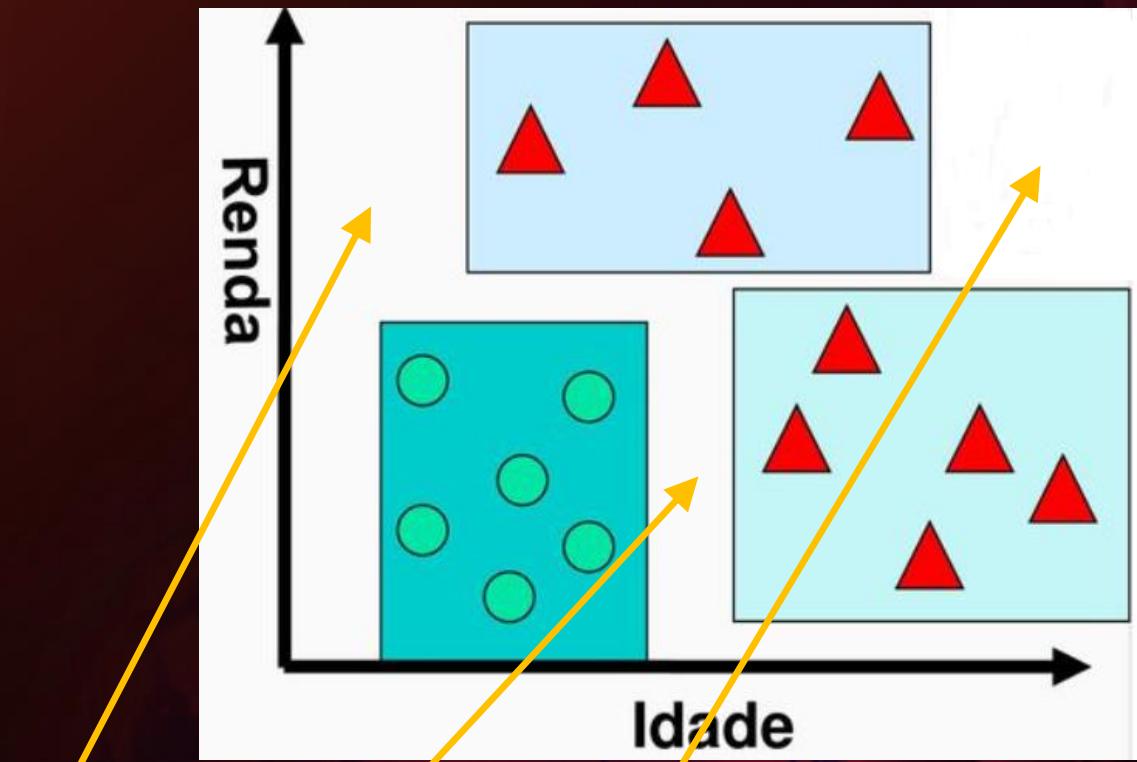
Ambas definem superfícies de decisão similares.

As superfícies de decisão definidas pelas regras de decisão correspondem a hiper-retângulos no espaço definido pelos atributos.

Cada regra cobre uma região específica do espaço de instâncias.

A união de todas as regras pode ser menor do que o Universo.

Regras são modulares e podem ser interpretadas isoladamente



Regiões não cobertas pelas regras

ÁRVORES DE DECISÃO PARA REGRAS DE DECISÃO

Cada regra corresponde a um percurso, desde a raiz da árvore até uma folha.

Existem tantas regras quanto folhas da árvore.

Alguns algoritmos utilizam um processo de otimização para simplificar o conjunto de regras removendo condições irrelevantes.

Processo de otimização

Cada regra é generalizada pela eliminação de condições que não contribuem para discriminar as classes (Busca Gulosa), utilizando-se um mecanismo de poda semelhante ao utilizado pelo algoritmo C4.5

As regras são agrupadas pela classe que preveem e a seguir são eliminadas as regras que não contribuem para a taxa de acerto do conjunto

ALGORITMO DE COBERTURA

Tipicamente o algoritmo procura regras da forma:

Se Atributo_i = valor_j e Atributo_l = valor_k ... Então Classe_z

Entrada: Um conjunto de treinamento $D = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$

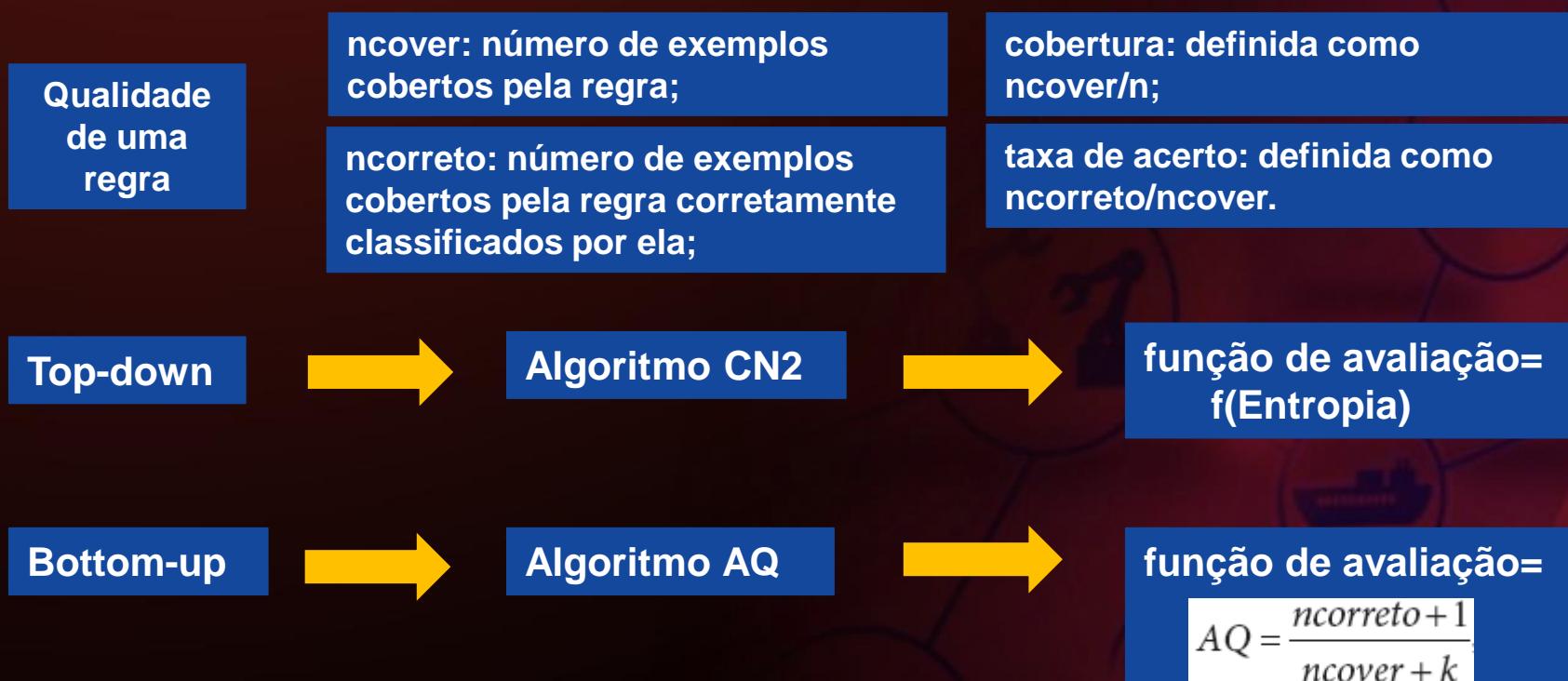
Saída: Um conjunto de regras: **Regras**

