Conceitos Básicos de Aprendizado de Máquina

Inteligência Artificial – Prof. Flávio Varejão
Programa de Pós-Graduação em Informática
Universidade Federal do Espírito Santo

Introdução

- Em geral, é difícil articular o conhecimento que precisamos para construir um sistema de IA
- Na verdade, algumas vezes, nem temos este conhecimento
- Em alguns casos, podemos construir sistemas capazes de “aprender” o conhecimento necessário

Introdução

- Utilização de técnicas de Aprendizado de Máquina para a construção de sistemas capazes de adquirir conhecimento de forma automática
- Um sistema de aprendizado é um programa que toma decisões baseado em experiências acumuladas por meio da solução bem sucedida de problemas anteriores

Indução

- Dicionário Aurélio

  “Indução é a operação mental que consiste em se estabelecer uma verdade universal ou uma proposição geral com base no conhecimento de certo número de dados singulares ou de proposições de menor generalidade”

Indução

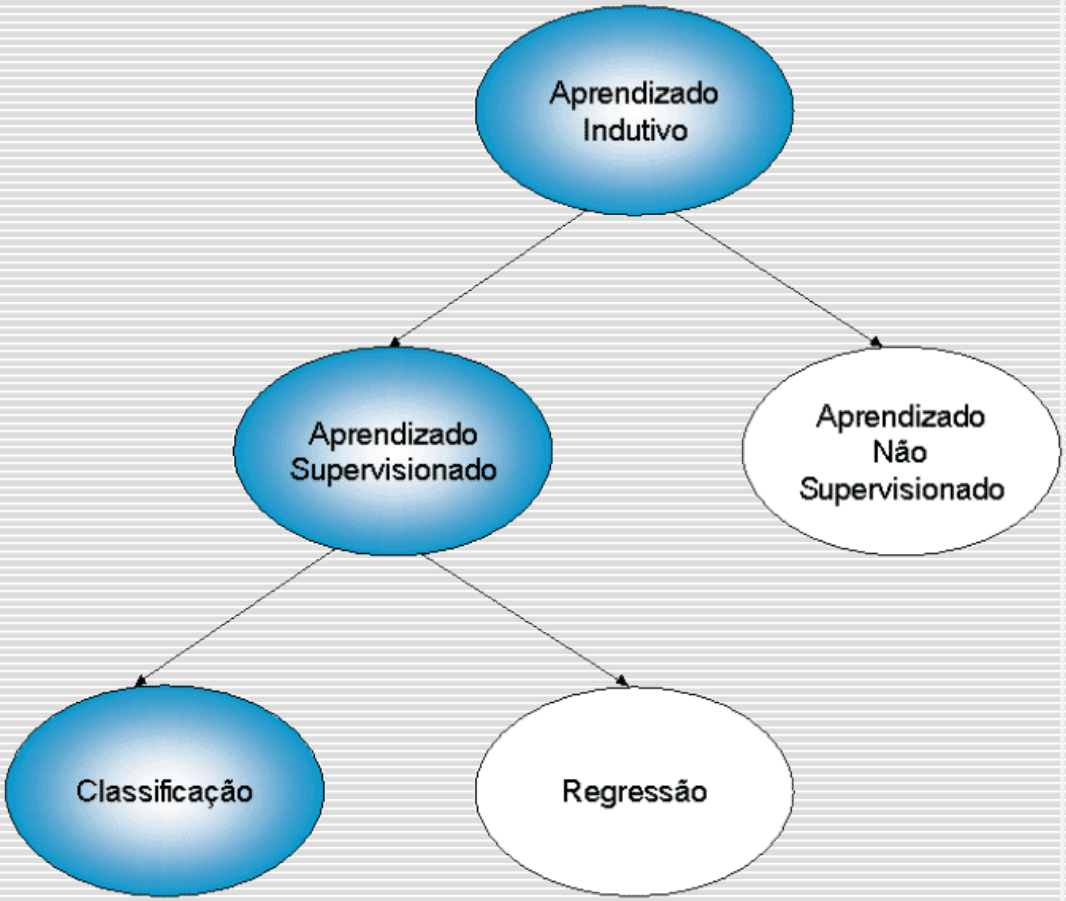
- Forma de inferência lógica que permite obter conclusões genéricas sobre um conjunto particular de exemplos
- Exemplo
  Pedaço de pão 1 foi nutritivo quando o comi
  Pedaço de pão 2 foi nutritivo quando o comi
  …
  Pedaço de pão 100 foi nutritivo quando o comi

  Portanto, todos os pedaços de pão serão nutritivos se eu comê-los

Indução

- Na indução, um conceito é aprendido efetuando-se inferência indutiva sobre os exemplos apresentados
- Portanto, as hipóteses geradas podem ou não representar a realidade

