

APRENDER A PARTIR DE EXEMPLOS

Capítulo 18, secções 1 – 2

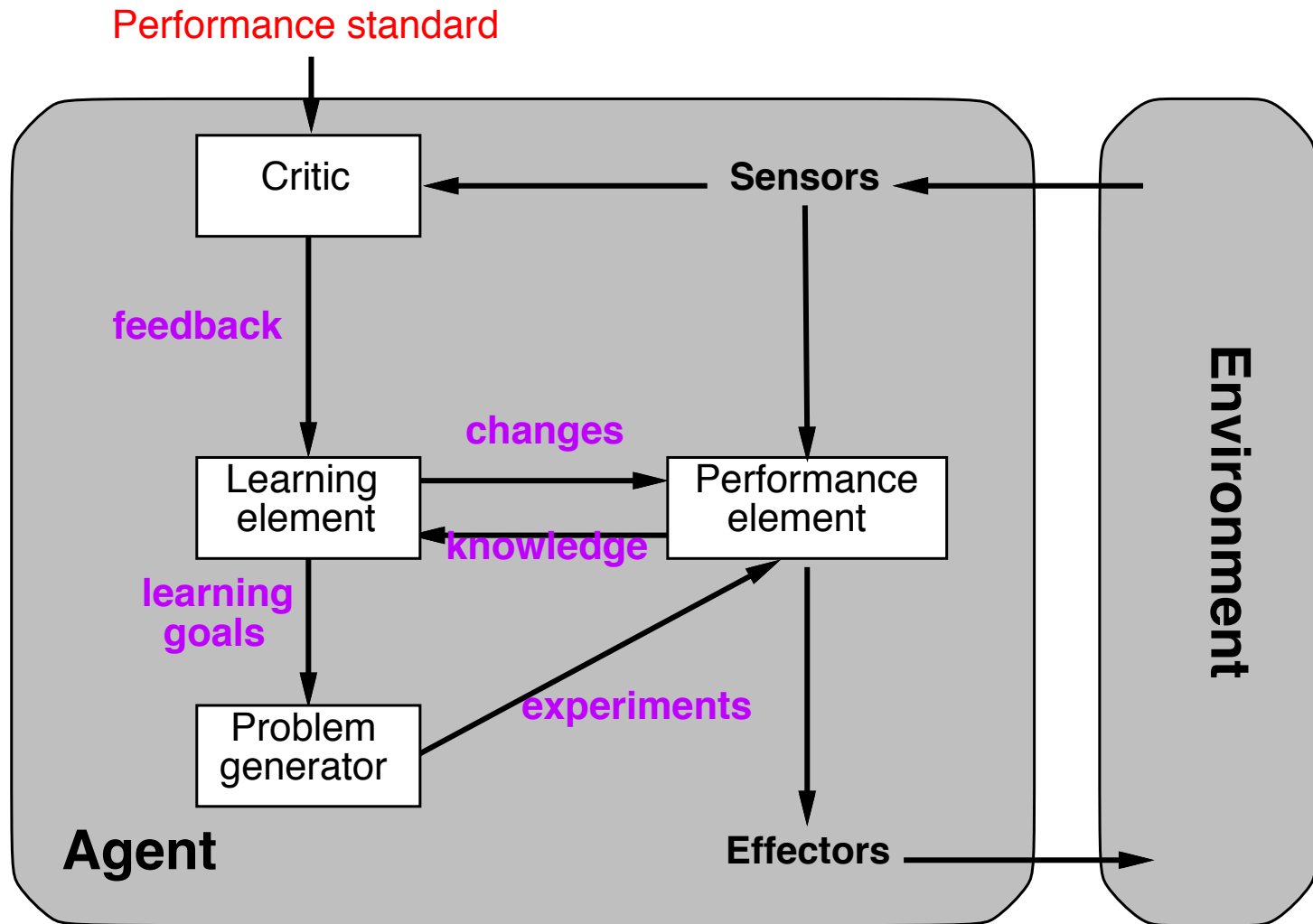
Resumo

- Agentes aprendizes
- Problemas de aprendizagem
- Abordagens
- Aprendizagem Indutiva
- Aprendizagem de Conceitos
- Aprendizagem baseada em Exemplos

Aprendizagem

- A aprendizagem é essencial em ambientes desconhecidos, i.e., quando o projectista do sistema não possui omnisciência
 - Não consegue antecipar todos os cenários possíveis (e.g. aprender um mapa de um labirinto)
 - Não consegue antecipar todas as alterações com o tempo (e.g. adaptar a novas circunstâncias na bolsa)
 - Não consegue programar a solução (e.g. reconhecimento de imagens)
- A aprendizagem é útil como um método de construção de sistemas,
 - i.e., expõe o agente à realidade em vez de tentar descrevê-lo
- A aprendizagem modifica os mecanismos de decisão do agente visando o aumento de desempenho

Agentes aprendizes



Elementos de aprendizagem

- O desenho do elemento de aprendizagem é ditado por
 - Quais os **componentes** do elemento de desempenho que devem ser aprendidos
 - Que **conhecimento prévio** o agente já detém
 - Qual é a **representação utilizada** pelos componentes
 - Qual o **feedback disponível** para aprender esses componentes

Componentes a serem aprendidos

- Os componentes de um agente que podem ser aprendidos incluem:
 - Mapeamento entre condições sobre o estado atual e ações
 - Meios de inferência de propriedades relevantes a partir da sequência de percepções
 - Informação sobre a forma como o mundo evolve e sobre os resultados de acções possíveis
 - Informação sobre utilidade, indicando quão desejáveis são os estados do mundo
 - Informação sobre o valor das acções, indicando quão desejáveis são
 - Objectivos que descrevem estados que maximizam a utilidade do agente

Conhecimento prévio e representação

- Tipos de representação:
 - Frases em lógica proposicional
 - Frases em lógica de primeira ordem
 - Redes de Bayes
 - ...
- Existem algoritmos de aprendizagem para estas formas de representação.
- Vamos focar numa representação factorizada:
 - Inputs: vectores de atributos
 - Outputs: valor numérico contínuo ou valor discreto.

Feedback disponível

- **Aprendizagem Não Supervisionada**

- Descobrir padrões nos valores de entrada sem que sejam especificados os valores de saída (exemplos de treino não estão classificados, por exemplo).
 - Clustering: detectar clusters de exemplos potencialmente úteis
 - Por exemplo, um taxista poderá desenvolver uma noção de “dias com mau trânsito” sem que lhe tenham sido dados exemplos classificados como tal.

- **Aprendizagem por Reforço**

- Aprendizagem por reforço, através de recompensas ocasionais. É a forma mais geral de aprendizagem.
 - Por exemplo, a ausência de gorjeta no final de uma corrida de táxi poderá indicar que o taxista fez algo de errado

- **Aprendizagem Supervisionada**

- Aprendizagem de uma função a partir de exemplos de valores de entrada e respectivos resultados (respostas correctas para cada instância). Necessita de “professor”.

