

Aprendizado de Máquina	Conceitos
<p>Universidade de São Paulo</p> <p>Instituto de Ciências Matemáticas e de Computação</p> <p>Departamento de Ciências de Computação</p> <p>Rodrigo Fernandes de Mello</p> <p>mello@icmc.usp.br</p>	<ul style="list-style-type: none"> • Surge com a intenção de explorar possibilidades de máquinas “aprenderem” sobre diversos assuntos e contextos <ul style="list-style-type: none"> • Reconhecimento de discurso • Detecção de fraude em sistemas de crédito • Veículos autônomos • Jogos • Área multidisciplinar: <ul style="list-style-type: none"> • Inteligência artificial • Estatística • Matemática • Teoria de Controle • Teoria da Informação • Biologia, etc.

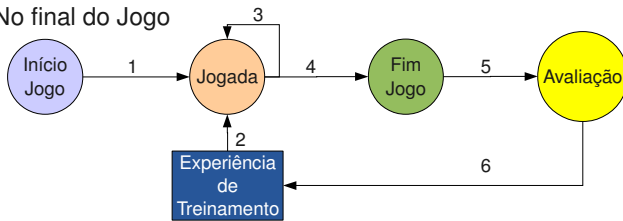
	Conceitos
<p>Introdução...</p>	<ul style="list-style-type: none"> • Afinal de contas, como definimos “aprendizado”? <ul style="list-style-type: none"> • Um programa “aprende” a partir de experiências E com respeito a alguma classe de tarefas T e medida de desempenho P, se o desempenho de tarefas em T, medido por P, melhora com experiências E • Exemplo: <ul style="list-style-type: none"> • Jogo de computador <ul style="list-style-type: none"> – Tarefa T <ul style="list-style-type: none"> • Jogar Xadrez – Medida de desempenho P <ul style="list-style-type: none"> • Partidas ganhas contra oponentes versus perdas – Experiência de Treinamento E <ul style="list-style-type: none"> • Prática de jogo contra ele mesmo

Conceitos	Projeto de um Sistema de Aprendizado
<ul style="list-style-type: none"> • Logo, antes de abordar um problema devemos definir: <ul style="list-style-type: none"> • Tarefa • Medida de Desempenho • Experiência de Treinamento 	<ul style="list-style-type: none"> • Qual o problema que desejamos abordar? <ul style="list-style-type: none"> • Por exemplo: Tarefa: Jogo de Xadrez • Informações prévias a serem utilizadas para Treinamento <ul style="list-style-type: none"> • Influenciam fortemente no aprendizado • Experiência de Treinamento – aprende-se a partir de feedbacks: <ul style="list-style-type: none"> – A partir de cada movimento – Somente no final da partida <ul style="list-style-type: none"> • Nessa situação pode-se inferir, do final para o início, a importância de cada jogada

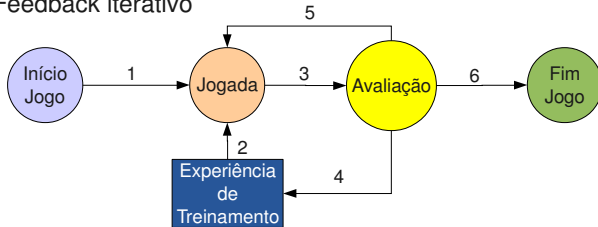
Projeto de um Sistema de Aprendizado

- Aprende-se com feedbacks:

- No final do Jogo



- Feedback iterativo



Projeto de um Sistema de Aprendizado

- Avaliação pode ser conduzida com base em:

- Um conjunto de experiências anteriores (Treinamento)**
 - Sistema executa e toma suas próprias decisões
 - Essas execuções auxiliam a formar uma base de experiências
 - Pode-se formar essa base
 - Disponibilizar essa base no início
 - Disponibilizar alguma base no início, a qual evolui no tempo
- Intervenção de um professor**
 - O “professor” aguarda uma solução do programa e apenas dá feedback da qualidade da solução

Avaliação

Projeto de um Sistema de Aprendizado

- Informações prévias a serem utilizadas para Treinamento

- Quão bem distribuídos são os exemplos de treinamento?

- Exercitam somente um ou mais tipos de jogadas/possibilidades?
- Exemplos de treinamento exprimem bem os casos que serão exercitados na situação de jogo?
- O que ocorre se situação real for muito distinta daquelas consideradas em treinamento?
 - Problemas!
 - Por isso devemos projetar sistemas com boa capacidade de generalização
 - Capacidade de se adaptar a situações reais

Experiência de Treinamento

Projeto de um Sistema de Aprendizado

- Como escolhemos uma futura jogada?

