Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Descrição da Implementação dos Métodos Estatísticos de Resampling do Ambiente AMPSAM

Gustavo Enrique de Almeida Prado Alves Batista
Maria Carolina Monard

Nº 68

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos

Fevereiro/1998

Descrição da Implementação dos Métodos Estatísticos de Resampling do Ambiente AMPSAM [1]

Gustavo Enrique de Almeida Prado Alves Batista
Maria Carolina Monard
Universidade de São Paulo/ILTC
Instituto de Ciências Matemáticas de São Carlos
Departamento de Ciências de Computação e Estatística
Caixa Postal 668, 13560-970 - São Carlos, SP
e-mail:{gbatista, mcmonard}@icmsc.sc.usp.br

Sumário

Neste trabalho nos concentramos em uma tarefa de aprendizado de máquina conhecida como classificação ou predição, cujo problema consiste na construção de um procedimento de classificação a partir de um conjunto de casos nos quais as classes verdadeiras são conhecidas, chamado de aprendizado supervisionado. O maior objetivo de um classificador é ser capaz de predizer com sucesso a respeito de novos casos. O desempenho de um classificador pode ser medido em termos da taxa de erro. Técnicas experimentais para estimar a taxa de erro para casos não vistos não somente provêm uma base para comparar objetivamente as performances de diversos algoritmos de aprendizado no mesmo conjunto de exemplos, mas também podem ser uma ferramenta poderosa para projetar um classificador. As técnicas para estimar a taxa de erro são baseados na teoria estatística de resampling.

Um ambiente chamado AMPSAM foi implementado para ajudar na aplicação dos métodos de resampling em conjuntos de exemplos do mundo real. AMPSAM foi projetado como uma coleção de programas independentes, os quais podem interagir entre si através de scripts pré-definidos ou de novos scripts criados pelo usuário. AMPSAM inclui ferramentas para particionar conjuntos de exemplos em conjuntos de treinamento e teste utilizando diferentes métodos de resampling. Além do método holdout, que é o estimador de taxa de erro mais comum, AMPSAM suporta os métodos n-fold cross-validation — incluindo o leaning-one-out — e o método bootstrap.

O objetivo deste trabalho é fornecer um panorama geral do funcionamento do ambiente AMPSAM , e descrever a implementação dos métodos estatísticos de resampling pertencentes a esse ambiente.

Fevereiro de 1998

[1] Trabalho realizado com auxílio do CNPq e FAPESP.

Sumário

1 Introdução **1**

2 Sistemas de Aprendizado e Classificadores **1**

3 A Taxa de Erro **4**

4 Tipos de Erros **5**

5 Técnicas para Estimar a Taxa de Erro Verdadeira **7**

5.1 Probably Approximately Correct (PAC) 7

5.2 Estimação da Taxa de Erro Através de Treinar-e-Testar 8

5.3 Métodos Resampling 10

5.3.1 Random Subsampling 10

5.3.2 Cross-validation 11

5.3.3 Bootstraping 12

6 O Ambiente Computacional AMPSAM **14**

7 Implementação **15**

7.1 Programa para Embaralhar o Arquivo de Exemplos - `embaralha.c` . . . 16

7.2 Métodos Holdout e Random Subsampling - `hold.c` 20

7.3 Métodos N-Fold Cross Validation - `part.c` e `nfoldcv.c` 22

7.4 Método Bootstrap - `bootstra.c` 26

8 Conclusão **27**

Lista de Figuras

1 Sistema de Aprendizado 2

2 Sistema de Classificação 2

3 Representação da Classificação 3

4 Conjuntos de Treinamento e Teste 8

5 Número de Casos de Teste e Qualidade da Predição 9

6 Processo de Particionamento dos Exemplos 14
7 Métodos Estatísticos do Ambiente AMPSAM 16
8 Procedimento `embaralha` 17
9 Procedimento `sorteio` 18
10 Procedimento `holdout` 21
11 Procedimento `particiona` 23
12 Procedimento `dividelementos` 24
13 Procedimento `crossvalidation` 25
14 Procedimento `bootstrap` 27

Lista de Tabelas

1 Exemplo de Matriz de Confusão para Três Classes 5
2 Tipos de Erros para Classificação com Duas Classes 5
3 Exemplo de Matriz de Custo de Classificação Incorreta para Três Classes 6
4 Comparação entre Holdout e Random Subsampling 11
5 Comparação entre Métodos de Cross-validation 12
6 Características dos Estimadores Bootstrap 13

1 Introdução

Neste trabalho nos concentramos em uma tarefa de aprendizado de máquina conhecida como classificação ou predição, cujo problema consiste na construção de um procedimento de classificação a partir de um conjunto de casos nos quais as classes verdadeiras são conhecidas, chamado de aprendizado supervisionado. O maior objetivo de um classificador é ser capaz de predizer com sucesso a respeito de novos casos. A performance de um classificador é medida em termos da taxa de erro. Técnicas experimentais para estimar a taxa de erro verdadeira não somente provêm uma base para comparar objetivamente as performances de diversos algoritmos de aprendizado no mesmo conjunto de exemplos, mas também podem ser uma ferramenta poderosa para projetar um classificador. As técnicas para estimar a taxa de erro são baseados na teoria estatística de resampling.

Um ambiente chamado AMPSAM (Ambiente para Medir a Performance de Sistemas de Aprendizado de Máquina) foi implementado para ajudar na aplicação dos métodos de resampling em conjuntos de exemplos do mundo real. AMPSAM foi projetado como uma coleção de programas independentes, os quais podem interagir entre si através de scripts pré-definidos ou de novos scripts criados pelo usuário. AMPSAM inclui ferramentas para particionar conjuntos de exemplos em conjuntos de treinamento e teste utilizando diferentes métodos de resampling. Além do método holdout, que é o estimador de taxa de erro mais comum, AMPSAM suporta os métodos n-fold cross-validation — incluindo o leaving-one-out — e o método bootstrap.