- 1 **Regras** $\leftarrow \{\}$;
- 2 Seja Y o conjunto das classes em D ;
- 3 **seja cada** $y_i \in Y$ **faça**
- 4 **repita**
- 5 *Regra* = Aprende_Uma_Regra(D, y_i);
- 6 **Regras** \leftarrow **Regras** $\cup \{\textit{Regra}\}$
- 7 $D \leftarrow$ Remove exemplos cobertos pela *Regra* em D ;
- 8 **até não haver exemplos de** y_i ;
- 9 **fim**
- 10 **Retorna:** **Regras**;

ESTRATÉGIAS

Top-down: inicia a busca da regra mais geral, {} → Classe, e aplica operadores de especificação, acrescentando condições à parte condicional da regra (orientada pelo modelo)

Bottom-up: começa pela regra mais específica (é escolhido um dos exemplos aleatoriamente, o que implica restrições em todos os atributos) e aplica operadores de generalização, removendo restrições (orientada a dados)



ALGORITMO TOP-DOWN

Entrada: Um conjunto de treinamento $D = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$

y : classe da regra

Saída: *Regra*: Uma regra de classificação

```
1 Seja  $Avs$  o conjunto de atributo_valores em  $D$ ;  
2 Regra  $\leftarrow \{ \}$ ;  
3  $v \leftarrow Avalia(Regra, D, y)$ ;  
4  $melhor \leftarrow v$ ;  
5  $continua \leftarrow$  Verdadeiro;  
6 enquanto  $continua$  faça  
7    $continua \leftarrow$  Falso;  
8   para cada  $av_i \in Avs$  faça  
9      $val \leftarrow Avalia(Regra \cup av_i, D, y)$ ;  
10    se  $val < melhor$  então  
11       $melhor \leftarrow val$ ;  
12       $Cond \leftarrow av_i$ ;  
13       $continua \leftarrow$  Verdadeiro;  
14    fim  
15  fim  
16  se  $continua$  então  
17     $Regra \leftarrow Regra \cup Cond$ ;  
18  fim  
19 fim  
20 Retorna: Regra;
```



ALGORITMO BOTTOM-UP

Entrada: Um conjunto de treinamento $D = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$

y : classe da regra

Saída: *Regra*: Uma Regra de classificação

```
1 Escolhe, aleatoriamente, um exemplo da classe  $y$  em  $D$ ;  
2 Seja Regra o conjunto de atributo_valor desse exemplo;  
3  $v \leftarrow Avalia(Regra, D, y)$ ;  
4  $melhor \leftarrow v$ ;  
5  $continua \leftarrow$  Verdadeiro;  
6 enquanto continua faça  
7   continua  $\leftarrow$  Falso;  
8   para cada  $av_i \in$  Regra faça  
9      $val \leftarrow Avalia(Regra \setminus av_p, D, y)$ ;  
10    se  $val < melhor$  então  
11       $melhor \leftarrow val$ ;  
12       $Cond \leftarrow av_p$ ;  
13      continua  $\leftarrow$  Verdadeiro;  
14    fim  
15  fim  
16  se continua então  
17    Regra  $\leftarrow Regra \setminus Cond$ ;  
18  fim  
19 fim  
20 Retorna: Regra;
```

Fonte: Inteligência Artificial – Uma abordagem de aprendizado de máquina, Faceli et al

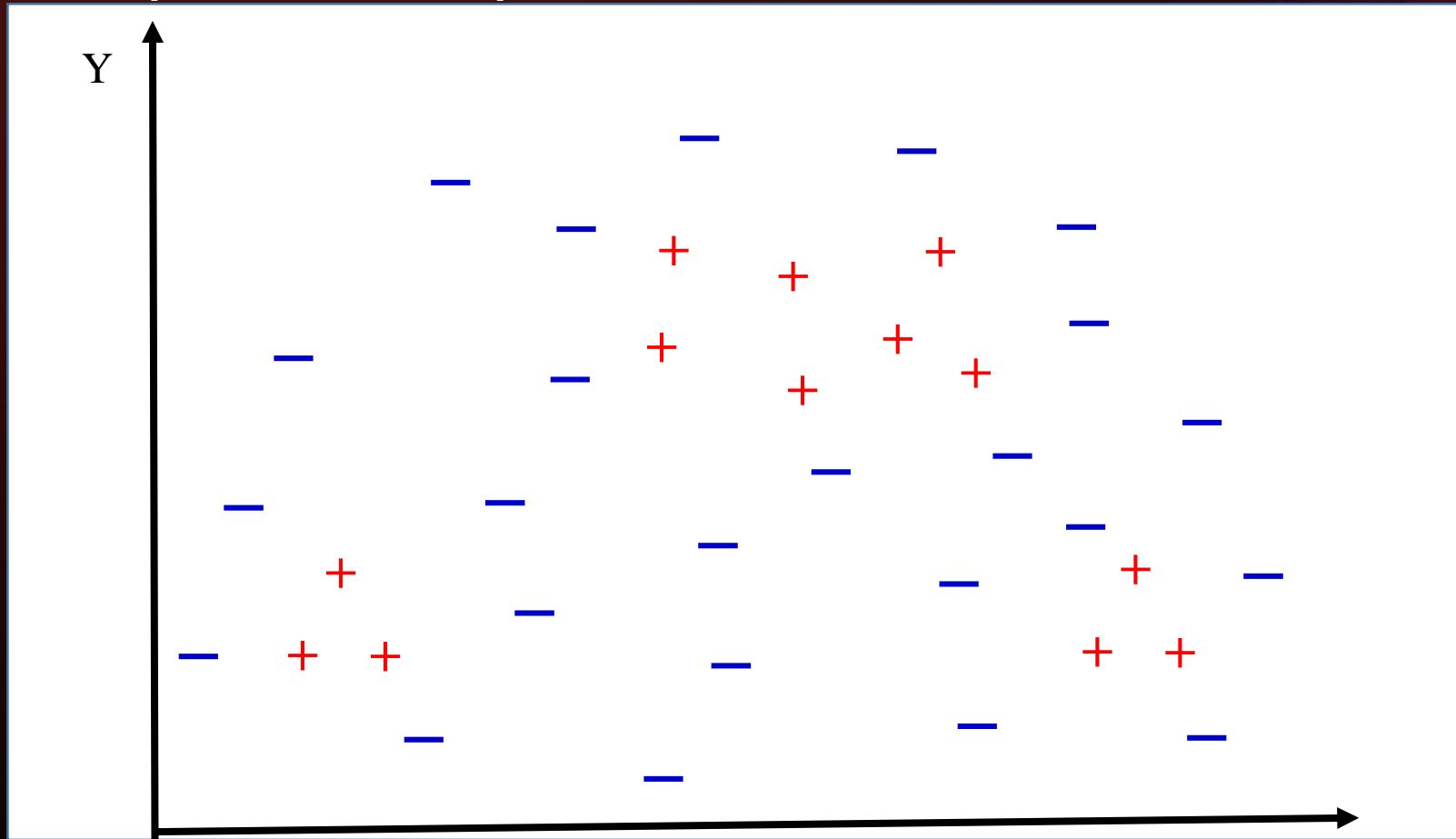


ESTRATÉGIA TOP-DOWN

Começar com a regra mais geral (vazia).

Repetidamente, adicionar restrições ao antecedente usando características que eliminem o máximo de negativos enquanto mantêm o maior número de positivos.