- Testar N possíveis jogadas permitidas
- Verificar a aptidão (fitness) ou custo de cada jogada
 - Como definir tal função?
 - Aptidão** – Busca maximizar as boas possibilidades a partir de tal jogada
 - Custo** – Busca reduzir custos, tal como a eliminação de uma peça

- Pode-se, avaliar **o jogo como um todo e não cada jogada**

- Considerado em situações em que há dificuldade em definir a aptidão ou o custo por jogada

Projeto de um Sistema de Aprendizado

- Em um cenário hipotético:

- Programa poderia avaliar todas as combinações de possíveis jogadas tanto dele quanto de seu oponente antes de definir uma nova jogada
 - Custo muito alto, tende a ser proibitivo

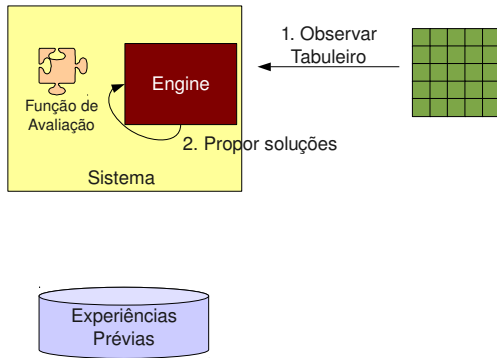
Projeto de um Sistema de Aprendizado

- Como definir a função de aptidão ou de custo?

- Poderíamos ter uma **longa base de dados** com diversas jogadas que poderiam ser feitas a partir de cada jogada do oponente
 - Cada entrada na base de dados teria uma aptidão ou custo associado
- Poderíamos ter **regras if-then-else**
 - Jogar a partir dessas regras
- Poderíamos definir uma **função matemática** para avaliar cada jogada
 - Assim não precisaríamos manter uma extensa base de dados ou regras
 - Exemplo de função simples: avaliar peças que correm ou não risco

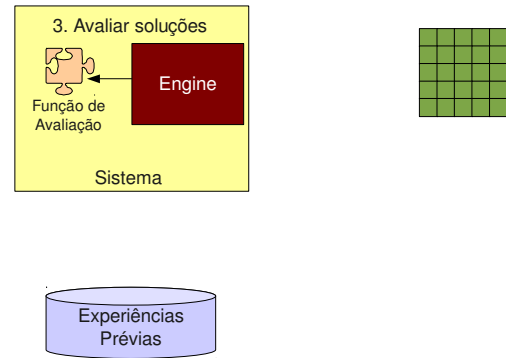
Projeto de um Sistema de Aprendizado

- Sendo assim:



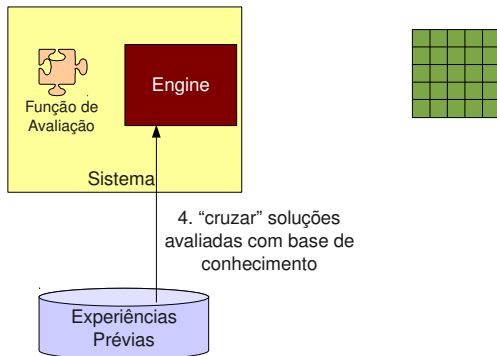
Projeto de um Sistema de Aprendizado

- Sendo assim:



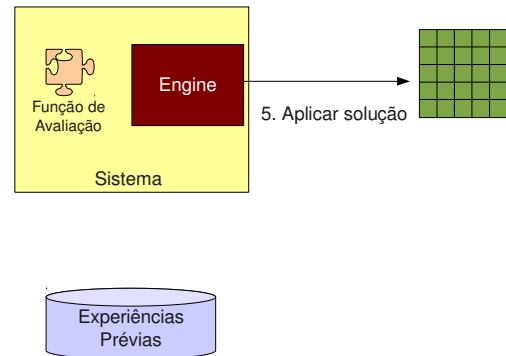
Projeto de um Sistema de Aprendizado

- Sendo assim:



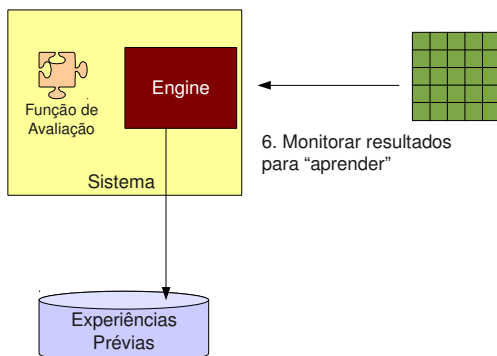
Projeto de um Sistema de Aprendizado

- Sendo assim:



Projeto de um Sistema de Aprendizado

- Sendo assim:



Projeto de um Sistema de Aprendizado

- Vamos projetar um jogo?
 - Xadrez, Damas, Jogo da Velha, etc.
- Partes:
 - Como representar conhecimento prévio?
 - Como representar soluções?
 - Como avaliar se uma possível solução vale a pena?
 - Função de custo ou aptidão
 - Aplicar solução
 - Obter retorno:
 - Sucesso ou Fracasso?

