

Aprendizado de Máquina

Aula 1

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

Aula 1 - 16/03/10

1

Referências

- **Slides das aulas:** na página
<http://www.ic.uff.br/~bianca/aa/>
- **Livro-texto:**
Machine Learning
Tom M. Mitchell
McGraw-Hill, 1997
<http://www.cs.cmu.edu/~tom/mlbook.html>
(Pasta 524, xerox do Bloco E)
- **Artigos que serão citados ao longo do curso.**

Aula 1 - 16/03/10

2

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 (06/04)
5. Avaliação Experimental – Cap. 5 (13/04)
6. Teoria do Aprendizado – Cap. 7 (20/04)
7. Aprendizado de Regras – Cap. 10 (27/04)
8. Redes Neurais – Cap. 4 (04/05)
9. Máquinas de Vetor de Suporte – Artigo (11/05)
10. Aprendizado Bayesiano – Cap. 6 e novo cap. online (18/05)
11. Aprendizado Baseado em Instâncias – Cap. 8 (25/05)
12. Classificação de Textos – Artigo (01/06)
13. Aprendizado Não-Supervisionado – Artigo (08/06)

Aula 1 - 16/03/10

3

Avaliação

- Trabalhos Teórico (TT)
- Trabalho Prático (TP)
- Listas de Exercício (LE) = média das notas das listas que serão dadas ao longo do curso

$$M = (TT + TP + LE)/3$$

Aula 1 - 16/03/10

4

Trabalhos

- **Trabalho Prático**
 - Cada aluno deverá propor e executar um projeto de aprendizado automático especificando o tipo de problema, base de dados, metodologia de avaliação e algoritmo(s) a serem utilizados.
 - Desenvolvido ao longo do semestre.
 - Apresentação de 20 minutos no final do curso.
- **Trabalho Teórico**
 - Apresentação de um artigo da área de aprendizado automático, escolhido pela professora de acordo com o interesse do aluno.
 - Cada apresentação terá duração de 45 minutos.

Aula 1 - 16/03/10

5

O que é “aprendizado automático”?

- Sub-área da inteligência artificial, também conhecida como “machine learning” ou “aprendizado de máquina”.
- Surgiu da ideia de criar programas que **aprendam** um determinado **comportamento ou padrão** automaticamente a partir de exemplos ou observações.
- Guarda alguma relação com o aprendizado humano.
 - Seres humanos (e outros animais) são capazes de **generalizar** a partir de exemplos.

Aula 1 - 16/03/10

6

Qual é a relação com a mineração de dados?

- A mineração de dados é o processo de extração automática de **conhecimento** a partir de grandes bases de dados.
- Algoritmos de aprendizado automático podem ser vistos como algoritmos que extraem um padrão de comportamento a partir de dados (exemplos).
 - Logo, podem ser utilizados como algoritmos de mineração de dados.
- Porém, algoritmos de aprendizado nem sempre utilizam bases de dados.
 - Podem aprender diretamente a partir da interação com o ambiente ou com um simulador.
- Ambas as áreas “emprestam” muitos métodos da área de estatística.

Aula 1 - 16/03/10

7

O que é “aprendizado automático”?

- Herbert Simon: “Aprendizado é qualquer processo no qual um sistema melhora seu desempenho através da experiência.”
- Qual é a tarefa?
 - Classificação
 - Resolução de problemas / planejamento / controle

Aula 1 - 16/03/10

8

Classificação

- Atribuir um objeto/evento a uma categoria, pertencente a um conjunto finito de categorias)
 - Diagnóstico médico
 - Detecção de fraude em cartões de crédito
 - Detecção de vírus em redes de computadores
 - Filtragem de spam em e-mails
 - Recomendação de produtos em e-commerce
 - Investimentos financeiros
 - Bioinformática (sequências de DNA)
 - Reconhecimento de voz
 - Reconhecimento de caracteres
 - Reconhecimento de imagens