O objetivo deste trabalho é fornecer um panorama geral do funcionamento do ambiente AMPSAM , e descrever a implementação dos métodos estatísticos de resampling pertencentes a esse ambiente.

Este trabalho está organizado da seguinte forma: a seção 2 apresenta uma introdução aos sistemas de aprendizado e classificadores. A seção 3 discute a taxa de erro como índice para medir a performace de sistemas de aprendizado. A seção 4 mostra como diferentes erros cometidos por classificadores podem possuir diferentes importâncias em diversas aplicações. A seção 5 descreve como a taxa de erro verdadeira pode ser estimada, com confiança, por métodos estatísticos. Esta seção também brevemente apresenta a análise teórica PAC (Probably Approximately Correct). A seção 6 fornece uma visão geral do ambiente AMPSAM , e a seção 7 mostra a implementação das funções mais importantes desse ambiente.

2 Sistemas de Aprendizado e Classificadores

O objetivo fundamental de um sistema de Aprendizado de Máquina — AM — está em extrair conhecimento de um conjunto de dados, isto é, construir um classificador ou preditor, na forma de, por exemplo, regras de decisão, tal que essas regras possam ser aplicadas a novos dados — Figura 1. Um sistema de aprendizado típico é projetado para trabalhar sob algum paradigma de aprendizado, tais como o paradigma simbólico, estatístico, conexionista, e outros. "Aprender" consiste em escolher ou

adaptar parâmetros da representação do modelo, no paradigma escolhido, de forma que o modelo classifique corretamente os exemplos conhecidos bem como novos exemplos [Weiss 91]. De certa forma, classificação pode ser resumida em encontrar uma função que mapeie objetos com características comuns, na mesma classe.

Figura 1: Sistema de Aprendizado

Sob a óptica do problema de classificação, um sistema de aprendizado pode ser visto como um sistema de alto nível que ajuda a construir um sistema de tomada de decisões chamado de *classificador*. A Figura 2 ilustra a estrutura simples de um sistema de classificação que aceita um padrão de dados como entrada e produz uma decisão como saída.

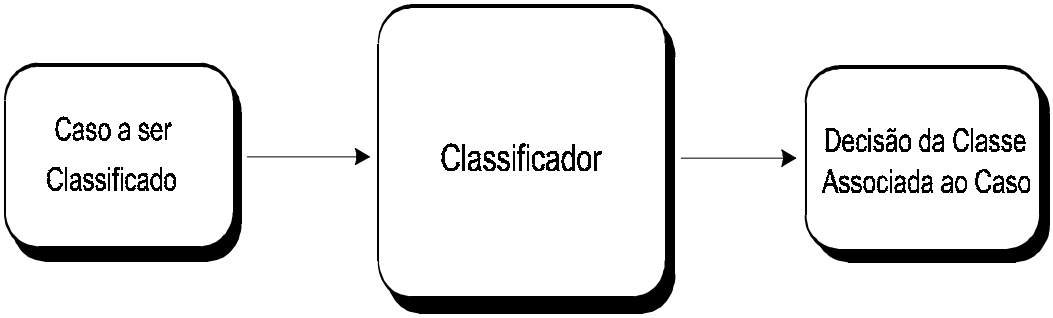

Figura 2: Sistema de Classificação

No caso do sistema de aprendizado ser do tipo *supervisionado*, objeto de estudo deste trabalho, o sistema tem disponível um conjunto finito de exemplos de casos resolvidos. Cada exemplo de um problema resolvido consiste de observações e a correta classe correspondente[2].

Desta forma, o conjunto de exemplos contém os dados que o sistema de aprendizado irá usar para encontrar, por exemplo, as regras de decisão generalizadas para o classificador. A Figura 3 ilustra os elementos representacionais de uma coleção de casos resolvidos, e a

[2]No caso do sistema de aprendizado não ter disponível a classe a qual pertence cada exemplo, o sistema é denominado *não supervisionado*. Neste caso, o objetivo do sistema é estabelecer a existência de classes ou *clusters* nos dados.

relação entre o sistema de aprendizado e o classificador. Cada observação, do conjunto de observações, potencialmente relevante à solução do problema é chamada de *atributo*. Atributos também são chamados de *features*, variáveis, testes e medidas.

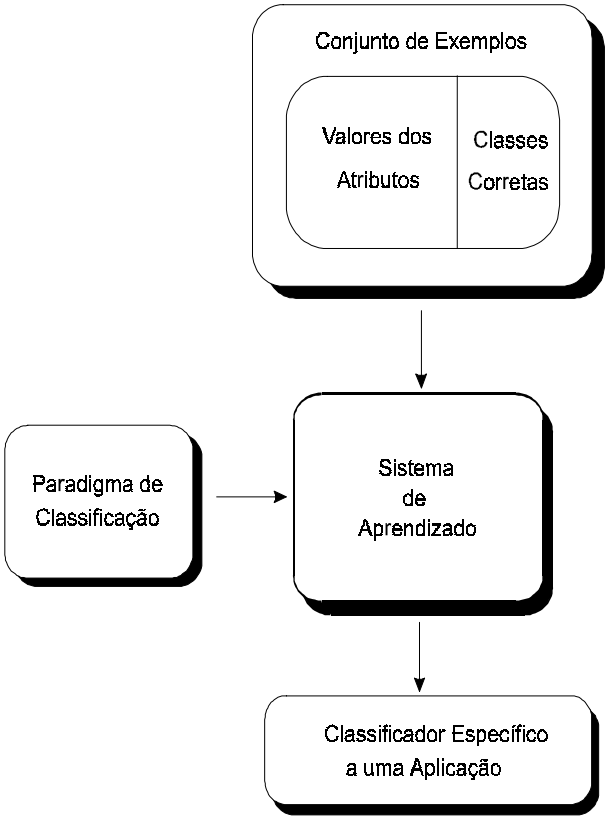

Figura 3: Representação da Classificação

Como somente casos corretamente resolvidos serão utilizados na construção de um classificador específico, o padrão de valores de atributos para cada caso é associado com uma classificação correta. Desta forma, aprender neste tipo de sistema pode ser visto como um processo de generalização das associações observadas empíricamente, mas sob restrições impostas pelo paradigma de classificação escolhido.

