Universidade de São Paulo (USP)
Universidade Federal de São Carlos (UFSCar)
Universidade Metodista de Piracicaba (Unimep)

Relatório Técnico "Conceitos sobre Aprendizado de Máquina"

Projeto "Um Ambiente para Análise de Dados da Doença Anemia Falciforme"

Pablo Freire Matos (UFSCar)
Leonardo de Oliveira Lombardi (Unimep)
Prof. Dr. Ricardo Rodrigues Ciferri (UFSCar)
Prof. Dr. Thiago Alexandre Salgueiro Pardo (USP/ICMC)
Profª. Drª. Cristina Dutra de Aguiar Ciferri (USP/ICMC)
Profª. Drª. Marina Teresa Pires Vieira (Unimep)
pablo_matos@dc.ufscar.br, lolombardi@unimep.br, ricardo@dc.ufscar.br,
{taspardo, cdac}@icmc.usp.br, mtvieira@unimep.br

São Carlos
Novembro/2009

RESUMO

Este relatório técnico visa apresentar as principais técnicas da abordagem baseada em aprendizado de máquina (AM). O foco de estudo deste trabalho é o aprendizado supervisionado no qual as classes estão previamente definidas. O clássico algoritmo Naïve Bayes é utilizado como um exemplo do aprendizado supervisionado. Busca-se com este relatório propiciar aos docentes, discentes, pesquisadores e pessoas interessadas no AM a conhecerem as medidas de desempenho utilizadas para avaliar um classificador, quais os possíveis métodos de particionamento que podem ser utilizados e algumas das técnicas de seleção de características utilizadas com o objetivo de reduzir a dimensionalidade dos dados.

ufscar
UNIMEP
Universidade Metodista de Piracicaba
USP

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
FD	Frequência de Documento
GI	Ganho de Informação
IM	Informação Mútua

SUMÁRIO

1 Introdução

Este relatório técnico tem por objetivo apresentar as principais técnicas utilizadas em uma abordagem baseada em aprendizado de máquina (AM). Este conhecimento é necessário para os integrantes do projeto "Um Ambiente para Análise de Dados da Doença Anemia Falciforme" compreenderem como é realizado aprendizado de máquina supervisionado. Este trabalho está sendo desenvolvido em conjunto com a Universidade de São Paulo (Campus de Ribeirão Preto e São Carlos), Fundação Hemocentro de Ribeirão Preto, Universidade Federal de São Carlos e Universidade Metodista de Piracicaba.

Os dados iniciais da doença Anemia Falciforme (PINTO et al., 2009) foram definidos em cinco classes identificados como importantes para serem utilizados na extração de informação: paciente, sintoma, fator de risco, tratamento e efeitos. Como objeto de estudo deste trabalho – que visa extrair informação de artigos científicos relacionados a essa doença – são utilizadas as seguintes classes: fator de risco, efeitos positivo e negativo.

Como as classes já estão definidas, o interesse deste trabalho encontra-se no aprendizado supervisionado, mais especificamente na classificação que trabalha com rótulos de classes discretos (e.g., paciente normal, paciente com doença A), diferentemente da regressão que lida com valores contínuos (e.g., pacientes maiores de 18 anos com altura de 1,8 metros).

2 Aprendizado de Máquina

Aprendizado de Máquina (AM) é uma área da Inteligência Artificial que lida com problemas de aprendizado computacional a fim de adquirir conhecimento de forma automática. Um sistema de aprendizado tem a função de analisar informações e generalizá-las, para a extração de novos conhecimentos. Para isso usa-se um programa de computador para automatizar o aprendizado (MONARD; BARANAUSKAS, 2003).

O aprendizado utiliza do princípio da indução (inferência lógica) com o intuito de obter conclusões genéricas a partir de um conjunto de exemplos. Um conceito é aprendido efetuando-se inferência indutiva sobre os exemplos apresentados. As hipóteses geradas através dessa inferência podem ou não preservar a verdade.

Para a indução derivar conhecimento novo representativo, os exemplos das classes têm que estar bem-definidos e ter uma quantidade suficiente de exemplos, obtendo assim hipóteses úteis para um determinado tipo de problema. Quanto mais exemplos relevantes selecionados para treinamento no indutor, mais bem classificado será o novo conjunto de dados. O objetivo do algoritmo de indução é construir um classificador que possa determinar a classe que um exemplo não rotulado pertence. É possível rotular um novo exemplo devido à generalização.

O aprendizado indutivo pode ser dividido em supervisionado (AS) e não-supervisionado (ANS), Figura 1. O AS é utilizado para classificação dos exemplos em classes predefinidas: resolve problemas preditivos. O ANS é utilizado para agrupamento, agrupando exemplos semelhantes: resolve problemas descritivos. Classificação e agrupamento são, respectivamente, exemplos desses dois tipos de aprendizado.

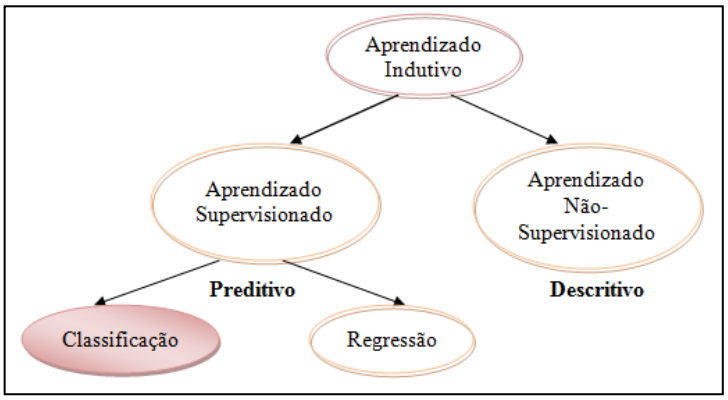

Figura 1 – Hierarquia do aprendizado.
Fonte: Adaptado de Monard e Baranauskas (2003).