A Hierarquia do Aprendizado



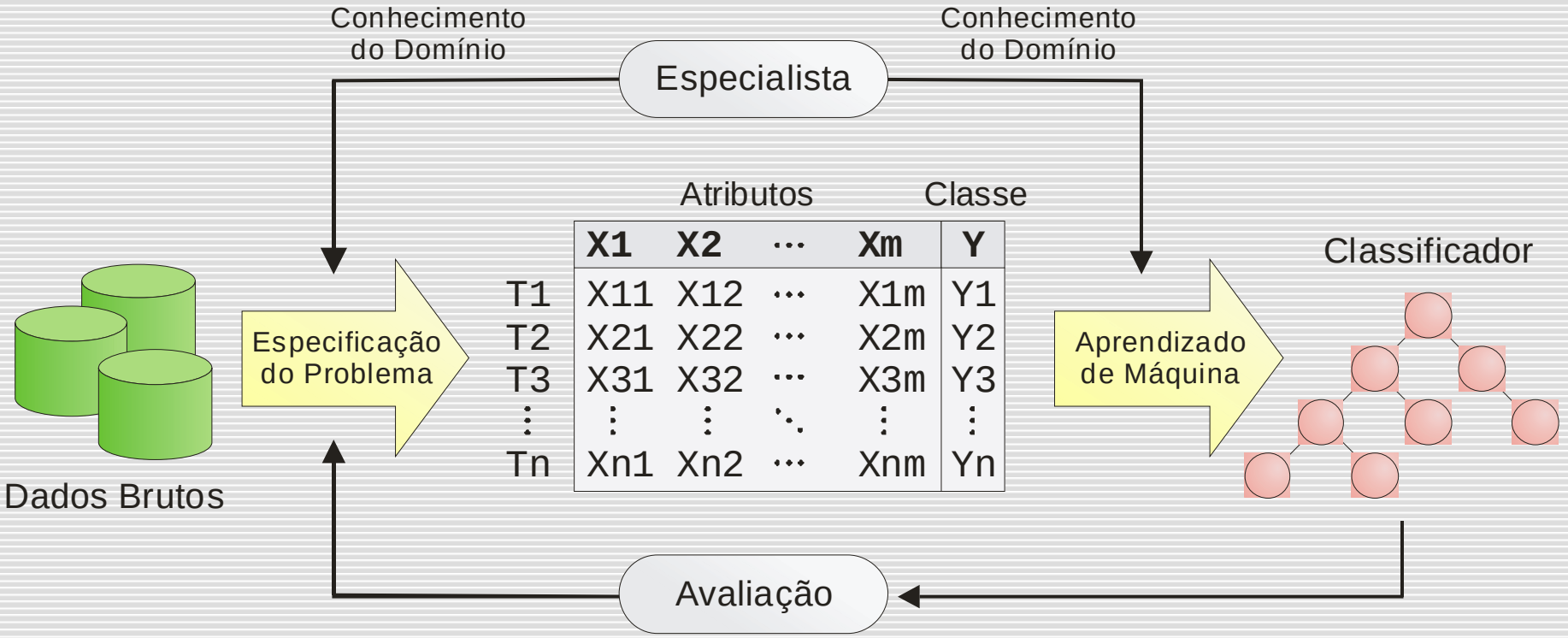
A Hierarquia do Aprendizado

- Aprendizado Supervisionado
  - Dado um conjunto de exemplos rotulados (entrada, saída), encontre uma regra que faça um bom trabalho em prever a saída associada com uma nova entrada
- Aprendizado não Supervisionado
  - Dado um conjunto de exemplos, mas nenhum rótulo associado a eles, identifique relações características entre esses exemplos

Tarefas Comuns

- Aprendizado Supervisionado
  - Classificação
  - Regressão
- Aprendizado não Supervisionado
  - Agrupamento
  - Associação
  - Sumarização

Processo de Classificação



Distribuição de Classes

- Dado um conjunto de exemplos T, para cada classe C_j sua distribuição *distr(C_j)* é calculada como sendo o número de exemplos em T que possuem classe C_j dividido pelo número total de exemplos *n*, ou seja, a proporção de exemplos em cada classe

$$\text{distr}(C_j) = \frac{1}{n} \sum_{i=1}^{n} \| y_i = C_j \|$$

Prevalência de Classes

- Quando existe um grande desbalanceamento das classes no conjunto de exemplos
- Classificador que sempre classifica exemplo na classe majoritária pode ter alta precisão
- Isso pode ser indesejável
  - Detecção de fraudes

Matriz de Confusão

- Mostra o número de classificações corretas versus as classificações preditas para cada classe em um conjunto de exemplos T
- Oferece uma medida efetiva do modelo de classificação

Classe	predita C_1	predita C_2	$\cdots$	predita C_k
verdadeira C_1	$M(C_1, C_1)$	$M(C_1, C_2)$	$\cdots$	$M(C_1, C_k)$
verdadeira C_2	$M(C_2, C_1)$	$M(C_2, C_2)$	$\cdots$	$M(C_2, C_k)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
verdadeira C_k	$M(C_k, C_1)$	$M(C_k, C_2)$	$\cdots$	$M(C_k, C_k)$

$$M(C_i, C_j) = \sum_{\{\forall (x,y) \in T : y = C_i\}} \| \, h(x) = C_j \, \|$$

Matriz de Confusão Binária

Classe	predita C_+	predita C_-	Taxa de erro da classe	Taxa de erro total
	Verdadeiros positivos	Falsos negativos		
verdadeira C_+	T_P	F_N	$\frac{F_N}{T_P+F_N}$	$\frac{F_P+F_N}{n}$
	Falsos positivos	Verdadeiros negativos		
verdadeira C_-	F_P	T_N	$\frac{F_P}{F_P+T_N}$	

Métricas Derivadas de Matriz de Confusão Binária

- Acurácia ou Taxa de Acertos Total

$$\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

- Precisão ou Valor Preditivo Positivo

$$\mathrm{prel}(h) = \frac{T_P}{T_P + F_P}$$

- Revocação ou Sensitividade ou Taxa de Verdadeitos Positivos

$$\mathrm{sens}(h) = \frac{T_P}{T_P + F_N}$$

Métricas Derivadas de Matriz de Confusão Binária

- Especificidade ou Taxa de Verdadeiros Negativos

$$\frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

- Acurácia Balanceada
  - Para bases desbalanceadas
  - Média aritmética de Revocação e Especificidade

Métricas Derivadas de Matriz de Confusão Binária

- F-Measure ou F1-Score

  Média Harmônica: quadrado da geométrica sobre aritmética

$$: 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$\frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

- F-Measure Generalizada
  - Valores comuns
    - $F_{0.5}$

      Maior peso para precisão
    - F_2

      Maior peso para revocação

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

Generalização de Métricas
Matriz de Confusão Multiclasse

Classe	predita C_1	predita C_2	$\cdots$	predita C_k
verdadeira C_1	$M(C_1, C_1)$	$M(C_1, C_2)$	$\cdots$	$M(C_1, C_k)$
verdadeira C_2	$M(C_2, C_1)$	$M(C_2, C_2)$	$\cdots$	$M(C_2, C_k)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
verdadeira C_k	$M(C_k, C_1)$	$M(C_k, C_2)$	$\cdots$	$M(C_k, C_k)$

- Acurácia = soma diagonal / soma de todas células
- Precisão por classe
  - Ex. (C_2) = M(C_2, C_2) / soma (M(C_i, C_2))
- Revocação por classe
  - Ex. (C_2) = M(C_2, C_2) / soma (M(C_2, C_j))
- Fmeasure por classe

Generalização de Métricas
Matriz de Confusão Multiclasse

$$PRE_{macro} = \frac{PRE_1 + \cdots + PRE_k}{k}$$

$$PRE_{micro} = \frac{TP_1 + \cdots + TP_k}{TP_1 + \cdots + TP_k + FP_1 + \cdots + FP_k}$$

- Class A: 1 TP and 1 FP
- Class B: 10 TP and 90 FP
- Class C: 1 TP and 1 FP
- Class D: 1 TP and 1 FP

You can see easily that $Pr_A = Pr_C = Pr_D = 0.5$, whereas $Pr_B = 0.1$.