Aula 1 - 16/03/10

9

Resolução de problemas / planejamento / controle

- Executar ações em um ambiente para atingir um determinado objetivo.
 - Resolver problemas de matemática
 - Jogar xadrez, damas ou gamão
 - Dirigir um carro
 - Pilotar um avião, helicóptero ou foguete
 - Controlar um elevador
 - Controlar um personagem em um jogo
 - Controlar um robô móvel

Aula 1 - 16/03/10

10

Medindo o desempenho

- Acurácia da classificação
- Corretude da solução encontrada
- Qualidade da solução (comprimento, eficiência)
- Tempo de execução

Aula 1 - 16/03/10

11

Por que estudar aprendizado de máquina?

- Desenvolver sistemas que são muito difíceis/caros de construir manualmente porque requerem conhecimentos detalhados de uma determinada tarefa.
- Desenvolver sistemas que possam se adaptar a usuários individualmente.
 - Filtro de notícias ou e-mail personalizado.
 - Sistemas de educação personalizados.
- Extrair conhecimento de grandes bases de dados (**mineração de dados**).
 - Análise de cestas de compras (ex.: fraldas e cerveja)

Aula 1 - 16/03/10

12

Qual é a relação com a mineração de dados?

- A mineração de dados é o processo de extração automática de **conhecimento** a partir de grandes bases de dados.
- Algoritmos de aprendizado automático podem ser vistos como algoritmos que extraem um padrão de comportamento a partir de dados (exemplos).
 - Logo, podem ser utilizados como algoritmos de mineração de dados.
- Porém, algoritmos de aprendizado nem sempre utilizam bases de dados.
 - Podem aprender diretamente a partir da interação com o ambiente ou com um simulador.
- Ambas as áreas “emprestam” muitos métodos da área de estatística.

Aula 1 - 16/03/10

13

Por que estudar aprendizado de máquina?

- Estudos computacionais do aprendizado podem ajudar a entender o aprendizado em humanos e outros organismos.
 - Aprendizado neural “Hebbiano”
 - “Neurônios que disparam juntos se conectam.”
 - A dificuldade relativa de humanos em aprender conceitos disjuntivos em relação a conceitos conjuntivos.
 - Lei da prática



Aula 1 - 16/03/10

14

Por que estudar aprendizado de máquina?

- Grande quantidade de algoritmos efetivos e eficientes.
- Grande quantidade de dados disponíveis (principalmente online).
- Grande quantidade de recursos computacionais disponíveis.

Aula 1 - 16/03/10

15

Definindo a tarefa de aprendizado

Melhorar na tarefa, T , com respeito à métrica de desempenho, D , baseado na experiência, E .

T: Jogar Damas

D: Percentagem de jogos ganhos contra um adversário arbitrário

E: Jogar jogos contra ele próprio

T: Reconhecer palavras escritas a mão

D: Percentagem de palavras corretamente reconhecidas

E: Base de dados de imagens com as palavras correspondentes

T: Dirigir em estradas usando sensores de visão

D: Distância média dirigida antes de um erro

E: Sequência de imagens e comandos de direção gravados de uma pessoa dirigindo

T: Categorizar mensagens de e-mail como spam ou legítimas.

D: Percentagem de mensagens de e-mail corretamente classificadas

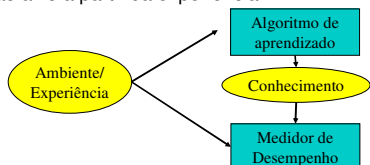
E: Base de dados de mensagens, algumas classificadas por humanos

Aula 1 - 16/03/10

16

Projetando um Sistema de Aprendizado

- Definir a experiência de treinamento.
- Definir exatamente o que será aprendido, i.e., a **função alvo**.
- Escolher como representar a função alvo.
- Escolher um algoritmo de aprendizado para inferir a função alvo a partir da experiência.



