



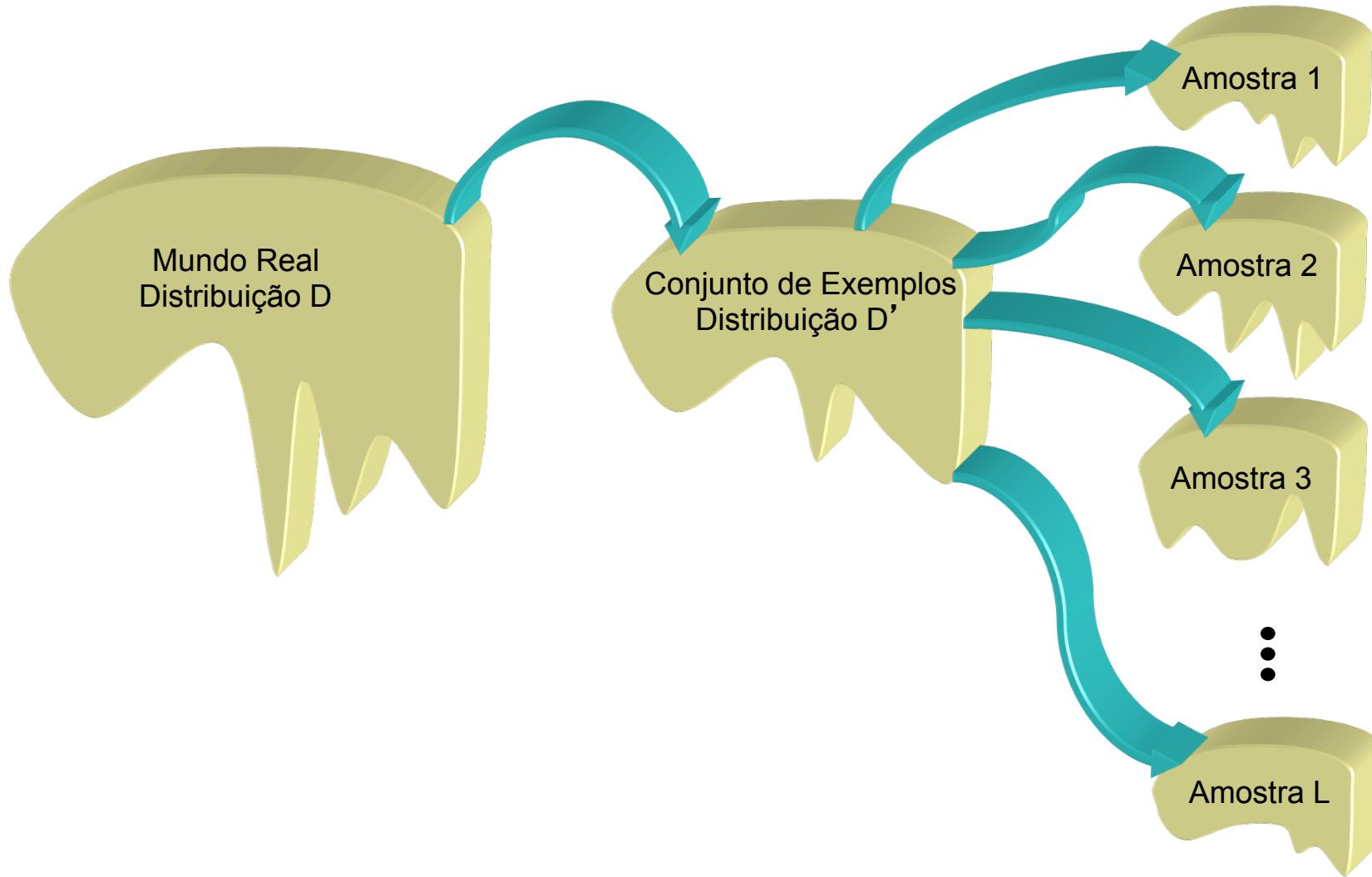
Métodos de Amostragem e Avaliação de Algoritmos

- 
- ❑ 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 **predizer**)
- 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