- A macro-average will then compute: $Pr = \frac{0.5+0.1+0.5+0.5}{4} = 0.4$
- A micro-average will compute: $Pr = \frac{1+10+1+1}{2+100+2+2} = 0.123$

Custos de Erro

- Representa uma penalidade aplicada quando o classificador faz um erro ao rotular exemplos
- Erros considerados mais graves tem maior penalidade

$$\text{err-cost}(h) = \frac{1}{n}\sum_{i=1}^{n} \parallel y_i \neq h(x_i) \parallel \times \text{cost}(y_i, h(x_i))$$

$$\text{err-cost}(h) = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{k} M(C_i, C_j) \times \text{cost}(C_i, C_j)$$

Avaliação de Algoritmos

Avaliação de Algoritmos

- Não existe um único algoritmo que apresente o melhor desempenho para todos os problemas
- É importante utilizar alguma metodologia de avaliação que permita comparar algoritmos
- Permite compreender o poder e a limitação dos diferentes algoritmos

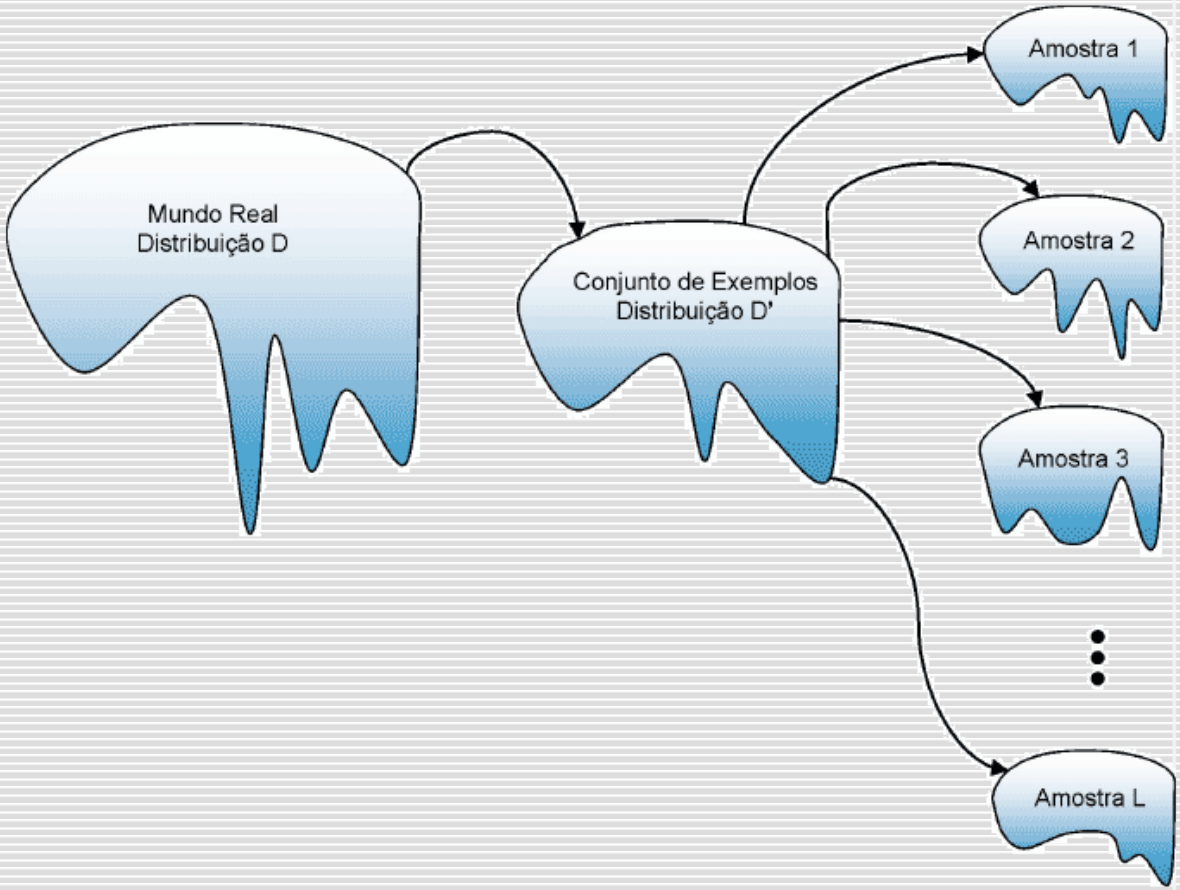
Métodos de Amostragem

- O mundo real apresenta uma distribuição de exemplos D em um dado domínio
  - A distribuição D é desconhecida
- Obtém-se uma distribuição de exemplos D‘ ao se extrair um conjunto de exemplos do mundo real
  - Supostamente D‘ é similar à distribuição D

Métodos de Amostragem

- Estimativa de precisão ou erro de indutores treinados com base na distribuição D'
  - Amostras obtidas de D‘
  - Treina-se um indutor com essas amostras
  - Testa-se seu desempenho em exemplos de D'
- Desta forma, simula-se o processo de amostragem que ocorre no mundo real, assumindo que D' representa o mundo real

Métodos de Amostragem



Amostra – Conjunto de Dados

Exemplo	Atributo 1	Atributo 2	Atributo 3	Classe
1	855	5142	2708	safra 95
2	854	23155	2716	safra 95
3	885	16586	2670	safra 95
4	877	16685	2677	safra 95
5	839	5142	2708	safra 95
6	854	5005	2685	safra 95
7	885	19455	2708	safra 95
8	839	5027	2708	safra 95
9	877	16823	2677	safra 95
10	892	19180	2716	safra 95
11	24628	39437	381	safra 96
12	43183	39277	328	safra 96
13	27871	39712	389	safra 96
14	42329	40307	328	safra 96
15	41627	40032	335	safra 96
16	39399	40322	335	safra 96
17	33677	40375	328	safra 96
18	33539	40078	335	safra 96
19	34150	40353	358	safra 96
20	34485	40742	358	safra 96