Aula 1 - 16/03/10

17

Exemplo de Problema de Aprendizado

- Aprender a jogar damas jogando contra si mesmo.
- Veremos uma abordagem similar a usada no primeiro sistema de aprendizado automático desenvolvido por Arthur Samuels na IBM em 1959.

Aula 1 - 16/03/10

18

Experiência para Treinamento

- **Experiência direta:** Dar alguns exemplos de pares de entrada e saída da função alvo.
 - Tabuleiros de damas identificados com a jogada correta, por exemplo extraídos do jogo de um especialista.
- **Experiência indireta:** Dar feedback que **não** é um par de E/S para a função alvo.
 - Sequências potencialmente arbitrárias de jogadas e o resultado final do jogo.
- **Problema da atribuição de crédito:** Como atribuir crédito a jogadas individuais quando a experiência é indireta?

Aula 1 - 16/03/10

19

Fonte dos dados de treinamento

- Exemplos aleatórios.
 - Exemplos negativos também ou só positivos?
- Exemplos de treinamento escolhidos por um professor benevolente.
 - Exemplos "didáticos".
- Algoritmo pode pedir o rótulo de um exemplo qualquer no ambiente.
- Algoritmo pode construir um exemplo e pedir o rótulo desse exemplo.
- Algoritmo pode conduzir experimentos no ambiente sem ajuda humana.

Aula 1 - 16/03/10

20

Distribuição de Treinamento vs. Distribuição de Teste

- Geralmente supomos que os exemplos de treinamento e de teste são derivados independentemente da mesma distribuição
 - IID: Independentemente e identicamente distribuídos.
- Se os exemplos não são independentes, precisamos de **classificação coletiva**.
- Se a distribuição de teste é diferente, precisamos de **aprendizado de transferência**.

Aula 1 - 16/03/10

21

Escolhendo uma função alvo

- Que função deve ser aprendida e como ela será usada pelo sistema?
- Para damas, supomos que é dada uma função para gerar jogadas possíveis a partir de uma dada posição do tabuleiro e queremos decidir qual é a melhor jogada.
 - Podemos aprender a função:
Escolher jogada(tabuleiro, jogadas) → melhor jogada
 - Ou podemos aprender uma **função de avaliação**, $V(\text{tabuleiro}) \rightarrow R$, que dá para cada posição do tabuleiro um score do quão favorável ela é. V pode ser usada para escolher uma jogada.

Aula 1 - 16/03/10

22

Definição ideal de $V(b)$

- Se b é um tabuleiro final ganhador, então $V(b) = 100$
- Se b é um tabuleiro final perdedor, então $V(b) = -100$
- Se b é um tabuleiro final de empate, então $V(b) = 0$
- Senão, $V(b) = V(b')$, onde b' é a posição final do tabuleiro com maior score possível que é atingida começando de b e jogando de forma ótima até o fim do jogo (supondo que o oponente joga de forma ótima também).
 - Pode ser computado usando a árvore de busca minimax.

Aula 1 - 16/03/10

23

Aproximando $V(b)$

- Calcular $V(b)$ não é tratável porque envolve buscar em uma árvore completa exponencial.
- Logo essa definição é **não-operacional**.
- Uma definição **operacional** pode ser calculada em tempo razoável (polinomial).
- Precisamos aprender uma **aproximação operacional** da função de avaliação ideal.

Aula 1 - 16/03/10

24

Representando a função alvo

- A função alvo pode ser representada de muitas formas: tabela, regras simbólicas, função numérica, rede neural.
- Existe um trade-off entre a expressividade da representação e a facilidade de aprendizado.
- Quanto mais expressiva for a representação, melhor ela será em representar uma função arbitrária; porém, mais exemplos serão necessários para aprender a função corretamente.