Aprendizagem supervisionada

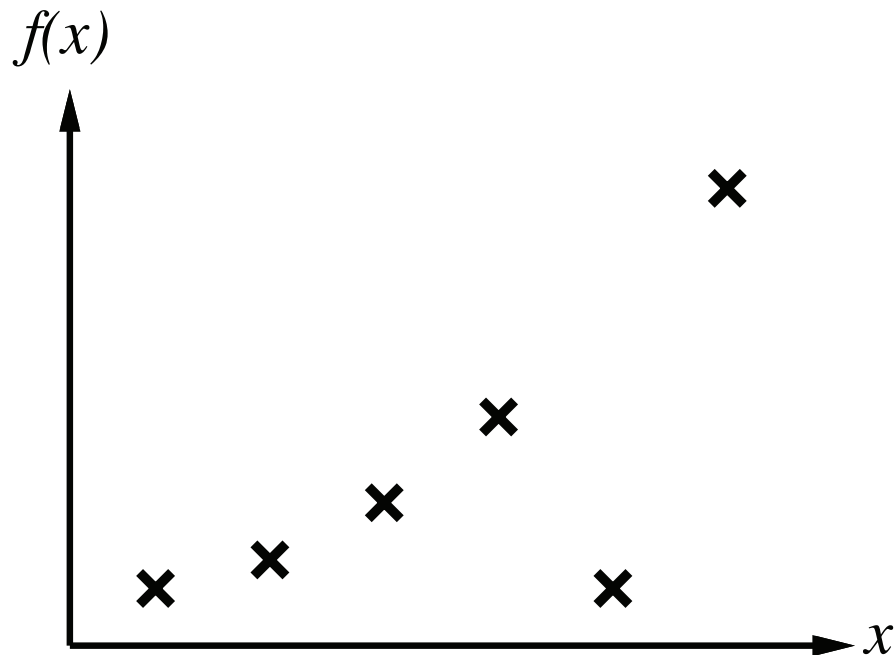
- Aprender uma função a partir de exemplos dados
 - f é uma **função alvo**
 - um **exemplo** é um par $(x, f(x))$
- **Problema da Aprendizagem supervisionada (Aprendizagem Indutiva)**: encontrar uma hipótese h no **espaço de hipóteses** tal que $h \approx f$, dado um **conjunto de treino** de exemplos
 - h diz-se **consistente** quando concorda com f em todos os exemplos
 - h diz-se que **generaliza** bem se classifica corretamente o valor de y para exemplos novos.
 - Por vezes a função é **estocástica**: nesse caso, temos que aprender uma distribuição de probabilidade condicional $\mathbf{P}(Y|x)$.
- Modelo muito simplificado da aprendizagem humana:
 - ignora conhecimento prévio
 - assume que os exemplos são fornecidos

Aprendizagem Supervisionada

- Se a função alvo que estivermos a aprender for discreta estamos na presença de um **problema de classificação**.
- Em particular, se estivermos interessados em funções booleanas dizemos que estamos na presença de um **problema de aprendizagem de conceitos**.
- O caso de aprendizagem de funções contínuas é designado por **regressão** (**interpolação** quando se impõe consistência)
- Todos eles assumem a **Hipótese da Aprendizagem Indutiva**:
“uma hipótese h que aproxime bem a função alvo num conjunto de treino suficientemente grande, também aproximará bem a função alvo num conjunto de exemplos não observados previamente.”

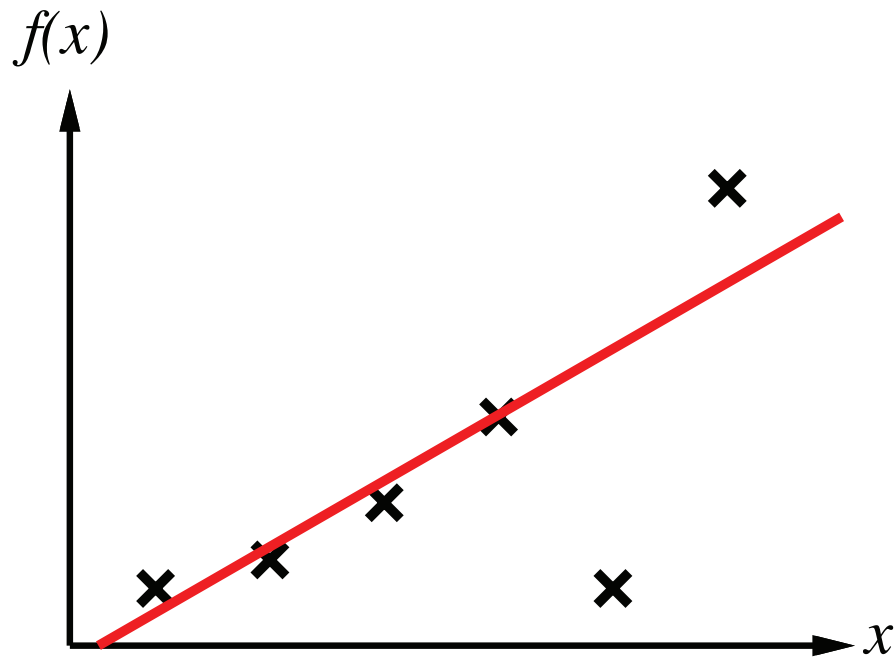
Método da Aprendizagem Indutiva

- Escolher um elemento h do espaço de hipóteses H tal que aproxime f no conjunto de treino
- E.g., regressão:



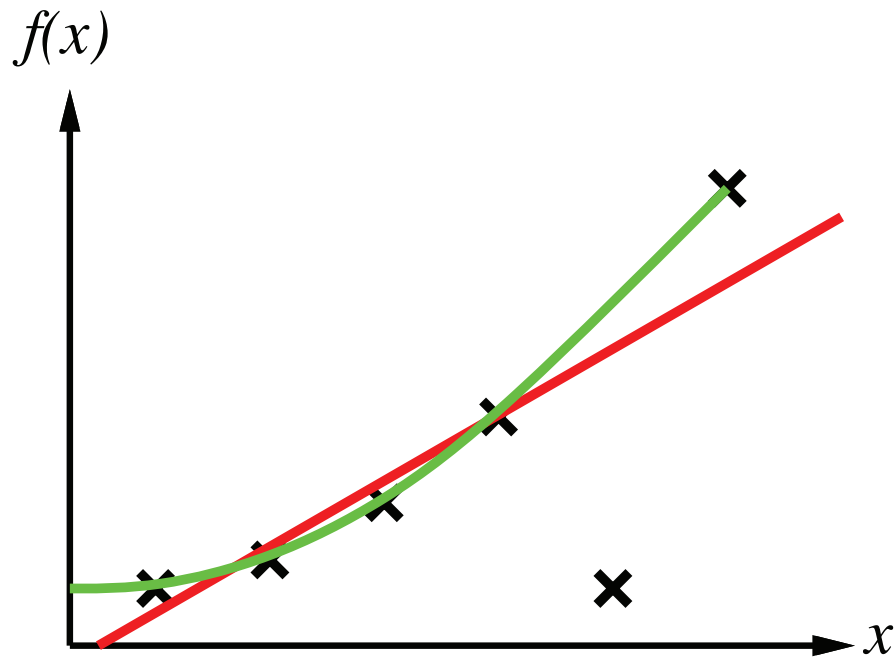
Método da Aprendizagem Indutiva

- Escolher um elemento h do espaço de hipóteses H tal que aproxime f no conjunto de treino
- E.g., regressão:



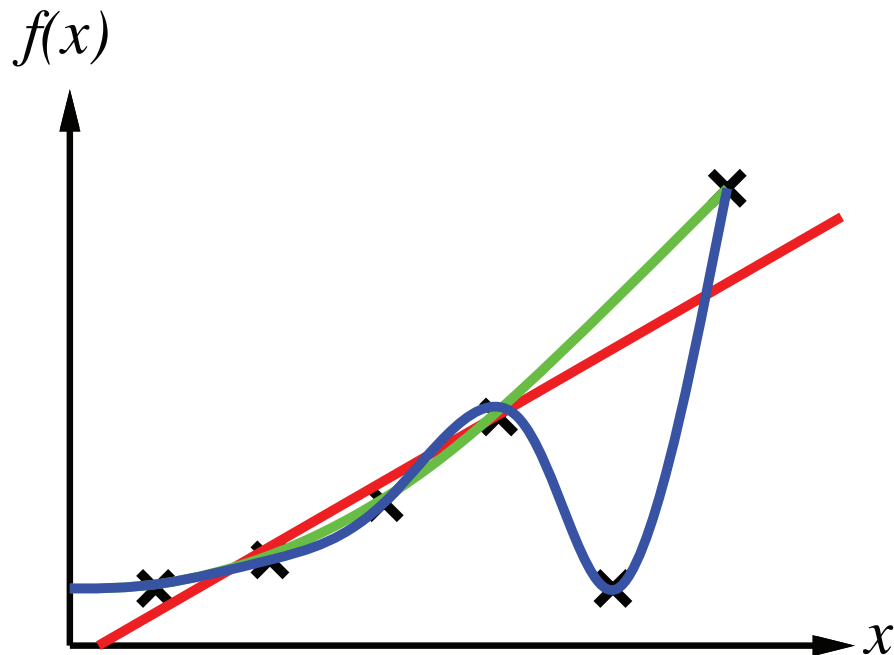
Método da Aprendizagem Indutiva

- Escolher um elemento h do espaço de hipóteses H tal que aproxime f no conjunto de treino
- E.g., regressão:



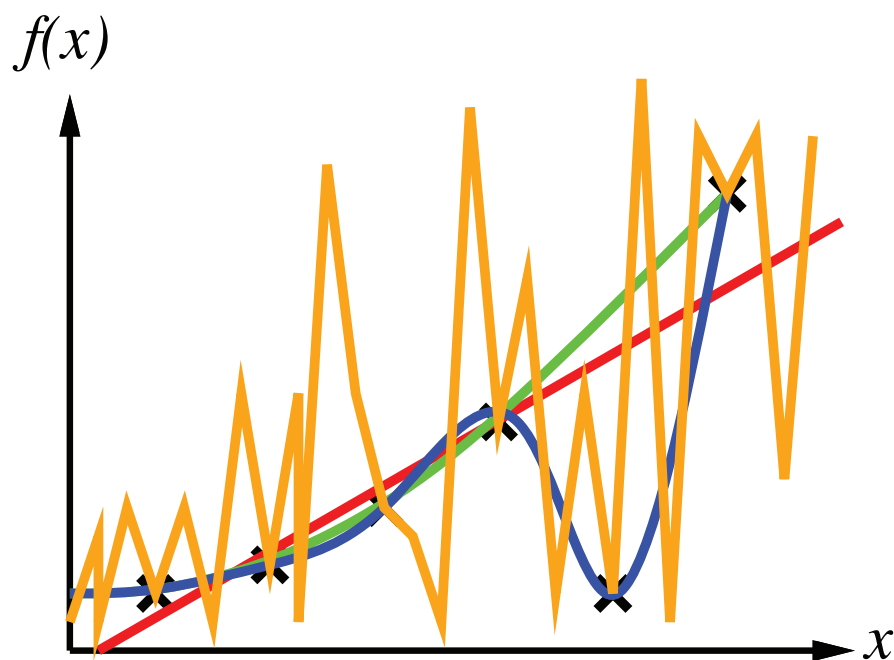
Método da Aprendizagem Indutiva

- Escolher um elemento h do espaço de hipóteses H tal que aproxime f no conjunto de treino
- E.g., regressão:



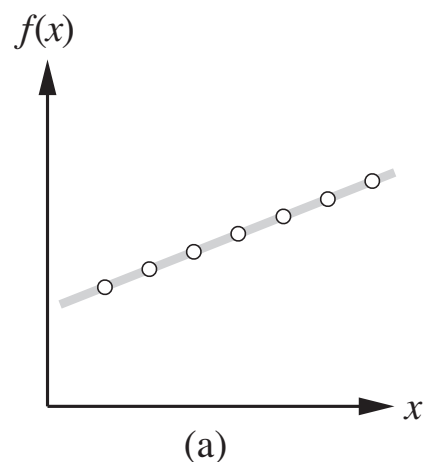
Método da Aprendizagem Indutiva

- Escolher um elemento h do espaço de hipóteses H tal que aproxime f no conjunto de treino
- E.g., regressão:



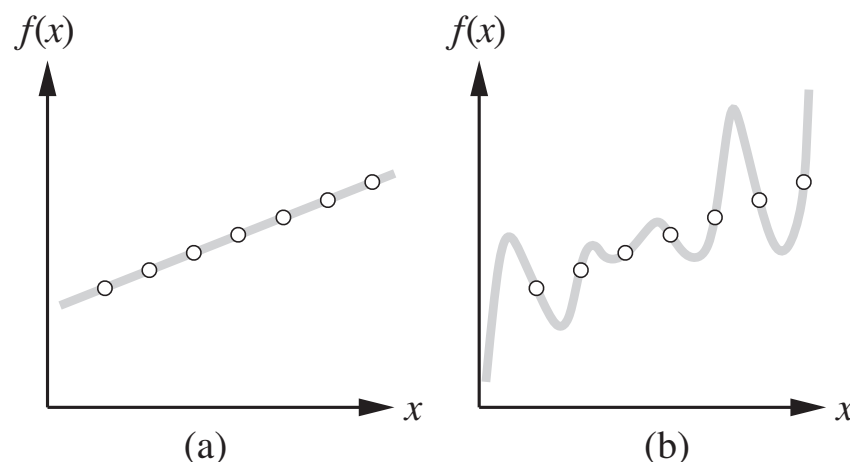
- **Navalha de Ockham:** preferir a hipótese mais simples consistente com os dados

Espaço de hipóteses



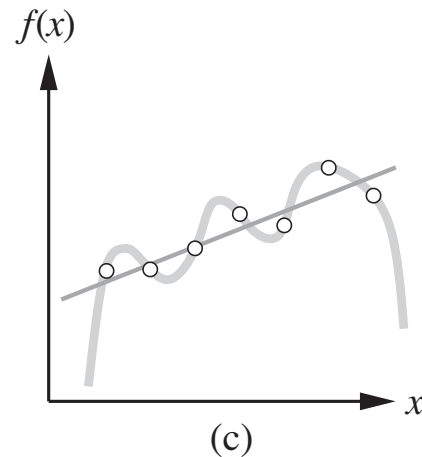
- (a) o polinómio $0.4x+3$ é consistente

Espaço de hipóteses



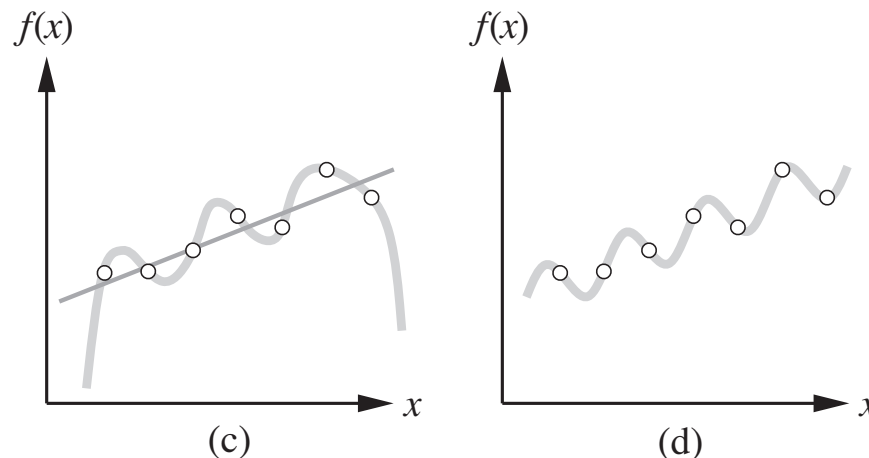
- (a) o polinómio $0.4x+3$ é consistente
- (b) polinómio de grau 7 também é consistente.
- Hipótese (a) é preferível (Navalha de Ockham)