Estratégias para Partição de Exemplos

- Qual a melhor estratégia para utilizar esses exemplos tanto para gerar um classificador quanto para estimar seu desempenho?
  - Depende do tamanho da amostra
  - Ideal
    - Separar treinamento de teste
      - Reduz possibilidade de overfitting
    - Várias amostras
      - Baixa variância indica estabilidade

Estratégias para Partição de Exemplos

- Estratégias Mais Comuns
  - Resubstituição
  - Divisão Percentual
    - 2/3 + 1/3
  - Aleatória
  - Validação Cruzada
  - Validação Cruzada Estratificada
  - Exclusão de Um
  - Bootstrap

Resubstituição (Resubstitution)

- Conjunto de treinamento é o mesmo do conjunto de teste
  - Produz estimativa otimista frequentemente causando overfitting
  - Alguns tipos de indutores produzem classificadores que acertam 100% dos exemplos
    - Caso não haja exemplos conflitantes
- Não deve ser usado para avaliação de desempenho
  - Eventualmente pode ser usado em teste de correção dos algoritmos

Divisão Percentual (Holdout)

- Divide o conjunto de dados S em uma porcentagem fixa de exemplos *p* para o treinamento e (*1 - p*) para teste
  - Valor típico
    - 2/3 treinamento
    - 1/3 teste
  - Aceitável *p* = 75%
    - Treinamento com 75% dos dados
    - Teste com 25% dos dados
      - Restantes

Divisão Percentual (Treinamento)

Exemplo	Atributo 1	Atributo 2	Atributo 3	Classe
4	877	16685	2677	safra 95
7	885	19455	2708	safra 95
8	839	5027	2708	safra 95
2	854	23155	2716	safra 95
10	892	19180	2716	safra 95
1	855	5142	2708	safra 95
6	854	5005	2685	safra 95
18	33539	40078	335	safra 96
15	41627	40032	335	safra 96
19	34150	40353	358	safra 96
12	43183	39277	328	safra 96
17	33677	40375	328	safra 96
20	34485	40742	358	safra 96
11	24628	39437	381	safra 96

Divisão Percentual (Teste)

Exemplo	Atributo 1	Atributo 2	Atributo 3	Classe
3	885	16586	2670	safra 95
5	839	5142	2708	safra 95
9	877	16823	2677	safra 95
13	27871	39712	389	safra 96
14	42329	40307	328	safra 96
16	39399	40322	335	safra 96

Amostragem Aleatória

- O conjunto de dados S disponível é dividido aleatoriamente em um conjunto de treinamento R e em um conjunto de teste T
- Esse processo é repetido *k* vezes
  - $k << n$
  - *n* é o número total de exemplos em S
- Resultado é a média dos resultados obtidos em cada rodada
  - Importante considerar o desvio padrão ou variância

Amostragem Aleatória

- K diferentes hipóteses geradas
  - Uma para cada base de treinamento
  - Mesmos parâmetros mas exemplos diferentes
- Hipótese final deve ser escolhida dentre aqueles cujo
  - Resultado esteja no intervalo
    - Resultado médio ± desvio padrão

Validação Cruzada (Cross Validation)

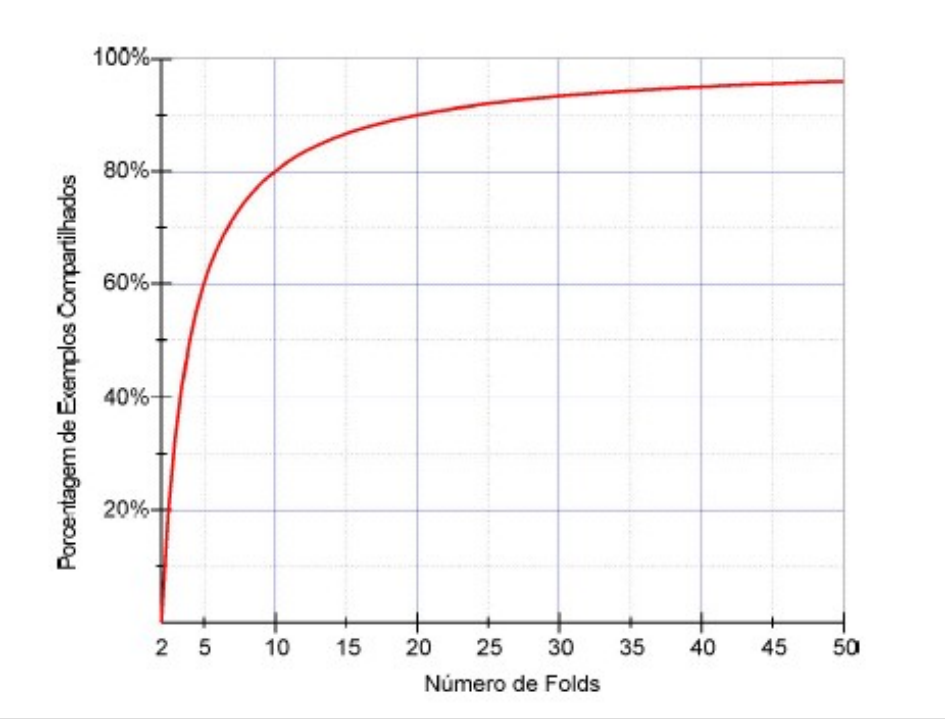
- Similar ao método de amostragem aleatória
- k divisões aleatórias do conjunto S em conjuntos de treinamento e teste
- S é dividido em *k* conjuntos disjuntos de tamanhos iguais
  - T_1 , T_2 , ..., T_k
  - Algoritmo executado *k* vezes
    - Em cada execução o conjunto de teste é T_i e conjunto de treinamento é a união dos outros T_j (j≠i)
  - Resultado médio das execuções

Validação Cruzada

- Divide-se a amostra em k partes
- Realiza-se k etapas de geração e validação
- Em cada etapa
  - 1 parte para validação
  - $k - 1$ partes para geração do classificador
- Vantagem
  - Mais exemplos para a etapa de validação
- Desvantagem
  - Distribuição de classes nas partes pode não corresponder a distribuição completa

