

Aprendizado de Máquina

Aula 2

<http://www.ic.uff.br/~bianca/aa/>

Aula 2 - 23/03/10

1

Classificação

- Dados:
 - A descrição de uma instância, $x \in X$, onde X é um **espaço de instâncias**.
 - Um conjunto fixo de classes: $C = \{c_1, c_2, \dots, c_n\}$
- Determine:
 - A classe de x : $c(x) \in C$, onde $c(x)$ é uma função de classificação cujo domínio é X e cuja imagem é C .
 - Se $c(x)$ é uma função binária $C = \{0, 1\}$ ($\{\text{verdadeiro, falso}\}$, $\{\text{positivo, negativo}\}$) então é chamada de **conceito**.

Aula 2 - 23/03/10

2

Aprendizado para Classificação

- Um exemplo de treinamento é uma instância $x \in X$, com sua classe correta $c(x)$: $\langle x, c(x) \rangle$ para uma função de classificação desconhecida, c .
- Dado um conjunto de exemplos de treinamento, D .
- Encontre uma função de classificação hipotética, $h(x)$, tal que:

$$\forall \langle x, c(x) \rangle \in D : h(x) = c(x)$$

Consistência

Aula 2 - 23/03/10

3

Exemplo

- Espaço de instâncias: $\langle \text{tamanho, cor, forma} \rangle$
 - tamanho $\in \{\text{pequeno, médio, grande}\}$
 - cor $\in \{\text{vermelho, azul, verde}\}$
 - forma $\in \{\text{quadrado, círculo, triângulo}\}$
- $C = \{\text{positivo, negativo}\}$
- D :

Exemplo	Tamanho	Cor	Forma	Classe
1	pequeno	vermelho	círculo	positivo
2	grande	vermelho	círculo	positivo
3	pequeno	vermelho	triângulo	negativo
4	grande	azul	círculo	negativo

Aula 2 - 23/03/10

4

Seleção de Hipóteses

- Muitas hipóteses são normalmente consistentes com os dados de treinamento.
 - vermelho & círculo
 - (pequeno & círculo) ou (grande & vermelho)
 - (pequeno & vermelho & círculo) ou (grande & vermelho & círculo)
 - não [(vermelho & triângulo) ou (azul & círculo)]
 - não [(pequeno & vermelho & triângulo) ou (grande & azul & círculo)]
- Viés ("bias")
 - Qualquer critério além de consistência com os dados de treinamento que é usado para selecionar uma hipótese.

Aula 2 - 23/03/10

5

Generalização

- Hipóteses têm que generalizar pra classificar instâncias que não estejam nos dados de treinamento.
- Simplesmente memorizar os exemplos de treinamento é uma hipótese consistente que não generaliza.
- Lâmina de Occam**:
 - Encontrar uma hipótese *simples* ajuda na generalização.

Aula 2 - 23/03/10

6

Espaço de Hipóteses

- Restringir as funções a serem aprendidas a um **espaço de hipóteses**, H , de funções $h(x)$ que serão consideradas para definir $c(x)$.
- Para conceitos sobre instâncias descritas por n características discretas, considere o espaço de hipóteses conjuntivas representadas por um vetor de n restrições $\langle c_1, c_2, \dots, c_n \rangle$ onde cada c_i pode ser:
 - $?$, indica que não há restrição para a i -ésima característica
 - Um valor específico do domínio da i -ésima característica
 - $\emptyset$, indicando que nenhum valor é aceitável
- Exemplos de hipóteses conjuntivas:
 - $\langle \text{grande}, \text{vermelho}, ? \rangle$
 - $\langle ?, ?, ? \rangle$ (hipótese mais geral)
 - $\langle \emptyset, \emptyset, \emptyset \rangle$ (hipótese mais específica)

Aula 2 - 23/03/10

7

Suposições do Aprendizado Indutivo

- Qualquer função que aproxima o conceito alvo bem num conjunto suficientemente grande de exemplos de treinamento também irá aproximar o conceito bem para exemplos que não foram observados.
- Supõe que os exemplos de treinamento e teste foram obtidos independentemente a partir da mesma distribuição.
- Essa é uma hipótese que não pode ser provada a não ser que suposições adicionais sejam feitas sobre o conceito alvo e a noção de "aproximar o conceito bem para exemplos não-observados" seja definida de forma apropriada.