Entretanto, o principal objetivo a ser atingido por um sistema de aprendizado está em classificar com sucesso casos nunca vistos. Este objetivo se torna mais problemático quando estão disponíveis apenas pequenos conjuntos de exemplos para treinamento. Nestas situações não se sabe ao certo se tal número reduzido de exemplos é ou não suficiente para extrair conceitos genéricos. Este tipo de indagação, entre outros, motivou a pesquisa de métodos capazes de apresentar índices que estimam a performace de classificadores para exemplos não vistos.

Existem várias formas de estimar a performace de um sistema de aprendizado. Algumas formas são muito melhores que outras. Em termos estatísticos, certos métodos são considerados *tendensiosos*. Alguns tendem a apresentar estimativas de performace muito otimistas e outros estimativas muito pessimistas. Encontrar algum índice capaz

de estimar verdadeiramente a performace de um sistema de aprendizado é a base para uma comparação adequada dos diversos sistemas de aprendizado.

3 A Taxa de Erro

Quando é fornecido um padrão de entrada a um classificador, este responde com a classe a que supostamente este padrão pertence. Algumas vezes o classificador responde corretamente, outras vezes não. Seguramente, a taxa de erro é a forma mais utilizada para medir a performace de um classificador. Pode-se calcular a taxa de erro através da equação

$$taxa\ de\ erro = \frac{número\ de\ erros}{número\ de\ casos} \tag{1}$$

É conhecida por *taxa de erro aparente* de um classificador a taxa de erro calculada utilizando somente os exemplos de treinamento, ou seja, inicialmente o sistema é treinado com um conjunto de exemplos e posteriormente é verificado quantos erros o classificador cometeu classificando os mesmos exemplos com que foi treinado. No outro extremo encontra-se a *taxa de erro verdadeira*, que pode ser entendida como a taxa de erro obtida sobre um número muito grande de novos casos, selecionados independentemente dos casos usados para treinar o classificador. A taxa de erro verdadeira difere da taxa de erro aparente, pois a segunda exige um grande número de exemplos para testar o classificador, independentes dos exemplos de treinamento .

A taxa de erro verdadeira fornece uma excelente medida da performace de um classificador. Entretanto, em casos reais, conjuntos com grande número de exemplos são raros, o que inviabiliza o seu cálculo na maioria das vezes. Resta, então, saber se é possivel aproximar a taxa de erro verdadeira através de taxas de erro empíricas calculadas com pequenos conjuntos de exemplos.

Uma tentativa inicial seria tentar estimar a taxa de erro verdadeira através da taxa de erro aparente. Entretanto, é bastante comum que os classificadores possuam uma taxa de erro aparente muito inferior a taxa de erro verdadeira, pois pode ocorrer o processo de overfitting. Esse processo ocorre quando o classificador se torna extremamente especializado aos exemplos de treinamento, resultando em baixas taxas de erro aparente, mas incapaz de classificar corretamente novos casos. Desta forma, se a taxa de erro aparente fosse utilizada, a estimativa seria pouco confiável e extremamente otimista se comparada com a taxa de erro verdadeira.

Através de técnicas estatísticas que apresentam de diferentes formas os exemplos ao classificador é possível obter aproximações da taxas de erro verdadeira bem mais confiáveis que a taxa de erro aparente [Dietterich 97, Salzberg 97, Michie 94]. Tais técnicas são objetos de discussão das próximas seções.

4 Tipos de Erros

Se todos os erros são de igual importância, a performace de um classificador pode ser calculada através da equação 1. Entretanto, distinções entre os tipos de erros são importantes para muitas aplicações. Por exemplo, o erro cometido no diagnóstico de uma pessoa como sendo sadia, quando na verdade esta pessoa está correndo risco de vida (conhecido como decisão *falsa positiva*) é muito mais grave que o tipo oposto de erro, ou seja, diagnosticar alguém doente quando, na verdade, esta pessoa está sadia (conhecido como *falso positivo*). Freqüentemente, com testes futuros e o passar do tempo o falso diagnóstico da pessoa sadia será corrigido e nenhum dano sério será causado a sua saúde. Entretanto, a pessoa doente que foi encaminhada para casa poderá adoecer, e no pior dos casos até mesmo morrer, o que pode tornar o erro muito sério.

Se é necessário fazer distinções entre os tipos de erros, então a *matriz de confusão* pode ser usada para separar os diferentes erros. A Tabela 1 mostra um exemplo deste tipo de matriz para três classes. O número de predições corretas para cada classe está representada ao longo da diagonal da matriz. Todos os outros números são erros de diferentes tipos. Por exemplo, na Tabela 1 a classe ♠ foi corretamente classificada 53 vezes, mas foi erroneamente classificada como classe ♢ 5 vezes.