Parar quando somente positivos estiverem cobertos.

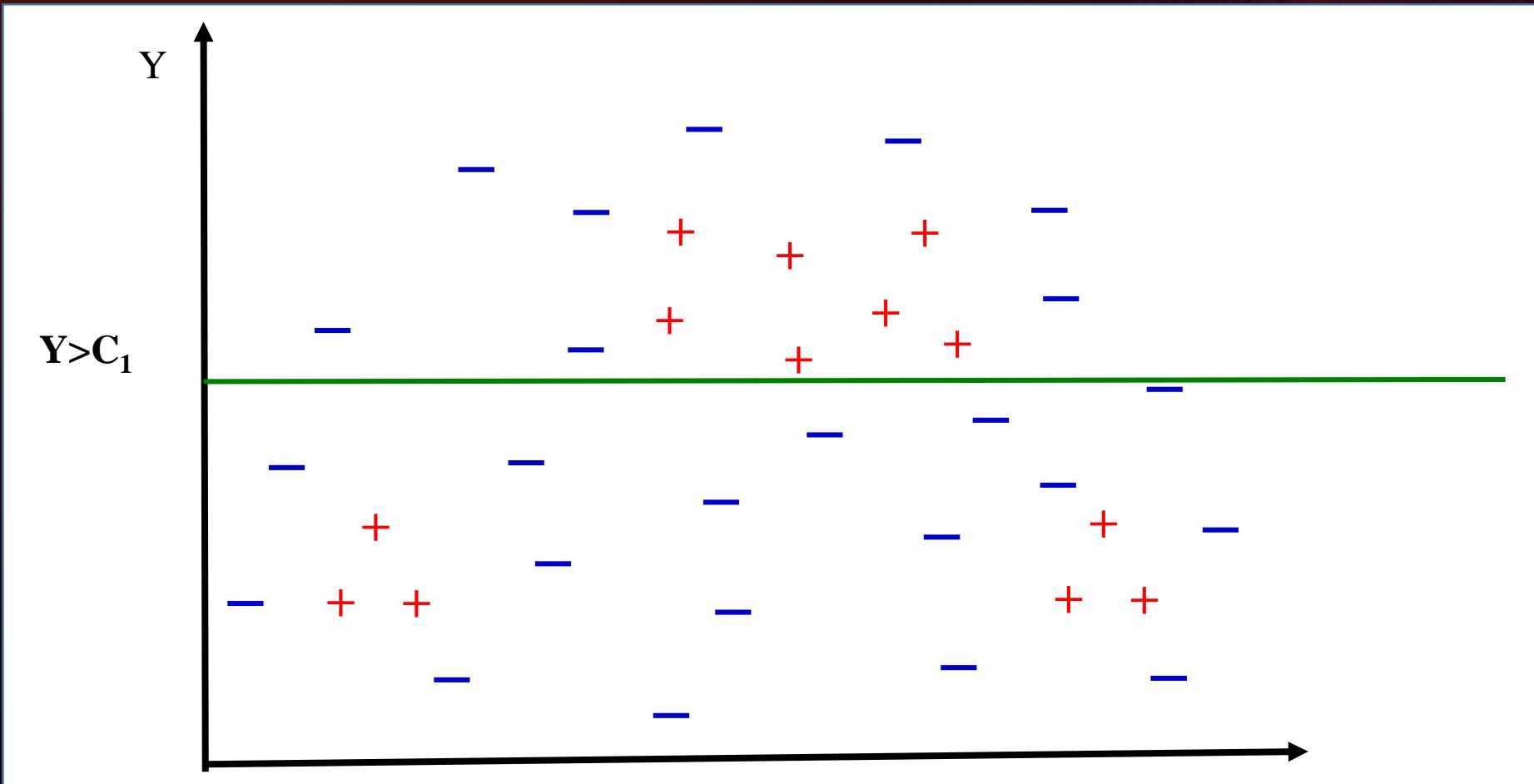


ESTRATÉGIA TOP-DOWN

Começar com a regra mais geral (vazia).

Repetidamente, adicionar restrições ao antecedente usando características que eliminem o máximo de negativos enquanto mantêm o maior número de positivos.

Parar quando somente positivos estiverem cobertos.

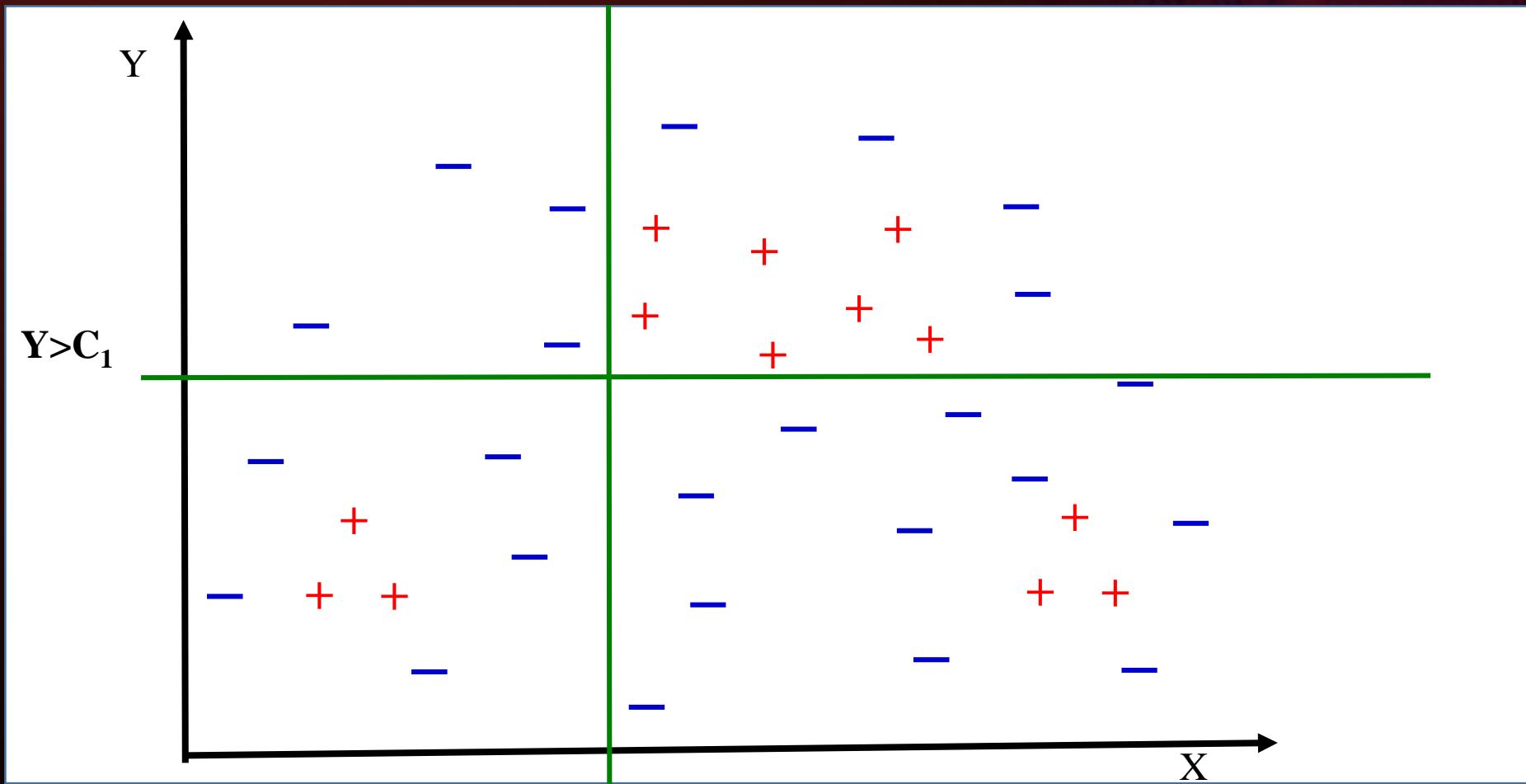


ESTRATÉGIA TOP-DOWN

Começar com a regra mais geral (vazia).