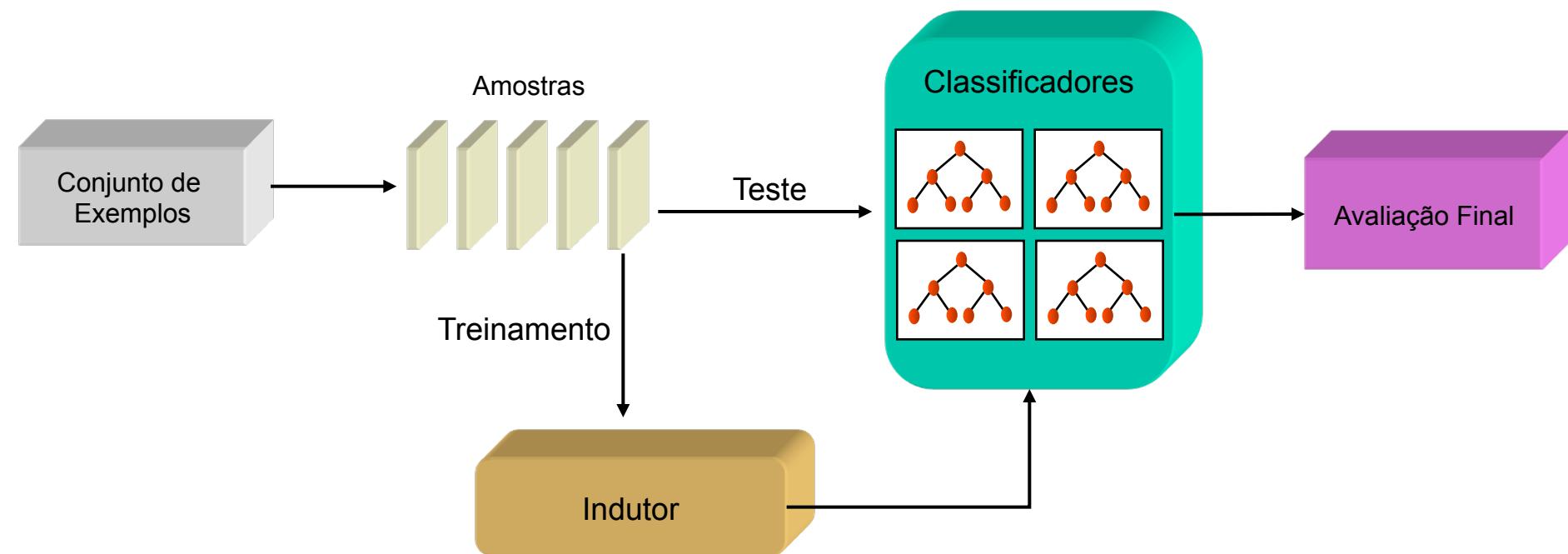
Métodos de Amostragem

- 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

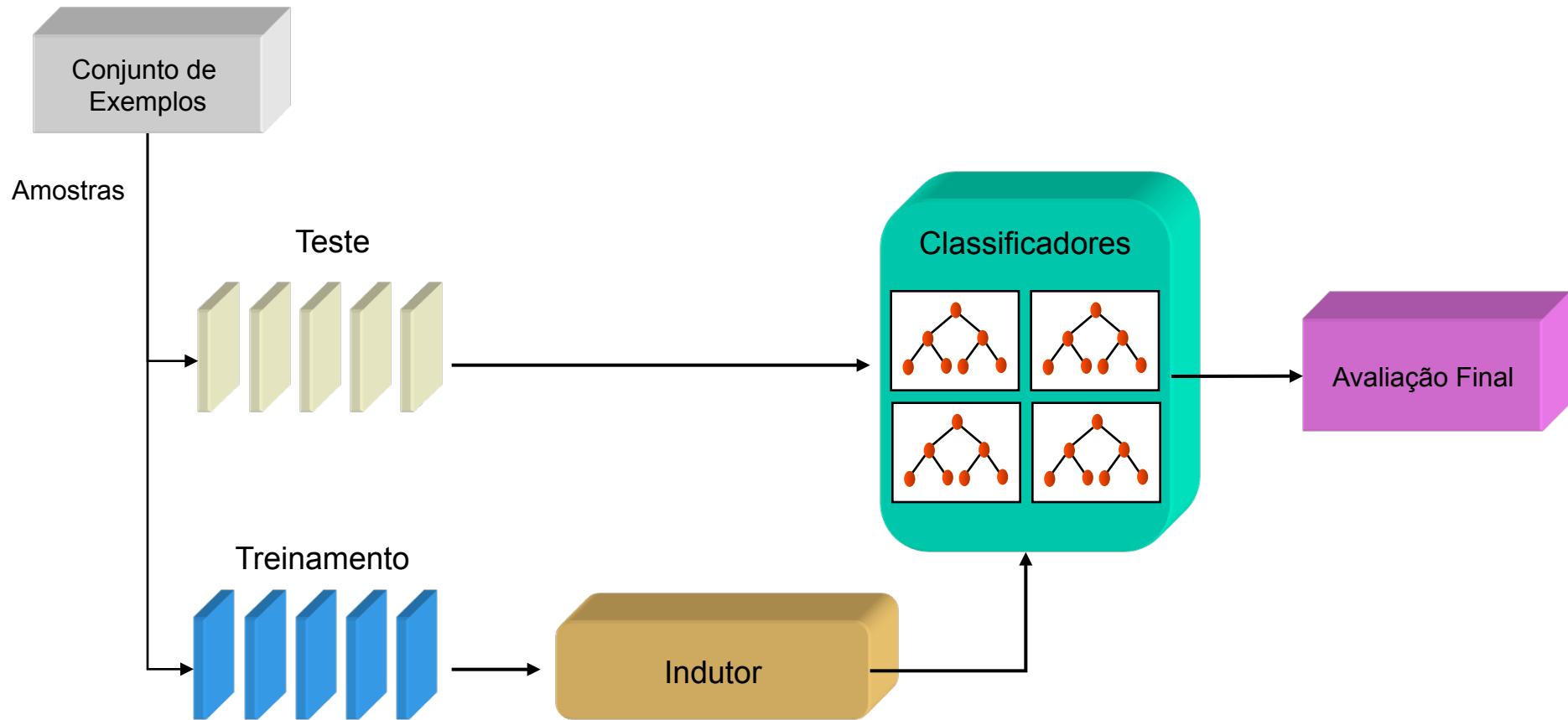
Métodos de Amostragem (Resubstituição)



Resubstituição

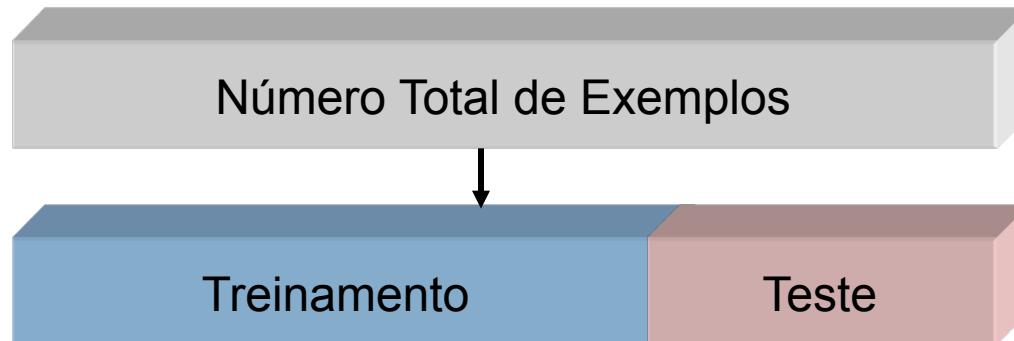
- Este método consiste em construir o classificador e testar seu desempenho no mesmo conjunto de exemplos, ou seja, o conjunto de teste é idêntico ao conjunto de treinamento
- Este estimador fornece uma **medida aparente**, possuindo uma estimativa altamente otimista da precisão, devido ao fato de que 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 **p** para treinamento e $(1-p)$ para teste, considerando normalmente $p > 1/2$
- ☐ Valores típicos são $p = 2/3$ e $(1-p) = 1/3$, embora não existam fundamentos teóricos sobre estes valores