	Classes Verdadeiras		
Classes Predizidas	♣	♠	♢
♣	36	0	1
♠	0	53	2
♢	1	5	72

Tabela 1: Exemplo de Matriz de Confusão para Três Classes

Problemas com apenas duas classes são mais comuns, pois geralmente se está interessado em predizer a ocorrência ou a não-ocorrência de um simples evento ou hipótese. A Tabela 2 mostra os tipos de erros que podem ocorrer através de uma matriz de confusão para duas classes.

	Classe Positiva	*Classe Negativa*
Predição Positiva	Verdadeiro Positivo	Falso Positivo
Predição Negativa	Falso Negativo	Verdadeiro Negativo

Tabela 2: Tipos de Erros para Classificação com Duas Classes

Quando é necessário distinguir entre os diferentes tipos de erros, o *custo de classificação incorreta* é uma alternativa natural à taxa de erro. Neste caso, ao invés de projetar um classificador que minimize a taxa de erro, o objetivo está em minimizar os custos de classificação incorreta. O custo de classificação incorreta é simplesmente uma penalização pela ocorrência de um erro. Por exemplo, no caso com duas classes, um custo pode ser associado ao erro falso positivo, e outro custo pode ser associado ao erro falso negativo. Um custo médio de classificação incorreta pode ser obtido multiplicando

cada custo pela respectiva taxa de erro. Mais formalmente, para qualquer matriz de confusão, se E_{ij} é o número de erros na matriz de confusão e C_{ij} é o custo para cada tipo de classificação incorreta, o custo total de classificação incorreta C_{total} é dado pela equação 2, onde o custo de classificação correta (C_{ij}, para $i = j$) é 0.

$$C_{total} = \sum_{i=1}^{n} \sum_{j=1}^{n} E_{ij} C_{ij} \tag{2}$$

Para exemplificar a utilização do custo para medir a performace de um classificador, suponha que um sistema de aprendizado foi treinado para aprender a classificar três classes: ♣, ♠ e ♢. A Tabela 3 mostra a matriz de custo definida para o problema.

	Classes Verdadeiras		
Classes Predizidas	♣	♠	♢
♣	0	3	5
♠	2	0	1
♢	5	2	0

Tabela 3: Exemplo de Matriz de Custo de Classificação Incorreta para Três Classes

Desta forma, considerando os valores apresentados nas Tabelas 1 e 3, o custo total (C_{total}) do classificador é

$$C_{total} = (36*0)+(0*3)+(1*5)+(0*2)+(53*0)+(2*1)+(1*5)+(5*2)+(72*0) = 22$$

e o custo médio por decisão ($C_{médio}$) é

$$C_{médio} = \frac{22}{170} \cong 0.13$$

É importante resaltar que este resultado é muito diferente do resultado que seria obtido se os custos fossem considerados todos iguais a 1, neste caso o custo total (C_{total}) seria 9 e o custo médio por decisão ($C_{médio}$) seria aproximadamente 0.05.

Até então foi considerada somente a possibilidade de penalizar as classificações incorretas, mas é possível também estabelecer ganhos para classificações corretas. Na *análise de risco* ou *análise de decisão*, ambos custos e benefícios são usados para avaliar a performace do classificador.

Neste tipo de análise, os custos são chamados de *riscos*. Se os custos de classificação incorreta são associados a números negativos, e os ganhos para decisões corretas são associados a números positivos, então a equação 2 pode ser reescrita em termos de riscos, da seguinte forma

$$R_{total} = \sum_{i=1}^{n} \sum_{j=1}^{n} E_{ij} R_{ij} \tag{3}$$

Entretanto, no caso de aplicações reais, é muito difícil estabelecer os valores correspondentes a custos e riscos para avaliar a performace de um classificador. Por isso a taxa de erro é mais utilizada como padrão para medir a performace de classificadores. Contudo, deve ser resaltado que as técnicas descritas neste trabalho para estimar a taxa de erro verdadeira podem ser facilmente modificadas para estimar custos e riscos verdadeiros, sempre que esta informação esteja disponível.

5 Técnicas para Estimar a Taxa de Erro Verdadeira

Uma vez que a taxa de erro aparente não é uma boa estimativa para a taxa de erro verdadeira, alguma alternativa deve ser encontrada. O termo *estimativa de taxa de erro honesta* é muitas vezes utilizado, mas não deve ser mal interpretado, pois se refere a estimadores imprecisos e não desonestos. Algumas vezes, taxas de erro aparente têm sido usadas para estimar a performace de classificadores, mas freqüentemente estes fatos estão relacionados com a falta de familiaridade com métodos estatísticos adequados para estimação de erros.

Um dos requisititos para estimar a taxa de erro verdadeira é manter a amostra de exemplos em ordem aleatória. Isto significa que a amostra de exemplos não deve ser pre-selecionada, de forma a evitar que seja feita qualquer suposição sobre qualidade de certos exemplos.

5.1 Probably Approximately Correct (PAC)

Como já mencionado, a performace de um classificador é geralmente medida através de sua taxa de erro. Se estão disponíveis um número ilimitado de casos para testar e treinar, a taxa de erro aparente será a taxa de erro verdadeira. Infelizmente, conjuntos reais de exemplos são finitos e geralmente pequenos. Desta forma, seria interessante responder a indagação de quantos casos são necessários para que a taxa de erro aparente se torne efetivamente a taxa de erro verdadeira.

Existem alguns resultados teóricos nesta área. Dada uma amostra retirada de uma população, e uma taxa de erro relativamente baixa, quantos casos são necessários na amostra de forma que o número de erros para novos casos seja também um valor baixo? Tipicamente, a taxa de erro em novos casos não supera em duas vezes a taxa de erro para os exemplos da amostra. É importante notar que a questão foi feita independentemente de qualquer distribuição da população, desta forma não foi assumida nenhuma característica da amostra.