Monard e Baranauskas (2003) classificam AM em alguns paradigmas, a saber: **Simbólico**, representações simbólicas de um problema através da análise de exemplos e contra-exemplos como expressão lógica, árvore de decisão, regras ou rede semântica. Exemplo: Algoritmos de árvore de decisão como ID3, C4.5; **Estatístico**, utiliza modelos estatísticos para encontrar uma aproximação do conceito induzido. Exemplo: *Support Vector Machines* (SVM) e aprendizado Bayesiano; **Baseado em Exemplos**, classifica um novo exemplo com base em uma classificação similar conhecida. Exemplo: Raciocínio baseado em caso e método do *k*-vizinhos mais próximos (*k-nearest neighbor*, *kNN*); **Conexionista**, inspirada no modelo biológico do sistema nervoso. Exemplo: Redes Neurais; e **Evolutivo**, modelo biológico de aprendizado. Exemplo: Analogia com a teoria de Darwin.

Alguns métodos típicos de classificação têm sido usados de forma bem-sucedida na classificação textual: kNN, métodos de seleção de características, SVM, classificação Bayesi-

ana e baseada em associação (HAN; KAMBER, 2006). Os algoritmos de aprendizado supervisionado C4.5, SVM, kNN, Naïve Bayes foram escolhidos entre os dez algoritmos mais influentes na área de mineração de dados (WU, X. et al., 2007).

O SVM pode ser utilizado para classificação e trabalha bem em espaço de alta dimensionalidade, atuando em problemas de duas classes, por exemplo, identificação de genes em pacientes normais e com câncer (GUYON et al., 2002). Naïve Bayes é fácil de interpretar e frequentemente funciona surpreendentemente bem; pode não ser o melhor classificador em alguma aplicação específica, mas normalmente é robusto. Xindong Wu et al. (2007) descrevem mais detalhes desses algoritmos.

3 Classificador Supervisionado: Naïve Bayes

Classificadores Bayesianos são classificadores estatísticos supervisionados, cuja função é predizer a probabilidade de um exemplo pertencer a uma determinada classe. Exemplo (instância ou padrão) é descrito por um vetor de valores (atributos) e pelo rótulo da classe associada. Neste trabalho, exemplo estará associado à *sentença* de um artigo e atributo a *termo* de uma determinada classe. Para mais informações sobre "exemplo", "atributo" e também sobre "conceito" consultar Witten e Frank (2005). É baseado no teorema de Bayes (apresentado em seguida), idealizado por Thomas Bayes, sacerdote e matemático inglês, que trabalhou com probabilidade e teoria da decisão durante o século $XVIII$.

Classificação Bayesiana é uma das muitas técnicas populares que pode ser usada para classificação de documento eficientemente. Considerado algoritmo de aprendizado indutivo eficiente e eficaz para AM e mineração de dados (ZHANG, H., 2004). Em alguns domínios o desempenho tem mostrado comparável com aprendizado de redes neurais e árvore de decisão (MITCHELL, 1997). É frequentemente utilizado em aplicações de classificação de texto devido à simplicidade e eficácia (IKONOMAKIS; KOTSIANTIS; TAMPAKAS, 2005).

Uma implementação simples do classificador Bayesiano é conhecido como Naïve Bayes (*Naïve* significa simples em inglês). O termo "simples" surge a partir da *independência condicional da classe*, isto é, o efeito de um valor de atributo de uma dada classe é independente dos valores de outros atributos.

3.1 Teorema de Bayes

Para a explicação do teorema de Bayes considera-se que o termo t ("*splenic sequestration*") é uma complicação e a sentença, no contexto de artigos científicos, representa informação de complicação da Anemia Falciforme.

O evento A ocorre quando uma sentença contém o termo t. $P(A)$ é a probabilidade da sentença A conter o termo t. O evento B ocorre quando uma sentença é de complicação. $P(B)$ é a probabilidade da sentença B ser uma complicação. $P(A|B)$ é a probabilidade de uma sentença A conter o termo t dado que B é uma complicação.

Em problemas de classificação, procura-se determinar $P(B|A)$ que é a probabilidade de ocorrer o evento B dado que o evento A aconteceu. $P(B|A)$ é a probabilidade da sentença B ser uma complicação dado que A contém o termo t.

O algoritmo de Bayes conhece a priori $P(A)$, $P(B)$, $P(A|B)$ analisando os exemplos de treinamento. $P(B|A)$ é calculado pela fórmula de Bayes, Equação (1).

$$P(B|A) = \frac{P(A|B)\,P(B)}{P(A)} \tag{1}$$

3.2 Classificação com Naïve Bayes

Segundo Mitchell (1997), o classificador Naïve Bayes está entre os mais eficazes algoritmos para aplicações de classificação de documentos textuais. Han e Kamber (2006) apresentam o processo de classificação do Naïve Bayes dividida em cinco passos, a saber:

1. Seja D um conjunto de treinamento de sentenças distribuídas nas respectivas classes. Cada sentença é representada por um vetor de termos n-dimensional, $X = (X_1, X_2, \dots, X_n)$ e cada termo está relacionado à sentença, respectivamente, por $A_1, A_2, \dots, A_n$.
2. Suponha que há m classes $C_1, C_2, \dots, C_m$. Dado uma sentença X, o classificador irá predizer que X pertence a classe que tiver a maior probabilidade posterior, condicionada a X. Isto é, a sentença X pertence a classe C_i se e somente se

   $$P(C_i|X) > P(C_j|X)\ para\ 1 \le j \le m, j \ne i$$

   Portanto, pelo teorema de Bayes para $P(C_i|X)$ a classe C_i é maximizada pela Equação (2):

   $$P(C_i|X) = \frac{P(X|C_i)\,P(C_i)}{P(X)} \tag{2}$$

3. Como $P(X)$ é constante para todas as classes, somente $P(X|C_i)\,P(C_i)$ necessita ser maximizada. Se a probabilidade prévia da classe não é conhecida, então é comumente assumido que as classes têm probabilidades iguais, isto é, $P(C_1) = P(C_2) = \cdots = P(C_m)$. Portanto, somente é necessário maximizar $P(X|C_i)$. Caso contrário, é maximizado $P(X|C_i)\,P(C_i)$. Note que a probabilidade prévia da

classe pode ser estimada por $P(C_1) = \left|C_{i,D}\right| \div |D|$, onde $\left|C_{i,D}\right|$ é o número de sentenças de treinamento da classe C_i em D.

4. Considere um conjunto de dados com muitos termos. Seria computacionalmente caro calcular $P(X|C_i)$ para cada termo. A hipótese simples de *independência condicional da classe* é usada a fim de reduzir o custo computacional para avaliar $P(X|C_i)$. Presume-se que os valores dos termos são condicionalmente independentes um do outro. Assim, pela Equação (3) tem-se a probabilidade de X condicionada a classe C_i.

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i) \quad (3)$$

Lembrando que x_k refere-se ao valor do termo A_k da sentença X. Para cada termo computa-se $P(X|C_i)$ da seguinte forma: x_k dividido por $\left|C_{i,D}\right|$.

5. $P(X|C_i)\,P(C_i)$ é avaliada para cada classe C_i a fim de predizer a qual classe a sentença X pertence. O classificador prediz a classe C_i para a sentença X para a classe que tiver a probabilidade mais alta, Equação (4).

$$P(X|C_i)\,P(C_i) > P(X|C_j)\,P(C_j)\ para\ 1 \leq j \leq m, j \neq i \quad (4)$$

4 Medidas de Desempenho do Classificador

Medida comumente utilizada para avaliar um classificador é a taxa de erro (também conhecida como taxa de classificação incorreta). Sendo *n* o número de exemplos, o $erro(h)$, calculado pela Equação (5), compara a classe verdadeira de cada exemplo y_i com o rótulo atribuído pelo classificador induzido $h(x_i)$. A expressão $\|y_i \neq h(x_i)\|$ retorna 1 se a condição for verdadeira e zero caso contrário.

$$erro(h) = \frac{1}{n}\sum_{i=1}^{n} \|y_i \neq h(x_i)\| \quad (5)$$

O complemento da taxa de erro é a precisão do classificador, denotada por $precisão(h)$, Equação (6).

$$precisão(h) = 1 - erro(h) \quad (6)$$

Há um limiar (erro máximo) que é estabelecido para um classificador. O erro chamado de erro majoritário é calculado em um conjunto de exemplos T a partir da distribuição das classes, Equação (7).

$$erro\text{-}maj(T) = 1 - \max_{i=1,\cdots,k} dist(C_i) \quad (7)$$

O erro majoritário é um *baseline*. O classificador mais simples que se pode construir sempre escolhe exemplos da classe majoritária e comete o erro majoritário. Os classificadores construídos idealmente devem errar menos que esse classificador "ingênuo". Por exemplo, considere a seguinte distribuição de classes $dist(C_1, C_2, C_3) = (0{,}65, 0{,}15, 0{,}20) = (65\%, 15\%, 20\%)$ em um conjunto de 100 exemplos. A classe C_1 é a classe majoritária. Portanto, o erro majoritário do conjunto de exemplos T é $erro\text{-}maj(T) = 1 - 0{,}65 = 35\%$.

Para avaliar o desempenho dos classificadores utiliza-se a taxa de erro, entretanto há a necessidade de usar uma medida de custo nas seguintes situações: quando há a prevalência de uma classe sobre a outra (por exemplo, prevalência da classe C_1 com 89% no conjunto com a seguinte distribuição de classe $dist(C_1, C_2, C_3) = (89\%, 8\%, 3\%)$); ou quando cada tipo de classificação incorreta (isto é, falsos positivos e falsos negativos, conceitos apresentados em (MATOS et al., 2009)) possui um custo diferente. O custo é um número que representa uma penalidade aplicada quando um classificador faz um erro ao rotular exemplos, Equação (8).

$$erro(h) = \frac{1}{n}\sum_{i=1}^{n} \|y_i \neq h(x_i)\| * custo\big(y_i, h(x_i)\big) \quad (8)$$

Portanto, o objetivo do AM é construir classificadores com baixa taxa de erro ou baixos custos de classificação incorreta.

5 Métodos de Particionamento

Avalia-se o resultado obtido pelos classificadores para compreender a abrangência e a limitação dos diferentes algoritmos. Vários métodos são usados em conjunto com uma medida de desempenho, geralmente a precisão ou o erro, para fazer essa avaliação.

A seguir serão apresentados alguns métodos de particionamento de amostragem randômico (*Holdout*, Amostra Aleatória, *Cross-Validation* e *Bootstrap*). O *Cross-Validation* é um dos métodos mais utilizados para particionamento de exemplos (KOHAVI, 1995; MANNING; SCHÜTZE, 1999; CHEN et al., 2005; KANYA; GEETHA, 2007).