Repetidamente, adicionar restrições ao antecedente usando características que eliminem o máximo de negativos enquanto mantêm o maior número de positivos.

Parar quando somente positivos estiverem cobertos.

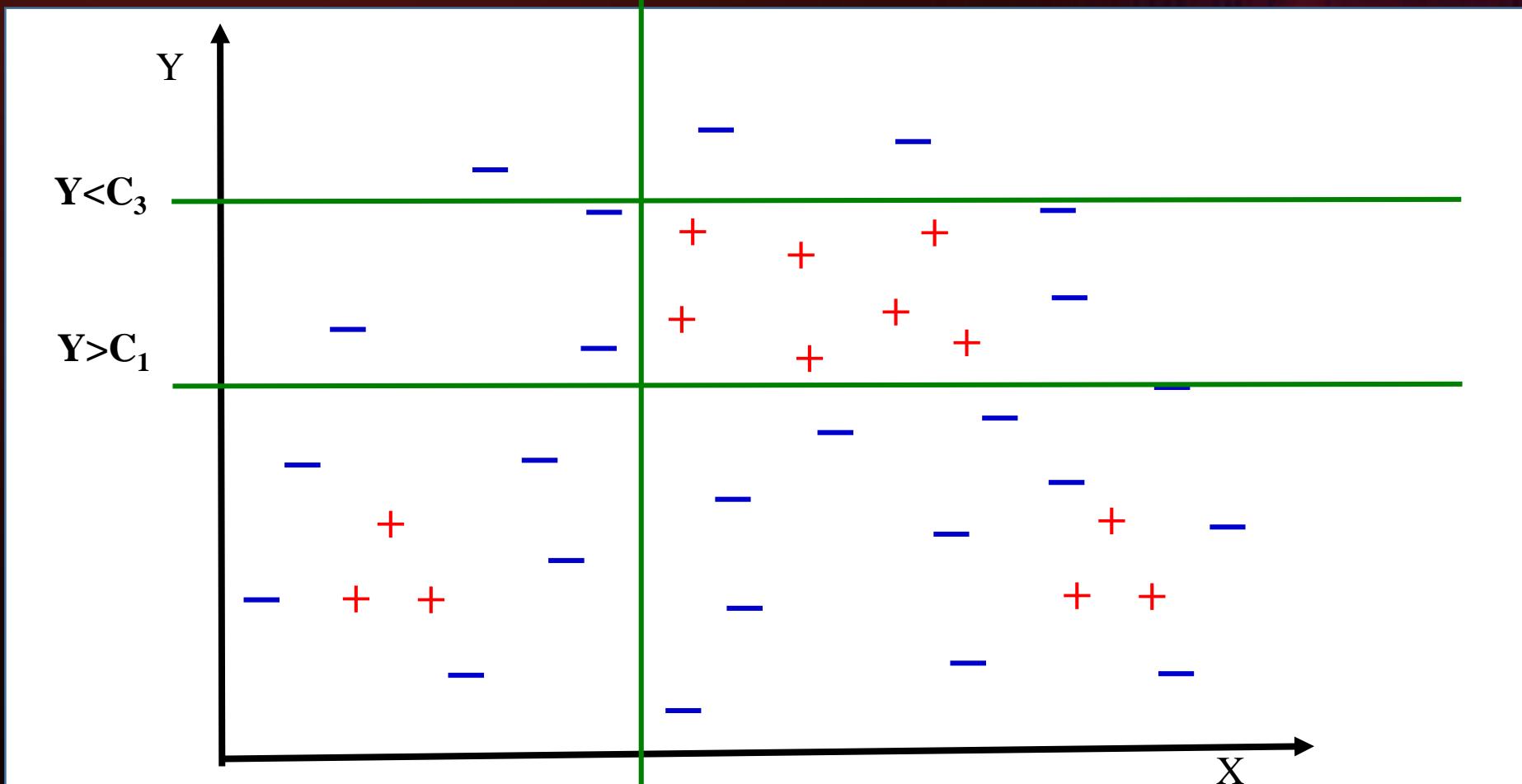


ESTRATÉGIA TOP-DOWN

Começar com a regra mais geral (vazia).

Repetidamente, adicionar restrições ao antecedente usando características que eliminem o máximo de negativos enquanto mantêm o maior número de positivos.

Parar quando somente positivos estiverem cobertos.