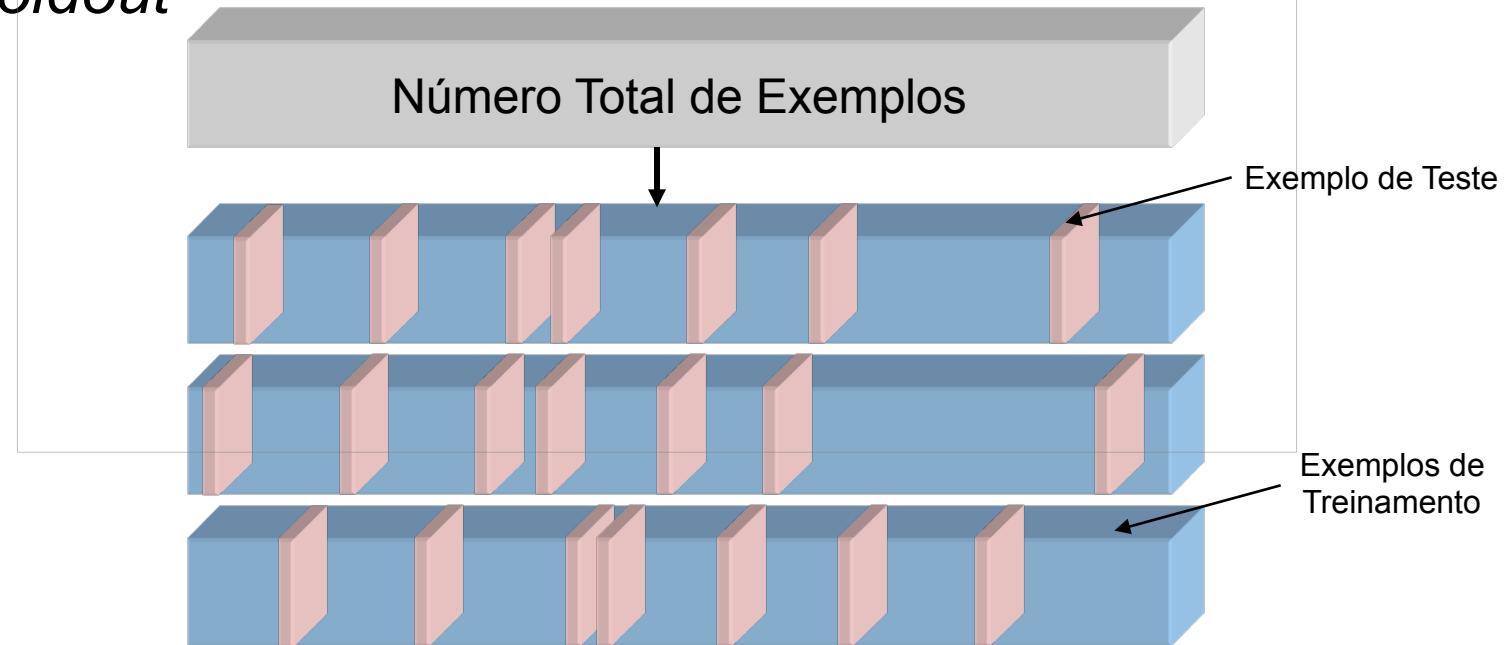


Holdout

- ❑ Uma vez que uma hipótese construída utilizando todos os exemplos, em média, apresenta desempenho melhor que uma hipótese construída utilizando apenas uma parte dos exemplos, este método tem a tendência de super estimar o erro verdadeiro
- ❑ Para pequenos conjuntos, nem sempre é possível separar uma parte dos exemplos

Holdout

- ☐ De forma a tornar o resultado menos dependente da forma de divisão dos exemplos, pode-se calcular a média de vários resultados de *holdout* através da construção de várias partições obtendo-se, assim, uma estimativa média do *holdout*