***Holdout*:** Os exemplos são divididos em uma percentagem fixa p de treinamento e $(1-p)$ para teste, considerando normalmente $\left(p > \frac{1}{2}\right)$. O valor típico para p é $\frac{2}{3}$, representando aproximadamente 67% para treinamento e 33% para teste. A vantagem é a independência dos exemplos e o tempo para computar não é muito grande. No entanto, a avaliação

pode ter uma alta variância, dependendo de como é feita a divisão do conjunto de treinamento e teste.

Amostra Aleatória: Consiste na múltipla aplicação do método *holdout.* Em cada iteração, os exemplos são particionados em conjuntos de treinamento e teste. Após o treinamento é obtida a taxa de erro do conjunto de teste. São realizadas tipicamente 20 iterações deste método e a estimativa da taxa de erro verdadeira é a média das taxas de erro do conjunto de teste de cada iteração (BATISTA; MONARD, 1998). Pode produzir melhores estimativas de erro que o estimador *holdout*. A vantagem é a independência dos exemplos.

***Cross-Validation (Validação Cruzada)*:** Uso dos mesmos dados, repetidas vezes, divididos diferentemente. Em *k-fold cross-validation* o conjunto de dados (os exemplos) é aleatoriamente dividido em k partições mutuamente exclusivas (*folds*). De tamanho aproximadamente igual a $\frac{n}{k}$ exemplos. As $(k-1)$ *folds* são usadas para treinamento e o *fold* restante para teste. Este processo é repetido k vezes, cada vez considerando um *fold* diferente para teste. O erro é a média dos erros calculados em cada um dos *k folds*. Veja exemplo a seguir.

Considere o conjunto de dados de 168 exemplos (n) distribuído em três classes relacionadas à Anemia Falciforme:

a) Tratamento (50);
b) Sintoma (38);
c) Complicação (80).

Suponha que o valor de k (*folds*) é 3. Valor comumente usado para k é 10, conhecido como 10-*fold cross-validation*. Porém, este valor depende da amostra dos dados. O tamanho de cada *fold* é calculado por $\frac{n}{k} = \frac{168}{3} = 56$ exemplos. Os 56 exemplos são selecionados aleatoriamente no conjunto total dos 168 exemplos e são exclusivos, isto é, cada exemplo pertence somente a um único *fold*. Assim, uma possível distribuição dos exemplos é mostrada na Figura 2.

		Classes			
		T	S	C	
Folds	1	20	6	30	= 56
	2	15	6	35	= 56
	3	15	26	15	= 56
Exemplos		50	38	80	= 168

Figura 2 – Exemplo de *Cross-Validation*.

Depois da distribuição dos valores entre as classes em cada *fold*, dois *folds* $(k-1)$ são escolhidos para treinamento e um para teste (*fold* restante). O processo é repetido k vezes testando o *fold* que ainda não foi usado para teste. Pode-se adotar o seguinte algoritmo. Para $k = 1$ treinar os dois primeiros *folds* e testar com o terceiro; $k = 2$ treinar o 1º e o 3º e testar com o segundo; e para $k = 3$ treinar o 2º e o 3º e testar com o primeiro.

A vantagem do *cross-validation* é na divisão dos dados. Cada partição é testada exatamente uma vez para o conjunto de treinamento $(k-1)$ vezes. A variância é reduzida a medida que o k é aumentado. A desvantagem é que o algoritmo de treinamento tem que repetir k vezes, o que significa um custo computacional elevado.

Existem duas variações comumente usadas do *cross-validation*: **S*tratified Cross-Validation*** e ***Leave-One-Out*** *(LOO)*. A primeira considera a percentagem de distribuição das classes. No exemplo anterior, as classes tratamento, sintoma e complicação representam, respectivamente, 30%, 22% e 48% do valor total da amostra. Isto significa que cada *fold* terá esta proporção de exemplos em relação ao tamanho da *fold* que é de 56 (Figura 3).

		Classes			
		T	S	C	
Folds	1	17	12	27	= 56
	2	17	12	27	= 56
	3	16	14	26	= 56
Exemplos		50	38	80	= 168

Figura 3 – Exemplo de *Stratified Cross-Validation.*

LOO é um caso particular da *cross-validation* quando *k* for igual a quantidade de exemplos. Considerando 168 exemplos, a quantidade de *folds* seria então 168. Para treinamento são os mesmos $(k-1)$ exemplos e um *fold* para teste. O processo é executado 168 vezes, sendo que cada *fold* conterá somente uma classe (Figura 4). É computacionalmente custoso e por isso é usado em amostras pequenas.

		Classes
Folds	1	T, S ou C
	2	T, S ou C
	3	T, S ou C
	.	.
	.	.
	.	.
	168	T, S ou C

Figura 4 – Exemplo de *Leave-One-Out*.

Bootstrap: Consiste em construir um conjunto de treinamento através da amostragem com reposição de n casos a partir de um conjunto de exemplos de tamanho n. Amostrar com reposição significa que os exemplos de treinamento são retirados do conjunto de exemplos, mas os elementos selecionados permanecem no conjunto de exemplos, de forma que um mesmo elemento possa ser escolhido aleatoriamente mais de uma vez.

Por exemplo, considere dez exemplos de treinamento (cada número representando uma sentença). A Figura 5 mostra uma iteração, sendo que as sentenças $2, 6, 8\ e\ 9$ não foram selecionadas para treinamento, enquanto que a sentença 1 foi selecionada três vezes e as sentenças 3 e 7 foram selecionadas, cada uma, duas vezes. As sentenças de teste são formadas pelas sentenças que não foram selecionadas para treinamento.

