

Aprendizado de Máquina

Aula 8

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

Tópicos

1. Introdução – Cap. 1 (16/03)
2. Classificação Indutiva – Cap. 2 (23/03)
3. Árvores de Decisão – Cap. 3 (30/03)
4. Ensembles - Artigo (13/04)
5. Avaliação Experimental – Cap. 5 (20/04)
6. Aprendizado de Regras – Cap. 10 (27/04)
7. Redes Neurais – Cap. 4 (04/05)
8. **Teoria do Aprendizado – Cap. 7 (11/05)**
9. Máquinas de Vetor de Suporte – Artigo (18/05)
10. Aprendizado Bayesiano – Cap. 6 e novo cap. online (25/05)
11. Aprendizado Baseado em Instâncias – Cap. 8 (01/05)
12. Classificação de Textos – Artigo (08/06)
13. Aprendizado Não-Supervisionado – Artigo (15/06)

Aula 8 - 11/05/2010

2

Teoria do Aprendizado

- Teoremas que caracterizam classes de problemas de aprendizado ou algoritmos específicos em termos da complexidade computacional ou da **complexidade de amostras**, i.e. o número de exemplos necessários para aprender hipóteses com uma dada acurácia.
- Complexidade do problema de aprendizado depende de:
 - Tamanho ou expressividade do espaço de hipóteses.
 - Precisão com a qual o conceito deve ser aproximado.
 - Probabilidade com a qual o aprendiz deve produzir o conceito desejado.
 - Maneira pela qual os exemplos são obtidos (aleatoriamente ou através de consultas do aprendiz).

3

Tipos de Resultados

- **Aprendendo no limite**: há garantia de que o aprendiz irá convergir para a hipótese correta quando o número de exemplos aumenta infinitamente?
- **Complexidade de amostras**: Quantos exemplos são necessários para construir (com alta probabilidade) um conceito altamente preciso?
- **Complexidade computacional**: Que recursos computacionais (tempo e espaço) são necessários para um aprendiz construir (com alta probabilidade) um conceito altamente preciso?
- **Quota de erros**: Aprendendo incrementalmente, quantos exemplos o aprendiz irá errar até chegar em um conceito altamente preciso.

4

Aprendendo no limite

- Dado um fluxo contínuo de exemplos, o aprendiz em algum momento converge a um conceito e não erra mais.
- Não há limite para o número de exemplos ou recursos computacionais, mas em algum momento deve aprender o conceito de forma exata.
- Por enumeração simples, conceitos de qualquer espaço de hipótese finito podem ser aprendidos no limite, apesar de tipicamente precisarem de um número exponencial de exemplos e tempo.
- Classe de funções totalmente recursivas (funções computáveis de Turing) não pode ser aprendido no limite.

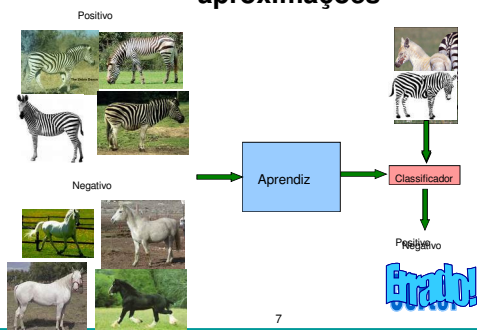
5

Aprendendo no limite vs. Modelo PAC

- Aprendizado no limite é muito forte
 - Exige que conceito exato seja aprendido
- Aprendizado no limite é muito fraco
 - Permite dados e recursos computacionais ilimitados
- Modelo PAC
 - Só é necessário aprender um conceito **Provavelmente Aproximadamente Correto**: Aprender uma aproximação razoável a maior parte do tempo.
 - Exige complexidade de amostra e computacional polinomial.