Validação Cruzada

- Número de exemplos compartilhados em função do número de partes



Validação Cruzada em 10 Partes

Partições	Exemplo	Atributo 1	Atributo 2	Atributo 3	Classe
Fold 1	4	877	16685	2677	safra 95
	18	33539	40078	335	safra 96
Fold 2	9	877	16823	2677	safra 95
	11	24628	39437	381	safra 96
Fold 3	1	855	5142	2708	safra 95
	14	42329	40307	328	safra 96
Fold 4	3	885	16586	2670	safra 95
	20	34485	40742	358	safra 96
Fold 5	7	885	19455	2708	safra 95
	15	41627	40032	335	safra 96
Fold 6	10	892	19180	2716	safra 95
	19	34150	40353	358	safra 96
Fold 7	6	854	5005	2685	safra 95
	17	33677	40375	328	safra 96
Fold 8	2	854	23155	2716	safra 95
	12	43183	39277	328	safra 96
Fold 9	5	839	5142	2708	safra 95
	16	39399	40322	335	safra 96
Fold 10	8	839	5027	2708	safra 95
	13	27871	39712	389	safra 96

Validação Cruzada em 10 Partes (versão 1)

Partições	Exemplo	Atributo 1	Atributo 2	Atributo 3	Classe
Fold 1	4	877	16685	2677	safra 95
	18	33539	40078	335	safra 96
Fold 2	9	877	16823	2677	safra 95
	11	24628	39437	381	safra 96
Fold 3	1	855	5142	2708	safra 95
	14	42329	40307	328	safra 96
Fold 4	3	885	16586	2670	safra 95
	20	34485	40742	358	safra 96
Fold 5	7	885	19455	2708	safra 95
	15	41627	40032	335	safra 96
Fold 6	10	892	19180	2716	safra 95
	19	34150	40353	358	safra 96
Fold 7	6	854	5005	2685	safra 95
	17	33677	40375	328	safra 96
Fold 8	2	854	23155	2716	safra 95
	12	43183	39277	328	safra 96
Fold 9	5	839	5142	2708	safra 95
	16	39399	40322	335	safra 96

Conjunto de Treinamento

Partições	Exemplo	Atributo 1	Atributo 2	Atributo 3	Classe
	4	877	16685	2677	safra 95

Conjunto de Teste

Validação Cruzada em 10 Partes (versão 2)

Partições	Exemplo	Atributo 1	Atributo 2	Atributo 3	Classe
Fold 1	4	877	16685	2677	safra 95
	18	33539	40078	335	safra 96
Fold 2	9	877	16823	2677	safra 95
	11	24628	39437	381	safra 96
Fold 3	1	855	5142	2708	safra 95
	14	42329	40307	328	safra 96
Fold 4	3	885	16586	2670	safra 95
	20	34485	40742	358	safra 96
Fold 5	7	885	19455	2708	safra 95
	15	41627	40032	335	safra 96
Fold 6	10	892	19180	2716	safra 95
	19	34150	40353	358	safra 96
Fold 7	6	854	5005	2685	safra 95
	17	33677	40375	328	safra 96
Fold 8	2	854	23155	2716	safra 95
	12	43183	39277	328	safra 96
Fold 9	5	839	5142	2708	safra 95
	16	39399	40322	335	safra 96

Conjunto de Treinamento

Fold 9	5	839	5142	2708	safra 95
	16	39399	40322	335	safra 96

Conjunto de Teste

Validação Cruzada Estratificada

- Caso especial da validação cruzada
- Mantém-se a distribuição de classes originais em cada parte

Um Excluído (Leave-One-Out)

- Caso especial da validação cruzada em que *k* é igual |S|
  - Por exemplo, no caso do nosso exemplo a amostra tem 20 padrões, então *k*=20
  - Cada conjunto de treinamento será formado por 19 padrões e o conjunto de teste por apenas um único padrão
  - Assim, o processo todo será repetido 20 vezes
  - E se o conjunto de dados tiver 500 padrões?

Um Excluído

- A amostra é composta por m exemplos,

$$\mathcal{T} = \left\{ \left(\vec{x}_i, f(\vec{x}_i) \right) \right\}_{i=1}^{m}$$

- Realiza-se m iterações de geração/validação
- Na primeira iteração
  - gera-se um classificador com os exemplos x2, ..., xm
  - valida-o com o exemplo x1
- Numa i-ésima iteração
  - gera-se um classificador com os exemplos x1, , xi−1, xi+1..., xm
  - valida-o com o exemplo xi
- Na última iteração
  - gera-se o classificador com os exemplos x1, x,..., xm−1
  - valida-o com o exemplo xm

Bootstrap

- Baseado em procedimento estatístico de amostragem com reposição
  - Nos métodos anteriores, sempre que uma amostra foi retirada de um conjunto de dados para formar o conjunto de treinamento ou de teste, isso foi feito sem reposição
  - Uma vez selecionado, ele não poderia ser usado novamente

Bootstrap

- A idéia do *bootstrap* é formar um conjunto de treinamento a partir do conjunto de dados usando reposição
- Um conjunto de dados S com *n* padrões é amostrado *n* vezes, com reposição, a fim de formar um outro conjunto de dados com *n* padrões

Bootstrap

- Alguns padrões no conjunto de dados gerado serão repetidos (quase certamente)
- Haverá padrões no conjunto de dados original que não foram escolhidos
  - Esses padrões serão usados para formar o conjunto de teste
- Este processo é repetido várias vezes

Quadro Comparativo – Estratégias de Partição

	holdout	aleatória	*leave-one-out*	*r-fold cv*	*r-fold strat cv*	*bootstrap*
Treinamento	pn	t	$n-1$	$n(r-1)/r$	$n(r-1)/r$	n
Teste	$(1-p)n$	$n-t$	1	n/r	n/r	$n-t$
Iterações	1	$L \ll n$	n	r	r	$\simeq 200$
Reposição	não	não	não	não	não	sim
Prevalência de Classe	não	não	não	não	sim	sim/não

