



SCC-630 - Capítulo 10

Métodos de Amostragem e Avaliação de Algoritmos

João Luís Garcia Rosa¹

¹Departamento de Ciências de Computação
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - São Carlos
joaoluis@icmc.usp.br

2011

Agradecimento

Agradeço à Profa. Maria Carolina Monard, que gentilmente permitiu que eu usasse seus slides [1] para preparação deste capítulo.

Sumário

1 Métodos de Amostragem e Avaliação de Algoritmos

Material do Prof. José Augusto Baranauskas

Os próximos 42 slides contêm material do Prof. José Augusto Baranauskas, do Departamento de Física e Matemática - FFCLRP-USP, com atualização da Profa. Maria Carolina Monard.



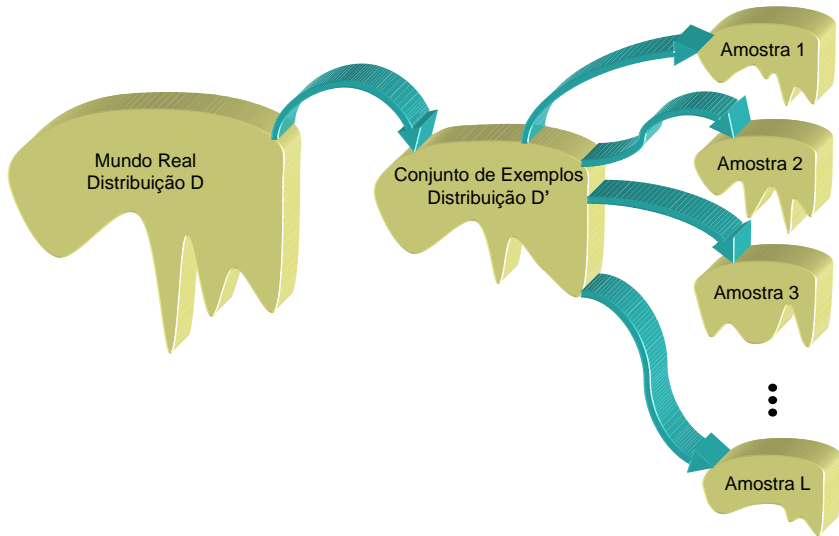
Métodos de Amostragem e Avaliação de Algoritmos

- ❑ AM é uma ferramenta poderosa, mas não existe um único algoritmo que apresente o melhor desempenho para todos os problemas
- ❑ Assim, é importante compreender o poder e a limitação dos diferentes algoritmos utilizando alguma metodologia de avaliação que permita comparar algoritmos
- ❑ Veremos uma metodologia de avaliação, freqüentemente utilizada pela comunidade de AM, para comparar dois algoritmos, a qual se baseia na idéia de amostragem (*resampling*)

Métodos de Amostragem

- ❑ O classificador por si só não fornece uma boa estimativa de sua capacidade de previsão (ele possui boa capacidade de **descrever** os dados, não de **prever**)
- ❑ Uma vez que o classificador conhece todos os dados é inevitável que super-estime sua capacidade de previsão
 - Por exemplo, a taxa de erro será super-otimista (abaixo da taxa de erro verdadeira) e não é raro obter 100% de precisão no conjunto de treinamento
- ❑ Assim, dados um conjunto de exemplos de tamanho finito e um indutor, é importante **estimar** o desempenho futuro do classificador induzido utilizando o conjunto de exemplos
- ❑ Todos os métodos não paramétricos descritos a seguir, exceto pelo método de resubstituição, estão baseados na idéia de **amostragem**