Espaço de hipóteses



- (a) polinómio de grau 6 – consistente. Dado que apenas temos 7 exemplos, um polinómio que requer 7 parâmetros não parece representar quaisquer padrões nos dados, sendo provável que não generalize bem

Espaço de hipóteses



- (a) polinómio de grau 6 – consistente. Dado que apenas temos 7 exemplos, um polinómio que requer 7 parâmetros não parece representar quaisquer padrões nos dados, sendo provável que não generalize bem
- (b) se o conjunto de hipóteses contemplar polinómios em x e em $\sin(x)$, então a função $ax + b + c \sin(x)$ também é consistente, sendo significativamente mais simples do que (a).
- Demonstra a importância de uma boa escolha do conjunto de hipóteses H .
- É no entanto preciso não exagerar em H (e.g. aprender máquinas de Turing)
 - tradeoff entre expressividade e complexidade de aprender e de usar.

Aprendizagem de Conceitos

- Dados

- Instâncias X descritas por vetores de atributos
- Espaço de hipóteses H
- Conceito alvo $c: X \rightarrow \{0,1\}$
- Exemplos de treino D positivos e negativos

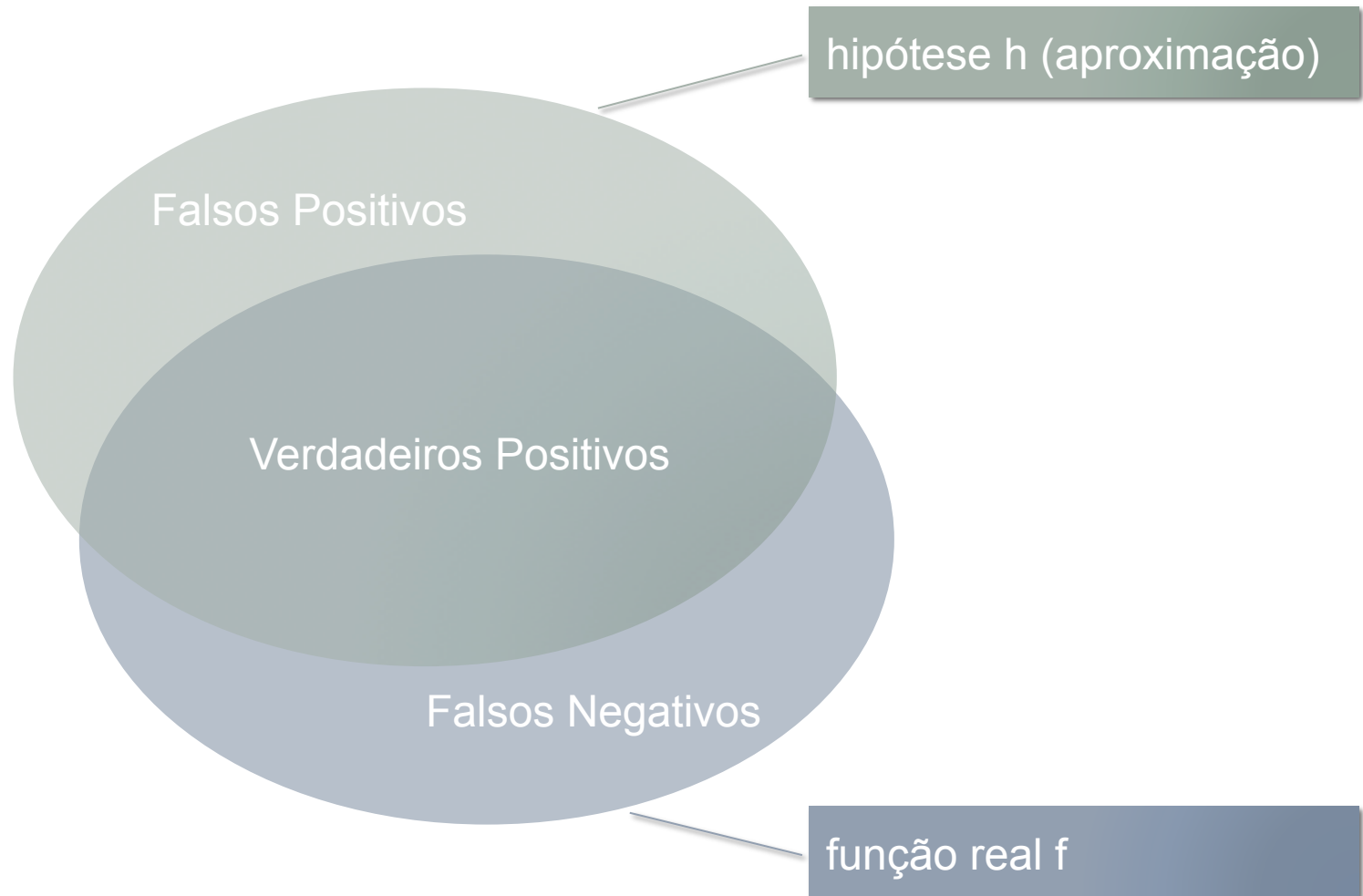
- Determinar

Uma hipótese h no espaço de hipóteses H tal que $h(x) = c(x)$ para todo o x .

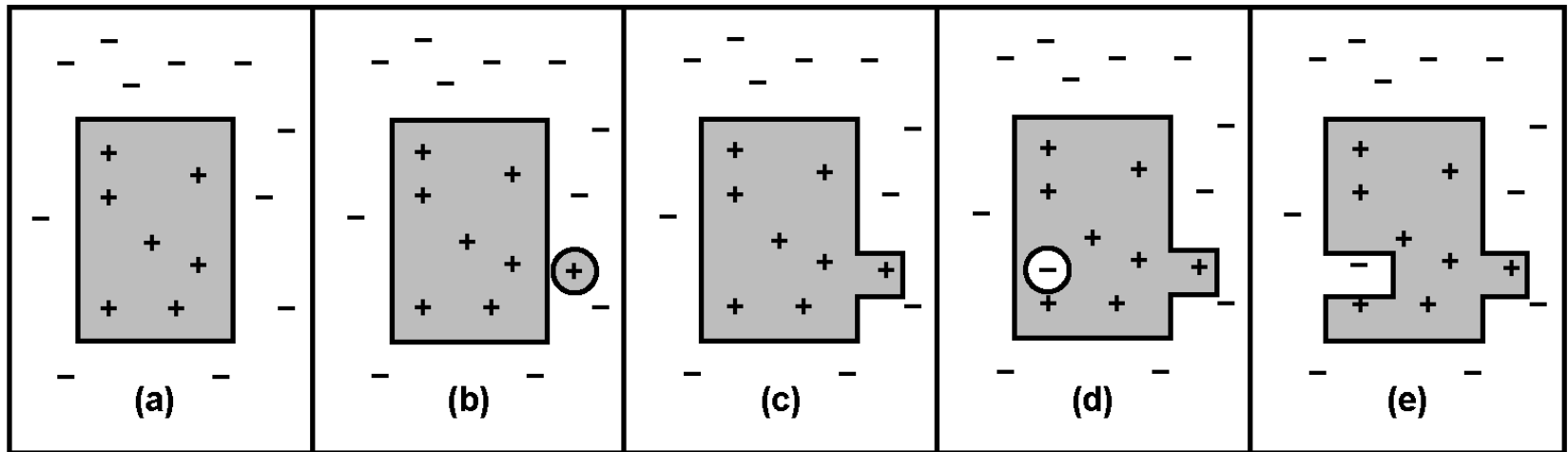
Exemplo (T. Mitchell)