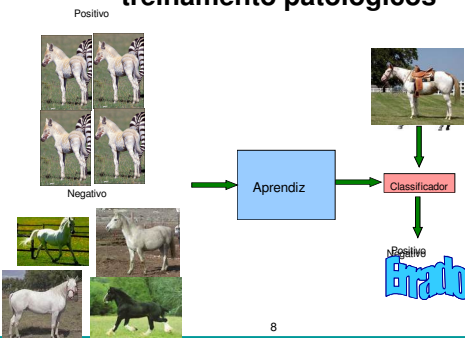
6

Não podemos aprender conceitos exatos com dados limitados; só aproximações



7

Não podemos aprender nem conceitos aproximados para conjuntos de treinamento patológicos



8

Aprendizado PAC

- A única expectativa razoável para um aprendiz é que com **alta probabilidade** ele aprenda **uma boa aproximação** do conceito alvo.
- No modelo PAC, especifica-se dois parâmetros, ϵ e δ , e exige-se que com probabilidade pelo menos $(1 - \delta)$ o sistema aprenda o conceito com erro no máximo ϵ .

9

Definição formal de aprendizado PAC

- Considere uma classe de conceitos C definida em um espaço de instâncias X contendo instâncias de comprimento n , e um aprendiz, L , usando espaço de hipóteses, H .
- C é **PAC-aprendível** por L usando H se e somente se para todo $c \in C$, distribuições D sobre X , $0 < \epsilon < 0.5$, $0 < \delta < 0.5$; o aprendiz L com amostras aleatórias de D , com probabilidade pelo menos $1 - \delta$ terá como saída uma hipótese $h \in H$ tal que $\text{error}_D(h) \leq \epsilon$, em tempo polinomial em $1/\epsilon$, $1/\delta$, n e tamanho(c).

10

Exemplo

- X : instâncias descritas por n características numéricas
- C : descrições conjuntivas usando essas características
- H : descrições conjuntivas usando essas características
- L : algoritmo de generalização mais específica (Find-S)
- tamanho(c): o número de literais em c (i.e. tamanho da conjunção).

Aula 8 - 11/05/2010

11

Resultado de Complexidade de Amostras

- Qualquer aprendiz consistente, dados pelo menos

$$\left(\ln \frac{1}{\delta} + \ln |H| \right) / \epsilon$$

exemplos irá produzir um resultado que é PAC.

- Só precisamos saber o tamanho do espaço de hipóteses para instanciar esse resultado para casos específicos.
- Dá um número de exemplos **suficientes** para o aprendizado PAC, mas **não** um número **necessário** (pode ser muito maior que o necessário).

12

Complexidade de Amostras do Aprendizado de Conjunções

- Considere conjunções com n características booleanas. Há 3^n conjunções já que cada característica pode aparecer positivamente, negativamente ou não aparecer. Logo $|H| = 3^n$, então um número suficiente de exemplos pra aprender o conceito de forma PAC é:

$$\left(\ln \frac{1}{\delta} + \ln 3^n \right) / \epsilon = \left(\ln \frac{1}{\delta} + n \ln 3 \right) / \epsilon$$

- Exemplos concretos:
 - $\delta = \epsilon = 0.05$, $n = 10$ dá 280 exemplos
 - $\delta = 0.01$, $\epsilon = 0.05$, $n = 10$ dá 312 exemplos
 - $\delta = \epsilon = 0.01$, $n = 10$ dá 1.560 exemplos
 - $\delta = \epsilon = 0.01$, $n = 50$ dá 5.954 exemplos
- Resultados válidos pra qualquer aprendiz consistente, incluindo FindS.

13

Complexidade de Amostras de Funções Booleanas Arbitrárias

- Considere qualquer função booleana de n características, tal como o espaço de hipóteses de DNF ou árvores de decisão. Há 2^{2^n} dessas, então um número suficiente de exemplos pra aprender o conceito de forma PAC é:

$$\left(\ln \frac{1}{\delta} + \ln 2^{2^n} \right) / \epsilon = \left(\ln \frac{1}{\delta} + 2^n \ln 2 \right) / \epsilon$$

- Exemplos concretos:
 - $\delta = \epsilon = 0.05$, $n = 10$ dá 14,256 exemplos
 - $\delta = \epsilon = 0.05$, $n = 20$ dá 14,536,410 exemplos
 - $\delta = \epsilon = 0.05$, $n = 50$ dá 1.561×10^{16} exemplos