Métodos de Amostragem



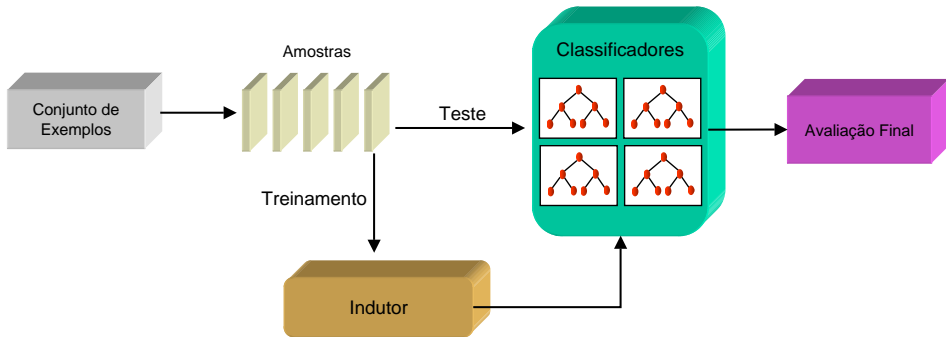
Métodos de Amostragem

- ❑ O mundo real apresenta uma distribuição de exemplos D em um dado domínio, a qual é desconhecida
- ❑ Ao extrair exemplos do mundo real, formando assim um conjunto de exemplos, obtém-se uma distribuição de exemplos D' , a qual é, supostamente, similar à distribuição D
- ❑ De modo a estimar uma medida, geralmente a precisão ou o erro, de indutores treinados com base na distribuição D' , extraem-se amostras a partir de D' , treina-se um indutor com essas amostras e testa-se seu desempenho em exemplos de D' (normalmente com exemplos fora da amostra utilizada para treinamento)
- ❑ Desta forma, simula-se o processo de amostragem que ocorre no mundo real, assumindo que D' representa o mundo real

Métodos de Amostragem

- ❑ É importante, ao estimar uma medida verdadeira (por exemplo, o erro verdadeiro), que a amostra seja **aleatória**, isto é, os exemplos não devem ser pré-selecionados
- ❑ Para problemas reais, normalmente é tomada uma amostra de tamanho **n** e o objetivo consiste em estimar uma medida para aquela população em particular (não para todas as populações)
- ❑ Alguns métodos para estimar medidas são descritos a seguir

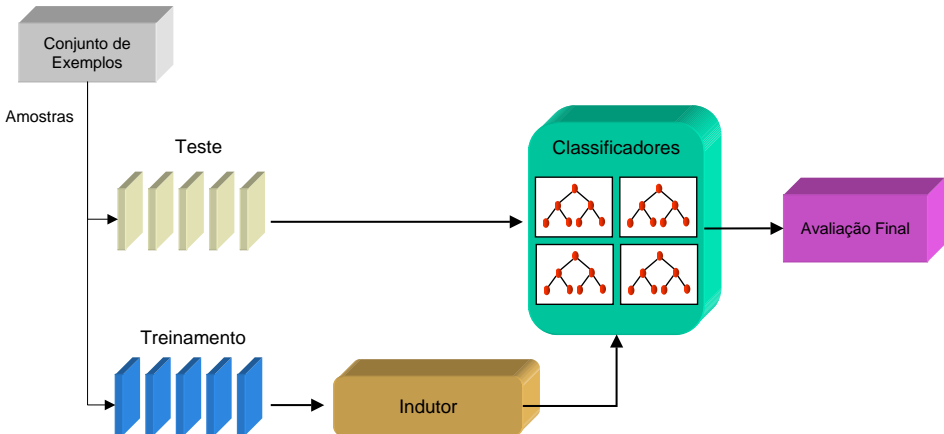
Métodos de Amostragem (Resubstituição)



Resubstituição

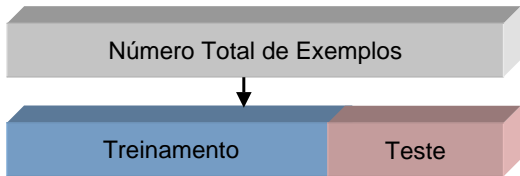
- ❑ Este método consiste em construir o classificador e testar seu desempenho no mesmo conjunto de exemplos, ou seja, o conjunto de teste é idêntico ao conjunto de treinamento
- ❑ Este estimador fornece uma **medida aparente**, possuindo uma estimativa altamente otimista da precisão, devido ao fato de que o processo de classificação tenta maximizá-la
- ❑ Para muitos algoritmos de indução que classificam corretamente todos os exemplos, tais como **1-Nearest Neighbors** ou árvores de decisão sem poda, esta estimativa é muito otimista: se não houver exemplos conflitantes, a estimativa de precisão atinge 100%
- ❑ Assim sendo, o desempenho calculado com este método possui um *bias* otimista, ou seja, o bom desempenho no conjunto de treinamento em geral não se estende a conjuntos independentes de teste
- ❑ Quando o *bias* do estimador de resubstituição foi descoberto, diversos métodos de **cross-validation** (validação cruzada) foram propostos, os quais são descritos a seguir
- ❑ Todos eles estão baseados no mesmo princípio: **não deve haver exemplos em comum entre o conjunto de treinamento (ou aprendizado) e o conjunto de teste**