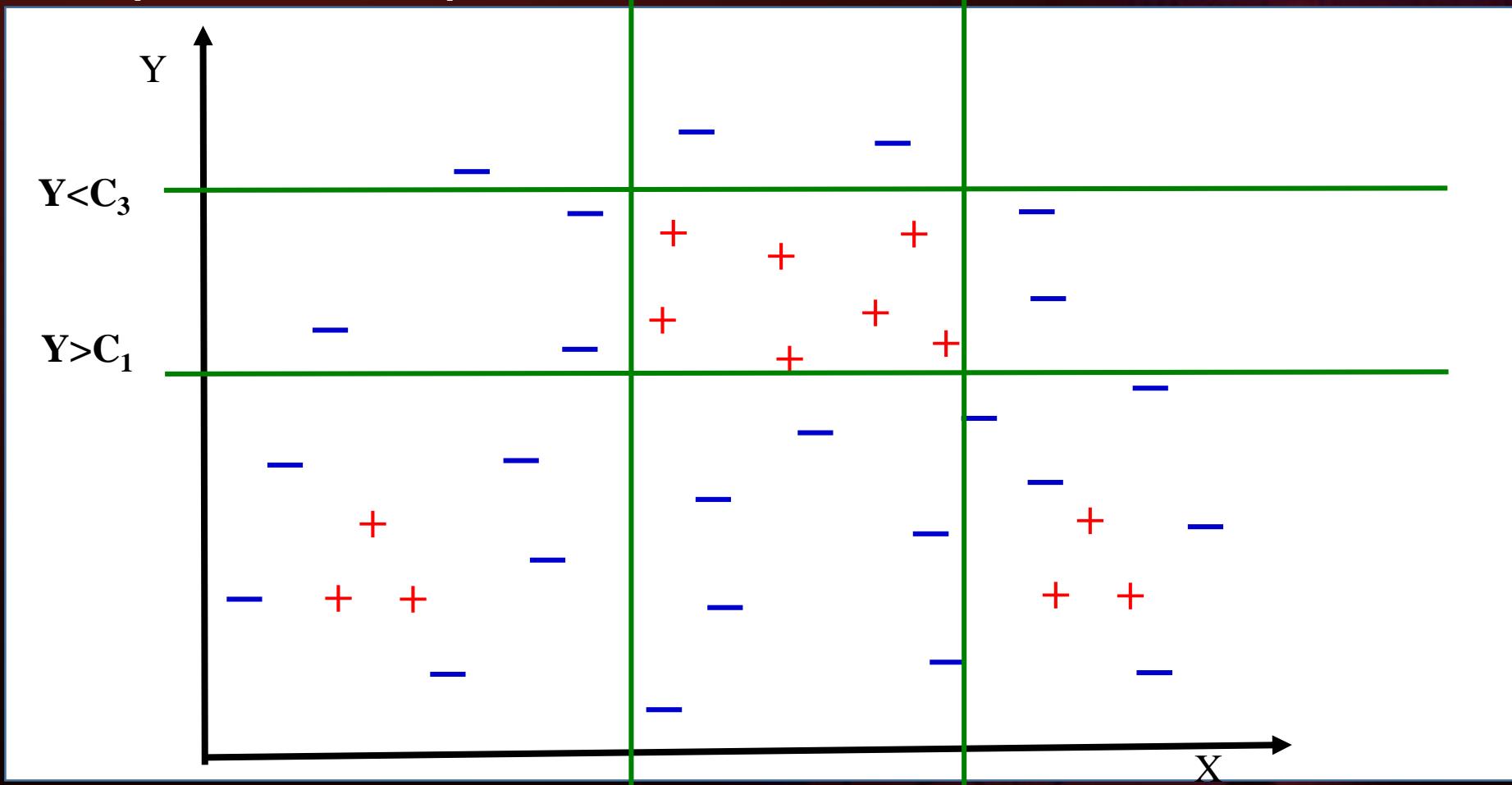


ESTRATÉGIA TOP-DOWN

Começar com a regra mais geral (vazia).

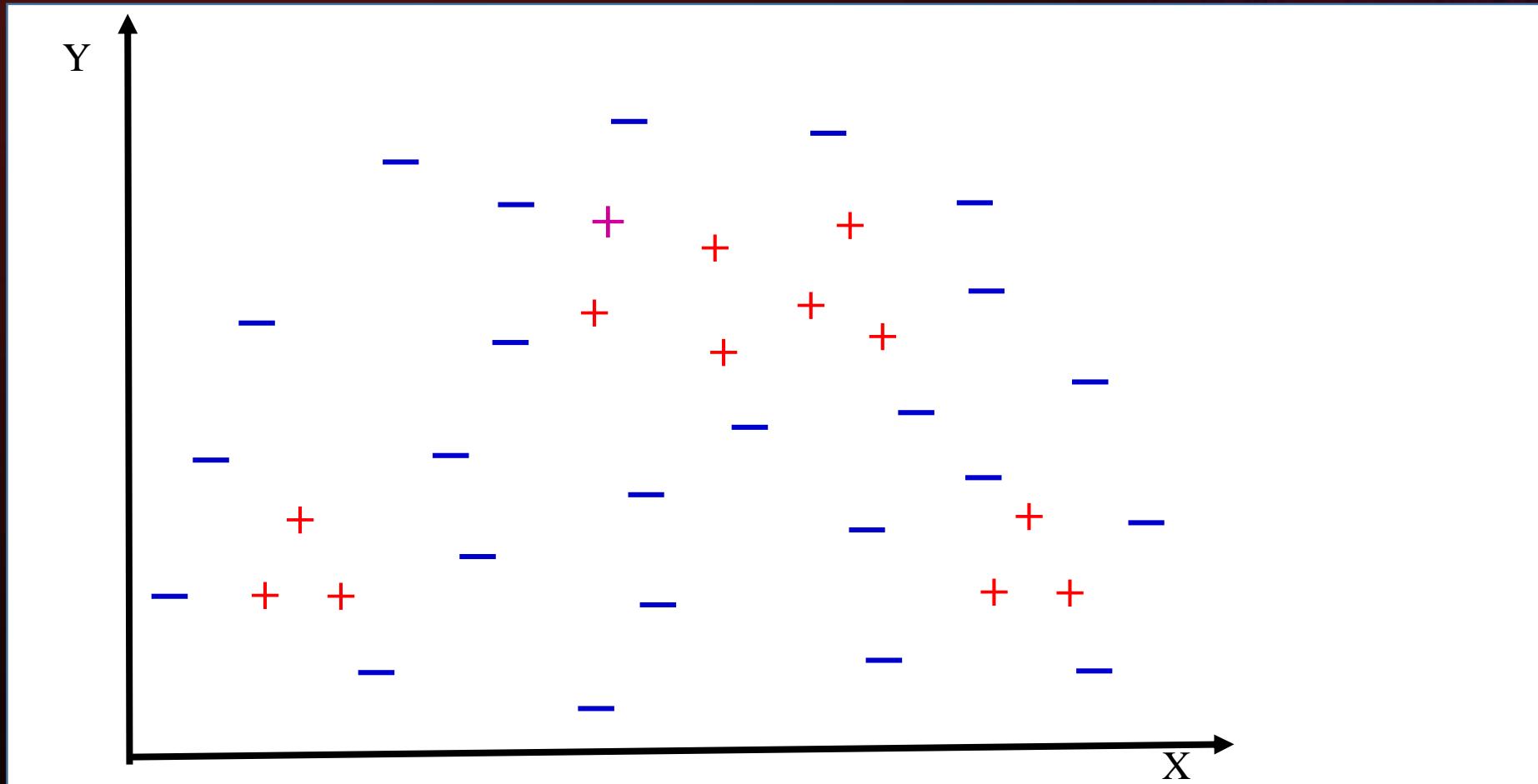
Repetidamente, adicionar restrições ao antecedente usando características que eliminem o máximo de negativos enquanto mantêm o maior número de positivos.

Parar quando somente positivos estiverem cobertos.