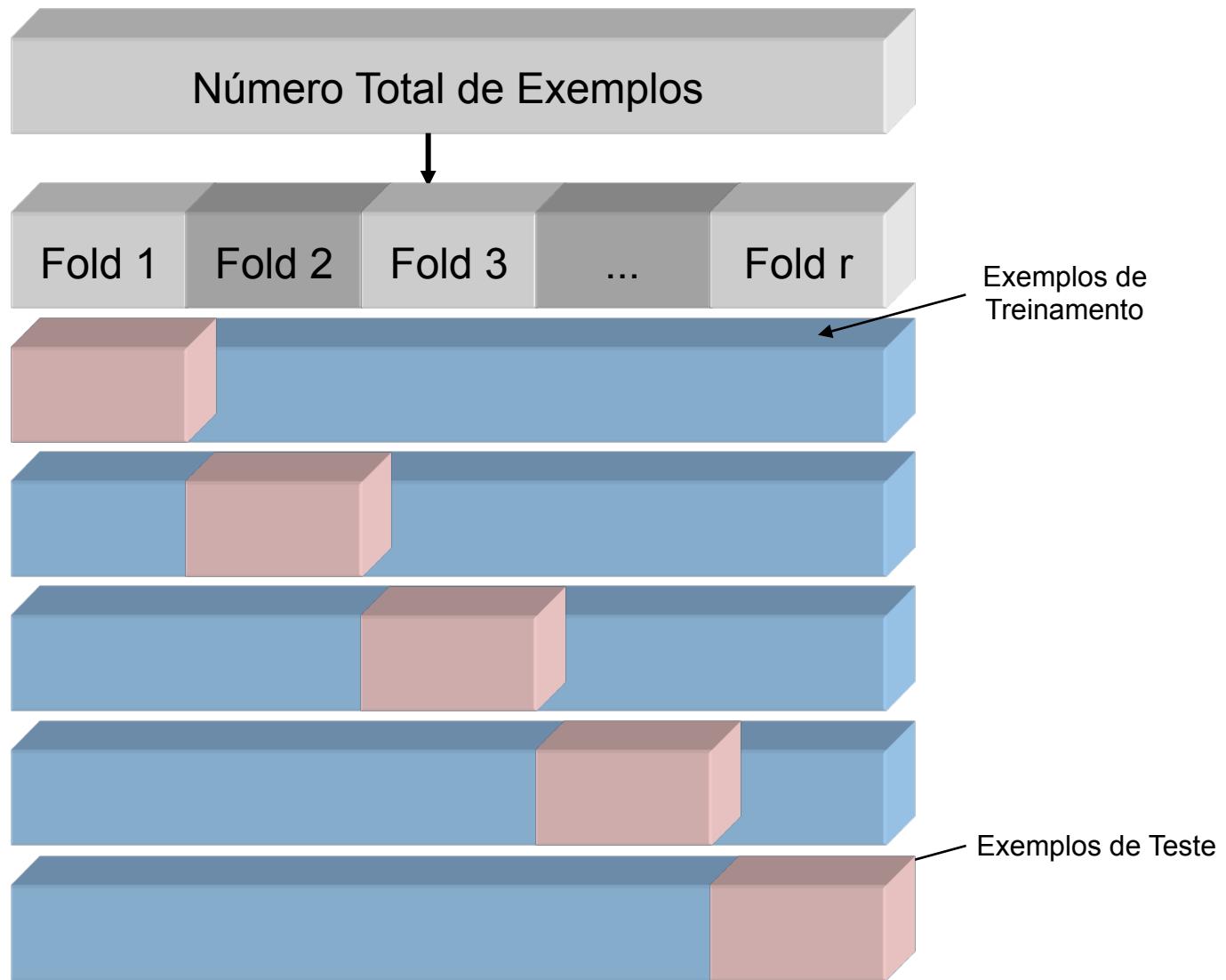
Amostragem Aleatória

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

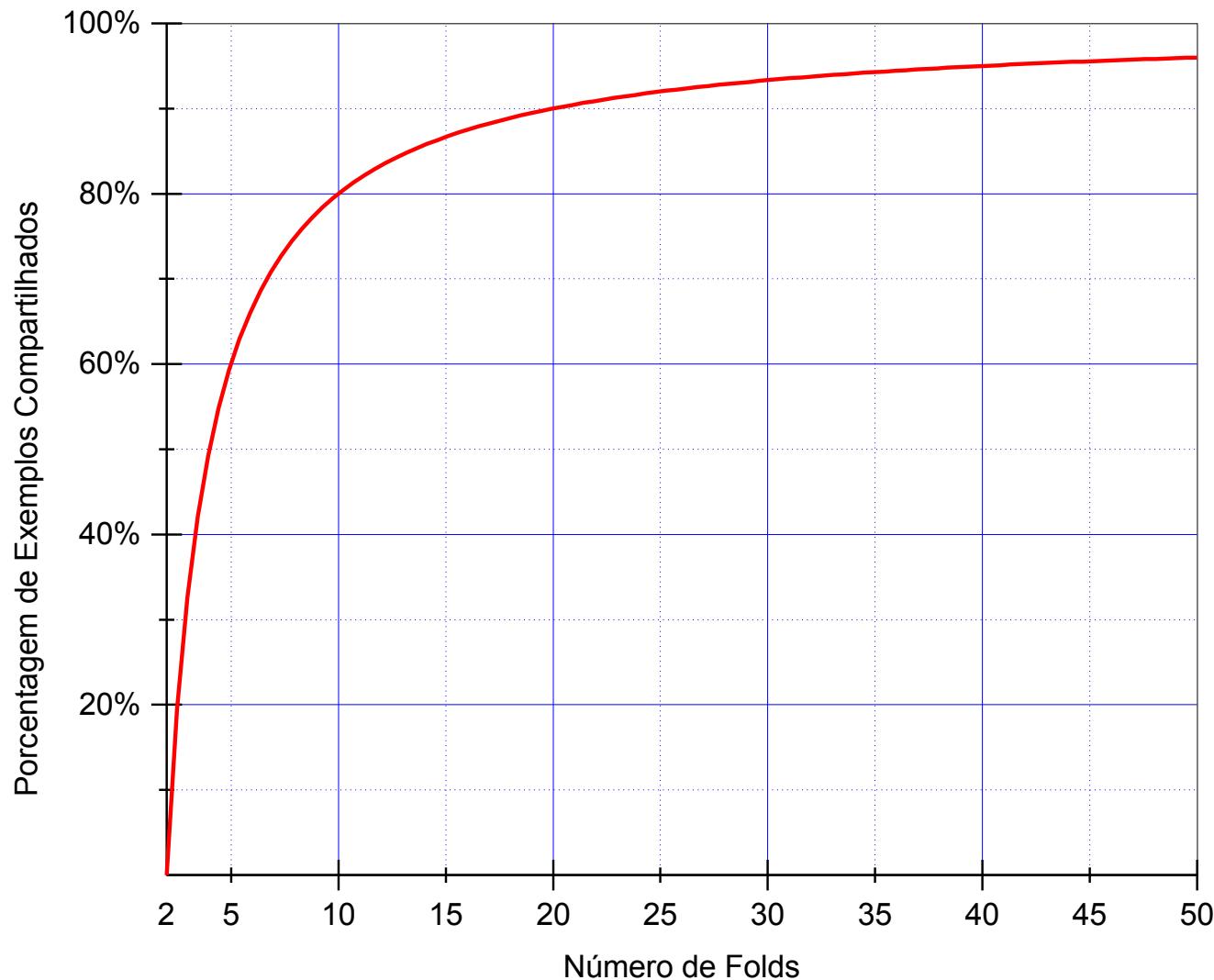
Cross-Validation



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 $(1 - \frac{2}{r})$ 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