Sky	AirTemp	Humidity	Wind	Water	Forecast	Sport?
Sunny	Warm	Normal	Strong	Warm	Same	1
Sunny	Warm	High	Strong	Warm	Same	1
Rainy	Cold	High	Strong	Warm	Change	0
Sunny	Warm	High	Strong	Cool	Change	1

Falsos positivos e negativos



Tratar falsos positivos e negativos



Hipótese
Consistente

Falso
Negativo

Hipótese
Consistente

Falso
Positivo

Hipótese
Consistente

Generalização
da Hipótese

Especialização
da Hipótese

Espaço de Hipóteses

- Começemos por considerar hipótese conjuntivas com restrições simples nos valores de atributos. Cada restrição pode ser:
 - Um valor específico (e.g. “Water=Warm”)
 - Indiferente (e.g. “Water=?”)
 - Nenhum valor permitido (e.g., “Water=_”)

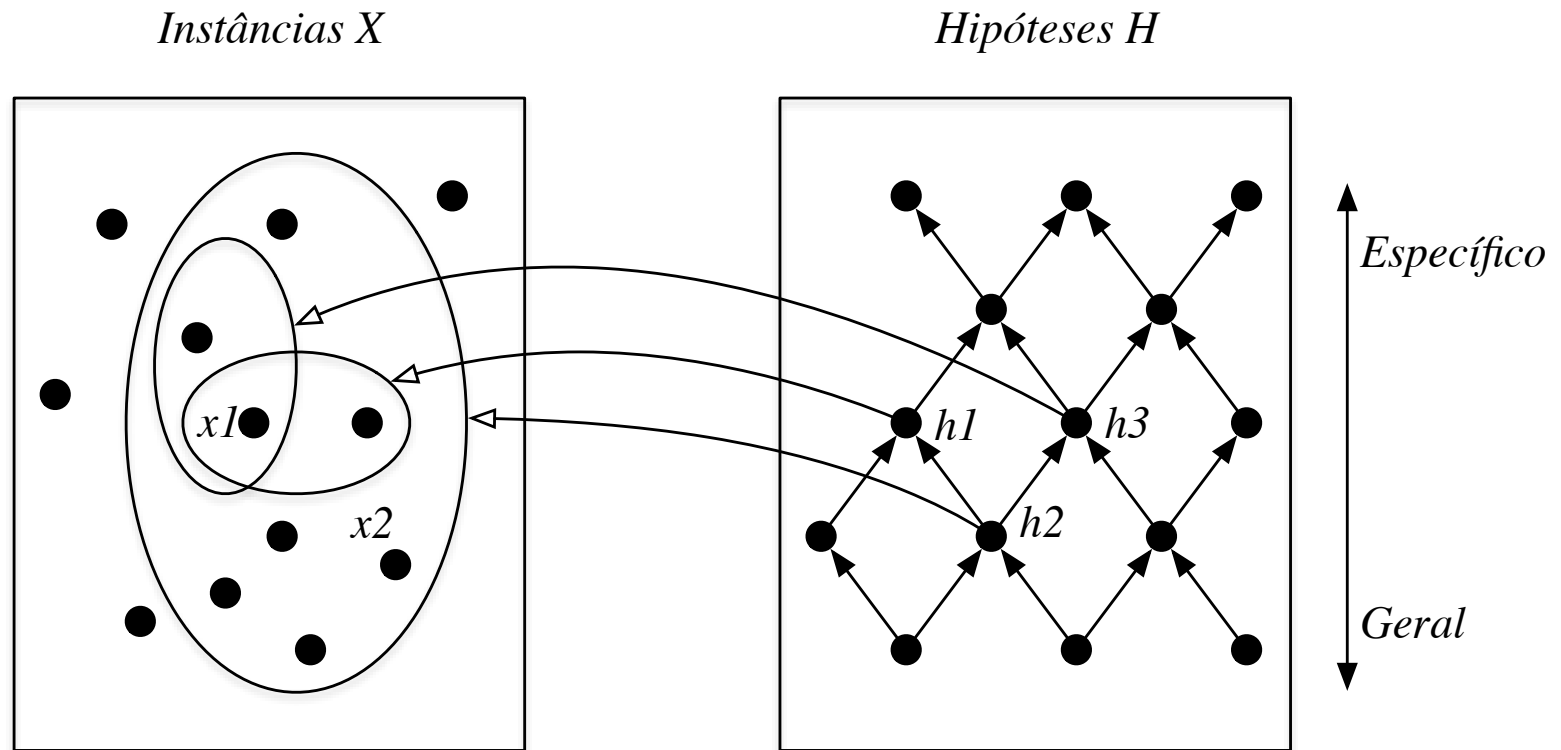
Exemplos

- $h_1 = \langle ?, \text{Cold}, \text{High}, ?, ?, ? \rangle$
- $h_2 = \langle ?, ?, ?, ?, ?, ? \rangle$
- $h_3 = \langle _, _, _, _, _, _ \rangle$

Satisfação e ordenação de hipóteses

- Uma instância x **satisfaz** uma hipótese h sse $h(x) = 1$
 - **Exemplo**
 - A hipótese $h = \langle \text{Sunny}, ?, \text{High}, ?, ?, ? \rangle$ é satisfeita pela instância $\langle \text{Sunny}, \text{Warm}, \text{High}, \text{Strong}, \text{Cool}, \text{Same} \rangle$
 - e não é satisfeita pela instância $\langle \text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Cool}, \text{Same} \rangle$
- Uma hipótese h_i é **mais geral** do que uma hipótese h_j sse todas as instâncias que satisfazem h_j também satisfazem h_i ($h_i \geq h_j$)
- Esta ordem parcial é induzida pela ordem parcial entre restrições simples nos valores do atributo.
 - **Exemplo**
 - $h_1 = \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$
 - $h_2 = \langle \text{Sunny}, ?, \text{High}, ?, ?, ? \rangle$
 - $h_3 = \langle \text{Rainy}, ?, \text{High}, ?, ?, ? \rangle$
 - Logo $h_1 \geq h_2$ (h_1 é mais geral do que h_2)
 - Nota: duas hipóteses podem ser incomparáveis (h_1 e h_3)

Satisfação e ordenação de hipóteses



$x1 = \langle \text{Sunny, Warm, High, Strong, Cool, Same} \rangle$
 $x2 = \langle \text{Sunny, Warm, High, Light, Warm, Same} \rangle$

$h1 = \langle \text{Sunny, ?, ?, Strong, ?, ?} \rangle$
 $h2 = \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle$
 $h3 = \langle \text{Sunny, ?, ?, ?, Cool, ?} \rangle$