Este tipo de análise teórica é conhecida como *Probably Approximately Correct (PAC)* [Kearns 94], e muitos sistemas de aprendizado, como as redes neurais, têm sido analisados através deste critério. A análise PAC é uma análise de pior caso. Para todas as possíveis distribuições resultantes de um conjunto de amostras, ela garante que os resultados de classificação serão corretos com uma pequena margem de erro. Embora este tipo de análise forneça limites teóricos interessantes para taxas de erro, mesmo

para classificadores simples os resultados indicam que um grande número de casos são necessários para garantir a performace.

Baseados nestes resultados teóricos, pode-se ficar desencorajado em tentar estimar a taxa de erro verdadeira para um classificador. Mas resultados melhores podem ser obtidos para problemas reais, uma vez que dada uma amostra de uma simples população, a tarefa está em estimar a taxa de erro para aquela população, não para todas as possíveis populações. Este tipo de análise requer muito menos casos, uma vez que uma única distribuição da população é considerada. Além disso, em vez de todos os casos serem usados para estimar a taxa de erro verdadeira, os casos podem ser particionados em dois grupos, alguns utilizados para projetar o classificador, outros para testar o classificador. Apesar de que esta forma de análise não fornece garantias de performace para todas as possíveis distribuições, ela fornece uma estimativa da taxa de erro verdadeira para a população considerada. Ao contrário da análise PAC, o número de casos necessários para os testes é supreendentemente pequeno.

Na próxima seção será apresentado o paradigma de *treinar-e-testar* que é capaz de estimar, com maior credibilidade, a taxa de erro verdadeira para problemas reais.

5.2 Estimação da Taxa de Erro Através de Treinar-e-Testar

O princípio básico que envolve os métodos de treinar-e-testar está em dividir a amostra de exemplos em dois grupos mutuamente exclusivos, como ilustrado na Figura 4. O primeiro, chamado de *conjunto de treinamento*, é usado para treinar o sistema de aprendizado. O segundo, chamado de *conjunto de teste*, é usado exclusivamente para medir a taxa de erro do classificador, chamada de *taxa de erro do conjunto de teste*. Se o conjunto de teste for "escondido" do classificador e mostrado somente depois do treinamento, então tem-se uma correspondência procedural direta à tarefa de determinar a taxa de erro em novos casos.

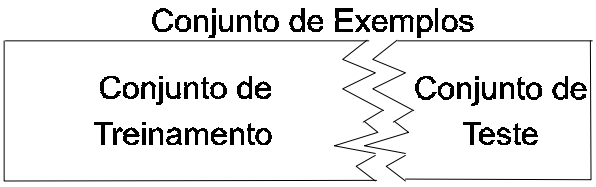

Figura 4: Conjuntos de Treinamento e Teste

Os dois conjuntos de casos devem ser amostras aleatórias de alguma população. Em adição, os casos dos dois conjuntos de amostras devem ser independentes. Independente significa que não há relações entre os conjuntos com excessão que ambos são amostras de uma mesma população. Para certificar que os conjuntos são independentes, eles podem ser coletados em datas diferentes ou por investigadores diferentes.

Na seção 5.1, uma questão muito abrangente foi levantada a respeito do número de casos que devem pertencer a uma amostra para garantir uma performace equivalen-

te no futuro. Se nenhuma suposição for feita a respeito da verdadeira distribuição da população, os resultados não são muito satisfatórios pois é necessário um número muito grande de casos. Entretanto, se conjuntos independentes de treinamento e teste são utilizados, excelentes resultados práticos são conhecidos. Com esta representação, surgem as seguintes questões

Quantos casos de teste são necessários para uma estimação de erro precisa?

A mesma questão pode ser reescrita como

Quantos casos de teste são necessários para que a taxa de erro do conjunto de teste seja essencialmente a taxa de erro verdadeira?

Surpreendentemente, pode-se obter excelentes aproximações da taxa de erro verdadeira, mesmo com amostras de exemplos pequenas. A Figura 5 apresenta um gráfico que mostra a relação entre a taxa de erro do conjunto do teste e a mais alta possível taxa de erro verdadeira, para exemplos de teste de vários tamanhos. Estes valores possuem 95% de confiabilidade, isto é, não há mais de 5% de probabilidade que a taxa de erro exceda os valores apresentados. Por exemplo, para um conjunto de teste de tamanho 50, se a taxa de erro do conjunto do teste apresentar um valor de 0%, então há uma grande probabilidade que a taxa de erro verdadeira seja no máximo de 10%. Para 1000 casos de teste e taxa de erro do conjunto de teste de 0%, certamente a taxa de erro verdadeira ficará abaixo de 1%. Estes resultados não foram conjecturados, mas sim derivados de considerações básicas de probabilidade e estatística.