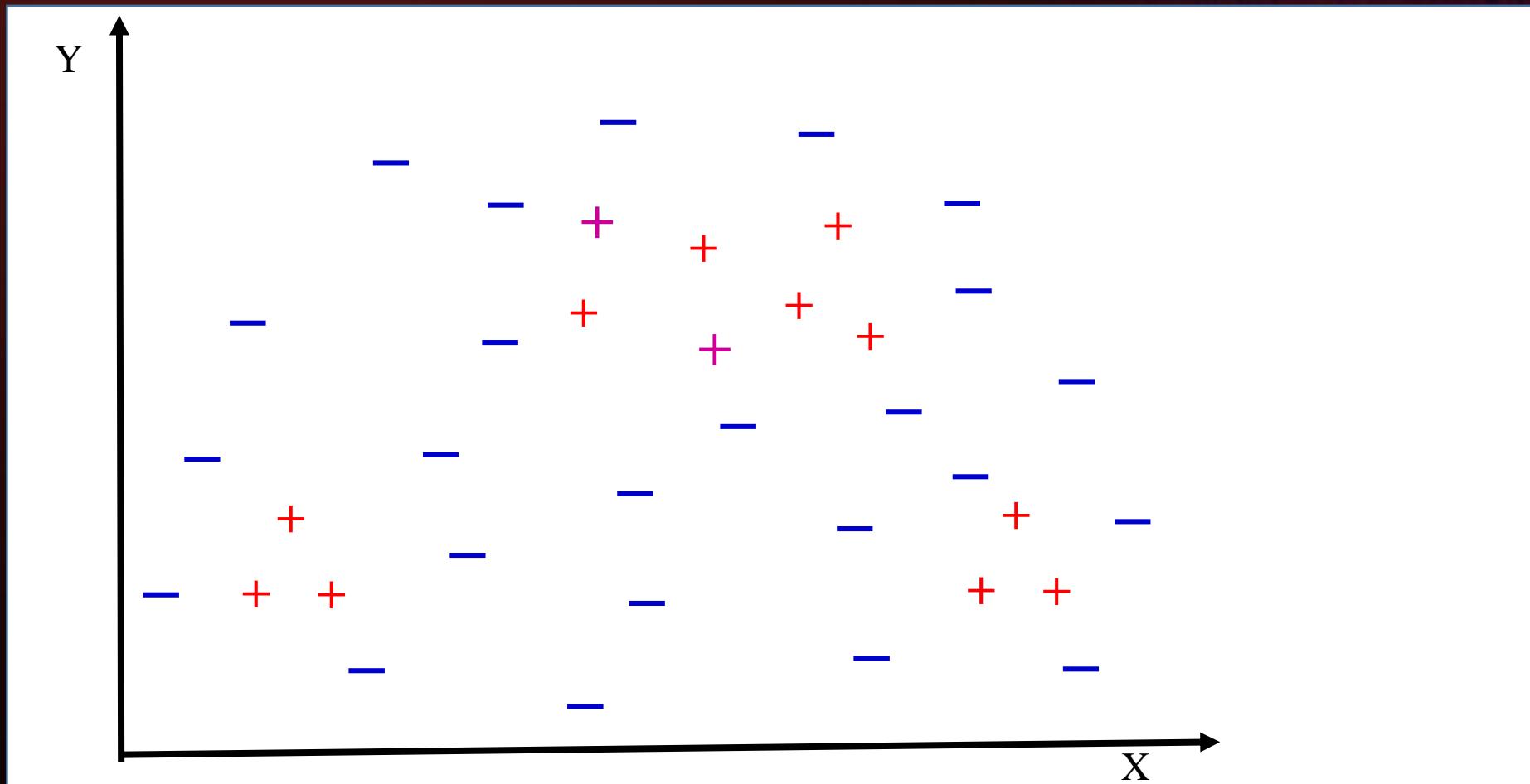
ESTRATÉGIA BOTTOM-UP

Começar com a regra mais específica (por exemplo, uma instância aleatória).
Repetidamente remover restrições do antecedente para cobrir mais positivos.
Parar quando generalização começar a cobrir negativos.



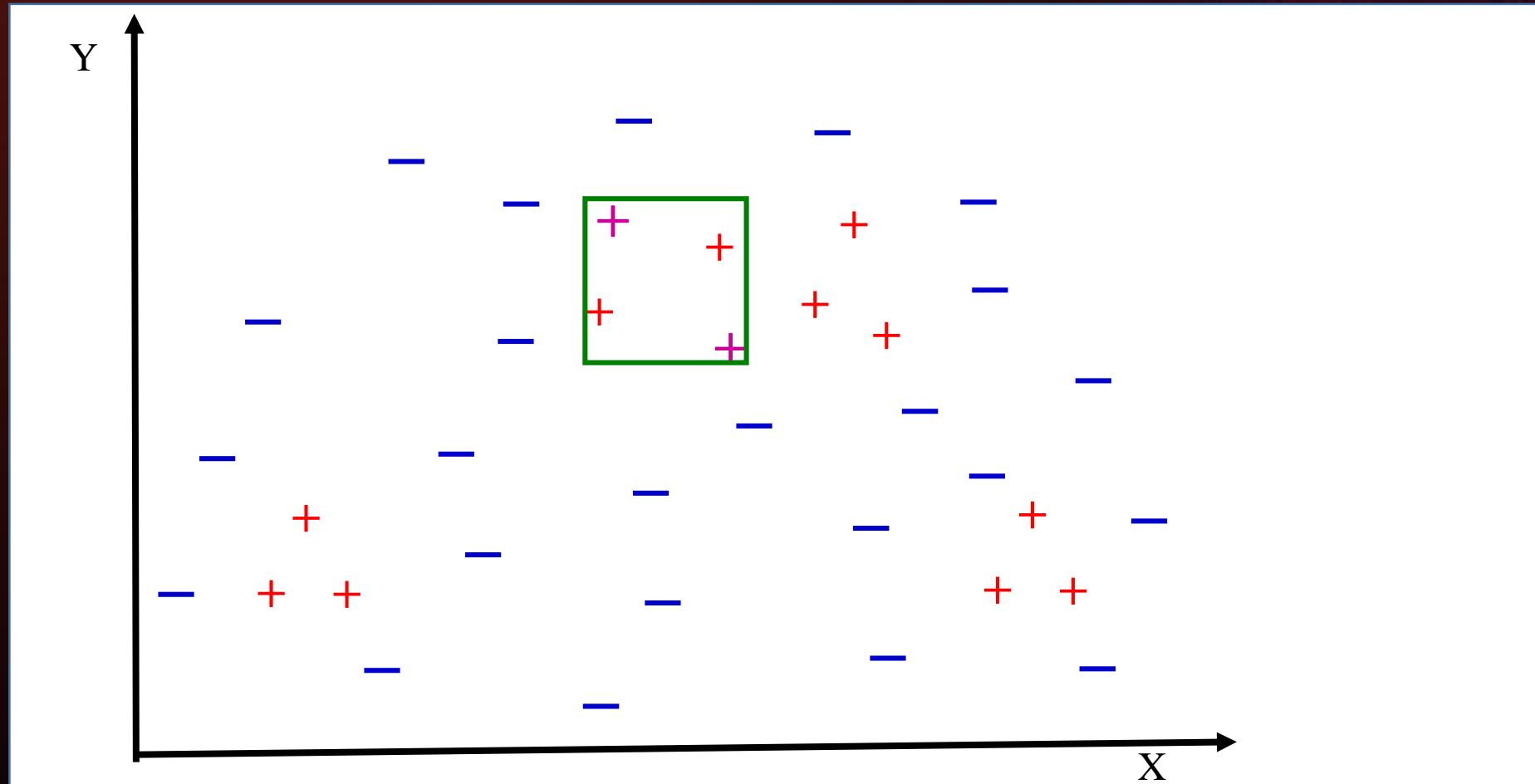
ESTRATÉGIA BOTTOM-UP

Começar com a regra mais específica (por exemplo, uma instância aleatória).
Repetidamente remover restrições do antecedente para cobrir mais positivos.
Parar quando generalização começar a cobrir negativos.