Aula 2 - 23/03/10

8

Avaliação do Aprendizado de Classificação

- Acurácia da classificação (% de instâncias classificadas corretamente).
 - Medida em dados de teste.
- Tempo de treinamento (eficiência do algoritmo de aprendizagem).
- Tempo de teste (eficiência da classificação).

Aula 2 - 23/03/10

9

Aprendizado de classificação visto como busca

- O aprendizado pode ser visto como uma busca no espaço de hipóteses por uma (ou mais) hipóteses que sejam consistentes com os dados de treinamento.
- Considere um espaço de instâncias consistindo de n características binárias que tem portanto 2^n instâncias.
- Para hipóteses conjuntivas, há 4 escolhas para cada característica: $\emptyset$, T, F, $?$, então há 4^n hipóteses sintaticamente distintas.
- Porém, todas as hipóteses com 1 ou mais $\emptyset$ s são equivalentes, então há $3^n + 1$ hipóteses semanticamente distintas.
- A função de classificação binária alvo em princípio poderia qualquer uma de 2^{2^n} funções de n bits de entrada.
- Logo, hipóteses conjuntivas são um subconjunto pequeno do espaço de possíveis funções, mas ambos são muito grandes.
- Todos os espaços de hipótese razoáveis são muito grandes ou até infinitos.

Aula 2 - 23/03/10

10

Aprendizado por Enumeração

- Para qualquer espaço de hipóteses finito ou infinito contável, podemos simplesmente enumerar e testar hipóteses uma de cada vez até que uma consistente seja encontrada.

Para cada h em H faça:

Se h é consistente com os dados de treinamento D ,
então termine e retorne h .
- Este algoritmo tem garantia de encontrar uma hipótese consistente se ela existir; porém é obviamente computacionalmente intratável para a maioria dos problemas práticos.

Aula 2 - 23/03/10

11

Aprendizado Eficiente

- Existe uma maneira de aprender conceitos conjuntivos sem enumerá-los?
- Como humanos aprendem conceitos conjuntivos?
- Existe uma maneira eficiente de encontrar uma função booleana não-restrita consistente com um conjunto de exemplos com atributos discretos?
- Se existir é um algoritmo útil na prática?

Aula 2 - 23/03/10

12

Aprendizado de Regras Conjuntivas

- Descrições conjuntivas podem ser aprendidas olhando as características comuns dos exemplos positivos.

Exemplo	Tamanho	Cor	Forma	Classe
1	pequeno	vermelho	círculo	positivo
2	grande	vermelho	círculo	positivo
3	pequeno	vermelho	triângulo	negativo
4	grande	azul	círculo	negativo

Regra aprendida: **vermelho & círculo** → positivo

- É necessário checar consistência com exemplos negativos. Se for inconsistente, não existe regra conjuntiva.

Aula 2 - 23/03/10

13

Limitações de Regras Conjuntivas

- Se o conceito não tiver um único conjunto de condições necessárias e suficientes, o aprendizado falha.

Exemplo	Tamanho	Cor	Forma	Classe
1	pequeno	vermelho	círculo	positivo
2	grande	vermelho	círculo	positivo
3	pequeno	vermelho	triângulo	negativo
4	grande	azul	círculo	negativo
5	médio	vermelho	círculo	negativo

Regra aprendida: **vermelho & círculo** → positivo

Inconsistente com exemplo negativo #5!

Aula 2 - 23/03/10

14

Conceitos disjuntivos

- Conceitos podem ser disjuntivos.