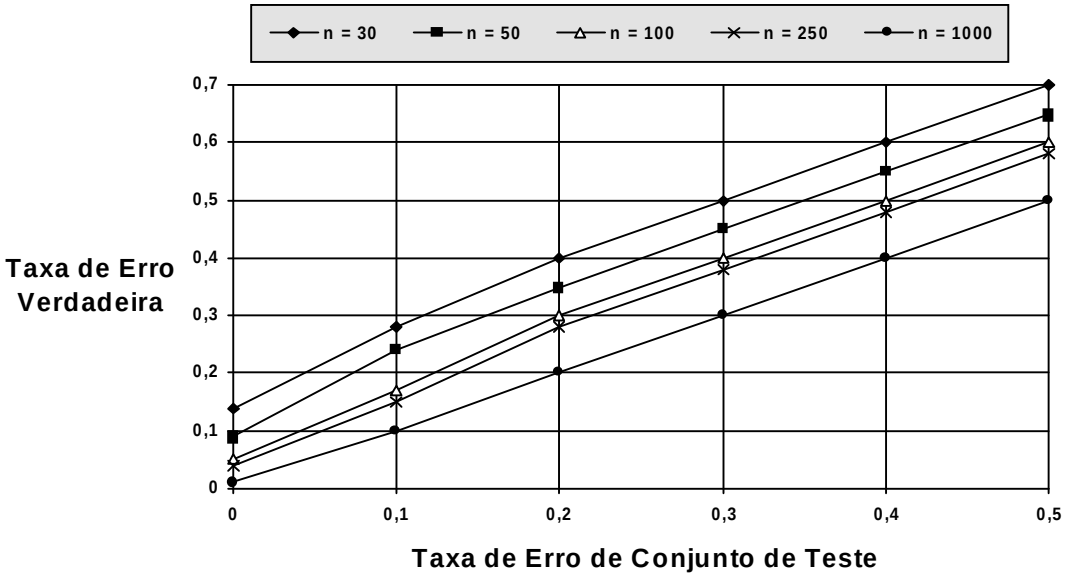

Figura 5: Número de Casos de Teste e Qualidade da Predição

Portanto, a qualidade da estimativa por taxa de erro do conjunto de teste é diretamente dependente do número de casos do conjunto de teste. Quando o tamanho do conjunto de teste atinge 1000 casos, a estimativa já é bastante precisa. Com 5000 casos, a taxa de erro do conjunto de teste é virtualmente idêntica à taxa de erro verdadeira.

Na prática o número de exemplos disponível é freqüentemente restrito. Desta forma, não só um número suficiente de casos de teste, mas também a forma de como devem ser escolhidos os casos para treinamento e teste é de essencial importância. Uma pergunta que ainda falta ser respondida é

Quantos casos deve conter cada conjunto de teste e treinamento?

Existem várias formas conhecidas para dividir a amostra em conjuntos de treinamento e teste. O *Método Holdout* ou *Método H* divide fixamente a amostra de exemplos, geralmente em $\frac{2}{3}$ para treinamento e $\frac{1}{3}$ para teste. Quando o número de casos para teste é grande (excede 1000 casos) costuma-se aumentar a fração do conjunto de treinamento.

Deve ser salientado que o método holdout apresenta uma taxa de erro do conjunto de teste que aproxima a taxa de erro verdadeira para grandes amostras de exemplos. Entretanto, para pequenas amostras de exemplos, os conjuntos de treinamento e teste ficam com tamanhos muito reduzidos, a aproximação se torna pessimista e o método holdout não apresenta resultados satisfatórios. Desta forma, quando o conjunto de exemplos disponível não é grande, podem ser aplicadas variações de treinar-e-testar conhecidos por métodos *resampling*. Esses métodos serão discutivos a seguir.

5.3 Métodos Resampling

O método holdout é a técnica mais simples de estimativa de erro "honesta" conhecida e consiste em realizar somente um experimento treinar-e-testar. Entretanto uma única partição pode levar a resultados imprecisos, principalmente para pequenas e médias amostras de exemplos. Os métodos de resampling são variações dos métodos de treinar-e-testar nos quais são realizados vários experimentos de treinar-e-testar com diferentes partições dos exemplos. A seguir são descritos os seguintes métodos de resampling

- Random subsampling
- Cross-validation
- Bootstraping

5.3.1 Random Subsampling

Quando múltiplos experimentos de treinar-e-testar são realizados, um novo classificador é projetado com cada conjunto de treinamento. A taxa de erro estimada é a média das taxas de erros dos classificadores calculada a partir de conjuntos de teste

independentes e aleatórios. O método que realiza tal procedimento é conhecido como *Random Subsampling* e pode produzir melhores resultados que uma simples partição treinar-e-testar.

A Tabela 4 mostra uma comparação entre os métodos Holdout e random subsampling para uma amostra de n casos. O método random subsamping soluciona um problema típico do método holdout, a escolha de uma partição dos exemplos que não é representativa aos conceitos. Com o método random subsampling este problema é contornado pois são criadas várias partições e o sistema é treinado e testado a cada iteração com uma partição diferente. No caso de uma partição ser pouco representativa, tal problema pode ser amenizado pois a taxa de erro final é calculada através das médias das taxas de erro de todas as partições.

	Holdout	*Random Subsampling*
Casos Treinamento	j	j
Casos de Teste	$n-j$	$n-j$
Iterações	1	Num. Iterações $\ll n$

Tabela 4: Comparação entre Holdout e Random Subsampling

5.3.2 Cross-validation

Cross-validation não é um único método, mas sim uma classe de métodos para estimar a taxa de erro verdadeira. Também são conhecidos como *k-fold cross-validation*, no qual k representa o número de partições geradas aleatóriamente a partir da amostra de exemplos para treinar e testar o sistema. Nestes métodos, a amostra de exemplos é dividida em k partições mutuamente exclusivas. A cada iteração, uma partição diferente é usada para testar o sistema de aprendizado e todas as outras $k-1$ partições são utilizadas para treinar o sistema. A taxa de erro é a média das taxas de erro calculadas para as diversas partições. Por exemplo, no caso do 10-fold cross-validation todo o conjunto de exemplos é dividido aleatoriamente em 10 partições diferentes mutuamente exclusivas. São realizadas 10 iterações, a cada iteração uma partição diferente é utilizada para testar e as outras 9 partições restantes são utilizadas para treinar o sistema. A taxa de erro é resultado da médias das taxas de erro das 10 iterações.