Nesse método são realizadas aproximadamente 200 iterações. A taxa de erro estimada é a média das taxas de erro de cada iteração.

Treinamento	1,3,7,1,4,1,5,7,3,10
Teste	2,6,8,9

Figura 5 – Exemplo de *Bootstrap*.

Kohavi (1995) faz uma comparação dos dois métodos mais comuns para estimar a precisão do classificador (*cross-validation* e *bootstrap*) e recomenda o método *stratified* 10-f*old cross-validation*. Han e Kamber (2006) corrobora o uso desse método mesmo que o poder computacional permita mais *folds*. Encontram-se mais detalhes dos métodos *holdout*, *cross-validation* e *bootstrap* em Stranieri e Zeleznikow (2005), no qual é ilustrado o erro verdadeiro dos métodos apresentados anteriormente.

Ainda segundo Stranieri e Zeleznikow (2005), o método que obteve o menor erro foi o *bootstrap* seguido do LOO, *cross-validation* e *holdout*. Entretanto, é difícil usar na prática o *bootstrap* ou LOO, porque estes métodos exigem que os dados sejam repetidos diversas vezes, usando um considerável esforço computacional. Na Tabela 1 é apresentado o resumo dos métodos de particionamento.

Tabela 1 – Resumo dos métodos de particionamento.
Fonte: Adaptado de Monard e Baranauskas (2003).

	Holdout	**Aleatória**	***LOO***	***k-Fold CV***	***k-Fold Stratified CV***	***Bootstrap***
Treinamento	pn	t	$n-1$	$n(k-1)/k$	$n(k-1)/k$	n
Teste	$(1-p)n$	$n-t$	1	n/k	n/k	$n-t$
Iterações	1	$\cong 20$	n	k	k	$\cong 200$
Reposição	não	não	não	não	não	sim
Prevalência de Classe	não	não	não	não	sim	sim/não

Parâmetros	n representa o número de exemplos, k o número de $folds$ (partições), p número entre $0 < p < 1$ e t número entre $0 < t < n$

6 Seleção de Características

Seleção de Características (*Feature Selection*) é o processo de selecionar um subconjunto de termos do conjunto de treinamento e usá-lo na classificação de texto. Serve para dois propósitos: diminuir a quantidade do vocabulário de treinamento, tornando o classificador mais eficiente (na maioria das vezes o custo computacional de treinar é caro); aumentar a precisão da classificação eliminando ruído (MANNING; RAGHAVAN; SCHÜTZE, 2008). Segundo Ikonomakis, Kotsiantis e Tampakas (2005), é a redução da dimensionalidade do conjunto de dados que tem o objetivo de excluir as características que são consideradas irrelevantes para a classificação.

Um algoritmo básico de seleção de características pode ser visto no Algoritmo 1. Para uma dada classe *c* é computado a medida de utilidade $A(t, c)$ (linha 4) para cada termo do vocabulário (linha 3) e é selecionado os *k* termos que tem os valores mais altos em relação ao cálculo da medida. Todos os outros termos são descartados e não são utilizados na classificação.

Algoritmo 1 – Seleção das melhores *k* características.
Fonte: Manning, Raghavan e Schütze (2008).

```
Entrada: D: conjunto de documentos textuais.
         c: classe.
Saída: Lista com os valores de k mais altos.

1 V ← ExtrairVocabulário(D)
2 L ← []
3 para cada t ∈ V
4 faça A(t,c) ← CalculaMedidaUtilidade(D,t,c)
5 | Append(L, ⟨A(t,c),t⟩)
6 retorna MaiorValorCaracterística(L,k)
```

Para redução das características existem várias medidas de utilidade que podem ser utilizadas: frequência, ganho de informação, informação mútua e X^2 (qui-quadrado). A seguir são apresentadas sucintamente cada uma delas.

Frequência de Documento (FD): É o número de documentos da classe c que contém o termo t, Equação (9). A hipótese é que termos raros não são importantes para a predição da categoria e não afeta o desempenho global. Não selecionando termos raros reduz a dimensionalidade do espaço de característica. A grande maioria das palavras que ocorrem em um documento tem uma FD muito baixa, o que significa que através da redução da frequência ape-

nas essas palavras são removidas, enquanto as palavras com frequência baixa, média e alta são preservadas (SEBASTIANI, 2002).

$$FD(t,c) = P(t|c) \tag{9}$$

Ganho de Informação (GI): Mede o número de bits de informação obtido por uma predição de categoria conhecendo a presença ou ausência do termo em um documento. A complexidade do tempo é $O(N)$ e a complexidade do espaço é $O(VN)$, onde N é o número de documento de treinamento e V é o tamanho do vocabulário. A computação da entropia tem um tempo de complexidade de $O(Vm)$. O GI do termo t com a classe c_i variando de $1 \leq i \leq m$ é definida pela Equação (10).

$$\begin{aligned} GI(t) = &-\sum_{i=1}^{m} P(c_i)\ log\ P(c_i) \\ &+P(t) \sum_{i=1}^{m} P(c_i|t)\ log\ P(c_i|t) \\ &+P(\bar{t}) \sum_{i=1}^{m} P(c_i|\bar{t})\ log\ P(c_i|\bar{t}) \end{aligned} \tag{10}$$

$P(t)$ é a probabilidade que o termo t ocorre e $\bar{t}$ é a probabilidade que o termo t não ocorre. $P(c_i|t)$ é a probabilidade condicional da ocorrência de um termo na classe c_i e $P(c_i|\bar{t})$ é a probabilidade condicional de não ocorrer o termo na classe c_i.