Exemplo	Tamanho	Cor	Forma	Classe
1	pequeno	vermelho	círculo	positivo
2	grande	vermelho	círculo	positivo
3	pequeno	vermelho	triângulo	negativo
4	grande	azul	círculo	negativo
5	médio	vermelho	círculo	negativo

Regras aprendidas: **pequeno & círculo** → positivo
grande & vermelho → positivo

Aula 2 - 23/03/10

15

Usando a estrutura de generalidade

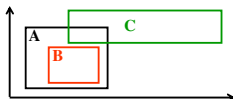
- Explorando a estrutura imposta pela generalidade das hipóteses, pode-se fazer a busca no espaço de hipóteses sem enumerar cada hipótese.
- Uma instância, $x \in X$, **satisfaz** uma hipótese, h , se e somente se $h(x)=1$ (positivo)
- Dadas duas hipóteses h_1 e h_2 , h_1 é **mais geral ou igual a** h_2 ($h_1 \geq h_2$) se e somente se cada instância que satisfaz h_2 também satisfaz h_1 .
- Dadas duas hipóteses h_1 e h_2 , h_1 é **(estritamente) mais geral que** h_2 ($h_1 > h_2$) se e somente se $h_1 \geq h_2$ e não $h_2 \geq h_1$.
- Generalidade define uma ordem parcial das hipóteses.

Aula 2 - 23/03/10

16

Exemplos de Generalidade

- Vetores de característica conjuntivos
 - $\langle ?, \text{vermelho}, ? \rangle$ é mais geral que $\langle ?, \text{vermelho}, \text{círculo} \rangle$
 - $\langle ?, \text{red}, ? \rangle$ e $\langle ?, ?, \text{circle} \rangle$ não são mais gerais um do que o outro.
- Retângulo com eixos paralelos no espaço 2-d:



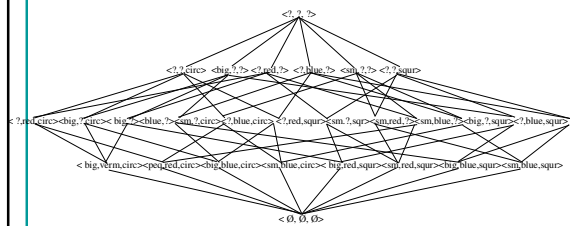
- A é mais geral que B
- Nem A nem C são mais gerais um do que o outro

Aula 2 - 23/03/10

17

Exemplo de látice de generalização

Size: {sm, big} Color: {red, blue} Shape: {circ, squ}



Número de hipóteses = $3^3 + 1 = 28$

Aula 2 - 23/03/10

18

Aprendizado Mais Específico (Encontre-S)

- Encontra a hipótese mais específica que seja consistente com os dados de treinamento (chamada de LGG = *Least General Generalization*).
- Atualiza a hipótese depois de examinar cada exemplo positivo, generalizando-a somente o suficiente para satisfazer o exemplo.
- Para vetores de características conjuntivas:
Inicialize $h = \langle \emptyset, \emptyset, \dots, \emptyset \rangle$
Para cada instância de treinamento positiva x em D
Para cada característica f_i
Se a restrição de f_i em h não for satisfeita por x
Se f_i em h é $\emptyset$
então faça f_i em h ter o valor de f_i em x
senão faça f_i em h ser "?"
Se h for consistente com as instâncias de treinamento em D
então retorne h
senão não existe hipótese consistente

Complexidade:
 $O(|D|n)$
onde n é o número de características

Aula 2 - 23/03/10

19

Propriedades de Encontre-S

- Para vetores de características conjuntivas, a hipótese mais específica é única e será encontrada por Encontre-S.
- Se a hipótese mais específica não for consistente com os exemplos negativos, então não há função consistente no espaço de hipóteses, já que, por definição, ela não pode ser mais específica e manter consistência com os exemplos positivos.
- Para vetores de características conjuntivas, se a hipótese mais específica for inconsistente, então o conceito alvo deve ser disjuntivo.