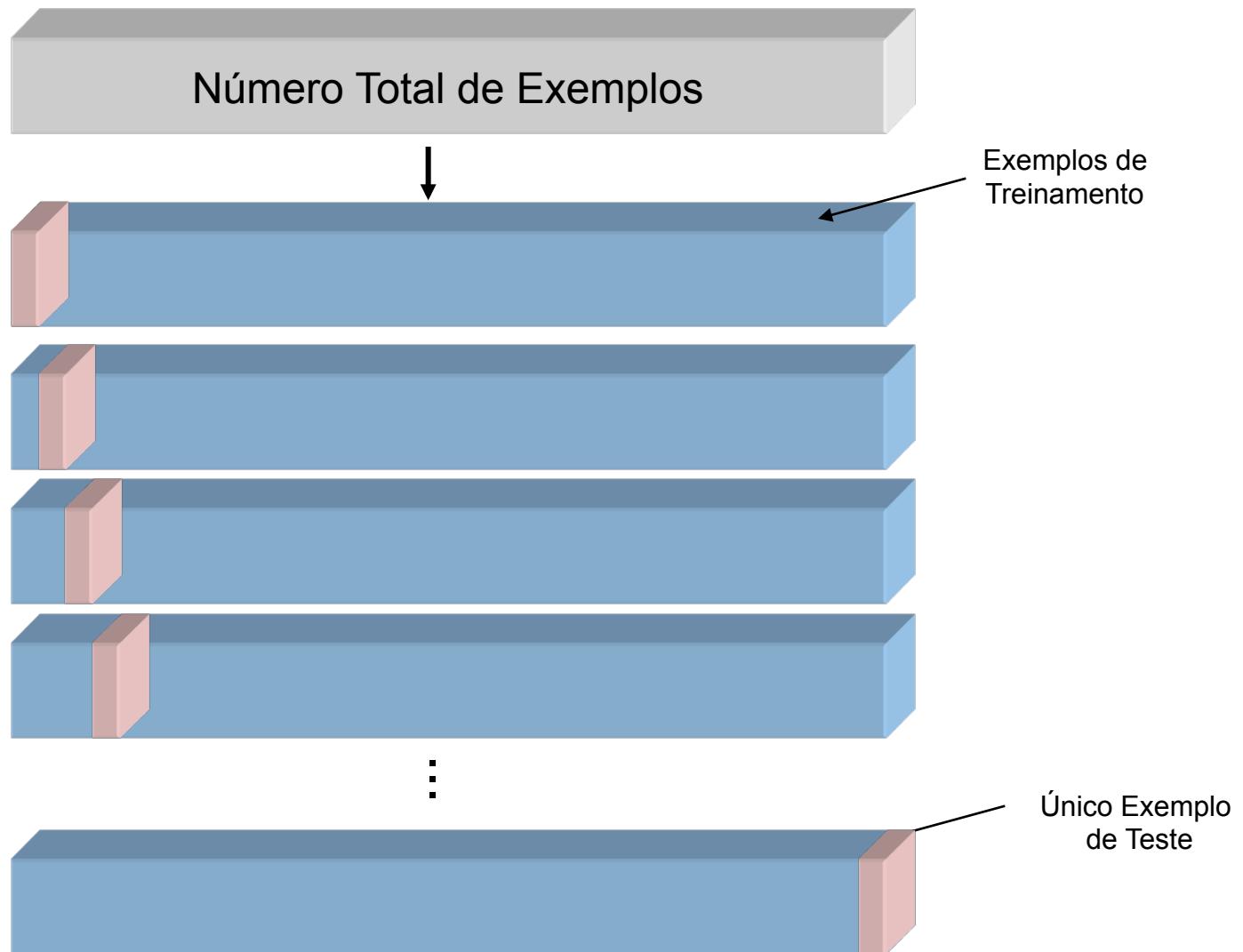
Stratified Cross-Validation

- 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 esta 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



Bootstrap

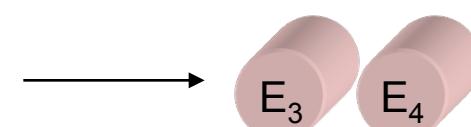
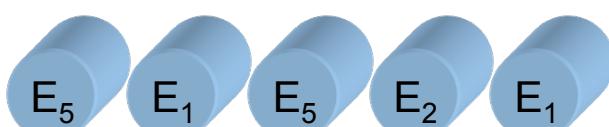
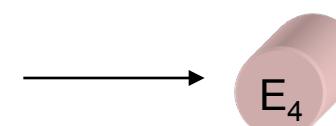
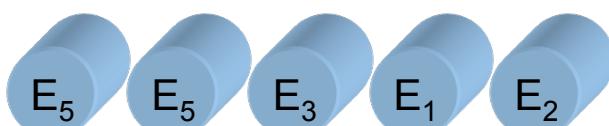
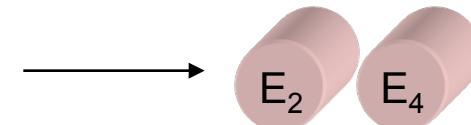
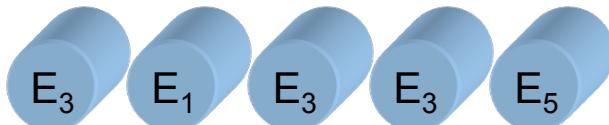
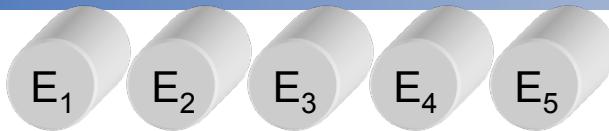
- ❑ No estimador *bootstrap*, a idéia básica consiste em repetir o processo de classificação um grande número de vezes
- ❑ Estima-se então valores, tais como o erro ou *bias*, a partir dos experimentos replicados, cada experimento sendo conduzido com base em um novo conjunto de treinamento obtido por amostragem **com reposição** do conjunto original de exemplos

Bootstrap e0

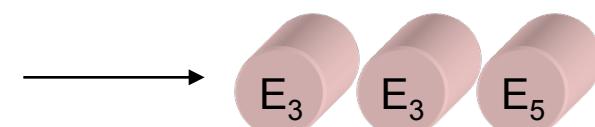
- Há muitos estimadores *bootstrap*, sendo o mais comum denominado *bootstrap e0*
- Um conjunto de treinamento *bootstrap* consiste de **n** exemplos (mesmo tamanho do conjunto original de exemplos) amostrados **com reposição** a partir do conjunto original de exemplos
- Isto significa que alguns exemplos 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