O método *Leaving-one-out* é um caso especial da classe de métodos cross-validation. Neste método, para uma amostra de n exemplos, são realizadas n iterações. A cada iteração um exemplo diferente é retirado do conjunto de exemplos e utilizado para testar o classificador. Os outros $n-1$ exemplos são usados para treinar o sistema. A taxa de erro é calculada dividindo-se o número de erros observados por n.

O leaving-one-out é muito utilizado pois sua estimativa de taxa de erro verdadeira é praticamente não tendenciosa, isto é, quando usados vários conjuntos de exemplos, a média das estimativas tenderá a taxa de erro verdadeira.

Por ser computacionalmente caro, o leaving-one-out é geralmente utilizado somente em pequenos conjuntos de exemplos. Para conjuntos maiores, o k-fold cross-validation,

com $k = 10$, apresenta uma aproximação quase tão apurada quanto o leaving-one-out, a um custo muito inferior. A Tabela 5 mostra uma comparação entre estes dois métodos de cross-validation.

	Leaving-One-Out	*10-fold Cross-Validation*
Casos Treinamento	$n - 1$	90%
Casos de Teste	1	10%
Iterações	n	10

Tabela 5: Comparação entre Métodos de Cross-validation

5.3.3 Bootstraping

O problema de encontrar o melhor estimador para pequenas amostras é particularmente intrigante uma vez que pequenas amostras são bastante comuns. Tradicionalmente são consideradas pequenas amostras aquelas que possuem 30 ou menos casos. Por muitos anos, o leaving-one-out foi a técnica recomendada para avaliar a performace de classificadores para pequenas amostras. Uma vez que o leaving-one-out é virtualmente não tendencioso, ele pode ser aplicado a amostras muito maiores, apresentando resultados precisos.

Embora o leaving-one-out seja um estimador quase não tendensioso da taxa de erro verdadeira, ainda existem dificuldades com esta técnica. Tendência e variância de um estimador de taxa de erro contribuem para a imprecisão da taxa de erro estimada. Enquanto que o leaving-one-out é praticamente não tendensioso, sua variância para pequenas amostras é alta. Lembre-se que não tendensioso significa que o estimador, após várias aplicações, tende para a taxa de erro verdadeira. Porém a variância do leaving-one-out é alta, o que pode ser comparado a um bêbado tentando andar sobre uma linha reta. Esta pessoa pode apresentar uma média exatamente sobre a linha, mesmo quando constantemente cambalea para a direita e esquerda da linha.

O efeito da variância tende a dominar em pequenas amostras. Desta forma, um estimador com baixa variância, mesmo que tendensioso, é potencialmente superior ao leaving-one-out para pequenas amostras.

Para amostras pequenas, o bootstrapping, um novo método de resampling, tem mostrado resultados promissores como estimador de taxa de erro. Esta é uma área de pesquisa ativa em estatística aplicada.

Existem vários estimadores bootstrap, mas dois estimadores em particular têm mostrado os melhores resultados, o *e0 bootstrap* e o *.632 bootstrap*. Para o estimador $e0$, o conjunto de treinamento consiste de n casos amostrados com substituição a partir de uma amostra de tamanho n. Amostrados com substituição significa que os exemplos de treinamento são retirados do conjunto inicial de exemplos mas cópias dos elementos retirados permanecem no conjunto inicial, de forma que elementos repetidos podem ser amostrados.

A fim de exemplificar, considerando-se existir uma amostra de casos contendo apenas

sete casos

$$\text{Amostra de casos: } \boxed{1\ 2\ 3\ 4\ 5\ 6\ 7}$$

Para contruir o conjunto de treinamento deve-se sortear um caso aleatóriamente da amostra de casos. Considerando que o caso número 6 é sorteado, ele é copiado para o conjunto de treinamento. Os casos devem ser copiados e não retirados da amostra inicial, desta forma a amostra inicial deve sempre conter os mesmos casos. Novamente, outros casos são sorteados aleatóriamente e copiados, existindo a possibilidade do caso número 6 ser copiado mais de uma vez para o conjunto de treinamento. Os casos são copiados até que o conjunto de treinamento possua o mesmo número de casos que a amostra de casos inicial, ou seja, sete casos. Por exemplo, o conjunto de treinamento poderia conter

$$\text{Conjunto de treinamento: } \boxed{1\ 2\ 3\ 3\ 3\ 6\ 6}$$

O conjunto de teste é composto por todos os casos que não pertencem ao conjunto de treinamento, neste exemplo este conjunto seria composto pelos seguintes casos

$$\text{Conjunto de teste: } \boxed{4\ 5\ 7}$$

Esta técnica possui uma fração média de casos não repetidos para o conjunto de treinamento de 0.632, e uma fração média de 0.368 para casos do conjunto de teste. O estimador .632 bootstrap é a combinação linear simples de

$$.368 * app + .632 * e0$$

Onde *app* é a taxa de erro aparente para todos os casos (treinamento e teste). Deve ser notado que $e0$ pode ser aproximando através da repetição do método 2-fold cross-validation, isto é, a divisão meio a meio da amostra inicial para treinamento e teste.

A taxa de erro final é calculada através da média das taxas de erro das várias iterações. Cerca de 200 iterações são consideradas necessárias para obter uma boa estimativa com o bootstrap. Portanto, este método pode ser considerado mais caro computacionalmente que o método leaving-one-out. A Tabela 6 resume as características básicas do método bootstrap.