Informação Mútua (IM): É um critério comumente usado em modelagem estatística de associação de palavras. Considera o termo t e a categoria c, sendo que A é o número de vezes que t e c coocorrem, B é o número de vezes que t ocorre sem c, C é o número de vezes que c ocorre sem t e N é o número total de documentos. A estimativa do termo t e categoria c é apresentada na Equação (11). O tempo de complexidade é $O(Vm)$, similar ao GI. É uma medida da quantidade de informação que uma variável contém sobre outra. A IM é maior quando todas as ocorrências de dois termos são adjacentes umas às outras, deteriorando-se em baixa frequência.

$$IM(t,c) \cong log\ \frac{A \times N}{(A+C) \times (A+B)} \tag{11}$$

Qui-quadrado (X^2): Mede a falta de independência do termo t e da categoria c. A medida X^2 tem valor zero se t e c são independentes. A computação tem complexidade quadrática, similar a IM e ao GI. *Considera* o significado de A, B e C explicado na medida ante-

rior. D é o número de vezes que não ocorrem nem c e t. A medida é definida pela Equação (12).

$$X^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \quad (12)$$

Yang e Pedersen (1997) apresentam um estudo comparativo dos quatros métodos apresentado anteriormente no qual o GI e X^2 foram os mais eficientes na redução da dimensionalidade. Os experimentos foram realizados com o classificador *nearest neighbor* no documento da Reuters, obtendo redução de 98% dos termos e o mais importante, houve melhora na precisão do classificador.

A FD, método com o mais baixo custo computacional, obteve um desempenho similar aos dois métodos citados no parágrafo anterior. O uso da FD é sugerido quando o custo de usar o GI e X^2 for alto. E diferentemente do que é senso comum na recuperação de informação – que termos comuns não são importantes – o bom desempenho da FD, do GI e do X^2 mostra que de fato os termos comuns (exceto as *stop words*) são informativos para tarefas de categorização de texto. A IM teve o pior desempenho comparado com os outros métodos devido ao favorecimento de termos raros e a forte sensibilidade dos erros estimados.

A principal diferença entre IM e X^2 é que esta última tem um valor normalizado que é comparado com termos da mesma categoria. Caso uma das variáveis (A, B, C ou D) desta última medida tenha valor desprezível (frequência baixa), não é possível obter o resultado precisamente. Portanto, X^2 não é uma medida confiável para termos com baixas frequências.

Sebastiani (2002) relata a partir de experimentos que algumas medidas se sobressaem mais do que outras no sentido de qualidade da redução da dimensionalidade. Todavia, faz uma observação ressaltando que os resultados obtidos com as diferentes medidas são apenas indicativos e que o mérito de cada medida somente poderia ser obtido com experimentos comparativos cuidadosamente controlados considerando a variedade de situações diferentes, por exemplo, classificadores e conjunto de dados diferentes.

Outras medidas de utilidade como *odds ratio*, *probability ratio* e p*ow* podem ser encontradas em Mladenic e Grobelnik (1999), Sebastiani (2002), Forman (2003) e Ikonomakis, Kotsiantis e Tampakas (2005). Os autores desta última referência destacam que não existe uma medida que sobressai mais do que as outras e por isso os pesquisadores, muitas vezes, combinam duas medidas a fim de obter benefícios de ambas.

7 Mover

O Mover é um sistema de classificação de aprendizado supervisionado de estruturas retóricas. Apresenta as organizações retóricas ou estruturas do texto, conhecidas como *moves*, usadas no texto, a fim de ajudar estudantes não-nativos que tenham dificuldades na leitura e escrita técnica seja por falta de conhecimento ou experiência. O sistema aprende características estruturais do texto usando um pequeno número de exemplos de treinamento e pode ser aplicado em diferentes textos (ANTHONY; LASHKIA, 2003).

Um modelo conhecido para representação de palavras é o *bag of words* (saco de palavras) que divide as sentenças em palavras individuais (toquenização). A representação *bag of words* não considera a ordem da palavra, a semântica ou a gramática da linguagem, porém para tarefas de processamento no nível de sentença tem mostrado sucesso (ANTHONY; LASHKIA, 2003).

Contudo, para representar o conhecimento dos *moves* estruturais foi utilizado a representação de grupos denominado modelo *bag of clusters*. Com este método é considerado o grupo de palavras. Considere a seguinte sentença "*once upon a time, there was an island*", uma faixa de grupos poderia incluir, por exemplo, "*once*, *upon*, *once upon* e *once upon a time"*. Cada sentença do conjunto de treinamento é dividida em grupos de palavras com tamanho que variam de uma a cinco palavras.

Alguns grupos, por exemplo, "*upon a"* e "*time there"* contribuem pouco para o processo de aprendizado. Esses termos irrelevantes são conhecidos como ruído. Removendo o ruído espera-se que o sistema melhore a velocidade de processamento e principalmente a precisão do classificador. Utilizou-se a medida *ganho de informação* que classifica os grupos de palavras de acordo com a pontuação. O ruído é reduzido excluindo os grupos que estão abaixo de um limiar.

O classificador utilizado é o Naïve Bayes combinado com uma das seguintes medidas de utilidade, *ganho de informação, qui-quadrado* ou *frequência,* para eliminar ruído. Experimentos com outros algoritmos de AM mais complexos, como Árvores de Decisão e Redes Neurais, foram realizados, no entanto o classificador Bayesiano apresentou melhor desempenho para a identificação das estruturas textuais (LEWIS, 1998 apud ANTHONY; LASHKIA, 2003). O método de particionamento utilizado é o 5-*fold cross-validation*. O treinamento é realizado com 100 resumos de artigos de pesquisa sobre Tecnologia da Informação publicados no IEEE em 1998. Tarefas de pré-processamento foram automatizadas com intuito de remover irrelevantes caracteres como espaço em branco, linhas irregulares, etc.