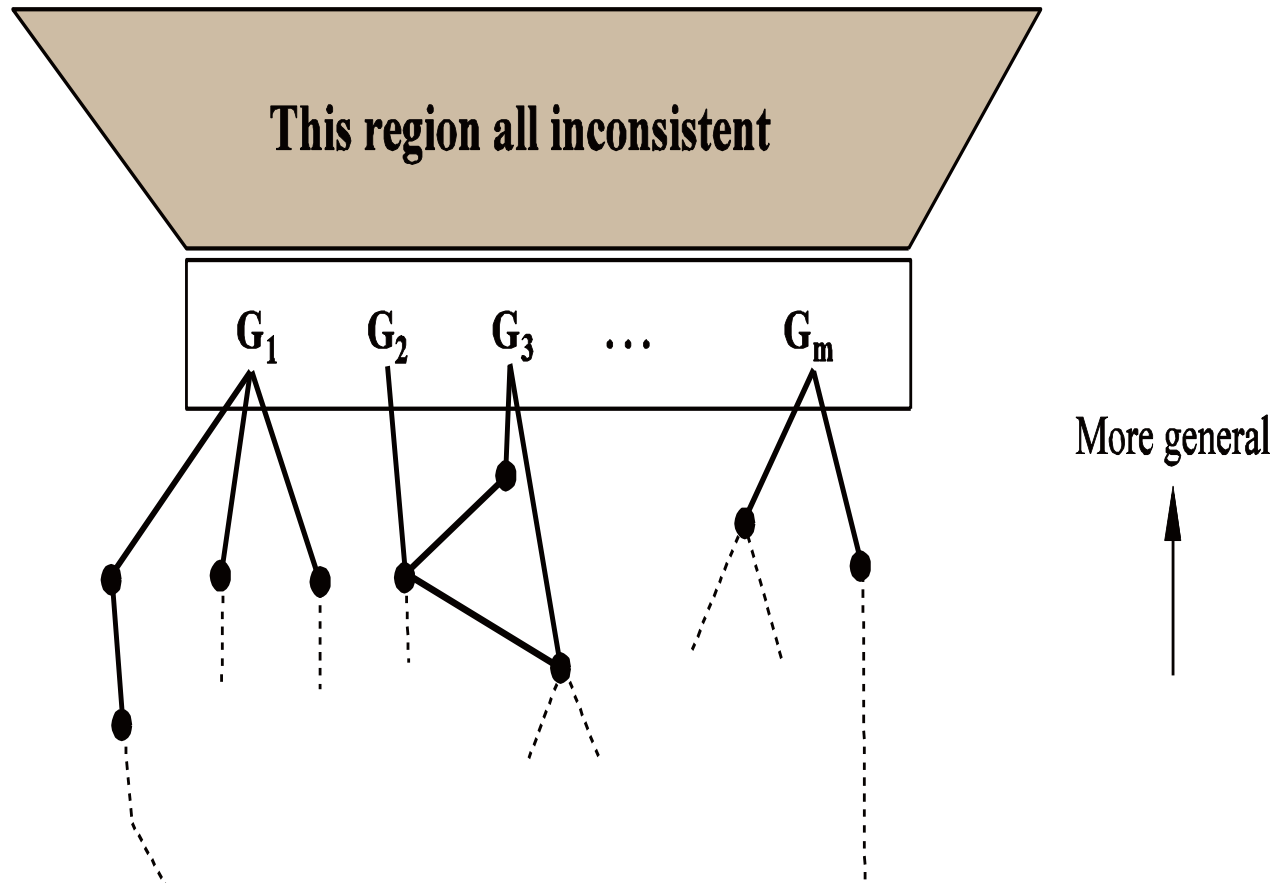
Espaço de Hipóteses

- Dado um conjunto de hipóteses qualquer, dizemos que h é **maximamente específica**, se não existir outra hipótese h' tal que $h \geq h'$
- Dado um conjunto de hipóteses qualquer, dizemos que h é **maximamente geral**, se não existir outra hipótese h' tal que $h' \geq h$

Eliminação de Candidatos

- Algoritmo proposto por Mitchell que efectua a procura no espaço de versões (Version Space - VS), o conjunto de **todas as hipóteses consistentes** com os exemplos apresentados.
- Uma hipótese h é **consistente** com um conjunto de treino D sse $h(x)=c(x)$ para todo o exemplo de treino $\langle x, c(x) \rangle$ em D .
- O algoritmo funciona mantendo duas fronteiras
 - **G**: fronteira mais geral das hipóteses consistentes com os exemplos D
 - **S**: fronteira mais específica das hipóteses consistentes com os exemplos D

O Espaço de Versões (VS)



Evolução das fronteiras

Seja S_i uma hipótese em S e um novo exemplo

- Se for um falso positivo para S_i , então eliminar S_i de S pois não se pode especializar S_i .
- Se for um falso negativo para S_i então substituímo-la pelas suas generalizações imediatas, desde que mais específicas do que algum elemento de G

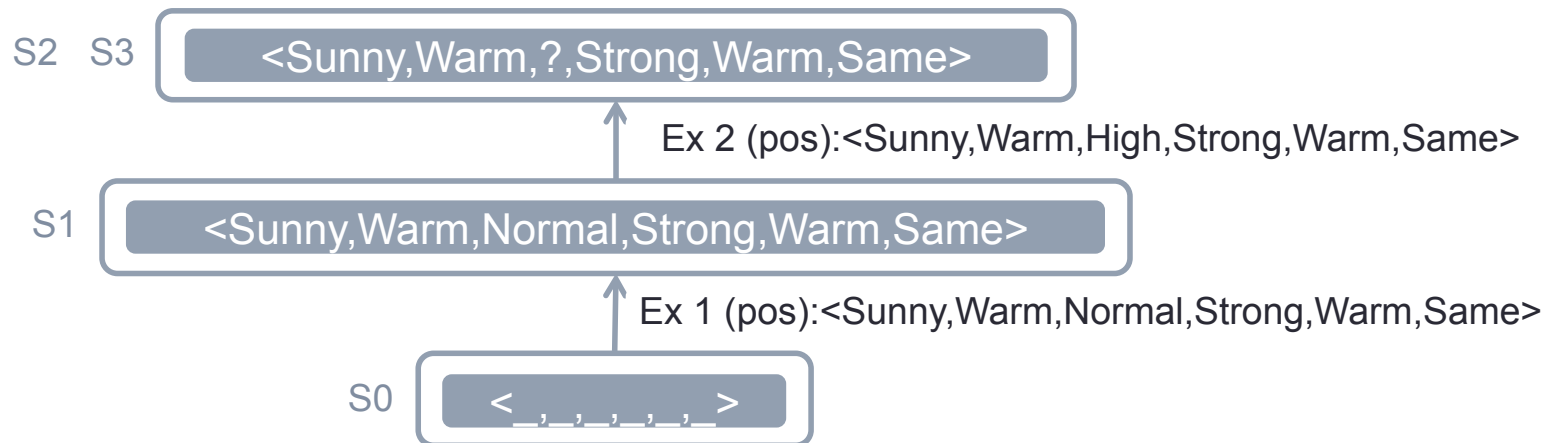
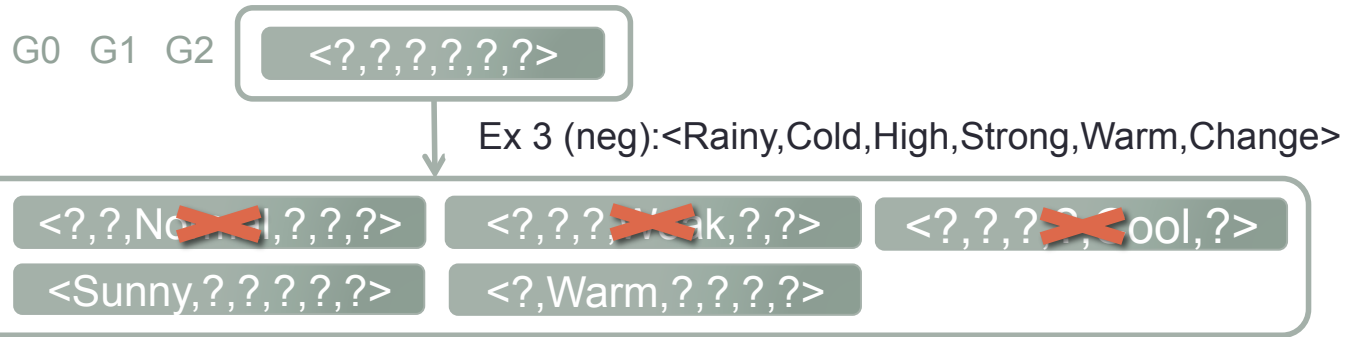
Seja G_i uma hipótese em G e um novo exemplo