- Veja os conjuntos de dados disponíveis no UCI (University of California at Irvine) Machine Learning Repository
 - <http://archive.ics.uci.edu/ml/>

Mais conceitos relacionados a hipóteses...

Conceitos Relevantes

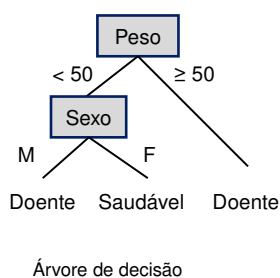
- Inferência
 - Processo de derivar conclusões lógicas a partir de premissas conhecidas ou decididamente verdadeiras
- Indução
 - Processo de inferência que parte do particular (com base no conhecimento de certo número de dados singulares) para o universal a fim de obter uma conclusão
- Dedução
 - Processo de inferência que parte do universal para o particular a fim de obter uma conclusão

Conceitos Relevantes

- Técnicas de Aprendizado de Máquina
 - Buscam induzir hipóteses a partir de um conjunto de exemplos
 - Uma hipótese também é chamada de modelo por diversos autores
- Nesse processo de indução a técnica de aprendizado de máquina:
 - Utiliza de uma linguagem para representar hipóteses
 - Privilegia determinadas hipóteses em detrimento de outras

Conceitos Relevantes

- Ao escolhermos uma linguagem de representação das hipóteses
 - Exemplos:



0.45	-0.40	0.54	0.12	0.98	0.37
-0.45	0.11	0.91	0.34	-0.20	0.83
-0.29	0.32	-0.25	-0.51	0.41	0.70

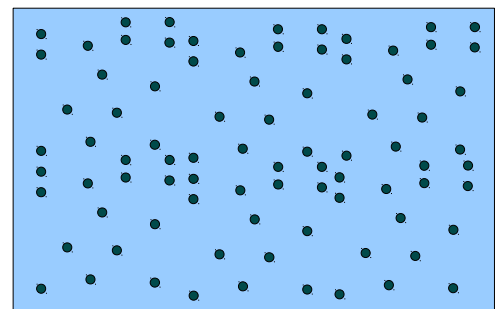
Redes neurais

Se $\text{Peso} \geq 50$ então Doente
 Se $\text{Peso} < 50$ e $\text{Sexo} = M$ então Doente
 Se $\text{Peso} < 50$ e $\text{Sexo} = F$ então Saudável

Conjunto de regras

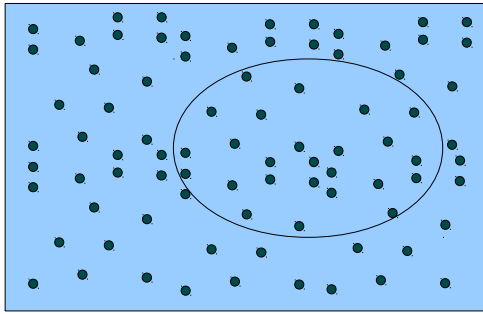
Conceitos Relevantes

- Conjunto universo de hipóteses



Conceitos Relevantes

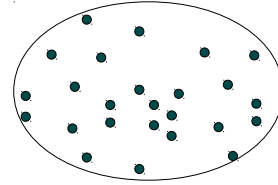
- Ao escolhermos uma linguagem de representação das hipóteses



- Conhecido como bias de representação ou bias de linguagem

Conceitos Relevantes

- A técnica pode preferir determinadas hipóteses em detrimento de outras
 - Conhecido como bias de preferência
- Bias de busca
 - Define como hipóteses serão procuradas no espaço



- Exemplo:
 - O ID3 e o C4.5 (Algoritmos de Árvore de Decisão) preferem árvores mais curtas em relação às mais longas

Conceitos Relevantes

- Bias de indutivo compreende essencialmente:
 - Bias de representação de hipóteses
 - Bias de preferência
 - Bias de busca
- Sem bias indutivo, uma técnica de aprendizado de máquina:
 - Não restringe o espaço para busca de possíveis hipóteses
 - Assim, o problema poderia ser de solução impraticável ou impossível
 - O bias indutivo é utilizado para generalizar o conhecimento a partir de exemplos conhecidos
 - Sem o bias indutivo, conheceríamos apenas os exemplos e estaríamos “amarrados” a eles

Grandes subáreas de Aprendizado de Máquina...

Conceitos Relevantes

- O Aprendizado de Máquina pode ser subdividido em duas categorias principais:
 - Supervisionado
 - Conhece-se a classe final para exemplos de treinamento
 - Assim, pode-se escolher hipóteses com base em acertos e erros dados os exemplos de treinamento
 - Não-supervisionado
 - Não se conhece a classe dos exemplos de treinamento
 - Considera-se relações de similaridade/dissimilaridade entre exemplos para separá-los em conjuntos distintos
 - Pode-se ter uma etapa posterior em que um especialista analisa esses conjuntos

O que significa aprender um conceito?