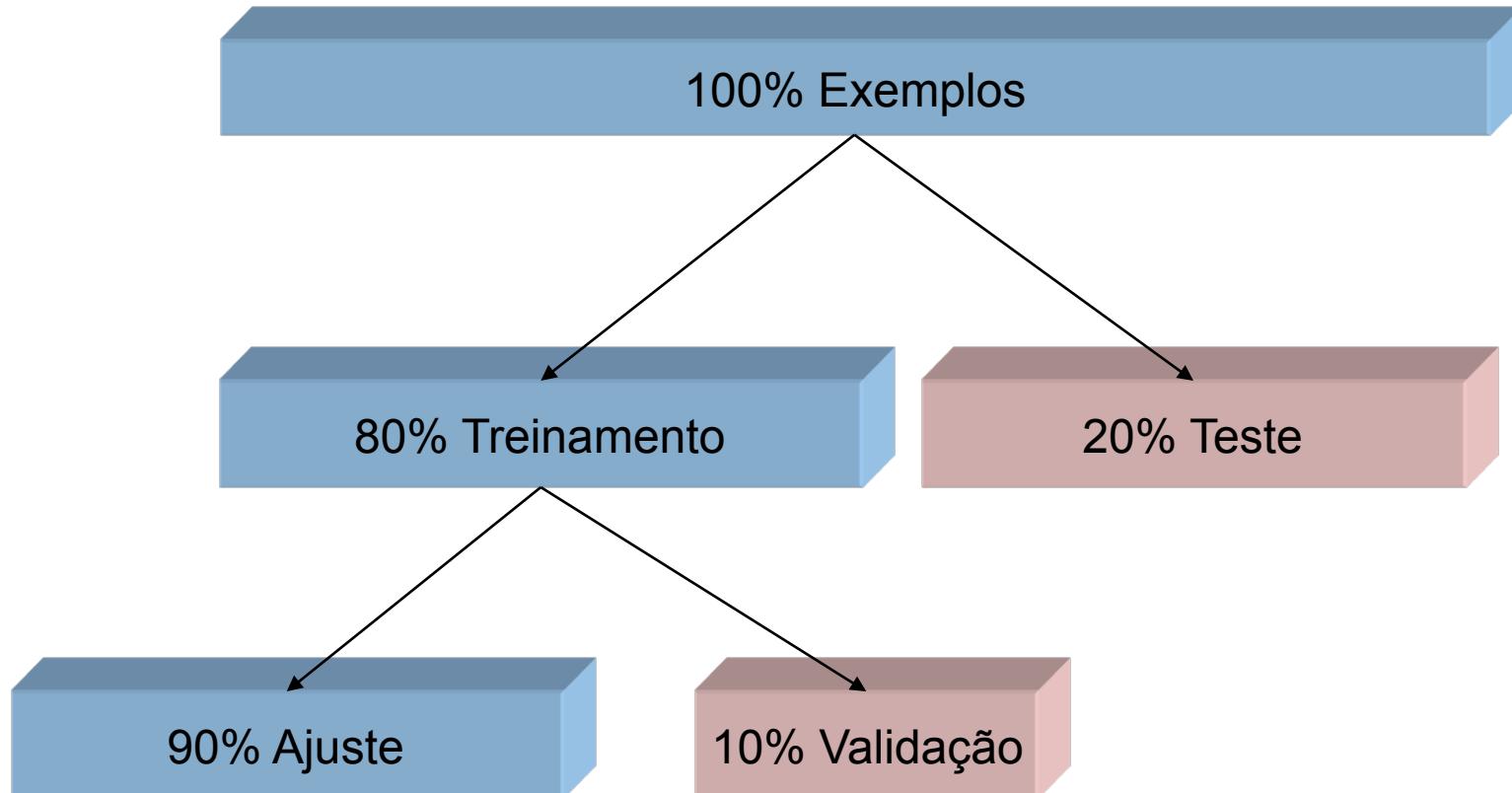
- Para uma dada amostra *bootstrap*, um exemplo de treinamento tem probabilidade $1-(1-1/n)^n$ de ser selecionado pelo menos uma vez em cada uma das **n** vezes nas quais os exemplos são aleatoriamente selecionados a partir do conjunto original de exemplos
- Para **n** grande, isto é aproximadamente $1-1/e = 0.632$
- Portanto, para esta técnica, a fração média de exemplos não repetidos é 63.2% no conjunto de treinamento e 36.8% no conjunto de teste
- Geralmente, o processo de *bootstrap* é repetido um número de vezes, sendo o erro estimado como a média dos erros sobre o número de iterações