Aula 2 - 23/03/10

20

Outro espaço de hipóteses

- Considere o caso de dois objetos (sem ordem) descritos por um conjunto fixo de atributos.
 - $\{ \langle \text{grande, vermelho, círculo} \rangle, \langle \text{pequeno, azul, quadrado} \rangle \}$
- Qual é a generalização mais específica de:
 - Positivo: $\{ \langle \text{grande, vermelho, triângulo} \rangle, \langle \text{pequeno, azul, círculo} \rangle \}$
 - Positivo: $\{ \langle \text{grande, azul, círculo} \rangle, \langle \text{pequeno, vermelho, triângulo} \rangle \}$
- LGG não é única, duas generalizações não comparáveis são:
 - $\{ \langle \text{grande, ?, ?} \rangle, \langle \text{pequeno, ?, ?} \rangle \}$
 - $\{ \langle \text{?, vermelho, triângulo} \rangle, \langle \text{?, azul, círculo} \rangle \}$
- Para esse espaço, Encontre-S teria que manter um conjunto continuamente crescente de LGGs e eliminar aquelas que cobrem exemplos negativos.
- Encontre-S não é tratável para esse espaço já que o número de LGGs pode crescer exponencialmente.

Aula 2 - 23/03/10

21

Problemas da Encontre-S

- Dados exemplos de treinamento suficientes, Encontre-S converge para uma definição correta do conceito alvo (supondo que ele está no espaço de hipóteses)?
- Como sabemos que houve convergência para a hipótese correta?
- Por que preferir a hipótese mais específica? Existem hipóteses mais gerais que sejam consistentes? E a hipótese mais geral? E a hipótese de treinamento?
- Se as LGGs não são únicas
 - Qual LGG deve ser escolhida?
 - Como uma única LGG consistente pode ser eficientemente calculada?
- E se houver ruído nos dados de treinamento e alguns exemplos estiverem com a classe errada?

Aula 2 - 23/03/10

22

Efeito do Ruído

- Frequentemente dados de treinamento reais têm erros (ruído) nas características ou classes.
- O ruído pode fazer com que generalizações válidas não sejam encontradas.
 - Por exemplo, imagine que há muitos exemplos positivos como #1 e #2, mas somente um exemplo negativo como o #5 que na verdade resultou de um erro de classificação.

Exemplo	Tamanho	Cor	Forma	Classe
1	pequeno	vermelho	círculo	positivo
2	grande	vermelho	círculo	positivo
3	pequeno	vermelho	triângulo	negativo
4	grande	azul	círculo	negativo
5	médio	vermelho	círculo	negativo

Aula 2 - 23/03/10

23

Espaço de Versões

- Dado um espaço de hipóteses, H , e dados de treinamento, D , o **espaço de versões** é o subconjunto completo de H que é consistente com D .
- O espaço de versões pode ser gerado de forma ingênua para qualquer H finito através da enumeração de todas as hipóteses e eliminando as inconsistentes.
- O espaço de versões pode ser calculado de forma mais eficiente do que com enumerações?

Aula 2 - 23/03/10

24

Espaço de versões com S e G

- O espaço de versões pode ser representado de forma mais compacta mantendo-se dois conjuntos de hipóteses: S , o conjunto de hipóteses mais específicas consistentes, e G , o conjunto de hipóteses mais gerais consistentes:

$$S = \{s \in H \mid \text{Consistent}(s, D) \wedge \neg \exists s' \in H[s' > s \wedge \text{Consistent}(s', D)]\}$$

$$G = \{g \in H \mid \text{Consistent}(g, D) \wedge \neg \exists g' \in H[g' > g \wedge \text{Consistent}(g', D)]\}$$

- S e G representam o espaço de versão inteiro através de seus limites na lattice de generalização:



Aula 2 - 23/03/10

25

Látice do Espaço de Versões

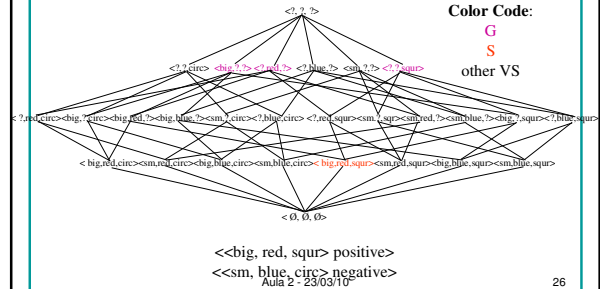
Size: {sm, big} Color: {red, blue} Shape: {circ, squar}

Color Code:

G

S

other VS



<<big, red, squar> positive>

<<sm, blue, circ> negative>

Aula 2 - 23/03/10

26

Algoritmo de Eliminação de Candidatos (Espaço de Versões)

Inicialize G como o conjunto de hipóteses mais gerais em H
Inicialize S como o conjunto de hipóteses mais específicas em H
Para cada exemplo de treinamento, d , faça:

Se d for um exemplo positivo então:

Remova de G qualquer hipótese que não esteja de acordo com d

Para cada hipótese s em S que não esteja de acordo com d

Remova s de S

Adicione a S todas as generalizações mínimas, h , de s tais que:

1) h está de acordo com d

2) alguns membros de G são mais gerais que h

Remover de S qualquer h que seja mais geral do que qualquer outra hipótese S

Se d for um exemplo negativo então:

Remova de S qualquer hipótese de acordo com d

Para cada hipótese g em G que está de acordo com d

Remova g de G

Adicione a G todas as especializações mínimas, h , de g tais que:

1) h não está de acordo com d

2) algum membro de S é mais específico que h

Remova de G qualquer h que seja mais específico que qualquer outra hipótese em G

Aula 2 - 23/03/10

27

Sub-rotinas necessárias

- Colocar o algoritmo em uma linguagem específica requer implementar os seguintes procedimentos:

- Hipóteses-igual(h_1, h_2)
- Mais geral(h_1, h_2)
- match(h, i)
- inicialize- $g()$
- inicialize- $s()$
- generalizar(h, i)
- especializar(h, i)

Aula 2 - 23/03/10

28

Especialização e Generalização Mínimas

- Os procedimentos de generalização e especialização são específicos para uma linguagem de hipóteses e podem ser complexos.
- Para vetores de características conjuntivos:
 - generalização: única, veja Encontre- S
 - especialização: não é única, podemos converter cada "?" pra valores diferentes..
 - Inputs:
 - $h = \langle ?, \text{red}, ? \rangle$
 - $i = \langle \text{small}, \text{red}, \text{triangle} \rangle$
 - Outputs:
 - $\langle \text{big}, \text{red}, ? \rangle$
 - $\langle \text{medium}, \text{red}, ? \rangle$
 - $\langle ?, \text{red}, \text{square} \rangle$
 - $\langle ?, \text{red}, \text{circle} \rangle$

Aula 2 - 23/03/10

29

Exemplo VS Trace

$S = \{ \langle \emptyset, \emptyset, \emptyset \rangle \}; G = \{ \langle ?, ?, ? \rangle \}$

Positivo: $\langle \text{big}, \text{red}, \text{circle} \rangle$

Nada a remover de G

Generalização mínima de cada elemento de S é $\langle \text{big}, \text{red}, \text{circle} \rangle$ que é mais específico do que G .

$S = \{ \langle \text{big}, \text{red}, \text{circle} \rangle \}; G = \{ \langle ?, ?, ? \rangle \}$

Negativo: $\langle \text{small}, \text{red}, \text{triangle} \rangle$

Nada a remover de S .