Aula 1 - 16/03/10

25

Função linear para representar $V(b)$

- Em damas, usamos uma aproximação linear da função de avaliação.

$$\hat{V}(b) = w_0 + w_1 \cdot bp(b) + w_2 \cdot rp(b) + w_3 \cdot bk(b) + w_4 \cdot rk(b) + w_5 \cdot bt(b) + w_6 \cdot rt(b)$$

- $bp(b)$: número de peças pretas no tabuleiro b
- $rp(b)$: número de peças vermelhas no tabuleiro b
- $bk(b)$: número de reis pretos no tabuleiro b
- $rk(b)$: número de reis vermelhos no tabuleiro b
- $bt(b)$: número de peças ameaçadas (i.e. que serão pegadas pelo vermelho na próxima jogada)
- $rt(b)$: número de peças vermelhas ameaçadas

Aula 1 - 16/03/10

26

Obtendo valores de treinamento

- Supervisão direta pode existir para a função alvo.
 - $\langle bp=3, rp=0, bk=1, rk=0, bt=0, rt=0 \rangle, 100 \rangle$ (preto ganhou)
- Com feedback indireto, valores de treinamento podem ser estimados usando **aprendizado de diferença temporal** (usado em **aprendizado por reforço**).

Aula 1 - 16/03/10

27

Aprendizado de Diferenças Temporais

- Estimar valores de treinamento para posições de tabuleiro não-terminais usando o valor estimado do seu sucessor em um jogo real.

$$V_{train}(b) = \hat{V}(\text{successor}(b))$$

onde $\text{successor}(b)$ é a próxima posição do jogo.

- Valores mais para o final do jogo são inicialmente mais corretos e o treinamento continuado faz com que eles sejam trazidos lentamente para as posições mais do início.

Aula 1 - 16/03/10

28

Algoritmo de Aprendizado

- Usa valores de treinamento da função alvo para induzir uma definição que esteja de acordo com esses exemplos e que esperamos que generalize para exemplos não vistos.
- Em estatística, aprender a aproximar uma função contínua é chamado de **regressão**.
- Tenta minimizar uma medida de erro tal como:

$$E = \frac{\sum_{b \in B} [V_{train}(b) - \hat{V}(b)]^2}{|B|}$$

Aula 1 - 16/03/10

29

Algoritmo Least Mean Squares (LMS)

- Um algoritmo de descida de gradiente que muda os pesos de uma função linear tentando minimizar o erro médio quadrático.
- Até que os pesos converjam:

Para cada exemplo de treinamento b faça :

- Calcule o erro absoluto :

$$\text{error}(b) = V_{train}(b) - \hat{V}(b)$$
- Para cada característica do tabuleiro, f_b , mude seu peso, w_i :

$$w_i = w_i + c \cdot f_i \cdot \text{error}(b)$$
 para alguma constante pequena c

Aula 1 - 16/03/10

30

Discussão do LMS

- Intuitivamente, LMS executa as seguintes regras:
 - Se a saída de um exemplo for correta, não faça modificações.
 - Se a saída for muito alta, diminua os pesos proporcionalmente aos valores das suas características correspondentes, de tal forma que a saída diminua.
 - Se a saída for muito baixa, aumente os pesos proporcionalmente aos valores das suas características correspondentes, de tal forma que a saída aumente.
- Sob suposições fracas, podemos provar que o LMS irá convergir a um conjunto de pesos que minimiza o erro médio quadrático.

Aula 1 - 16/03/10

31

Conclusões sobre o aprendizado

- O aprendizado pode ser visto como o uso de experiência direta ou indireta pra aproximar uma função.
- Aproximação de funções pode ser vista como uma busca num espaço de hipóteses (representações de funções) por uma que melhor se adapta ao conjunto de treinamento.
- Métodos de aprendizado diferentes usam espaços de hipóteses diferentes (linguagens de representação) e/ou usam técnicas de busca diferentes.