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

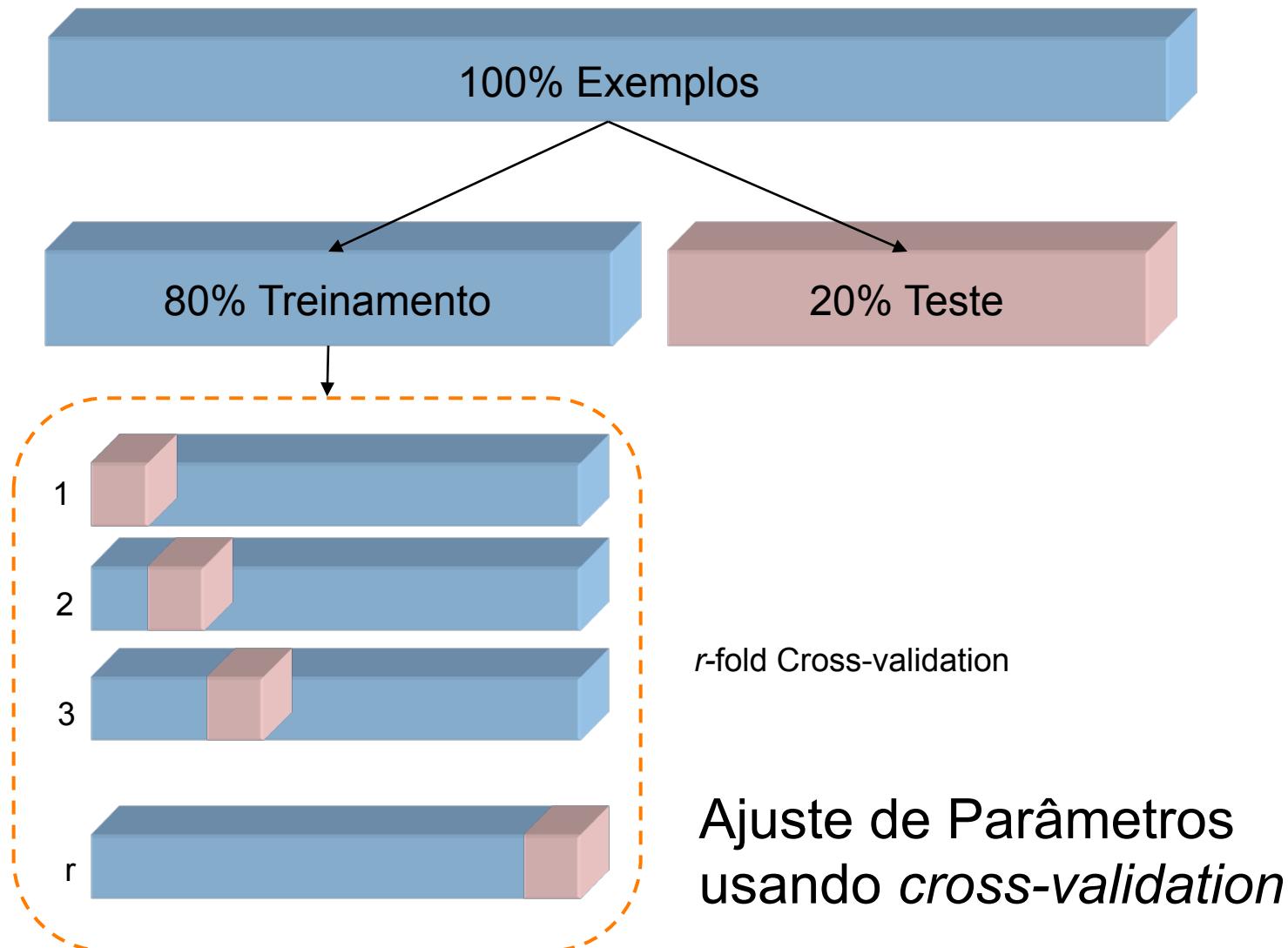
Variação Holdout 1



Ajuste de parâmetros
utilizando *holdout*

Ajuste de Parâmetros

Variação Holdout 2



Parâmetros Típicos de Estimadores

	<i>holdout</i>	aleatória	<i>leave-one-out</i>	<i>r-fold cv</i>	<i>r-fold strat cv</i>	<i>bootstrap</i>
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

□ Alguns parâmetros típicos de estimadores, onde

- n representa o número de exemplos;
- r o número de *folds* (partições);
- p um número tal que $0 < p < 1$;
- t um número tal que $0 < t < n$ e
- L o número de hipóteses induzidas

Desempenho de Algoritmos

- ❑ Veremos uma metodologia para a avaliação de algoritmos que é comumente utilizada em AM
- ❑ Veremos dois testes estatísticos para
 - estimar os limites para o desempenho de um algoritmo
 - estimar se desempenho entre quaisquer dois algoritmos é significativa ou não
- ❑ Existem muitos outros testes estatísticos além dos descritos aqui

Estimando o Desempenho de um Algoritmo

- Suponha que um classificador possua erro verdadeiro (avaliado no conjunto de teste) de 25%
- Lembrando que o aprendizado é efetuado em uma amostra (pequena) de uma população (grande), espera-se que o erro em toda população seja próximo de 25%
- A proximidade será maior dependendo do tamanho do conjunto de teste
- Naturalmente tem-se uma maior confiança no valor de 25% se ele foi obtido a partir de um conjunto de teste com 10000 do que com 10 exemplos

Estimando o Desempenho de um Algoritmo

- Em Estatística, uma sucessão de **N** eventos independentes que têm **sucesso** (ocorrência) ou **falham** (não ocorrência) é chamado de processo de Bernoulli
 - $\text{Pr}(\text{sucesso}) = p$
 - $\text{Pr}(\text{falha}) = 1-p$
- Assumindo **sucesso** como o evento de estudo, em um processo de Bernoulli
 - média = p
 - variância = $p * (1-p) / N$
- Para **N** grande, essa distribuição aproxima-se da distribuição normal

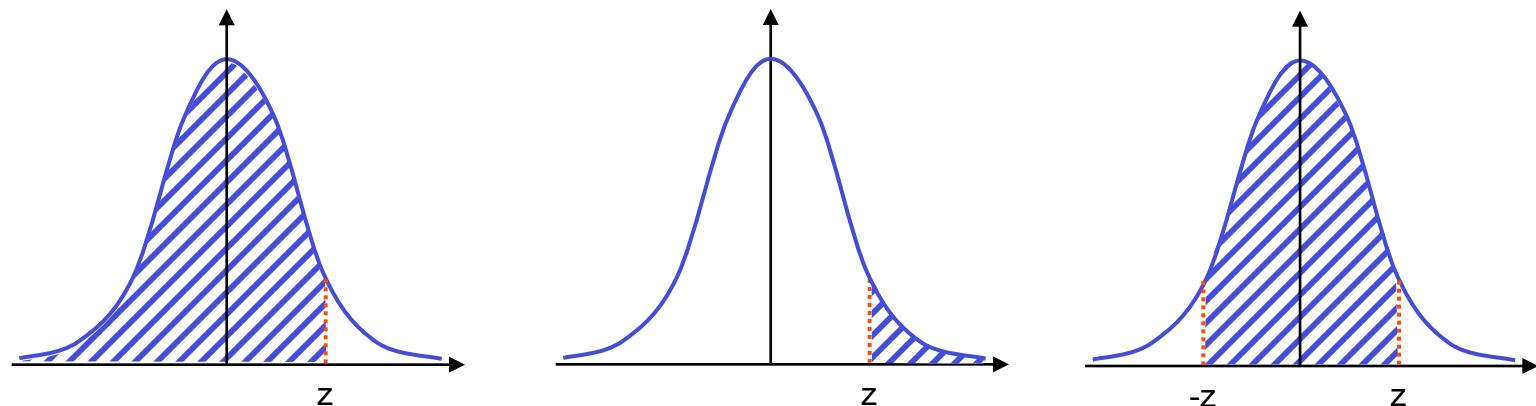
Estimando o Desempenho de um Algoritmo

- Suponha que de **N** tentativas, **S** são sucessos
- Assim, a taxa de sucesso observada é $f=S/N$
- A questão é determinar taxa de sucesso **p** na população a partir de **f**
- A solução é usualmente expressa como um intervalo de confiança, ou seja, o valor de **p** encontra-se dentro de um intervalo com um grau de confiança **c**

Estimando o Desempenho de um Algoritmo

- Na distribuição normal, a probabilidade que variável aleatória X com média 0 esteja dentro de um intervalo de confiança de tamanho 2^*z é
 - $\Pr(-z \leq X \leq z) = c$
- Os valores de c e os correspondentes valores de z são obtidos através de tabelas existentes na maioria dos livros de Estatística
- Os livros fornecem normalmente valores $\Pr(X \leq -z)$ ou $\Pr(X \geq z)$
- Lembrando que, por simetria da distribuição normal, $\Pr(X \leq -z) = \Pr(X \geq z)$
- Por exemplo, $\Pr(-1.65 \leq X \leq 1.65) = 90\%$

Distribuição Normal



z	$\Pr(X \leq z)$	$\Pr(X \geq z)$	$\Pr(-z \leq X \leq z)$
3.00	99.87%	0.13%	99.74%
2.00	97.73%	2.27%	95.46%
1.65	95.05%	4.95%	90.10%
1.29	90.15%	9.85%	80.30%
1.00	84.13%	15.87%	68.26%