Métodos de Amostragem (Exceto Resubstituição)



Holdout

- ❑ Este estimador divide os exemplos em uma porcentagem fixa de exemplos q para treinamento e $(1-q)$ para teste, considerando normalmente $q > 1/2$
- ❑ Valores típicos são $q = 2/3$ e $(1-q) = 1/3$, embora não existam fundamentos teóricos sobre estes valores



Holdout

- Entretanto, quando o conjunto de exemplos disponível é pequeno de exemplos, nem sempre é possível separar uma parte dos exemplos para utilizar como exemplos de teste

Métodos de Treinar-e-Testar

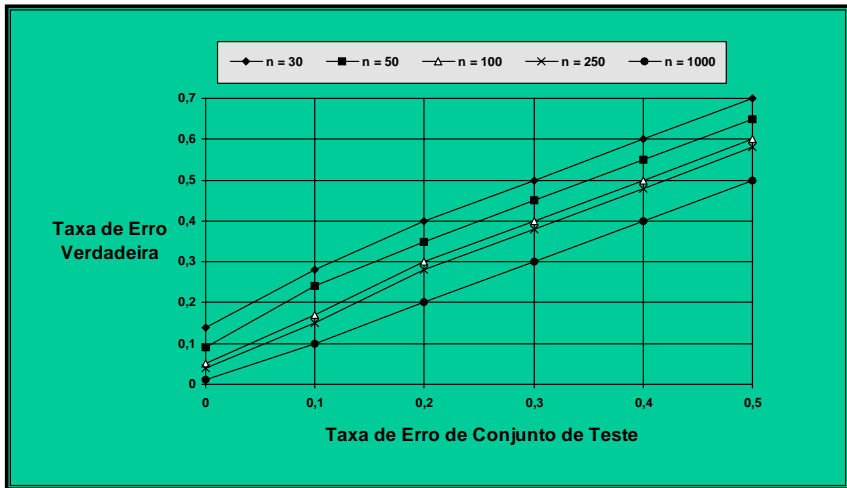
Quantos casos de teste são necessários para uma estimativa precisa?

Quantos casos deve conter cada conjunto de treinamento e teste?

Número de Casos de Teste e Qualidade da Predição

A seguir é mostrada a relação entre a taxa de erro do conjunto de teste e a mais alta possível taxa de erro verdadeira para conjuntos de teste de vários tamanhos. Esses valores têm 95% de confiabilidade. Isso significa que a probabilidade da taxa de erro ser maior que os valores apresentados no gráfico não excede 5%.

Número de Casos de Teste e Qualidade da Predição



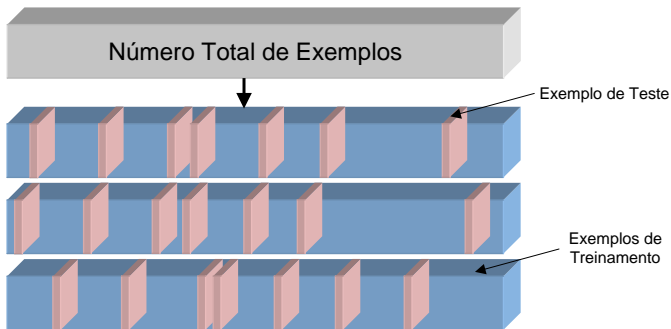
Número de Casos de Teste e Qualidade da Predição

Por exemplo, para um conjunto de teste de 50 exemplos, se a taxa de erro no conjunto de teste for 0%, há uma alta probabilidade (95%) que a taxa de erro verdadeira seja no máximo 10%.

Se isso acontecer com 1000 exemplos de teste, a taxa de erro verdadeira será, com alta probabilidade, menor que 1%

Amostragem Aleatória

- Na amostragem aleatória, L hipóteses, $L \ll n$, são induzidas a partir de cada um dos L conjuntos de treinamento