- O que é Aprendizado de um Conceito?
 - Definição que permite categorizar exemplos como positivos ou negativos
- Aprender um conceito:
 - Chamado de aprendizado Indutivo
 - Pois parte do particular para o geral
 - A partir de um conjunto de dados
 - Busca pela hipótese que melhor categorize esse conjunto
 - Buscamos “aprender” um conceito usando um conjunto de treinamento

- Considere a situação:
 - João gosta de nadar sob determinadas condições?

Ex	Céu	Temperatura	Umidade	Vento	Água	Previsão	Gostar de Nadar
1	Ensolarado	Quente	Normal	Forte	Quente	Mesma	Sim
2	Ensolarado	Quente	Alta	Forte	Quente	Mesma	Sim
3	Chuvoso	Fria	Alta	Forte	Quente	Mudar	Não
4	Ensolarado	Quente	Alta	Forte	Fria	Mudar	Sim

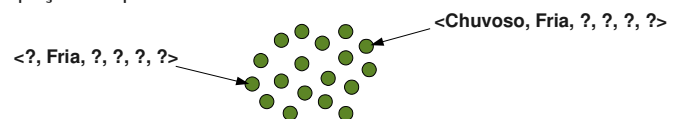
- Considere que há três possíveis opções para Céu e duas para Temperatura, Umidade, Vento, Água e Previsão
- Como “aprender” esse conceito?
 - Deve-se responder Sim para os casos positivos e Não para os casos negativos

- Possíveis Hipóteses:
 - A mais abrangente:
 - $\langle ?, ?, ?, ?, ?, ? \rangle$
 - O termo ? indica que qualquer valor é aceito
 - Essa hipótese não é capaz de categorizar!
 - A mais específica:
 - $\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$
 - O termo \emptyset indica vazio, ou seja, nenhum termo é aceito
 - A seguinte hipótese poderia categorizar:
 - $\langle ?, \text{Fria}, \text{Alta}, ?, ?, ? \rangle$
 - Somente essa?

- Outras possíveis Hipóteses que permitem categorizar nosso conjunto de dados:
 - $\langle ?, \text{Fria}, ?, ?, ?, ? \rangle$
 - $\langle \text{Chuvoso}, \text{Fria}, ?, ?, ?, ? \rangle$
 - $\langle \text{Chuvoso}, ?, ?, ?, ?, ? \rangle$
 - Etc...

- Considere um conceito c a ser aprendido:
 - Sejam exemplos de treinamento x em X
 - $c: X \rightarrow \{0,1\}$
 - $c(x) = 1$ se Gostar de Nadar = 1
 - $c(x) = 0$ se Gostar de Nadar = 0
 - Podemos ver como uma situação binária, em que categorizamos exemplos como pertencentes ou não a uma classe
- Em que:
 - $c(x) = 1$ representa um exemplo positivo
 - $c(x) = 0$ representa um exemplo negativo

- Espaço de hipóteses:



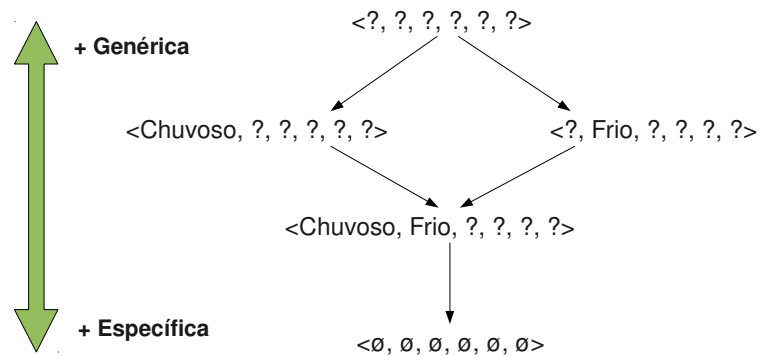
- Podemos:
 - Buscar pela melhor hipótese, mas o que ocorre se:
 - Espaço de hipóteses é muito grande?
 - Ex: Caixeiro Viajante
 - Se aptidão ou custo das hipóteses é dado por uma função convexa?
 - Função de aptidão ou custo das hipóteses é descontínua?
 - Não conhecemos essa função de aptidão ou custo?