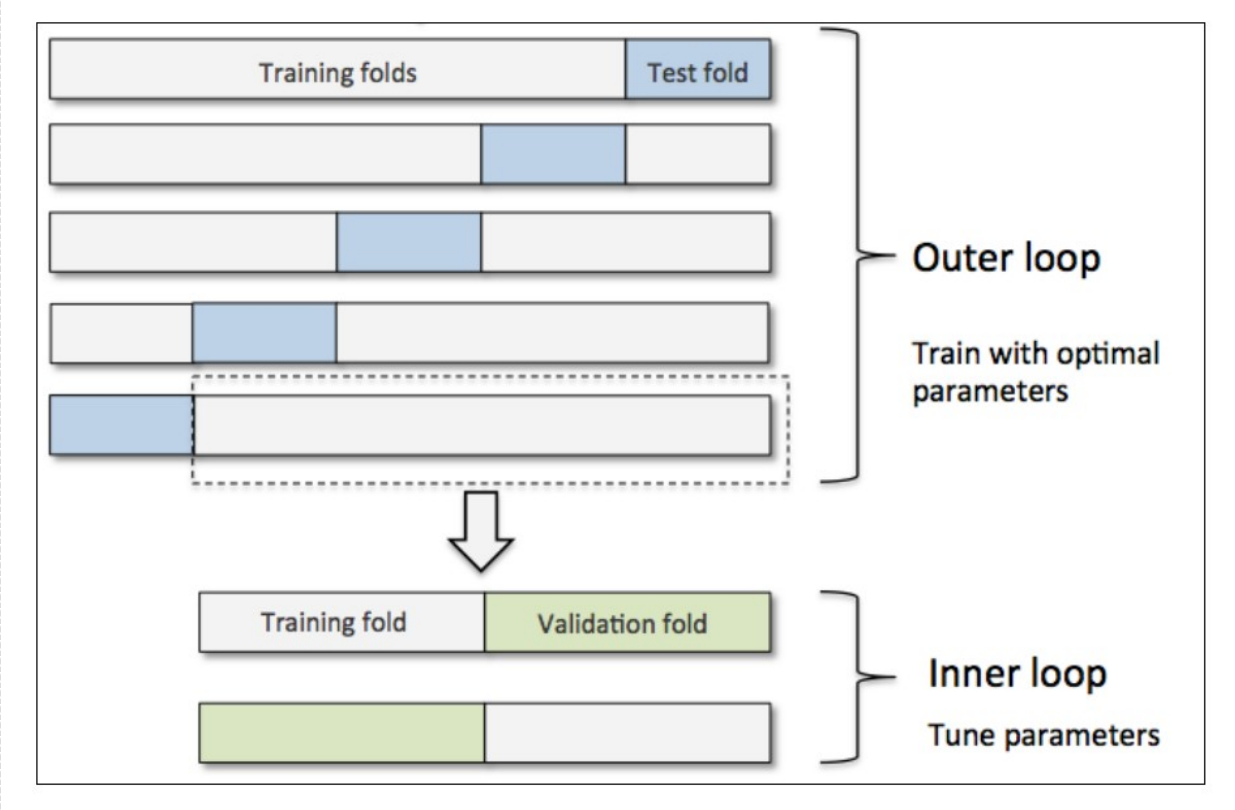
O Problema do Superajuste

- Métodos possuem hiperparâmetros
- Busca por valores de parâmetros que geram melhores resultados no conjunto de teste
  - Uso de busca em grade (grid search) ou metaheurística para encontrar melhores valores
    - De certo modo, se vê o conjunto de teste
    - Gera resultados otimistas quando aplicados em situação desconhecida

Contornando o Superajuste

- Conjunto de validação como parte do treinamento para definir valores dos hiperparâmetros
- Busca por valores dos hiperparâmetros
  - Métodos de Busca
    - Busca em grade (grid search)
    - Metaheurística
  - Método de Amostragem
    - Pode ser qualquer um
    - Validação cruzada é mais recomendado

Superajuste – Ciclos Aninhados



Ajuste de Hiperparâmetros

- Busca no Ciclo Interno
  - Busca em grade (grid search)
  - Metaheurística
- Avaliação (Função Objetivo)
  - Fixa Valor dos Hiperparâmetros
  - Executa Validação Cruzada
  - Média (+/- Desvio)

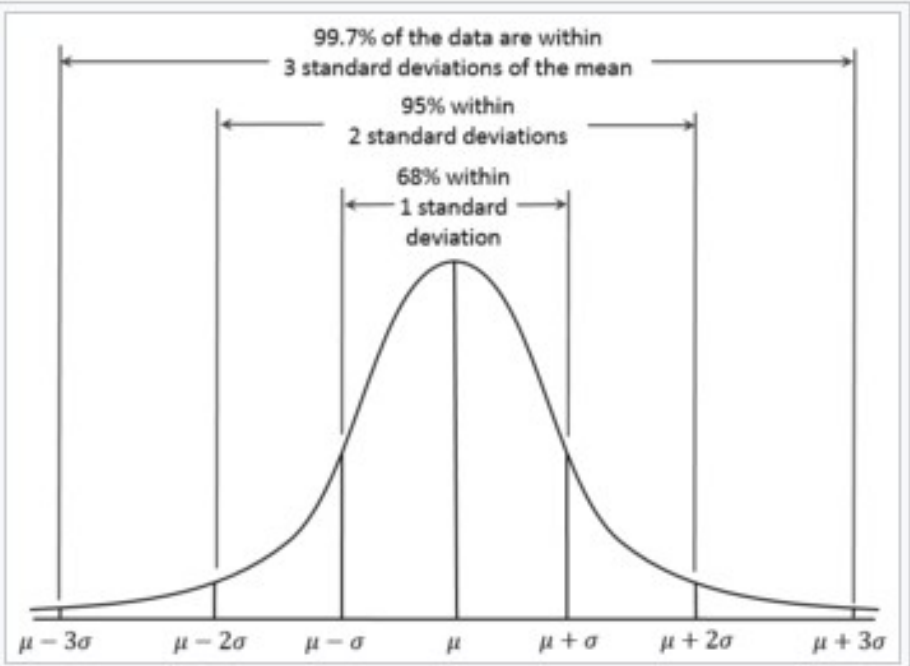
Avaliação de Desempenho

- Validação Cruzada no Ciclo Externo
  - Todo conjunto de treino
  - Valores ótimos dos hiperparâmetros
  - Média (+/- Desvio)
  - Intervalos de Confiança
  - Boxplot

Confiança nos Resultados

- Estimativas são mais confiáveis com amostras maiores (conjunto de teste T)
  - Acerto de 75% em T com 100 exemplos
  - Acerto de 75% em T com 1000 exemplos
- Há diferença nos intervalos de confiança
  - Com 80% de grau de confiança
    - [69.1%-80.1%] para amostra com 100 exemplos
    - [73.2%-76.7%] para amostra com 1000 exemplos