- Se for um falso negativo para G_i , então eliminar G_i de G pois não se pode generalizar G_i .
- Se for um falso positivo para G_i então substituímo-la pelas suas especializações imediatas, desde que mais gerais do que algum elemento de S

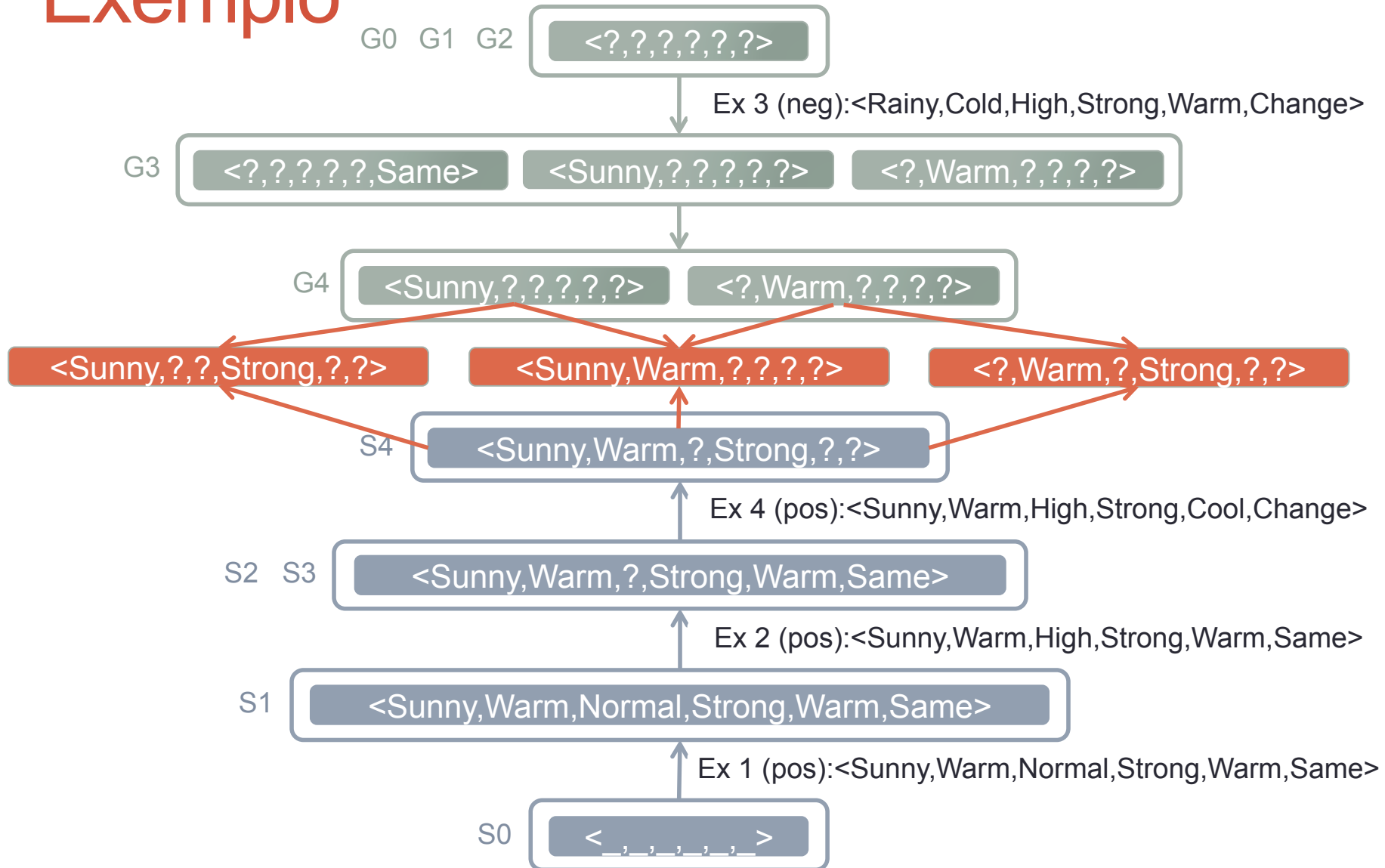
Algoritmo de Eliminação de Candidatos

- S contém a(s) hipóteses mais específicas de H
- G contém a(s) hipóteses mais gerais de H
- Para cada exemplo de treino d , fazer
 - Se d é um exemplo positivo
 - Remover de G qualquer hipótese inconsistente com d
 - Para cada hipótese s em S que não é consistente com d
 - Retirar s de S
 - Adicionar a S todas as generalizações minimais h de s tal que h é consistente com d , e algum membro de G é mais geral do que h
 - Retirar de S qualquer hipótese que seja mais geral do que outra hipótese em S
 - Se d é um exemplo negativo
 - Remover de S qualquer hipótese inconsistente com d
 - Para cada hipótese g em G que não é consistente com d
 - Retirar g de G
 - Adicionar a G todas as especializações minimais h de g tal que h é consistente com d e algum membro de S é mais específico do que h
 - Retirar de G qualquer hipótese que seja menos geral do que outra hipótese em G

Exemplo



Exemplo



Algoritmo de Eliminação de Candidatos

Árvore de especialização (G)

A hipótese mais geral é satisfeita por todos os exemplos

Instâncias negativas especializam hipóteses gerais

Instâncias positivas eliminam hipóteses gerais

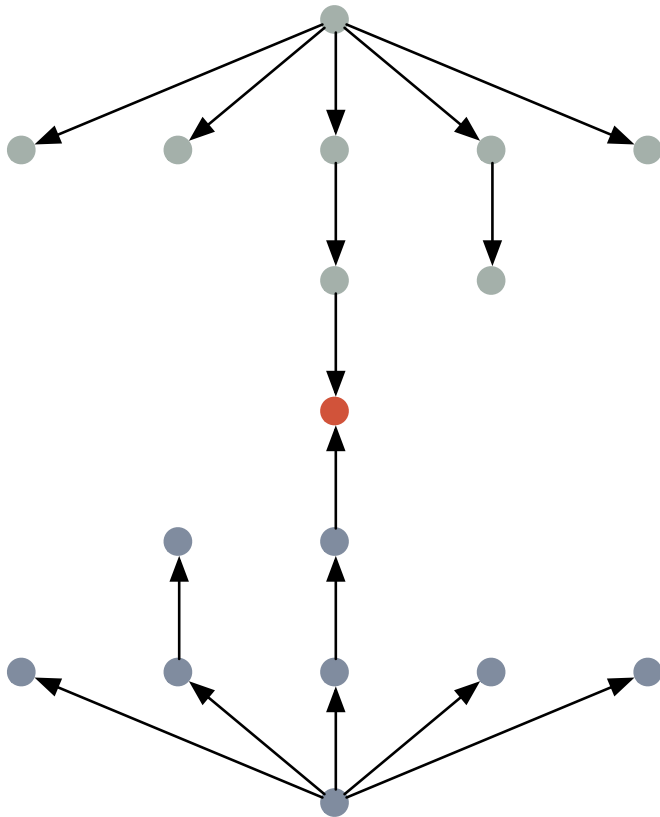
Eventualmente as instâncias positivas e negativas poderão forçar as hipóteses gerais e específicas a convergir para uma solução única