Especializações mínimas de $\langle ?, ?, ? \rangle$ são $\langle \text{medium}, ?, ? \rangle, \langle \text{big}, ?, ? \rangle,$

$\langle ?, \text{blue}, ? \rangle, \langle ?, \text{green}, ? \rangle, \langle ?, ?, \text{circle} \rangle, \langle ?, ?, \text{square} \rangle$ mas a maioria é mais geral do que algum elemento de S

$S = \{ \langle \text{big}, \text{red}, \text{circle} \rangle \}; G = \{ \langle \text{big}, ?, ? \rangle, \langle ?, ?, \text{circle} \rangle \}$

Aula 2 - 23/03/10

30

Exemplo VS Trace (cont)

$S = \{ \langle \text{big, red, circle} \rangle \}; G = \{ \langle \text{big, ?, ?}, \langle \text{?, ?, circle} \rangle \}$

Positivo: $\langle \text{small, red, circle} \rangle$

Remova $\langle \text{big, ?, ?} \rangle$ de G

Generalização mínima de $\langle \text{big, red, circle} \rangle$ é $\langle \text{?, red, circle} \rangle$

$S = \{ \langle \text{?, red, circle} \rangle \}; G = \{ \langle \text{?, ?, circle} \rangle \}$

Negativo: $\langle \text{big, blue, circle} \rangle$

Nada a remover de S

Especializações mínimas de $\langle \text{?, ?, circle} \rangle$ são $\langle \text{small, ? circle} \rangle$, $\langle \text{medium, ? circle} \rangle$, $\langle \text{?, red, circle} \rangle$, $\langle \text{?, green, circle} \rangle$ mas a maioria não é mais genérica que alguns elementos de S .

$S = \{ \langle \text{?, red, circle} \rangle \}; G = \{ \langle \text{?, red, circle} \rangle \}$

$S = G$; Convergência

Aula 2 - 23/03/10

31

Propriedades do Algoritmo VS

- S resume a informação relevante dos exemplos positivos (relativo a H) de tal forma que eles não precisem ser mantidos.
- G resume a informação relevante dos exemplos negativos, de tal forma que eles não precisem ser mantidos.
- O resultado não é afetado pela ordem dos exemplos, mas a eficiência computacional pode ser afetada pela ordem.
- Se S e G convergirem para a mesma hipótese, então ela é a única que é consistente com os dados em H .
- Se S e G ficam vazios então não há nenhuma hipótese em H que seja consistente com os dados.

Aula 2 - 23/03/10

32

Exemplo: Weka VS Trace 1

```
java weka.classifiers.vspace.ConjunctiveVersionSpace -t figure.arff -T figure.arff -v -P
```

Initializing VersionSpace

$S = \{ \{ \# \# \# \} \}$

$G = \{ \{ \text{?, ?, ?} \} \}$

Instance: big,red,circle,positive

$S = \{ \{ \text{big,red,circle} \} \}$

$G = \{ \{ \text{?, ?, ?} \} \}$

Instance: small,red,square,negative

$S = \{ \{ \text{big,red,circle} \} \}$

$G = \{ \{ \text{big, ?, ?}, \{ \text{?, ?, circle} \} \}$

Instance: small,red,circle,positive

$S = \{ \{ \text{?, red, circle} \} \}$

$G = \{ \{ \text{?, ?, circle} \} \}$

Instance: big,blue,circle,negative

$S = \{ \{ \text{?, red, circle} \} \}$

$G = \{ \{ \text{?, red, circle} \} \}$

Version Space converged to a single hypothesis.

Aula 2 - 23/03/10

33

Exemplo: Weka VS Trace 2

```
java weka.classifiers.vspace.ConjunctiveVersionSpace -t figure2.arff -T figure2.arff -v -P
```

Initializing VersionSpace

$S = \{ \{ \# \# \# \} \}$

$G = \{ \{ \text{?, ?, ?} \} \}$

Instance: big,red,circle,positive

$S = \{ \{ \text{big,red,circle} \} \}$

$G = \{ \{ \text{?, ?, ?} \} \}$

Instance: small,blue,triangle,negative

$S = \{ \{ \text{big,red,circle} \} \}$

$G = \{ \{ \text{big, ?, ?}, \{ \text{?, red, ?}, \{ \text{?, ?, circle} \} \}$

Instance: small,red,circle,positive

$S = \{ \{ \text{?, red, circle} \} \}$

$G = \{ \{ \text{?, red, ?}, \{ \text{?, ?, circle} \} \}$

Instance: medium,green,square,negative

$S = \{ \{ \text{?, red, circle} \} \}$

$G = \{ \{ \text{?, red, ?}, \{ \text{?, ?, circle} \} \}$

Aula 2 - 23/03/10

34

Exemplo: Weka VS Trace 3

```
java weka.classifiers.vspace.ConjunctiveVersionSpace -t figure3.arff -T figure3.arff -v -P
```