Aprendizado de Conceitos

- Exemplos:
 - Como fica o espaço de hipóteses para uma peça do tabuleiro de Damas em dada circunstância?
 - Como fica o espaço de hipóteses para uma peça do tabuleiro de Xadrez em dada circunstância?
 - Como é o espaço de hipóteses para o problema do Caixeiro Viajante?
- Perceber que hipóteses podem ser válidas ou inválidas

Hipóteses Genéricas e Específicas

- Há situações em que é possível definir uma ordem parcial entre hipóteses
 - Assim podemos definir quais hipóteses são mais específicas e quais são mais genéricas
 - Por exemplo:



Algoritmos para a Busca por Hipóteses

- Pode-se “aproveitar” essa ordem parcial entre hipóteses para buscar por uma hipótese consistente com os exemplos de treinamento
 - Mitchell propõe alguns algoritmos:
 - Find-S
 - List-Then-Eliminate
 - Candidate-Elimination

Find-S

- Inicializar **h** como a hipótese mais específica
- Para cada exemplo positivo **x** em **X**
 - Para cada atributo **a** em **h**
 - Se a restrição **a** é satisfeita por **x**
 - Não faça nada
 - Senão
 - Substitua **a** em **h** pela próxima restrição o mais específica possível, mas capaz de satisfazer **x**
- Saída dada por hipótese **h**

Find-S

- Inicializar **h** como a hipótese mais específica
 - h** ← <0, 0, 0, 0, 0, 0>
- Após observar o primeiro exemplo positivo:
 - h** ← <Ensolarado, Quente, Normal, Forte, Quente, Mesma>
- Após o segundo exemplo positivo:
 - h** ← <Ensolarado, Quente, ?, Forte, Quente, Mesma>

Find-S

- Após observar o terceiro exemplo, nesse caso negativo, **h** não muda:
 - h** ← <Ensolarado, Quente, ?, Forte, Quente, Mesma>
- Após observar o quarto exemplo, que é positivo, então:
 - h** ← <Ensolarado, Quente, ?, Forte, ?, ?>

Ex	Céu	Temperatura	Umidade	Vento	Água	Previsão	Gostar de Nadar
1	Ensolarado	Quente	Normal	Forte	Quente	Mesma	Sim
2	Ensolarado	Quente	Alta	Forte	Quente	Mesma	Sim
3	Chuvoso	Fria	Alta	Forte	Quente	Mudar	Não
4	Ensolarado	Quente	Alta	Forte	Fria	Mudar	Sim

Ex	Céu	Temperatura	Umidade	Vento	Água	Previsão	Gostar de Nadar
1	Ensolarado	Quente	Normal	Forte	Quente	Mesma	Sim
2	Ensolarado	Quente	Alta	Forte	Quente	Mesma	Sim
3	Chuvoso	Fria	Alta	Forte	Quente	Mudar	Não
4	Ensolarado	Quente	Alta	Forte	Fria	Mudar	Sim

- Garante convergência para uma hipótese válida?
 - Irá encontrar uma hipótese consistente com os exemplos de treinamento, no entanto não se sabe se é a única hipótese consistente (podem haver outras)
 - Desde que conjunto de treinamento não apresente erros ou ruídos
- Seria melhor um algoritmo capaz de obter todo o conjunto de hipóteses válidas
 - Assim definimos qual delas iremos, de fato, utilizar no futuro
 - Ou utilizamos um esquema de votação

- Apresenta todas as hipóteses válidas dado um conjunto de treinamento

Algoritmo

- Inicializa com espaço **V**:
 - contendo todas hipóteses possíveis em **H**
- Para cada exemplo no conjunto de treinamento:
 - Remova de **V** qualquer hipótese que não seja consistente com o exemplo
 - Em que consistência significa $h(x) = c(x)$
 - Ou seja, a hipótese produz o resultado esperado
- Saída: lista de hipóteses em **V**

List-Then-Eliminate

List-Then-Eliminate

- Questões:
 - Espaço H com todas hipóteses pode ser infinito
 - Por exemplo: encontrar um valor para um parâmetro real
 - Definir todas hipóteses pode demandar muito processamento mesmo quando isso é possível
 - Ex: Caixeiro Viajante
 - Pode ser viável para problemas com espaço finito e pequeno de hipóteses

- O que ocorreria com o problema do Caixeiro Viajante?
 - Consideremos $n = 3$ cidades
 - Quantas combinações são possíveis?
 - $n!$
 - Assumindo que o custo da solução é dado pelo custo do traslado, o qual é simétrico
 - Por exemplo, combustível consumido
 - Nem todas soluções são distintas:
 - $(n-1)! / 2$ soluções distintas

List-Then-Eliminate

Candidate-Elimination

- Tendências de custo
 - Tamanho do problema (PI), em que k é o número de cidades (exemplificando com $k!$)