	Bootstrap
Casos Treinamento	n (com j casos não repetidos)
Casos de Teste	$n - j$
Iterações	200

Tabela 6: Características dos Estimadores Bootstrap

6 O Ambiente Computacional AMPSAM

O ambiente computacional AMPSAM tem como objetivo principal automatizar, ao máximo, o processo de medir a performance de sistemas de aprendizado. Para isto, o ambiente provê um conjunto de métodos estatísticos capazes de estimar, com confiabilidade, a *taxa de erro verdadeira* — seção 3, pg 4. O problema geral para medir a performance de algoritmos de Aprendizado de Máquina por Exemplos, ou seja, encontrar uma boa estimativa da taxa de erro verdadeira é o seguinte [Batista 97b]

Dados

- O algoritmo A_a
- Um conjunto de n exemplos na sintaxe entendida pelo algoritmo A_a

utilizar alguma(s) técnica(s) estatística(s) para separar esses n exemplos disponíveis em diversos conjuntos de exemplos para treinamento e teste a fim de obter aproximações confiáveis para calcular a taxa de erro verdadeira do algoritmo A_a.

A Figura 6 ilustra este processo no qual o conjunto dos n exemplos disponíveis é separado, segundo alguns dos critérios estatísticos discutidos na seção 5.2, pg 8, em k pares

$$(TR_i, TE_i) \quad i = 1 \ldots k \tag{4}$$

de conjuntos de treinamento e teste respectivamente.

O algoritmo A_a é então treinado com cada um dos conjuntos de treinamento TR_i, produzindo o respectivo classificador $C_i(A_a)$ que logo após é alimentado com o conjunto de teste TE_i correspondente medindo assim o erro de classificação E_i do classificador $C_i(A_a)$, $i = 1 \ldots k$.

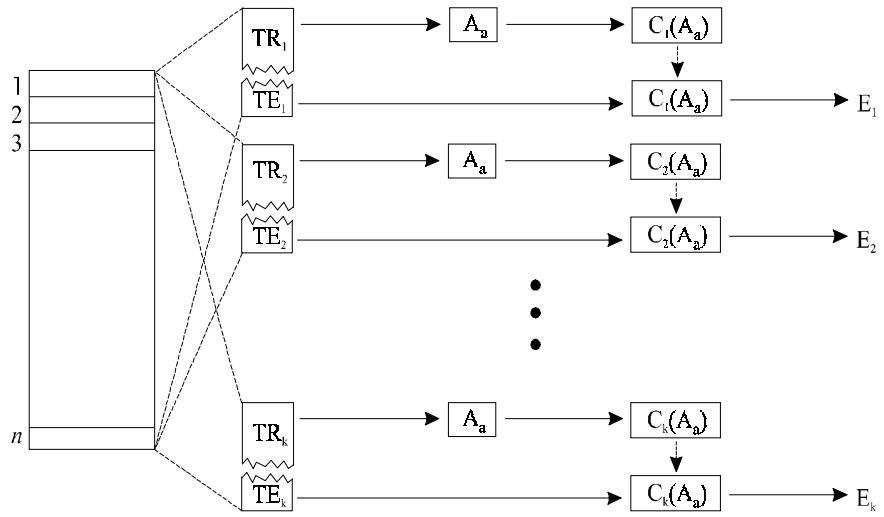

Figura 6: Processo de Particionamento dos Exemplos

Ao final deste processo, tem-se como resultado as taxas de erro E_i obtidas através da aplicação dos conjuntos de teste TE_i aos classificadores $C_i(A_a)$. A média das taxas de erro E_i é a estimativa da taxa de erro verdadeira. Por exemplo, pode-se utilizar a média aritmética para calcular o valor médio dos valores E_i encontrados, desta forma

$$m(E) = \frac{1}{k}\sum_{i=1}^{k} E_i \tag{5}$$

Outras duas medidas estatísticas conhecidas como variância e desvio padrão são importantes pois fornecem um índice de confiabilidade das medidas realizadas. Pode-se definir a variância da seguinte forma

$$\tau^2 = \frac{1}{k}\sum_{i=1}^{k}(E_i - m(E))^2 \tag{6}$$

e o desvio padrão

$$\tau = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(E_i - m(E))^2} \tag{7}$$

Como foi dito anteriormente, os métodos estatísticos de resampling criam diversas partições de treinamento e teste a partir de um conjunto inicial de exemplos. Na prática, estes conjuntos iniciais de exemplos estão armazenados na forma de arquivos. De uma forma geral, os arquivos de exemplos da maioria dos algoritmos de aprendizado de máquina proposicionais compartilham de um formato parecido com uma tabela. Os diferentes atributos são expressos em colunas e a classe está frequentemente posicionada na primeira ou na última coluna, por sua vez, cada exemplo está situado em uma linha diferente. Apesar de apresentar este formato geral, os arquivos de exemplos proposicionais não apresentam uma sintaxe comum, por exemplo, pode-se encontrar diferentes símbolos separando cada atributo tais como vírgulas, espaços ou tabulações.

Desta forma, foi decidido definir uma sintaxe padrão, utilizada pelo ambiente AMPSAM, para arquivos de exemplos. Assim, foram implementados conversores sintáticos que transformam os arquivos na sintaxe padrão para a sintaxe de um determinado sistema de aprendizado. Esses conversores sintáticos, descritos em [Batista 97d], fazem parte do ambiente AMPSAM.

Entretanto, para ilustrar o funcionamento dos métodos estatísticos implementados não são necessárias conversões sintáticas. Os métodos estatísticos implementados são independentes de qualquer sintaxe específica, pois eles consideram que cada exemplo está posicionado em uma linha do arquivo de exemplos, e manipulam cada linha de forma independente.

7 Implementação

Esta seção apresenta as implementações dos métodos estatísticos descritos nas seções anteriores. Estas implementações foram realizadas na linguagem de programação C

```
ERROR: undefined
OFFENDING COMMAND: sko

STACK:

0
0
0
```