Na Figura 6 é apresentada a tela principal do Mover, cuja ferramenta está disponível na web (ANTHONY, 2009). O conjunto de dados de treinamento foi separado em seis classes (*Claim centrality*, *Generalize topics*, *Indicate a gap*, *Announce research*, *Announce findings* e *Evaluate research)* conforme o espaço de pesquisa de Swales, *Create a Research Space* – CARS (ANTHONY, 1999).

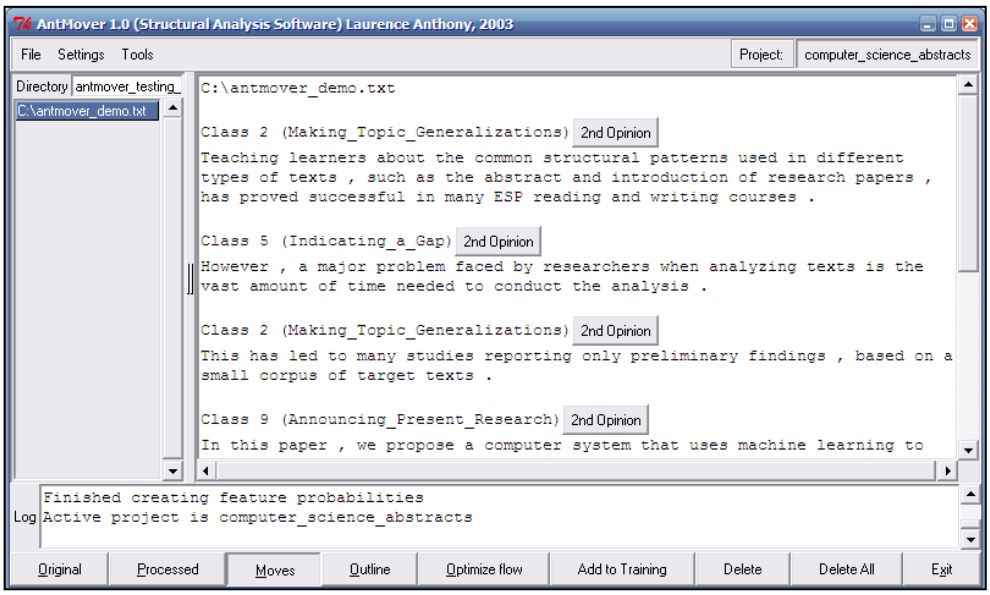

Figura 6 – Ferramenta Mover.

Foram realizados testes com os resumos dos artigos técnicos e obteve uma média de precisão de 68%. Segundo Anthony e Lashkia (2003), uma característica importante do classificador Naïve Bayes é a habilidade de pontuar decisões a partir da solução mais provável. Assim, duas técnicas podem ser utilizadas para melhorar a precisão da ferramenta: otimização de fluxo, aumenta a precisão do sistema em 2% a um pequeno custo computacional; e adição de treinamento, usuários podem corrigir manualmente classificação realizada pela ferramenta e adicionar nova informação para treinamento que em princípio pode obter melhores resultados ao custo de mais tempo. Com o uso dessas técnicas alcançou uma melhora na precisão de 86%. Resultado inexpressivo foi obtido para uma determinada classe que pode ser explicado pela pouca quantidade de exemplos de treinamento da mesma (ANTHONY; LASHKIA, 2003).

8 Considerações Finais

Neste relatório foi apresentada uma introdução sobre aprendizado de máquina no qual foram abordados os seguintes assuntos: classificador supervisionado Naïve Bayes; medidas comumente utilizadas para avaliar o desempenho do classificador; métodos de particionamento usados para separar o conjunto de dados em treinamento e teste; seleção de característica

inserida no contexto do aprendizado, explicando algumas medidas de utilidade que podem ser utilizadas para redução de características; e por fim, ferramenta Mover, a qual implementa o classificador Naïve Bayes e fornece a opção de usar uma de três medidas de utilidade, utilizada para auxiliar na leitura e escrita de artigos técnicos.

Referências

ANTHONY, L. Writing research article introductions in software engineering: how accurate is a standard model? **IEEE Transactions on Professional Communication**, v. 42, n. 1, p. 38-46, 1999. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=749366>. Acesso em: 20 fev. 2009.

_____. **AntMover 1.0**. 2009. Disponível em: <http://www.antlab.sci.waseda.ac.jp/software/antmover1.0.exe>. Acesso em: 09 mar. 2009.

ANTHONY, L.; LASHKIA, G. V. Mover: a machine learning tool to assist in the reading and writing of technical papers. **IEEE Transactions on Professional Communication**, v. 46, n. 3, p. 185-193, 2003. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1227591>. Acesso em: 23 out. 2008.

BATISTA, G. E. A. P. A.; MONARD, M. C. Utilizando métodos estatísticos de resampling para estimar a performance de sistemas de aprendizado. In: WORKSHOP DE DISSERTAÇÕES DEFENDIDAS, 1998, São Carlos. **Proceedings.** Instituto de Ciências Matemáticas e de Computação, 1998. p. 173-184. Disponível em: <http://www.icmc.usp.br/~mcmonard/public/icmcwshG1998.pdf>. Acesso em: 17 fev. 2009.

CHEN, H.; FULLER, S. S.; FRIEDMAN, C.; WILLIAM **Medical informatics:** knowledge management and data mining in biomedicine. Berlin: Springer, 2005. 624 p. Disponível em: <http://ai.arizona.edu/hchen/chencourse/MedBook/>. Acesso em: 23 out. 2008.