Capacidade do Computador (CC) em MIPS

PI (k)/CC	1,000	5,000	10,000	50,000	100,000	$20 \cdot 10^9$
$k = 20, 4 \cdot 10^{19}$ s	$4 \cdot 10^{10}$ s	$9 \cdot 10^9$ s	$4 \cdot 10^9$ s	$9 \cdot 10^8$ s	$4 \cdot 10^8$ s	$2 \cdot 10^7$ s
$k = 30, 7 \cdot 10^{23}$ m	$1 \cdot 10^{23}$ m	$2 \cdot 10^{22}$ m	$1 \cdot 10^{22}$ m	$2 \cdot 10^{21}$ m	$1 \cdot 10^{21}$ m	$6 \cdot 10^{15}$ m
$k = 40, 3 \cdot 10^{49}$ h	$8 \cdot 10^{36}$ h	$1 \cdot 10^{36}$ h	$8 \cdot 10^{35}$ h	$1 \cdot 10^{35}$ h	$8 \cdot 10^{34}$ h	$4 \cdot 10^{29}$ h
$k = 50, 1 \cdot 10^{66}$ y	$4 \cdot 10^{49}$ y	$9 \cdot 10^{48}$ y	$4 \cdot 10^{48}$ y	$9 \cdot 10^{47}$ y	$4 \cdot 10^{47}$ y	$2 \cdot 10^{42}$ y
$k = 60, 4 \cdot 10^{83}$ d	$1 \cdot 10^{66}$ d	$3 \cdot 10^{65}$ d	$1 \cdot 10^{65}$ d	$3 \cdot 10^{64}$ d	$1 \cdot 10^{64}$ d	$7 \cdot 10^{58}$ d
$k = 70, 8 \cdot 10^{101}$ c	10^{84} c	10^{84} c	10^{83} c	10^{83} c	10^{82} c	10^{77} c

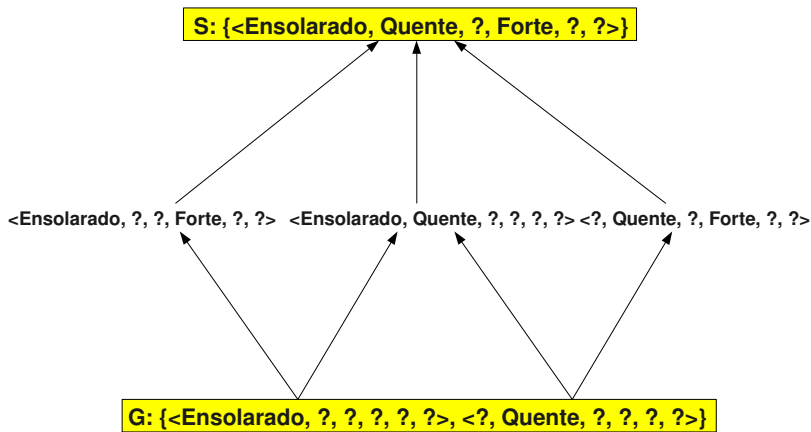
Processor	IPS	Year
Pencil and paper (for comparison)	0.0119 IPS	1892
Intel 8080	640 kIPS at 2 MHz	1974
Intel 286	2.66 MIPS at 12 MHz	1982
Intel 386DX	8.5 MIPS at 25 MHz	1988
Intel 486DX	54 MIPS at 66 MHz	1992
Intel Pentium Pro	541 MIPS at 200 MHz	1996
Intel Pentium III	1,354 MIPS at 500 MHz	1999
Pentium 4 Extreme Edition	9,726 MIPS at 3.2 GHz	2003
Intel Core 2 Extreme QX6700	49,161 MIPS at 2.66 GHz	2006
Intel Core 2 Extreme QX9770	59,455 MIPS at 3.2 GHz	2008
Intel Core i7 Extreme 965EE	76,383 MIPS at 3.2 GHz	2008

Obtido de: http://en.wikipedia.org/wiki/Instructions_per_second

- Mitchell propõe um algoritmo melhorado a partir do List-Then-Eliminate
 - Algoritmo Candidate-Elimination
 - Não requer gerar todas as hipóteses possíveis em **H**
 - Faz busca em duas direções:
 - Apenas mantém as hipóteses mais específicas e as mais genéricas
 - Logo, todas hipóteses entre essas duas, são válidas e contêm diferentes níveis de generalização e especialização
 - Lembre-se: desde que haja ordem parcial entre essas hipóteses

Candidate-Elimination

Candidate-Elimination



- Inicializar a hipótese mais genérica e a mais específica:
 - $G \leftarrow \{<?, ?, ?, ?, ?, ?>\}$
 - $S \leftarrow \{<\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset>\}$
 - Funcionam como limites superior e inferior para espaço de hipóteses válidas
 - Ao invés de produzir todas hipóteses, busca-se pelos limites, dada a ordem parcial de hipóteses
 - Observe que assumimos, neste cenário, uma ordem parcial entre hipóteses!
 - O que ocorre quando não há essa ordem?