Distribuição Normal



Intervalos de Confiança para Classificadores

$$IC(erro_X(\hat{f}), N\%) = erro_{\mathcal{T}}(\hat{f}) \pm Z_N \sqrt{\frac{erro_{\mathcal{T}}(\hat{f}).(1 - erro_{\mathcal{T}}(\hat{f}))}{|\mathcal{T}|}}$$

Nível de confiança $N\%$:	50%	68%	90%	95%	98%	99%
Constante Z_N:	0,67	1,00	1,64	1,96	2,33	2,58

Intervalos de Confiança para Classificadores

- Com grau de confiança de aproximadamente 95% o erro verdadeiro erro(f) estará no intervalo

$$erro_{\mathcal{T}}(\hat{f}) \pm 1,96\sqrt{\frac{erro_{\mathcal{T}}(\hat{f}).(1-erro_{\mathcal{T}}(\hat{f}))}{|\mathcal{T}|}}$$

Um Exemplo

- |T| = 40
- f comete 12 erros

- erro(f) = 12/40 = 0,30
- IC(erro(f), 95%) = 0,30 ± 1,96 . 0,07
  = 0,30 ± 0,14

Intervalos de Confiança para Classificadores

- Função da Média e Desvio Padrão

$$IC(erro_X(\hat{f}), N\%) = erro_{\mathcal{T}}(\hat{f}) \pm Z_N \sqrt{\frac{erro_{\mathcal{T}}(\hat{f}).(1 - erro_{\mathcal{T}}(\hat{f}))}{|\mathcal{T}|}}$$

$$IC(erro_X(\hat{f}), N\%) = erro_{\mathcal{T}}(\hat{f}) \pm Z_N \sqrt{\frac{1/4}{|\mathcal{T}|}}$$

$$IC(erro_X(\hat{f}), N\%) = \mu \pm Z_N.\sigma$$

Boxplot

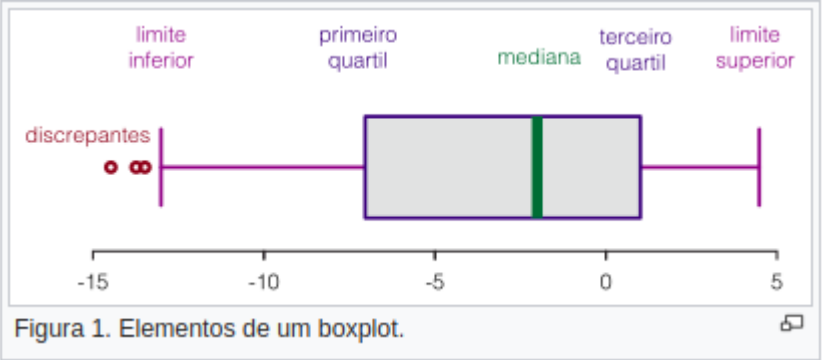
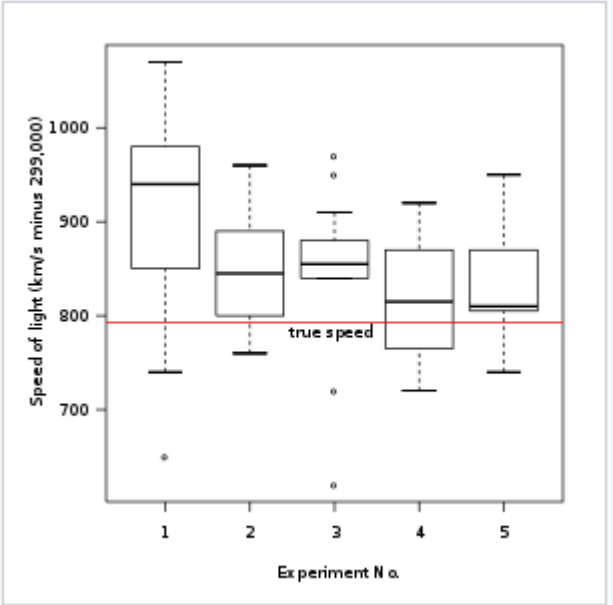


Figura 1. Elementos de um boxplot.



Comparação de Desempenho de Classificadores

- Comparação de taxas de erro não é suficiente
- Diferença de taxas pode não ter significado
- Pode ser dependente da amostra escolhida
  - Tamanho da amostra influencia
  - Iterações reduzem dependência mas aumentam a variância

Comparação de Desempenho de Classificadores

- Uma estratégia simples e comum
  - Estratégia de Partição com várias iterações
    - Normalmente validação cruzada
  - Uso de média, variância e desvio padrão

$$\text{mean}(A) = \frac{1}{r}\sum_{i=1}^{r}\text{err}(h_i)$$

$$\text{var}(A) = \frac{1}{r}\left[\frac{1}{r-1}\sum_{i=1}^{r}(\text{err}(h_i)-\text{mean}(A))^2\right]$$

$$\text{sd}(A) = \sqrt{\text{var}(A)}$$

Comparação de Desempenho de Classificadores

- Comparação de dois classificadores (A_s e A_p)
- Assume distribuição normal
  - Grau de confiança pré-estabelecido em 95%
  - Cálculo de diferença absoluta em desvios padrões (ad)

$$\text{mean}(A_S - A_P) = \text{mean}(A_S) - \text{mean}(A_P)$$

$$\text{sd}(A_S - A_P) = \sqrt{\frac{\text{sd}(A_S)^2 + \text{sd}(A_P)^2}{2}}$$

$$\text{ad}(A_S - A_P) = \frac{\text{mean}(A_S - A_P)}{\text{sd}(A_S - A_P)}$$

Comparação de Desempenho de Classificadores

- ad (A_s - A_p) > 0 => A_p supera A_s
- ad (A_s - A_p) >= 2 => A_p supera A_s com grau de confianca de 95%
- ad (A_s - A_p) <= 0 => A_s supera A_p
- ad (A_s - A_p) <= -2 => A_p supera A_s com grau de confianca de 95%

Projeto e Avaliação de Testes Estatísticos

- Uma das principais etapas é a identificação de fontes de variações que precisam ser controladas em cada teste
- Há quatro fontes principais de variações
  - Variação na seleção do conjunto de teste
  - Variação na seleção do conjunto de treinamento
  - Variação interna no algoritmo de aprendizado
  - Erro na rotulação de exemplos