FORMAN, G. An extensive empirical study of feature selection metrics for text classification. **The Journal of Machine Learning Research**, v. 3, p. 1289-1305, Mar., 2003. Disponível em: <http://portal.acm.org/citation.cfm?id=944974>. Acesso em: 16 fev. 2009.

GUYON, I.; WESTON, J.; BARNHILL, S.; VAPNIK, V. Gene selection for cancer classification using support vector machines. **Machine Learning**, v. 46, n. 1-3, p. 389-422, 2002. Disponível em: <http://dx.doi.org/10.1023/A:1012487302797>. Acesso em: 17 fev. 2009.

HAN, J.; KAMBER, M. **Data mining:** concepts and techniques. 2nd ed. San Francisco, CA: Morgan Kaufmann, 2006. 743 p.

IKONOMAKIS, M.; KOTSIANTIS, S.; TAMPAKAS, V. Text classification using machine learning techniques. **WSEAS Transactions on Computers**, v. 4, n. 8, p. 966-974, 2005. Disponível em: <http://www.math.upatras.gr/~esdlab/en/members/kotsiantis/Text%20Classification%20final%20journal.pdf>. Acesso em: 13 fev. 2009.

KANYA, N.; GEETHA, S. Information extraction - a text mining approach. In: IET-UK INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY IN ELECTRICAL SCIENCES (ICTES), 2007, Chennai, Tamilnadu, India. **Proceedings.** 2007. p. 1111-1118. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4735960>. Acesso em: 10 mar. 2009.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: INTERNATIONAL JOINT CONFERENCES ON ARTIFICIAL INTELLIGENCE (IJCAI), 14th, 1995, Montréal, Québec. **Proceedings.** Morgan Kaufmann, 1995. p. 1137-1145. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.529>. Acesso em: 13 fev. 2009.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to information retrieval**. Cambridge: Cambridge University Press, 2008. 482 p. Disponível em: <http://www-csli.stanford.edu/~hinrich/information-retrieval-book.html>. Acesso em: 28 nov. 2008.

MANNING, C. D.; SCHÜTZE, H. **Foundations of statistical natural language processing**. London, England: MIT Press, 1999. 680 p.

MATOS, P. F.; CAROSIA, A. E. O.; LOMBARDI, L. O.; CIFERRI, R. R.; PARDO, T. A. S.; CIFERRI, C. D. A.; VIEIRA, M. T. P. **Relatório Técnico "Métricas de Avaliação"**. São Carlos: Universidade Federal de São Carlos, 2009, p. 15.

MITCHELL, T. M. **Machine learning**. Boston: McGraw-Hill, 1997. 414 p.

MLADENIC, D.; GROBELNIK, M. Feature selection for unbalanced class distribution and Naive Bayes. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML), 16th, 1999, Bled, Slovenia. **Proceedings.** Morgan Kaufmann, 1999. p. 258-267. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.2544>. Acesso em: 16 fev. 2009.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: REZENDE, S. O. (Ed.). **Sistemas inteligentes:** fundamentos e aplicações. São Carlos: Manole, 2003. p. 89-114. cap. 4.

PINTO, A. C. S.; MATOS, P. F.; PERLIN, C. B.; ANDRADE, C. G.; CAROSIA, A. E. O.; LOMBARDI, L. O.; CIFERRI, R. R.; PARDO, T. A. S.; CIFERRI, C. D. A.; VIEIRA, M. T. P. **Technical Report "Sickle Cell Anemia"**. São Carlos: Federal University of São Carlos, 2009, p. 16. Disponível em: <http://sca.dc.ufscar.br/download/files/report.sca.pdf>. Acesso em: 30 out. 2009.

SEBASTIANI, F. Machine learning in automated text categorization. **ACM Computing Surveys**, v. 34, n. 1, p. 1-47, 2002. Disponível em: <http://doi.acm.org/10.1145/505282.505283>. Acesso em: 17 fev. 2009.

STRANIERI, A.; ZELEZNIKOW, J. **Knowledge discovery from legal databases**. Dordrecht, Netherlands: Springer, 2005. 294 p. (Law and Philosophy Library; v. 69).

WITTEN, I. H.; FRANK, E. **Data mining:** practical machine learning tools and techniques with Java implementations. 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005. 525 p.

WU, X.; KUMAR, V.; QUINLAN, J. R.; GHOSH, J.; YANG, Q.; MOTODA, H.; MCLACHLAN, G. J.; NG, A.; LIU, B.; YU, P. S.; ZHOU, Z.-H.; STEINBACH, M.; HAND, D. J.; STEINBERG, D. Top 10 algorithms in data mining. **Knowledge and Information Sys-**

tems, v. 14, n. 1, p. 1-37, 2007. Disponível em: <http://dx.doi.org/10.1007/s10115-007-0114-2>. Acesso em: 19 fev. 2009.

YANG, Y.; PEDERSEN, J. O. A comparative study on feature selection in text categorization. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML), 14th, 1997, San Francisco, CA. **Proceedings.** Morgan Kaufmann, 1997. p. 412-420. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.9956>. Acesso em: 16 fev. 2009.

ZHANG, H. The optimality of Naive Bayes. In: INTERNATIONAL FLAIRS CONFERENCE, 17th, 2004, Miami Beach, Florida. **Proceedings.** Menlo Park, CA: AAAI Press, 2004. p. 562-567. Disponível em: <http://www.cs.unb.ca/profs/hzhang/publications/FLAIRS04ZhangH.pdf>. Acesso em: 16 fev. 2009.