Aula 1 - 16/03/10

32

Funções de Representação

- Funções numéricas
 - Regressão linear
 - Redes neurais
 - Máquinas de vetor de suporte (SVMs)
- Funções simbólicas
 - Árvores de decisão
 - Regras em lógica proposicional
 - Regras em lógica de primeira ordem
- Funções baseadas em instância
 - Vizinho mais próximo
 - Baseadas em casos
- Modelos gráficos probabilísticos
 - Naïve Bayes
 - Redes Bayesianas
 - Modelos de Markov Ocultos (HMMs)
 - Gramáticas Probabilísticas Livres de Contexto (PCFGs)
 - Redes de Markov

Aula 1 - 16/03/10

33

Algoritmos de Busca

- Descida de gradiente
 - Perceptron
 - Backpropagation
- Programação dinâmica
 - Aprendizado de HMM
 - Aprendizado de PCFG
- Divisão e conquista
 - Indução de árvores de decisão
 - Aprendizado de regras
- Computação evolucionária
 - Algoritmos genéticos (GAs)
 - Programação genética (GP)

Aula 1 - 16/03/10

34

Avaliação de Sistemas de Aprendizado

- Experimental
 - Conduzir experimentos controlados com validação cruzada para comparar métodos em uma variedade de bases de dados.
 - Coletar dados sobre o seu desempenho, ex.: acurácia de teste, tempo de treinamento, tempo de teste.
 - Analisar diferenças quanto a significância estatística.
- Teórica
 - Analisar algoritmos matematicamente e provar teoremas sobre sua:
 - Complexidade computacional
 - Abilidade de se adaptar aos dados de treinamento
 - Número de exemplos de treinamento necessários para se aprender uma função correta

Aula 1 - 16/03/10

35

Histórico do Aprendizado de Máquina

- 1950s
 - Jogador de damas de Samuel
 - Pandemonium de Selfridge
- 1960s:
 - Redes neurais: Perceptron
 - Reconhecimento de padrões
 - Teoria do aprendizado no limite
 - Minsky e Papert provam limitações do Perceptron
- 1970s:
 - Indução de conceitos simbólicos
 - Algoritmo de Winston
 - Sistemas especialistas e gargalo da aquisição de conhecimento
 - Algoritmo de árvore de decisão ID3
 - Michalski's AQ e diagnóstico de soja
 - Descoberta científica com BACON
 - Descoberta matemática com AM

Aula 1 - 16/03/10

36

Histórico do Aprendizado de Máquina

- 1980s:
 - Aprendizado de regras e árvores de decisão avançado
 - Aprendizado em planejamento e resolução de problemas
 - Analogia
 - Arquiteturas cognitivas
 - Ressurgimento de redes neurais
 - Teoria do Aprendizado PAC
 - Foco na metodologia experimental
- 1990s
 - Mineração de dados
 - Agentes de software adaptativos e mineração web
 - Aprendizado de textos
 - Aprendizado por reforço (RL)
 - Programação lógica indutiva (ILP)
 - Ensembles: Bagging, Boosting e Stacking
 - Aprendizado de redes neurais

Aula 1 - 16/03/10

37

Histórico do Aprendizado de Máquina

- 2000s
 - Máquinas de Vetor de Suporte
 - Métodos de Kernel
 - Modelos Gráficos
 - Aprendizado Relacional
 - Aprendizado de Transferência
 - Rotulação de Sequências
 - Classificação Coletiva
 - Aplicações a Sistemas de Computação
 - Compiladores
 - Debugging
 - Computação Gráfica
 - Segurança
 - Gerenciamento de e-mail
 - Assistentes personalizados que aprendem
 - Aprendizado em visão e robótica

Aula 1 - 16/03/10

38

Recursos

- Bases de Dados
 - UCI Repository: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
 - UCI KDD Archive: <http://kdd.ics.uci.edu/summary.data.application.html>
 - Statlib: <http://lib.stat.cmu.edu/>
 - Delve: <http://www.cs.utoronto.ca/~delve/>
 - LETOR: <http://research.microsoft.com/en-us/um/beijing/projects/letor/index.html>
- Ferramenta
 - WEKA: <http://www.cs.waikato.ac.nz/ml/weka/>

Aula 1 - 16/03/10

39