Initializing VersionSpace

$S = \{ \{ \# \# \# \} \}$

$G = \{ \{ \text{?, ?, ?} \} \}$

Instance: big,red,circle,positive

$S = \{ \{ \text{big,red,circle} \} \}$

$G = \{ \{ \text{?, ?, ?} \} \}$

Instance: small,red,triangle,negative

$S = \{ \{ \text{big,red,circle} \} \}$

$G = \{ \{ \text{big, ?, ?}, \{ \text{?, ?, circle} \} \}$

Instance: small,red,circle,positive

$S = \{ \{ \text{?, red, circle} \} \}$

$G = \{ \{ \text{?, ?, circle} \} \}$

Instance: big,blue,circle,negative

$S = \{ \{ \text{?, red, circle} \} \}$

$G = \{ \{ \text{?, red, circle} \} \}$

Version Space converged to a single hypothesis.

Language is insufficient to describe this data.

Aula 2 - 23/03/10

35

Exemplo: Weka VS Trace 4

```
java weka.classifiers.vspace.ConjunctiveVersionSpace -t figure4.arff -T figure4.arff -v -P
```

Initializing VersionSpace

$S = \{ \{ \# \# \# \} \}$

$G = \{ \{ \text{?, ?, ?} \} \}$

Instance: small,red,square,negative

$S = \{ \{ \# \# \# \} \}$

$G = \{ \{ \text{?, ?, ?} \} \}$

Instance: big,blue,circle,negative

$S = \{ \{ \# \# \# \} \}$

$G = \{ \{ \text{?, ?, ?} \} \}$

Instance: medium,?,?, [big,?,?], [?,blue,?], [?,green,?], [?,?,circle], [?,?,triangle]

$S = \{ \{ \text{medium,?,?}, \{ \text{?, green,?}, \{ \text{?, ?, triangle}, \{ \text{big,red,?}, \{ \text{big,?,square}, \{ \text{small,blue,?}, \{ \text{?, blue,square}, \{ \text{small,?,circle}, \{ \text{?, red,circle} \} \}$

$G = \{ \{ \text{big,red,?}, \{ \text{?, red,circle} \} \}$

Instance: big,red,circle,positive

$S = \{ \{ \text{big,red,circle} \} \}$

$G = \{ \{ \text{big,red,?}, \{ \text{?, red,circle} \} \}$

Instance: small,red,circle,positive

$S = \{ \{ \text{?, red,circle} \} \}$

$G = \{ \{ \text{?, red,circle} \} \}$

Version Space converged to a single hypothesis.

Aula 2 - 23/03/10

36

Corretude do Aprendizado

- Como o espaço de versões inteiro é mantido, dado um fluxo contínuo de exemplos de treinamento sem ruído, o algoritmo VS algorithm vai convergir para o conceito correto se ele estiver no espaço de hipóteses, H , ou determinará que ele não está em H .
- A convergência é sinalizada quando $S=G$.

Aula 2 - 23/03/10

37

Complexidade Computacional do VS

- Calcular o conjunto S de vetores de características conjuntivos é linear em relação ao número de características e exemplos de treinamento.
- Calcular o conjunto G de vetores de características conjuntivos é exponencial em relação ao número de exemplos de treinamento no pior caso.
- Com linguagens mais expressivas, tanto S quanto G podem crescer exponencialmente.
- A ordem em que os exemplos são processados pode afetar significativamente a complexidade computacional.

Aula 2 - 23/03/10

38