Projeto e Avaliação de Testes Estatísticos

- Variação interna no algoritmo de aprendizado
  - Algoritmo *backpropagation* de redes neurais é geralmente inicializado com um conjunto de pesos aleatórios que são iterativamente ajustados (melhorados)
  - A rede neural resultante (treinada) depende criticamente dessa inicialização aleatória dos pesos
  - Neste caso, mesmo que o conjunto de treinamento não seja mudado, o algoritmo provavelmente irá produzir um conjuntos de pesos finais diferentes se for executado de novo a partir de uma nova inicialização aleatória

Projeto e Avaliação de Testes Estatísticos

- Erro na rotulação de exemplos
  - Se uma fração fixa η dos exemplos de teste estão rotulados aleatoriamente, então nenhum algoritmo de aprendizado conseguirá um taxa de erro menor que η

Projeto e Avaliação de Testes Estatísticos

- Um bom teste estatístico não deverá ser “enganado” por essas fontes de variação
- Um teste deverá concluir que dois algoritmos são diferentes se as suas porcentagens de classificações corretas sejam diferentes, em média, quando treinados com um conjunto de treinamento de um determinado tamanho fixo e avaliado com todos os exemplos da população

Projeto e Avaliação de Testes Estatísticos

- A fim de evitar as variações conseqüentes da escolha dos dados de teste e da possibilidade de erros aleatórios de classificação, o procedimento estatístico deve levar em consideração tanto o tamanho do conjunto de teste quanto as conseqüências de mudanças neste conjunto
- A fim de evitar as variações conseqüentes da escolha do conjunto de treinamento e a aleatoriedade interna, o procedimento estatístico deve executar o algoritmo de aprendizado várias vezes e medir a variação na precisão dos classificadores resultantes

Comparação de Desempenho de Classificadores

- Teste de Hipótese
  - Hipótese Nula: o desempenho de A é igual ao desempenho de B ($M_A = M_B$)
  - Hipótese Alternativa: o desempenho de A é diferente do desempenho de B ($M_A \neq M_B$)
- Vários testes estatísticos
  - Friedman
  - Wilcoxon Signed-Ranks Test
  - Nemenyi
  - Teste t Pareado

Comparação de Desempenho de Classificadores

- Teste *t* Pareado
  - Testa se o desempenho médio (e.g., taxa de erro) de dois algoritmos de aprendizado A e B é diferente para uma dada tarefa
  - Para testar essa hipótese, o pesquisador estabelece 5% como **nível de significância** (α), ou seja, ele rejeitará a hipótese nula SOMENTE se a probabilidade for menor ou igual 0.05 de que a diferença média amostral resulte de erro de amostragem (e.g., variabilidade do conjunto de teste)

Teste *t* Pareado

- Depois de ter estabelecido esse critério de significância, ele treina cada algoritmo com o k-*fold cross-validation* (gerado a partir de uma amostra fixa S)

	Algoritmo A	Algoritmo B
Fold1	1	3
Fold2	2	5
Fold3	1	5
Fold4	1	5
Fold5	2	4
Fold6	1	5

Erro no conjunto de teste

Etapas (1/4)

- Passo 1
  - Achar a média de cada amostra
    - $M_A=1,33$
    - $M_B=4,50$
- Passo 2
  - Achar o desvio-padrão de cada amostra
    - $s_A=0,48$
    - $s_B=0,76$

Etapas (2/4)

- Passo 3
  - Achar o erro padrão de cada média
    - $\sigma_{MA} = s_A/\text{sqrt}(5) = 0,21$
    - $\sigma_{MB} = s_B/\text{sqrt}(5) = 0,34$
- Passo 4
  - Achar o erro padrão da diferença
    - $\sigma_{dif} = \text{sqrt}(\sigma_{MA}^2 + \sigma_{MB}^2) = 0,40$

Etapas (3/4)

- Passo 5
  - Traduzir a diferença média amostral em unidades de erro padrão da diferença
    - $t = (M_A - M_B)/\sigma_{dif} = -7,93$
- Passo 6
  - Achar o grau de liberdade
    - $gl = N_A + N_B - 2 = 6 + 6 - 2 = 10$

Etapas (4/4)

- Passo 7
  - Comparar *t* obtida com o t adequado
    - t_O = t observado (obtido) = 7,93
    - t_c = t crítico (tabelado) = 2,228
    - gl = 10
    - α = 0,05

Conclusões do Teste t Pareado

- Para rejeitar a hipótese nula ao nível de significância de 0,05 com 10 graus de liberdade, o t observado (t_o) deve ser igual ou maior que 2,228
- No caso em foco, t=7,93
  - Rejeita-se a hipótese nula em favor da hipótese alternativa
  - A taxa de erro médio do algoritmo A é diferente do algoritmo B
  - Mais especificamente, o algoritmo B (4,50) produz significativamente mais erros do que o algoritmo A (1,33)

Conclusões do Teste t Pareado

- Caso o t observado fosse menor que 2,228 então não poder-se-ia rejeitar a hipótese nula
  - Não haveria evidências para afirmar que o desempenho do algoritmo A é significativamente diferente do algoritmo B

Levando em Conta Graus de Certeza

- Medida de Desempenho Usada
  - Taxa de Erro
  - Também chamada função de erro 0-1

$$\sum_i \begin{cases} 0 \text{ if prediction is correct} \\ 1 \text{ if prediction is incorrect} \end{cases}$$

- Boa parte dos classificadores produz graus de certeza para as classes
  - Pode-se querer levar em conta essas probabilidades
  - Classe predita com 99% de certeza deve ser melhor avaliada do que uma com 51%

Levando em Conta os Custos

- Diferentes tipos de erros de classificação implicam em custos diferentes
  - Detecção de fraudes
    - Custo de classificar normais como fraudadores é menor do que o de classificar fraudadores como normais
- Existem vários outros tipos de custo
  - Custo de coletar dados para treinamento

Considerações Finais

- Deve-se analisar desempenho de classificadores por uma abordagem estatística
- Há diversas estratégias para geração e validação de classificadores
- Desafios
  - Escolha da estratégia apropriada
  - Novas estratégias

Conceitos Básicos de Aprendizado de Máquina

Inteligência Artificial – Prof. Flávio Varejão

Programa de Pós-Graduação em Informática

Universidade Federal do Espírito Santo