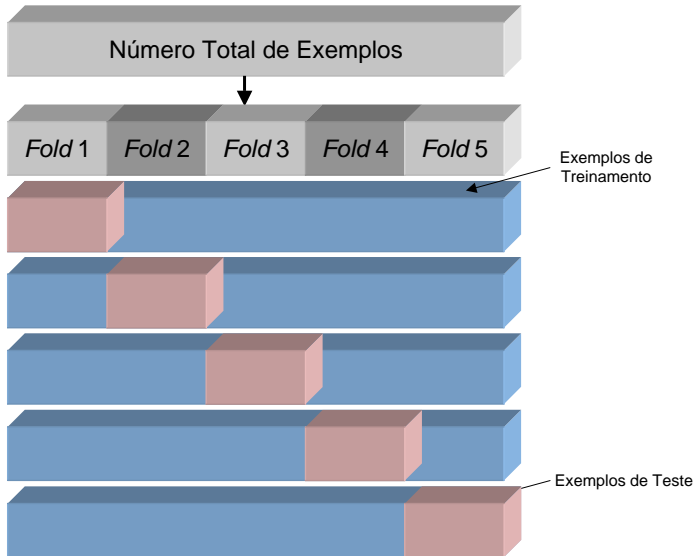
Amostragem Aleatória

- ❑ O erro final é calculado como sendo a média dos erros de todas as hipóteses induzidas e calculados em conjuntos de teste independentes e extraídos aleatoriamente
- ❑ Amostragem aleatória pode produzir melhores estimativas de erro que o estimador *holdout*

Cross-Validation

- ❑ Este estimador é um meio termo entre os estimadores *holdout* e *leave-one-out*
- ❑ Em *r-fold cross-validation* (CV) os exemplos são aleatoriamente divididos em *r* partições mutuamente exclusivas (*folds*) de tamanho aproximadamente igual a n/r exemplos
- ❑ Os exemplos nos $(r-1)$ *folds* são usados para treinamento e a hipótese induzida é testada no *fold* remanescente
- ❑ Este processo é repetido *r* vezes, cada vez considerando um *fold* diferente para teste
- ❑ O erro na *cross-validation* é a média dos erros calculados em cada um dos *r folds*

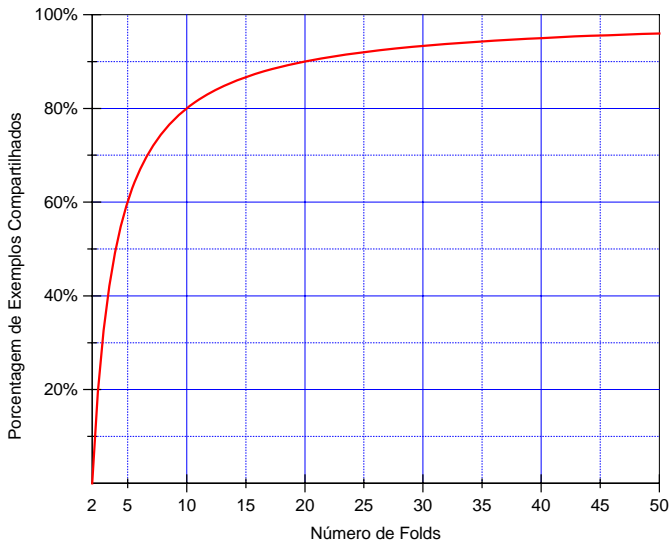
Exemplo de *Cross-Validation* com 5 *Folds* (5-*CV*)



Cross-Validation

- ❑ Este procedimento de rotação reduz tanto o *bias* inerente ao método de *holdout* quanto o custo computacional do método *leave-one-out*
- ❑ Entretanto, deve-se observar, por exemplo, que em *10-fold cross-validation*, cada par de conjuntos de treinamento compartilha 80% de exemplos
- ❑ É fácil generalizar que a porcentagem de exemplos compartilhados na *r-fold cross-validation* é dada por $(1 - 2/r)$ para $r \geq 2$ folds (figura seguinte)
- ❑ À medida que o número de *folds* aumenta, esta sobreposição pode evitar que os testes estatísticos obtenham uma boa estimativa da quantidade de variação que seria observada se cada conjunto de treinamento fosse independente dos demais