Candidate-Elimination

Candidate-Elimination

- Para cada exemplo x de treinamento
 - Se x é um exemplo positivo:
 - Remover de G qualquer hipótese inconsistente com x
 - Para cada hipótese s em S que não é consistente com x
 - Remover s de S
 - Adicionar a S todas generalizações mínimas h de s tal que:
 - h é consistente com x e algum membro de G é mais genérico que h
 - Remover de S qualquer hipótese que é mais genérica que outra em S

- Para cada exemplo x de treinamento
 - Se x é um exemplo negativo:
 - Remover de S qualquer hipótese inconsistente com x
 - Para cada hipótese g em G que não é consistente com x
 - Remover g de G
 - Adicionar a G todas especializações mínimas h de g tal que:
 - h é consistente com x e algum membro de S é mais específico que h
 - Remover de G qualquer hipótese que é menos genérica que outra em G

Candidate-Elimination

Candidate-Elimination

- Para o primeiro e segundo exemplos:

Ex	Céu	Temperatura	Umidade	Vento	Água	Previsão	Gostar de Nadar
1	Ensolarado	Quente	Normal	Forte	Quente	Mesma	Sim
2	Ensolarado	Quente	Alta	Forte	Quente	Mesma	Sim
3	Chuvoso	Fria	Alta	Forte	Quente	Mudar	Não
4	Ensolarado	Quente	Alta	Forte	Fria	Mudar	Sim

$$S_0 \leftarrow \{<\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset>\}$$

$$S_1 \leftarrow \{<\text{Ensolarado}, \text{Quente}, \text{Normal}, \text{Forte}, \text{Quente}, \text{Mesma}>\}$$

$$S_2 \leftarrow \{<\text{Ensolarado}, \text{Quente}, ?, \text{Forte}, \text{Quente}, \text{Mesma}>\}$$

$$G_0, G_1, G_2 : \{<?, ?, ?, ?, ?, ?>\}$$

- Para o terceiro exemplo:

Ex	Céu	Temperatura	Umidade	Vento	Água	Previsão	Gostar de Nadar
1	Ensolarado	Quente	Normal	Forte	Quente	Mesma	Sim
2	Ensolarado	Quente	Alta	Forte	Quente	Mesma	Sim
3	Chuvoso	Fria	Alta	Forte	Quente	Mudar	Não
4	Ensolarado	Quente	Alta	Forte	Fria	Mudar	Sim

$$S_2, S_3 : \{<\text{Ensolarado}, \text{Quente}, ?, \text{Forte}, \text{Quente}, \text{Mesma}>\}$$

$$G_3 : \{<\text{Ensolarado}, ?, ?, ?, ?, ?>, <?, \text{Quente}, ?, ?, ?, ?>, <?, ?, ?, ?, \text{Mesma}>\}$$

$$G_2 : \{<?, ?, ?, ?, ?, ?>\}$$

- Para o quarto exemplo:

Ex	Céu	Temperatura	Umidade	Vento	Água	Previsão	Gostar de Nadar
1	Ensolarado	Quente	Normal	Forte	Quente	Mesma	Sim
2	Ensolarado	Quente	Alta	Forte	Quente	Mesma	Sim
3	Chuvoso	Fria	Alta	Forte	Quente	Mudar	Não
4	Ensolarado	Quente	Alta	Forte	Fria	Mudar	Sim

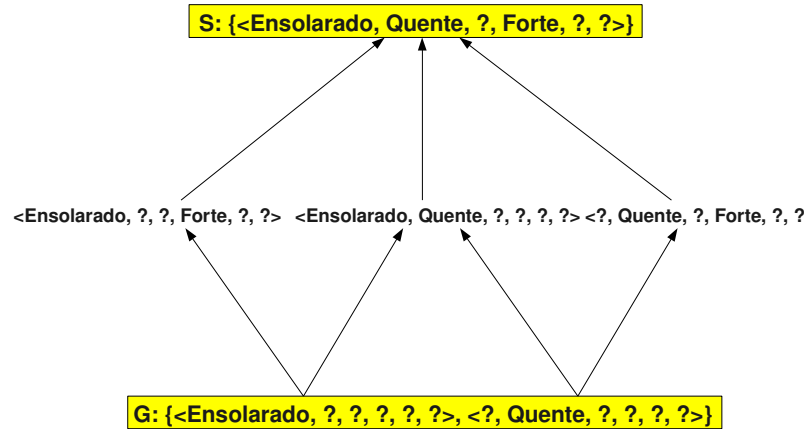
$S_3 : \{<\text{Ensolarado}, \text{Quente}, ?, \text{Forte}, \text{Quente}, \text{Mesma}>\}$

$S_4 : \{<\text{Ensolarado}, \text{Quente}, ?, \text{Forte}, ?, ?>\}$

$G_4 : \{<\text{Ensolarado}, ?, ?, ?, ?, ?>, <?, \text{Quente}, ?, ?, ?, ?>\}$

$G_3 : \{<\text{Ensolarado}, ?, ?, ?, ?, ?>, <?, \text{Quente}, ?, ?, ?, ?>, <?, ?, ?, ?, \text{Mesma}>\}$

Finalmente temos os limites superior e inferior:



Questões

- Questões:
 - Podemos usar todo o espaço entre a hipótese mais generalizada e especializada, inclusive elas mesmas, para avaliar dados desconhecidos
 - Assim poderíamos usar essas hipóteses para “votar” e, portanto, categorizar um exemplo nunca visto
 - Esses “votos” poderiam informar um grau de confiança ou certeza sobre um exemplo nunca visto