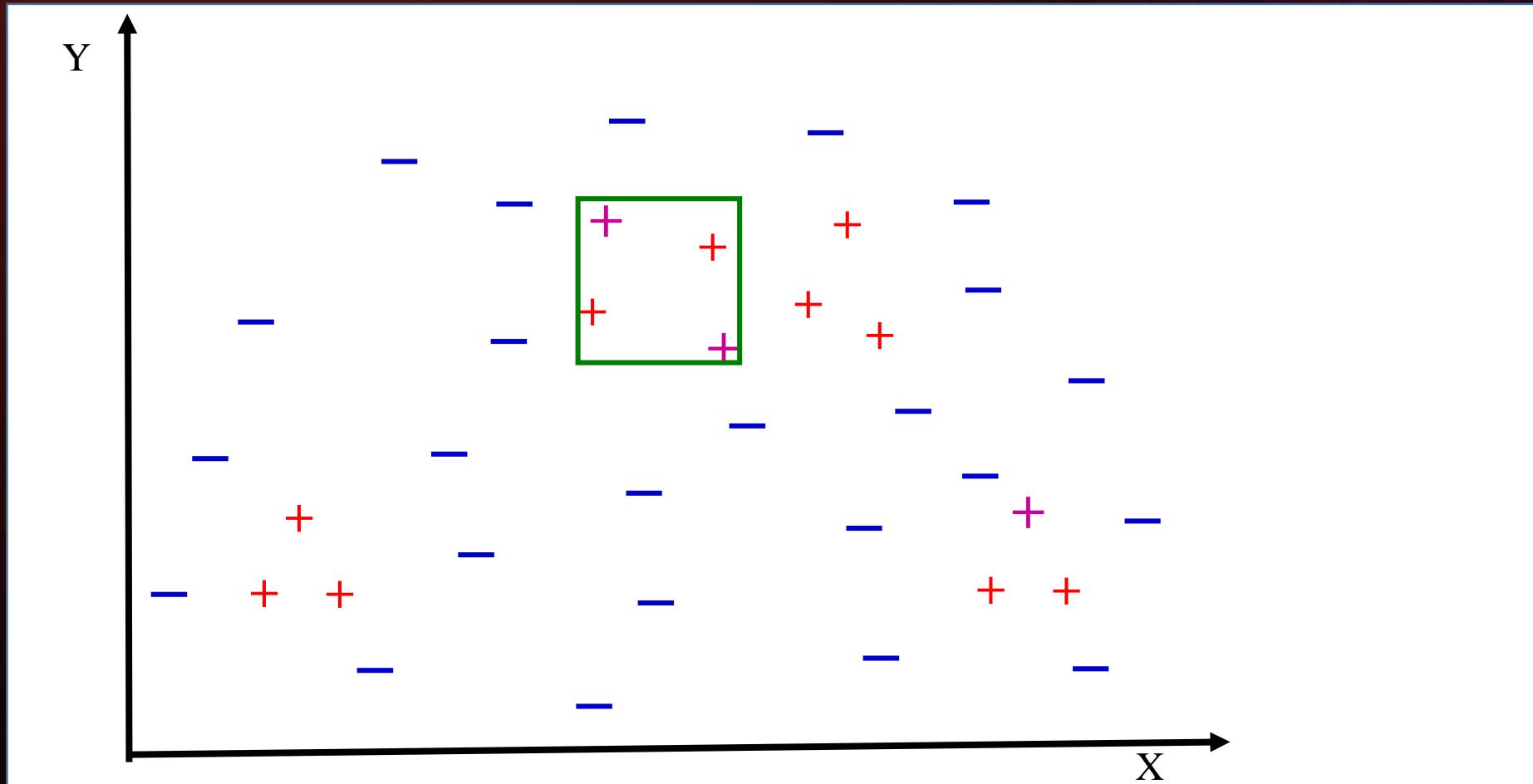
ESTRATÉGIA BOTTOM-UP

Começar com a regra mais específica (por exemplo, uma instância aleatória).
Repetidamente remover restrições do antecedente para cobrir mais positivos.
Parar quando generalização começar a cobrir negativos.



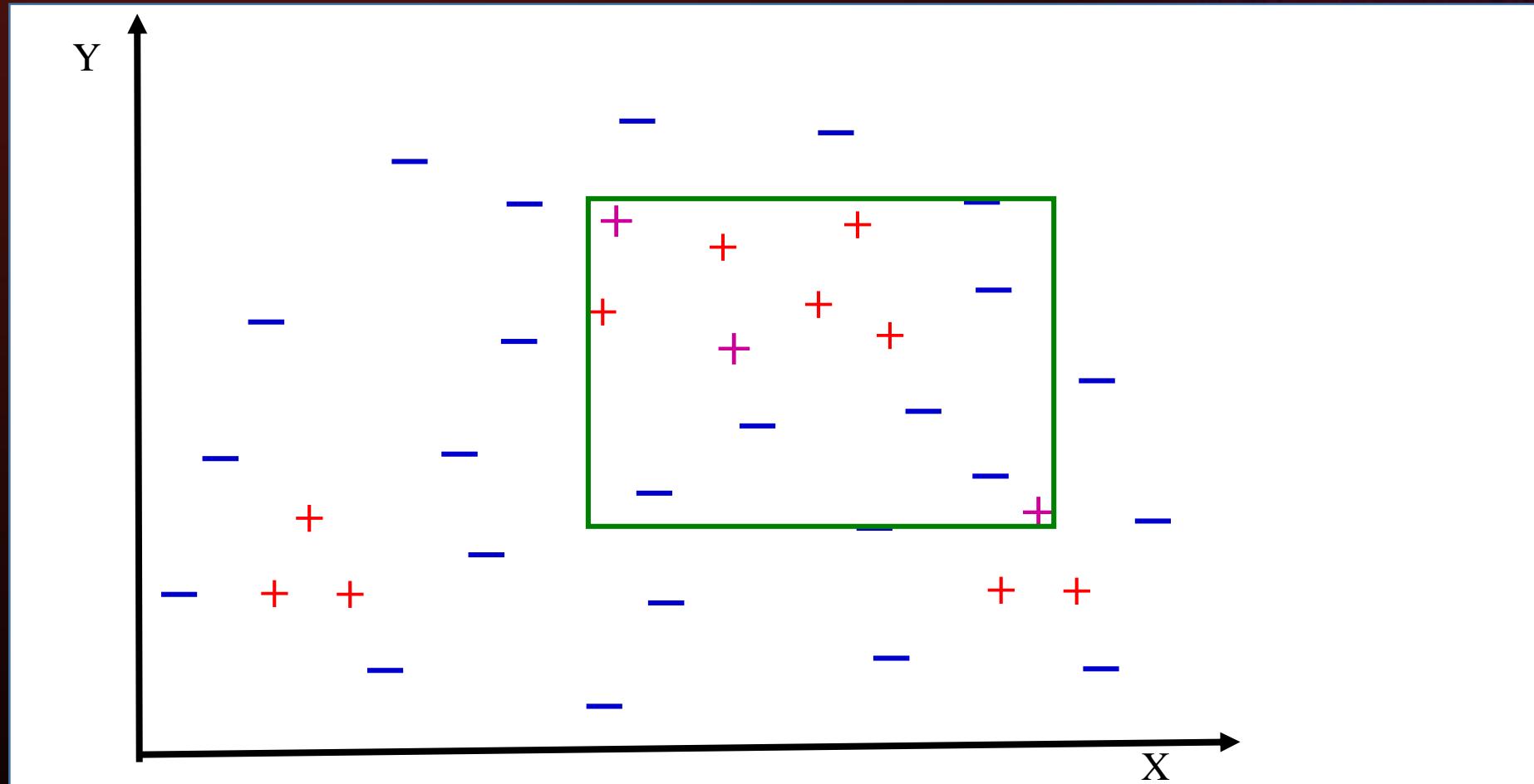
ESTRATÉGIA BOTTOM-UP

Começar com a regra mais específica (por exemplo, uma instância aleatória).
Repetidamente remover restrições do antecedente para cobrir mais positivos.
Parar quando generalização começar a cobrir negativos.



ESTRATÉGIA BOTTOM-UP

Começar com a regra mais específica (por exemplo, uma instância aleatória).
Repetidamente remover restrições do antecedente para cobrir mais positivos.
Parar quando generalização começar a cobrir negativos.



EXEMPLO – CN2

Exemplo:

Exemplos de Treino					
Dia	Aspecto	Temp.	Humidade	Vento	Jogar Tênis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

O objetivo é encontrar uma regra para Joga = Sim. Neste exemplo ilustrativo vamos usar como função de avaliação de hipóteses a taxa de erro. Obviamente, queremos minimizar essa função.