Estimando o Desempenho de um Algoritmo

- Basta reduzir a variável aleatória $f = S/N$ para média 0 e desvio padrão unitário, subtraindo a média **p** e dividindo pelo desvio padrão

$$\Pr(-z \leq X \leq z) = c \quad \Pr\left(-z \leq \frac{f - p}{\sqrt{p(1-p)/N}} \leq z\right) = c$$

- O passo final consiste em escrever a desigualdade como igualdade, resolvendo-a para encontrar o valor de **p**

$$p = \frac{f + \frac{z^2}{2N} - z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

Estimando o Desempenho de um Algoritmo

- Assuma $c=95\%$ ou seja, $z=2$
- Exemplo 1:
 - Se $S=250$ sucessos dentre $N=1000$ tentativas, a taxa de sucesso observada é $f=25\%$
 - Com $c=95\%$ de confiança o valor de p fica no intervalo entre 23.29% e 26.79%
- Exemplo 2:
 - Se $S=25$ sucessos observados dentre $N=100$ tentativas, f também é igual a 25%
 - Com $c=95\%$ de confiança o valor de p fica no intervalo entre 19.89% e 30.92%
- Observe que embora $f=25\%$ e $c=95\%$ nas duas situações, o intervalo do segundo exemplo é maior pois o experimento é menor
- Note que podemos considerar f como sendo a taxa de erro

$$p = \frac{f + \frac{z^2}{2N}}{1 + \frac{z^2}{N}} z \sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}$$

$$p = \frac{0.25 + \frac{2^2}{2 \cdot 1000}}{1 + \frac{2^2}{1000}} 2 \sqrt{\frac{0.25}{1000} - \frac{0.25^2}{1000} + \frac{2^2}{4 \cdot 1000^2}}$$

$$p = \frac{0.25 + \frac{2^2}{2 \cdot 100}}{1 + \frac{2^2}{100}} 2 \sqrt{\frac{0.25}{100} - \frac{0.25^2}{100} + \frac{2^2}{4 \cdot 100^2}}$$

Estimando o Desempenho de um Algoritmo

- O valor superior de **p** (calculado usando a parte “+” da equação para taxa de erro) é comumente conhecido como **erro pessimista (perr)**- (utilizado no C4.5)
- Assim, ele também pode ser empregado para a poda de árvores ou regras
 - No caso de árvores, para cada sub-árvore a ser podada, calcula-se o valor de **perr** (sub-árvore sem poda) e o valor de **perr'** (sub-árvore podada)
 - ❖ se **perr' < perr** então realiza-se a poda
 - O mesmo é válido para regras

Comparando Dois Algoritmos

- Antes de comparar dois algoritmos, algumas definições adicionais são necessárias
- Para tanto, assume-se o emprego de *cross-validation*, uma vez que é um método comumente utilizado pela comunidade de AM
- Entretanto, qualquer outro método de amostragem (exceto resubstituição) pode ser utilizado no lugar de *cross-validation* para calcular a média e desvio padrão de um algoritmo

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,...,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:

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

$$\text{sd}(A) = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (x_i - 9.00)^2} = \sqrt{\frac{1}{10} (5.5^2 + 11.4^2 + 12.7^2 + 5.2^2 + 5.9^2 + 11.3^2 + 10.9^2 + 11.2^2 + 4.9^2 + 11.0^2 - 10 \cdot 9.00^2)} = \sqrt{10.30} = 3.18$$

- Em geral, o erro é representado por sua média seguida pelo símbolo “ \pm ” e seu desvio padrão
 - 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 utilizada (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 com taxas de erro iguais a
 - 9.00 ± 1.00 e
 - 7.50 ± 0.80
- Para decidir qual deles é melhor que o outro (com grau de confiança de 95%) basta assumir o caso geral para determinar se a diferença entre dois algoritmos (A_S e A_P) é significante ou não, assumindo uma distribuição normal
- Em geral, a comparação é feita de forma que A_P é o algoritmo proposto e A_S o algoritmo padrão

Comparando Algoritmos

- Para isso, a média e desvio padrão combinados são calculados de acordo com as seguintes equações:

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

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

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

- A diferença absoluta (ad) é dada em desvios padrões

Comparando Algoritmos

- Se $ad(A_S - A_P) > 0$ então A_P supera A_S
 - se $ad(A_S - A_P) \geq 2$ desvios padrões então A_P supera A_S com grau de confiança de 95%
- Se $ad(A_S - A_P) \leq 0$ então A_S supera A_P
 - se $ad(A_S - A_P) \leq -2$ então A_S supera A_P com grau de confiança de 95%

Exemplo

- Retornando ao exemplo descrito anteriormente, assuma
 - $A_S = 9.00 \pm 1.00$ (algoritmo padrão)
 - $A_P = 7.50 \pm 0.80$ (algoritmo proposto)
- Apenas observando os valores, há uma tendência em se achar que A_P é melhor que A_S
- Entretanto $\text{mean}(A_S - A_P) = 9.00 - 7.50 = 1.50$

$$\text{sd}(A_S - A_P) = \sqrt{\frac{1.00^2 + 0.80^2}{2}} = 0.91$$

$$\text{ad}(A_S - A_P) = \frac{1.50}{0.91} = 1.65$$

- Conseqüentemente, como $\text{ad}(A_S - A_P) > 0$, A_P supera A_S , porém como $\text{ad}(A_S - A_P) = 1.65 < 2$ A_P **não** supera A_S significativamente (com grau de confiança de 95%)
- Normalmente, quando se avalia muitos conjuntos de exemplos e/ou algoritmos, é útil a utilização de gráficos, indicando o valor absoluto em desvios padrões

Exemplo

Conjunto de Exemplos	Algoritmo Padrão	Algoritmo Proposto	mean	sd	ad		
bupa	32.70	2.79	33.88	1.78	-1.18	2.34	-0.50
pima	25.87	1.28	26.83	0.76	-0.96	1.05	-0.91
breast-cancer	5.86	0.84	5.69	0.37	0.17	0.65	0.26
hungaria	20.08	2.69	22.95	1.52	-2.87	2.18	-1.31
crx	14.64	0.88	15.60	0.69	-0.96	0.79	-1.21
hepatitis	21.92	3.20	20.22	1.38	1.70	2.46	0.69
sonar	32.17	2.79	25.81	2.07	6.36	2.46	2.59
genetics	5.92	0.52	7.00	0.15	-1.08	0.38	-2.82
dna	7.50	0.63	7.55	0.31	-0.05	0.50	-0.10

Exemplo