Cross-Validation



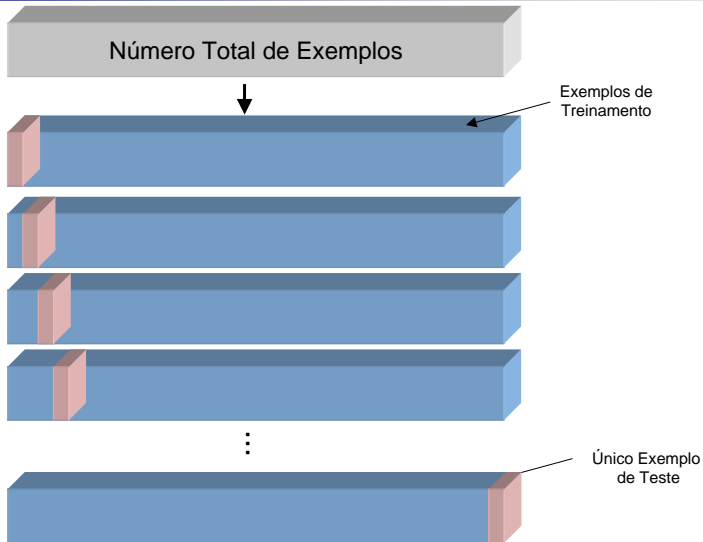
Stratified Cross-Validation

- ❑ O estimador *stratified cross-validation* é similar à *cross-validation*, mas ao gerar os *folds* mutuamente exclusivos, a distribuição de classe (proporção de exemplos em cada uma das classes) é considerada durante a amostragem
- ❑ Isto significa, por exemplo, que se o conjunto original de exemplos possui duas classes com distribuição de 20% e 80%, então cada *fold* também terá essa proporção de classes

Leave-one-out

- ❑ O estimador *leave-one-out* é um caso especial de *cross-validation*
- ❑ É computacionalmente dispendioso e freqüentemente é usado em amostras pequenas
- ❑ Para uma amostra de tamanho n uma hipótese é induzida utilizando $(n-1)$ exemplos; a hipótese é então testada no único exemplo remanescente
- ❑ Este processo é repetido n vezes, cada vez induzindo uma hipótese deixando de considerar um único exemplo
- ❑ O erro é a soma dos erros em cada teste dividido por n

Leave-one-out



Leave-one-out

- ❑ Por muitos anos o *leave-one-out* foi a técnica recomendada para avaliar a performance de classificadores para pequenas amostras de exemplos.
- ❑ Embora o *leave-one-out* é um estimador praticamente não tendencioso (ou seja, o estimador, após várias aplicações, tende para a taxa de erro verdadeira), sua variância para pequenas amostras é alta.

Leave-one-out

- ❑ O *leave-one-out* pode ser comparado a um bêbado tentando andar sobre uma linha reta. Na média, ele pode estar exatamente sobre a linha, mesmo quando constantemente cambaleia para a direita e a esquerda dessa linha.
- ❑ Lembrar que tendência e variância de um estimador de taxa de erro verdadeira contribuem para a imprecisão da taxa de erro estimada.

Bootstrapping

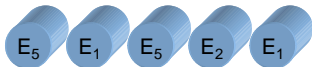
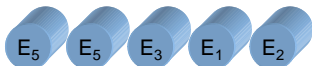
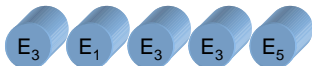
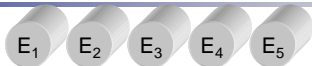
- ❑ Para amostras pequenas, o método de resampling *bootstrapping* tem mostrado resultados promissores como estimador da taxa de erro verdadeira.
- ❑ Cada experimento é conduzido com base em um novo conjunto de treinamento obtido por amostragem **com reposição** do conjunto original de exemplos
- ❑ Há muitos estimadores *bootstrap*, sendo o mais comum denominado *bootstrap e_0*

Bootstrap e0

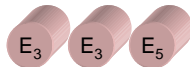
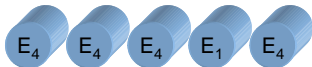
- ❑ Um conjunto de treinamento bootstrap consiste de n exemplos (mesmo tamanho do conjunto original de exemplos) amostrados **com reposição** a partir do conjunto original de exemplos
- ❑ Isto significa que alguns exemplos E_i podem não aparecer no conjunto de treinamento *bootstrap* e alguns E_i podem aparecer mais de uma vez
- ❑ Os exemplos remanescentes (aqueles que não aparecem no conjunto de treinamento *bootstrap*) são usados como conjunto de teste