A procura começa com a regra mais geral $\{\}$ → Sim. A regra não tem restrições, ou seja, tudo é da classe Sim. A taxa de erro é de 5/14

EXEMPLO – CN2

Introduzindo uma restrição, o conjunto de hipóteses é:

- Atributo *Tempo*
 - $\text{Tempo} = \text{Ensolarado} \rightarrow \text{Sim}; \left(\frac{3}{5}\right)$
 - $\text{Tempo} = \text{Nublado} \rightarrow \text{Sim}; \left(\frac{0}{4}\right)$
 - $\text{Tempo} = \text{Chuvoso} \rightarrow \text{Sim}; \left(\frac{2}{5}\right)$
- Atributo *Temperatura*
 - $\text{Temperatura} = \text{Quente} \rightarrow \text{Sim}; \left(\frac{3}{4}\right)$
 - $\text{Temperatura} = \text{Fresco} \rightarrow \text{Sim}; \left(\frac{1}{6}\right)$
 - $\text{Temperatura} = \text{Frio} \rightarrow \text{Sim}; \left(\frac{1}{4}\right)$
- Atributo *Umidade*
 - $\text{Umidade} = \text{Alta} \rightarrow \text{Sim}; \left(\frac{3}{7}\right)$
 - $\text{Umidade} = \text{Normal} \rightarrow \text{Sim}; \left(\frac{2}{7}\right)$
- Atributo *Vento*
 - $\text{Vento} = \text{Sim} \rightarrow \text{Sim}; \left(\frac{3}{6}\right)$
 - $\text{Vento} = \text{Não} \rightarrow \text{Sim}; \left(\frac{2}{8}\right)$

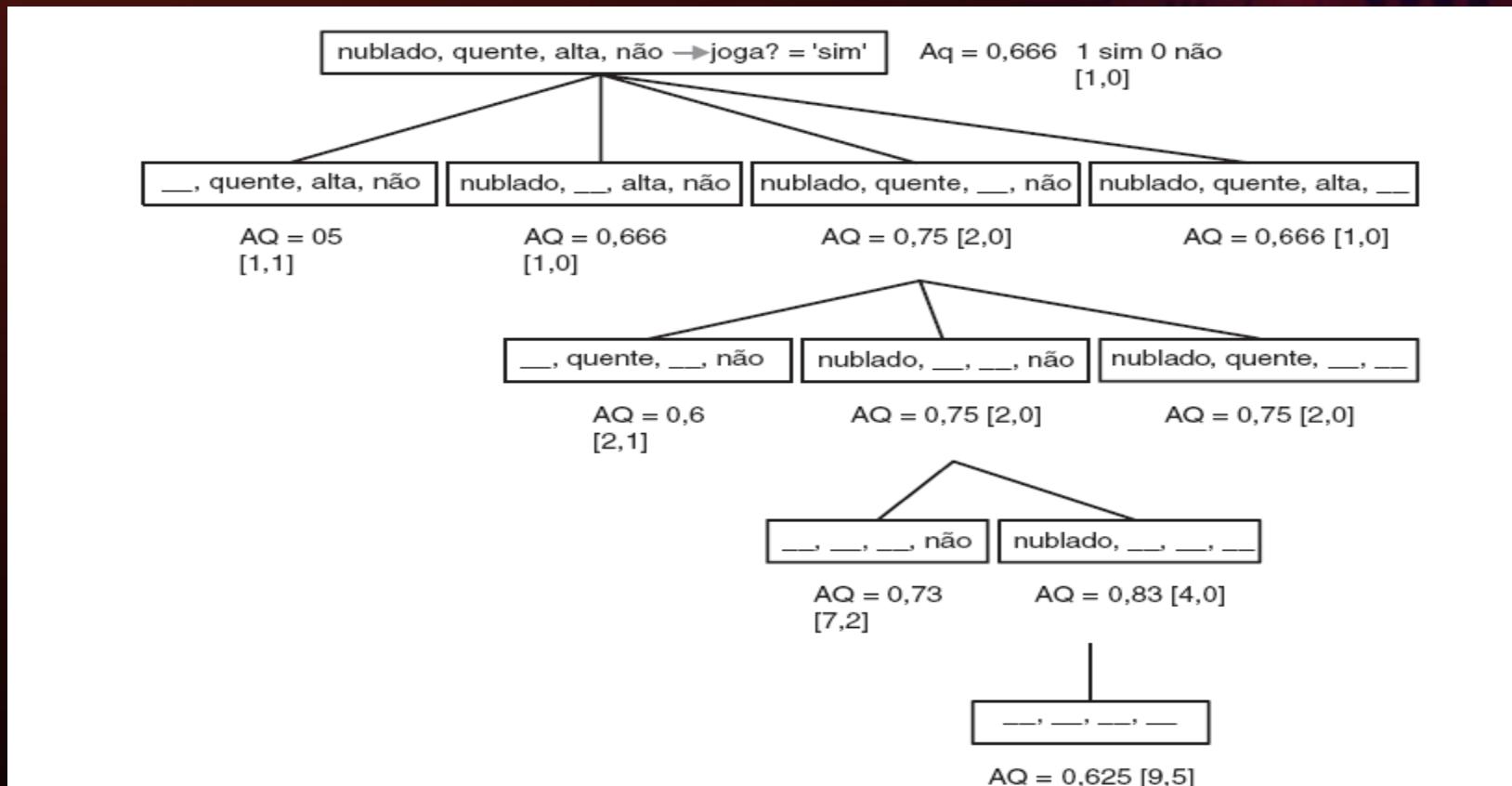
Como foi encontrada uma regra com taxa de erro 0, $\text{Tempo} = \text{Nublado} \rightarrow \text{Sim}$, o processo de encontrar uma regra termina. O algoritmo de cobertura remove os exemplos cobertos pela regra2 e retorna o processo de encontrar uma nova regra a partir do subconjunto de exemplos.

Dia	Aspecto	Temp.	Humidade	Vento	Jogar Ténis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

EXEMPLO - AQ

O algoritmo AQ segue uma estratégia bottom-up, a procura é efetuada do mais específico para o geral. Para construir uma regra para uma classe, o ponto de partida é um exemplo dessa classe. O operador de generalização remove condições da regra atual. No caso do sistema AQ, a função de avaliação é:

$$\delta(S) = \phi(p_1, p_2, \dots, p_k) - P_L \times \phi(p_{1L}, p_{2L}, \dots, p_{kL}) - P_R \times \phi(p_{1R}, p_{2R}, \dots, p_{kR})$$



APLICANDO REGRAS

Situações particulares:

- i) nenhuma regra dispara;
- ii) apenas uma regra dispara; ou
- iii) mais que uma regra dispara.

O primeiro caso pode ser evitado, acrescentando uma regra sem parte condicional e cuja conclusão seja, por exemplo, a classe majoritária. Se mais que uma regra dispara, podemos ter situações de conflito. As soluções mais usuais para esses casos consistem em ordenar as regras por prevalência, ou qualquer critério de mérito.

Enquanto o método top-down gera regras ordenadas pela ordem em que são induzidas, o método bottom-up gera um conjunto não ordenado de regras. Na aplicação do conjunto de regras a exemplos não classificados, há duas estratégias básicas. No caso de conjuntos ordenados de regras, cada exemplo é classificado pela primeira regra cuja parte condicional é satisfeita. Neste contexto, é frequente adicionar uma regra *default* sem parte condicional, que se aplica quando nenhuma das regras dispara. Como vimos, o algoritmo de cobertura termina quando existem apenas exemplos de uma classe. A regra *default* tem como conclusão essa classe. No caso de conjuntos de regras não ordenadas, todas as regras cuja parte condicional é verificada são utilizadas para classificar o exemplo, tipicamente por votação ponderada pela qualidade da regra.

APRENDIZADO DE MÁQUINAS

Aprendizado de reglas