Instâncias negativas eliminam hipóteses específicas

Instâncias positivas generalizam hipóteses específicas

A hipótese mais específica não é satisfeita por qualquer exemplo

Árvore de generalização (S)



Propriedades das fronteiras

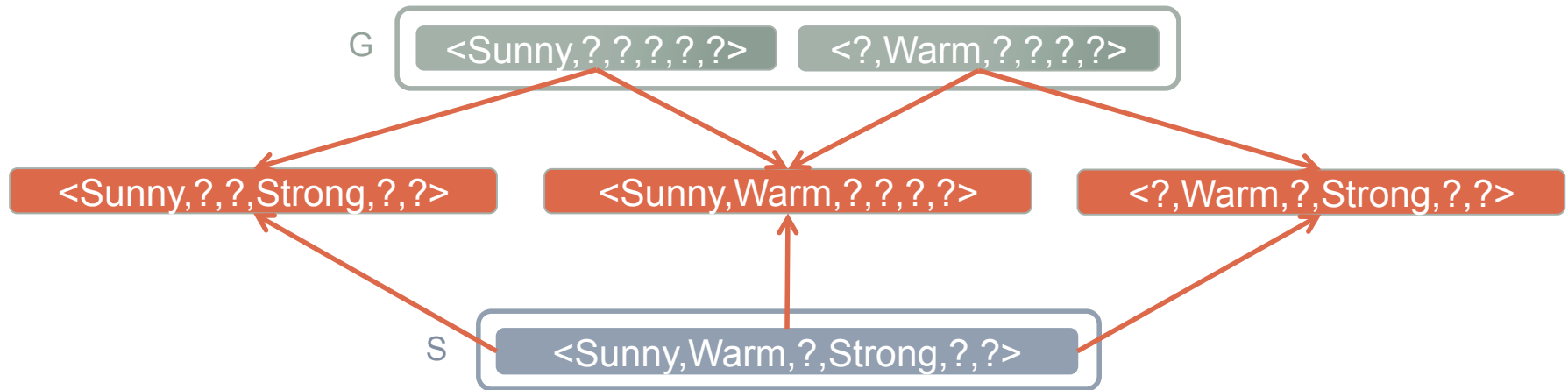
- S

- Pode ser vista como um sumário dos exemplos positivos
- Qualquer hipótese mais geral do que S cobre todos os exemplos positivos
- Outras hipóteses não conseguem cobrir pelo menos um exemplo positivo

- G

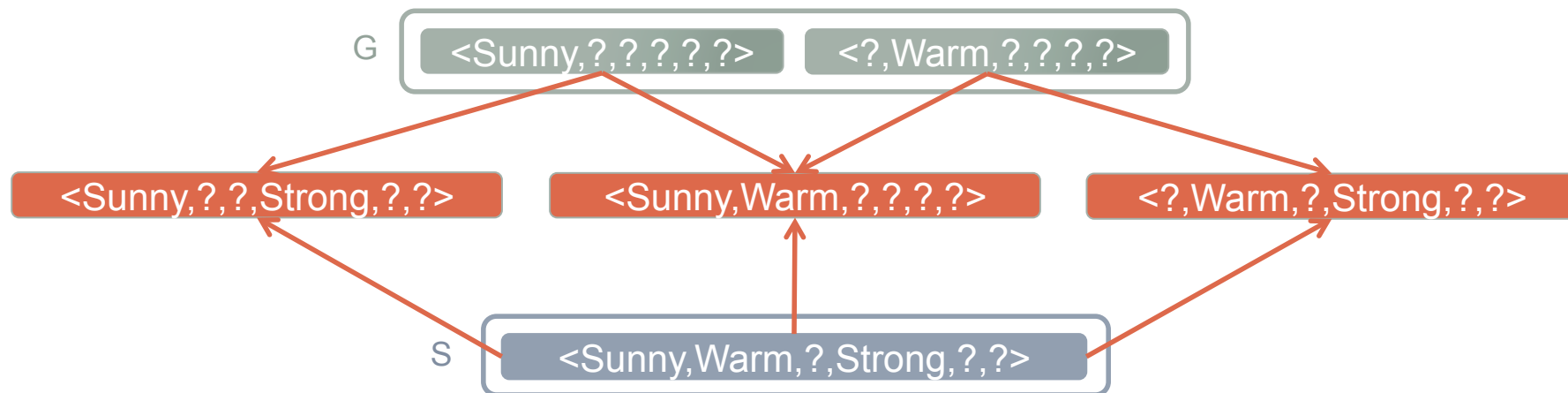
- Pode ser vista como um sumário dos exemplos negativos
- Qualquer hipótese mais específica do que G não cobre nenhum exemplo negativo
- Outras hipóteses cobrem pelo menos um exemplo negativo

Que exemplo tratar a seguir?



- Se o algoritmo pudesse escolher o próximo exemplo usado, qual seria o melhor?
- Idealmente, escolher um exemplo que fosse classificado positivamente por metade das hipóteses em VS. Em qualquer caso (exemplo positivo ou negativo), iria eliminar metade das hipóteses.
 - Por exemplo: `<Sunny, Warm, Normal, Light, Warm, Same>`

Classificação de novas instâncias



- Classificar um exemplo (pos ou neg) se todas as hipóteses do VS concordarem na sua classificação. Caso contrário:
 - Rejeitar ou
 - Maioria (vamos adoptar esta estratégia)

<Sunny, Warm, Normal, Strong, Cool, Change>

<Rainy, Cold, Normal, Weak, Warm, Same>

<Sunny, Warm, Normal, Weak, Warm, Same>

<Sunny, Cold, Normal, Strong, Warm, Same>

Positivo (classificado + por todas as hipóteses)

Negativo (classificado - por todas as hipóteses)

Indefinido (classificado + por 1/2 das hipóteses)

Negativo? (classificado - por 2/3 das hipóteses)

Propriedades do algoritmo

- 😊 O algoritmo de eliminação de candidatos é incremental
- 😊 Efectua o menor compromisso (tal como no POP)
- 😊 O algoritmo converge para o conceito alvo pretendido se forem dados exemplos de treino suficientes (pelo menos $\log_2 |VS|$)
- 😞 Se existir ruído ou o domínio não contiver atributos suficientes para a classificação exacta, o espaço de versões colapsa (um dos conjuntos S ou G fica vazio)
- 😞 O algoritmo com o espaço de hipóteses apresentado não permite aprender conceitos disjuntivos.
- 😞 Caso seja permitida disjunção ilimitada no espaço de hipóteses, então o algoritmo só conseguirá classificar os exemplos dados (não generaliza).
- 😞 Para alguns espaços de hipóteses o número de elementos de S e de G podem crescer exponencialmente.

Sumário

- Aprendizagem essencial para lidar com ambientes desconhecidos
- Agente aprendiz = elemento de desempenho + elemento de aprendizagem.
- No caso da aprendizagem indutiva, o objectivo consiste em encontrar uma hipótese simples que é aproximadamente consistente com os exemplos de treino.
- Aprendizagem de conceitos é um caso particular de aprendizagem indutiva onde se pretende aprender uma função booleana a partir de exemplos dados.
- O algoritmo de eliminação de candidatos mantém as fronteiras de hipóteses maximamente específicas e maximamente gerais.