Bootstrap e0

Conjunto Completo
de Exemplos



⋮



Conjuntos de Treinamento

Conjuntos de Teste

Bootstrap e0

- Em uma amostra com **n** exemplos, a probabilidade de escolher um exemplo é

$$1/n$$

- A probabilidade de não escolher um exemplo é

$$1-1/n$$

- Então, a probabilidade de não escolher um exemplo em **n** tentativas é

$$(1-1/n)^n$$

Bootstrap e0

- ❑ Ou seja, para uma dada amostra *bootstrap*, um exemplo de treinamento tem probabilidade $1 - (1 - 1/n)^n$ de ser selecionado pelo menos uma vez em cada uma das n vezes nas quais os exemplos são aleatoriamente selecionados a partir do conjunto original de exemplos
- ❑ Para n grande, isto é aproximadamente $1 - 1/e = 0.632$ (lembrar que $(1 - 1/n)^n \sim e^{-1}$)
- ❑ Assim, a técnica possui uma fração média de casos (exemplos) não repetidos para o conjunto de treinamento de 0.632, e uma fração média de 0.368 para casos do conjunto de teste.

Bootstrap e0

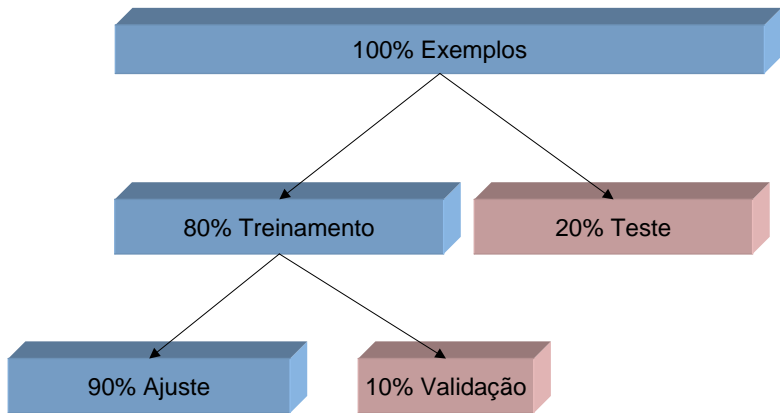
- A taxa de erro final é calculada através da média das taxas se erro das várias iterações. Cerca de 200 iterações são consideradas necessárias para uma boa estimativa com o *bootstrap e0*. Portanto, este método pode ser considerado mais caro que o método *leave-one-out*.

Ajuste de Parâmetros

- ❑ Em algumas situações torna-se necessário realizar um ajuste de parâmetros de um indutor
 - fator de confiança (poda), número mínimo de exemplos em cada folha, etc (DT)
 - número de condições por regra, suporte, etc (Indução de Regras)
 - número de neurônios, tipo de função de ativação, número de camadas, etc (RNA)
- ❑ Nesses casos, é necessário reservar uma parte dos exemplos para ajustar os parâmetros e outra parte para teste

Ajuste de Parâmetros

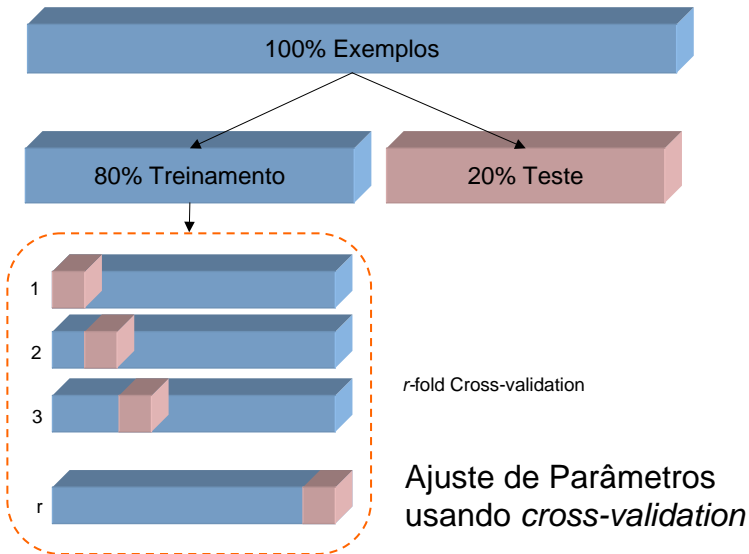
Variação *Holdout* 1



Ajuste de parâmetros
utilizando *holdout*

Ajuste de Parâmetros

Variação *Holdout* 2



Desempenho de Algoritmos

- ❑ Ainda que o erro de um algoritmo seja menor que outro usando o mesmo conjunto de dados, é necessário utilizar algum teste estatístico apropriado para verificar a probabilidade da validade dessa afirmação.
- ❑ **Os testes de hipóteses** têm como objetivo decidir, com base na informação fornecida pelos dados de uma amostra, sobre a aceitação ou não de uma dada hipótese
- ❑ Existem muitos testes estatísticos (e muitas discussões ao respeito) que podem ser utilizados para esse fim pela comunidade de AM.