Questões

- Questões:
 - Convergência do Candidate-Elimination poderia tendê-lo à uma única hipótese válida
 - Desde que haja elementos suficientes no conjunto de treinamento
 - Desde que o conjunto de treinamento não apresente erros ou ruídos
 - Desde que seja possível obter uma hipótese a partir dos atributos escolhidos para abordar o problema

Questões

- Questões:
 - Conjunto de treinamento deve ser bem balanceado
 - Ou seja, conter um volume aproximadamente similar de diferentes instâncias que desejamos “aprender”
 - Quanto exemplos são suficientes para treinar um algoritmo?
 - Quanto exemplos positivos?
 - Quanto exemplos negativos?
 - O que ocorre se treinarmos apenas com exemplos positivos ou somente negativos?

Questões

- Questões:
 - Considere que há três possíveis opções para Céu e duas para Temperatura, Umidade, Vento, Água e Previsão
 - Logo temos exemplos com as seguintes combinações:
 - $3 * 2 * 2 * 2 * 2 * 2 = 96$
 - Como fica o espaço de hipóteses?
 - Pode apresentar opções como ? e vazio
 - Logo podemos ter mais combinações:
 - $5 * 4 * 4 * 4 * 4 * 4 = 5120$

Ex	Céu	Temperatura	Umidade	Vento	Água	Previsão	Gostar de Nadar
1	Ensolarado	Quente	Normal	Forte	Quente	Mesma	Sim
2	Ensolarado	Quente	Alta	Forte	Quente	Mesma	Sim
3	Chuvoso	Fria	Alta	Forte	Quente	Mudar	Não
4	Ensolarado	Quente	Alta	Forte	Fria	Mudar	Sim

Questões	Questões
<ul style="list-style-type: none"> Questões: <ul style="list-style-type: none"> Esse espaço tem 5120 hipóteses sintaticamente distintas No entanto, qualquer hipótese com atributo vazio é similar, neste cenário temos, de fato: <ul style="list-style-type: none"> $1 + (4 * 3 * 3 * 3 * 3) = 973$ hipóteses distintas Até podemos verificar todas as possíveis e obter as mais adequadas para representar o conjunto de treinamento <ul style="list-style-type: none"> Podemos “votar” com base no conjunto de hipóteses válidas 	<ul style="list-style-type: none"> Mas como fica quando espaço de hipóteses é grande demais? <ul style="list-style-type: none"> Qual o espaço de hipóteses para 1 peça do jogo de Damas? <ul style="list-style-type: none"> Observar que há hipóteses válidas e inválidas! Qual o mesmo espaço para 1 peça do jogo de Xadrez? <ul style="list-style-type: none"> Depende da peça Qual o espaço de hipóteses para o Caixeiro Viajante?

Questões	Questões
<ul style="list-style-type: none"> Questões: <ul style="list-style-type: none"> Para alguns problemas, o espaço de possíveis soluções é infinito <ul style="list-style-type: none"> Por exemplo, busca por um número real entre 0 e 1 que satisfaça exemplos de treinamento 	<ul style="list-style-type: none"> Problemas de aprendizado de máquina e inteligência artificial clássica buscam por boas hipóteses para representar conhecimento, o qual, tipicamente, é dado por um conjunto de treinamento Há diferentes frentes para a busca de hipóteses: <ul style="list-style-type: none"> Utilizamos dados de treinamento para alcançar hipóteses <ul style="list-style-type: none"> Típico em: <ul style="list-style-type: none"> Redes Neurais Artificiais Aprendizado Baseado em Instâncias Não há dados de treinamento, mas sim uma função para avaliar a qualidade de hipóteses que sugerimos <ul style="list-style-type: none"> Típico em: <ul style="list-style-type: none"> Algoritmos Genéticos Inteligência de Enxames

Questões	Referências
<ul style="list-style-type: none"> Exemplificar a representação de hipóteses: <ul style="list-style-type: none"> Árvores de decisão <ul style="list-style-type: none"> Subdivisão do espaço por hiperplanos ortogonais aos eixos Redes Neurais Artificiais <ul style="list-style-type: none"> Subdivisão do espaço por quaisquer hiperplanos lineares Aprendizado Bayesiano <ul style="list-style-type: none"> Construção de hipóteses com base na entropia dos dados Aprendizado Baseado em Instâncias <ul style="list-style-type: none"> Hipóteses definidas por vizinhança do ponto de consulta Computação Evolutiva <ul style="list-style-type: none"> Hipóteses avaliadas por funções de energia Agrupamento de Dados <ul style="list-style-type: none"> Modelo ou hipótese resultante da organização dos dados 	<ul style="list-style-type: none"> Tom Mitchell, Machine Learning, 1997