14

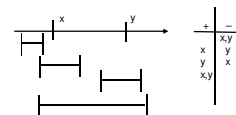
Espaço de Hipóteses Infinito

- A análise anterior estava restrito a espaços de hipóteses finitos.
- Alguns espaços de hipóteses infinitos (como os que usam valores reais como parâmetros) são mais expressivos do que outros.
 - Compare uma regra permitindo um limiar em uma variável contínua ($c < 3$) com outra que permita dois limiares ($1 < c < 3$).
- Precisamos de uma medida de expressividade.
- A **dimensão de Vapnik-Chervonenkis (VC)** é uma medida desse tipo, escrita como $VC(H)$.
- De forma análoga a $\ln |H|$, há quotas para complexidade amostral usando $VC(H)$.

15

“Despedaçando” instâncias

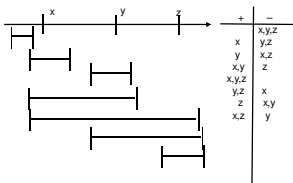
- Um espaço de hipóteses “despedaça” um conjunto de instâncias se e somente se pra cada partição das instâncias em positivos e negativos, há uma hipótese que produz essa partição.
- Por exemplo, considere 2 instâncias descritas usando um atributo com um único valor sendo “despedaçada” pelos intervalos.



16

“Despedaçando” instâncias

- Mas 3 instâncias não podem ser “despedaçadas” por um único intervalo.



- Como há 2^m partições de m instâncias, teríamos que ter: $|H| \geq 2^m$.

17

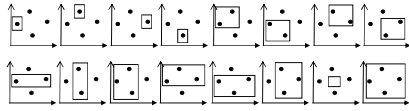
Dimensão VC

- Quanto maior o subconjunto de X que pode ser “despedaçado”, mais expressivo é o espaço de hipóteses.
- A dimensão VC de um espaço de hipóteses H definido sobre um espaço de instâncias X é o tamanho do maior subconjunto de X que pode ser “despedaçado” por H . (Note que podemos ter $VC(H) = \infty$)
- Se existe pelo menos um subconjunto de tamanho d então $VC(H) \geq d$. Se não existe nenhum, então $VC(H) < d$.
- Para intervalos únicos na reta, temos $VC(H) = 2$.
- Como $|H| \geq 2^m$, para “despedaçar” m instâncias, $VC(H) \leq \log_2 |H|$.

18

Exemplo

- Considere retângulos paralelos aos eixos no plano real, i.e. conjunções de intervalos de 2 características reais. Temos que 4 instâncias podem ser "despedaçadas"



Algumas configurações de 4 instâncias não podem ser despedaçadas:



19

Exemplo (cont)

- Nenhum conjunto de 5 instâncias pode ser despedaçado



- Logo $VC(H) = 4$
- Pode ser generalizado para hiper-retângulos (conjunções de intervalos em n dimensões): $VC(H)=2n$.

20

Quota superior para complexidade de amostras usando dimensão VC

- Usando dimensão VC como medida de complexidade, podemos mostrar que o seguinte número de exemplos é suficiente para o aprendizado PAC (Blumer *et al.*, 1989).

$$\frac{1}{\epsilon} \left(4 \log_2 \left(\frac{2}{\delta} \right) + 8VC(H) \log_2 \left(\frac{13}{\epsilon} \right) \right)$$

21

Conclusões

- A análise PAC é um arcabouço para o estudo teórico da efetividade de algoritmos de aprendizagem.
- A complexidade de amostra de qualquer aprendiz consistente usando um espaço de hipóteses, H , pode ser determinada a partir de $|H|$ or $VC(H)$.
- Resultados experimentais mostram que as quotas de complexidade superestimam o número de exemplos necessários na prática já que são quotas superiores de pior caso.

22

Conclusões (cont)

- Additional results produced for analyzing:
 - Learning with queries
 - Learning with noisy data
 - Average case sample complexity given assumptions about the data distribution.
 - Learning finite automata
 - Learning neural networks
- Analyzing practical algorithms that use a preference bias is difficult.
- Some effective practical algorithms motivated by theoretical results:
 - Boosting
 - Support Vector Machines (SVM)

23