Desempenho de Algoritmos

- Porém, esse tema foge do programa desta disciplina. Assim, no TP, somente deve ser apresentado o erro dos classificadores no conjunto de dados, sem análise dos resultados usando algum teste estatístico, como mostrado a seguir.

Calculando Média e Desvio Padrão Utilizando Amostragem

- ❑ Dado um algoritmo **A** e um conjunto de exemplos **T**, assume-se que **T** seja dividido em **r** partições
- ❑ Para cada partição **i**, é induzida a hipótese **h_i** e o erro denotado por **err(h_i)**, $i = 1, 2, \dots, r$, é calculado
- ❑ A seguir, a **média** (mean), **variância** (var) e **desvio padrão** (sd) para todas as partições são calculados utilizando-se:

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

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

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

Calculando Média e Desvio Padrão Utilizando Amostragem

- ❑ É possível denotar **mean(A)** como **mean(A,T)**, quando a intenção é tornar evidente o fato que o erro médio do algoritmo **A** foi calculado sobre o conjunto de exemplos **T**
- ❑ Alternativamente, **mean(A)** pode ser denotado por **mean(T)**, quando deseja-se evidenciar o fato que o erro médio foi calculado utilizando o conjunto particular de exemplos **T**, assumindo o algoritmo **A** fixo para um dado experimento
- ❑ Analogamente, essa notação se estende para **var(A)**, **sd(A)** ou outros valores que possam ser derivados a partir destes

Exemplo

- ❑ Para exemplificar o cálculo da média e desvio padrão de um algoritmo **A** utilizando um conjunto de exemplos **T**, considere *10-fold cross-validation*, isto é, $r=10$, para um algoritmo **A** com os seguintes erros em cada *fold*
 - (5.5, 11.4, 12.7, 5.2, 5.9, 11.3, 10.9, 11.2, 4.9, 11.0)

- ❑ Então:

$$\text{mean}(A) = \frac{90.00}{10} = 9.00$$

$$\text{sd}(A) = \sqrt{\frac{1}{10 \times 9} 90.30} = 1.00$$

- ❑ Em geral, o erro é representado por sua média seguida pelo seu desvio padrão entre parênteses
 - neste exemplo, o erro é 9.00 (1.00)

Comparando Algoritmos

- ❑ Ao tentar comparar dois algoritmos observando apenas valores, por exemplo, a taxa de erro em problemas de classificação ou o erro em problemas de regressão, não é fácil perceber se um algoritmo é melhor do que o outro
- ❑ Em várias situações, para comparar o erro (média e desvio padrão) obtido, *r-fold stratified cross-validation* é usualmente utilizado (para manter a distribuição de classes)
- ❑ De fato, a maioria dos trabalhos na área reportam erros utilizando *10-fold cross-validation* ou *stratified cross-validation*

Comparando Algoritmos

- Ao comparar dois indutores no mesmo domínio T , o **desvio padrão** pode ser visto como uma imagem da robustez do algoritmo: se os erros, calculados sobre diferentes conjuntos de teste, provenientes de hipóteses induzidas utilizando diferentes conjuntos de treinamento são muito diferentes de um experimento para outro, então, o indutor **não é robusto** a mudanças no conjunto de treinamento proveniente de uma mesma distribuição

Comparando Algoritmos

- ❑ Suponha por exemplo, que deseja-se comparar dois algoritmos A_S e A_P com taxas de erro iguais a
 - A_S : 9.00 (1.00) e
 - A_P : 7.50 (0.80)
- ❑ Parece que o segundo, A_P , é melhor que A_S nesse conjunto de dados. Mas se a gente aplicar algum teste estatístico apropriado para decidir qual deles é melhor que o outro (com grau de confiança de 95% por exemplo) veremos que A_P **não** supera A_S significativamente (com grau de confiança de 95%)

Referências I



[1] Monard, M. C.

Slides da disciplina SCC630 - Inteligência Artificial. ICMC - USP, 2010.