Redução de dimensionalidade aplicada na classificação de spams usando filtros bayesianos

Tiago A. Almeida [1]
Akebo Yamakami [1]

Resumo: Nos últimos anos, e-mails spams têm-se tornado um importante problema com enorme impacto econômico para a sociedade. Felizmente, existem métodos capazes de detectar automaticamente a maioria dessas mensagens, sendo as técnicas mais empregadas baseadas na Teoria da Decisão Bayesiana. Por outro lado, grande parte das abordagens probabilísticas apresenta uma dificuldade: a manipulação de dados em um espaço com alta dimensionalidade. Para contornar esse problema, muitas técnicas de seleção de termos têm sido propostas na literatura. Neste artigo, revisamos os métodos mais populares empregados como técnicas para seleção de termos em conjunto com sete modelos diferentes de filtros anti-spam Naïve Bayesianos.

Palavras-chave: Redução de dimensionalidade. Filtragem de spams. Aprendizagem de máquina.

Abstract: *In recent years, e-mail spam has become an increasingly important problem with a big economic impact in society. Fortunately, there are different approaches able to automatically detect and remove most of these messages, and the best-known ones are based on Bayesian decision theory. However, the most of these probabilistic approaches have the same difficulty: the high dimensionality of the feature space. Many term selection methods have been proposed in the literature. In this paper, we review the most popular methods used as term selection techniques with seven different versions of Naïve Bayes spam filters.*

Keywords: *Dimensionality reduction. Spam filtering. Machine learning.*

1 Introdução

E-mail é um dos meios de comunicação mais popular, mais rápido e mais barato. Tornou-se parte do cotidiano de milhares de pessoas, mudando a maneira como elas trabalham e colaboram. O e-mail não é mais empregado apenas para dar suporte à comunicação pessoal, mas também é utilizado como agenda, gerenciador de tarefas, organizador de contatos, sistema de envio e armazenamento de documentos etc. A desvantagem desse enorme sucesso é o volume sempre crescente de spams (mensagens eletrônicas comerciais não solicitadas) que recebemos. O problema do spam pode ser quantificado em termos econômicos, uma vez que muitas horas são desperdiçadas todos os dias por trabalhadores. Não se trata apenas do tempo que perdem com a leitura do spam, mas também do tempo que gastam para excluir essas mensagens. Isso sem considerar os inúmeros problemas com fraudes, roubos e danos diretos.

De acordo com relatórios anuais divulgados por grandes corporações de segurança computacional, a quantidade de spams em circulação está aumentando de maneira assustadora. A média de spams enviados por dia passou de 2,4 bilhões em 2002[2] para 300 bilhões em 2010[3]. Calcula-se que aproximadamente 96% de todos os e-mails recebidos por empresas são representados por spams[4]. Estima-se também que, atualmente, mais de 90% de todo o tráfego de e-mail seja representado por mensagens desse tipo[5].

[1]Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Av. Albert Einstein, 400 - Barão Geraldo - 13083-852, Campinas /SP - Brasil
{tiago,akebo@dt.fee.unicamp.br}

doi: 10.5335/rbca.2011.003
[2]Fonte: http://www.spamlaws.com/spam-stats.html
[3]Fonte: http://www.cisco.com/en/US/prod/collateral/vpndevc/cisco_2009_asr.pdf
[4]Fonte: http://www.sophos.com/articles/2008/dirtydozjul08.html
[5]Fonte: http://www.messagelabs.com/MLIReport_2009_FINAL.pdf

De acordo com o Relatório Americano de Tecnologia[6], o custo do spam em termos de perda de produtividade nos Estados Unidos chegou a US$ 21,58 bilhões anualmente, enquanto que o custo do spam em relação à produtividade mundial foi estimada em US$ 50 bilhões. Em escala global, o custo empregado em tecnologia de informação por consequência dos spams aumentou de US$ 20,5 bilhões em 2003, para US$ 130 bilhões em 2009[7].

O volume de spams enviado no Brasil é um problema crescente, com proporções alarmantes e enorme impacto econômico para toda a sociedade. Apesar de se tratar de um problema de escala global, tal fenômeno vem ganhando grande destaque internamente, fazendo com que o Brasil seja um dos países que mais enviam lixo virtual no mundo. Segundo fontes, o país foi o maior emissor de mensagens não solicitadas, sendo origem de aproximadamente 20% dos spams enviados nos dois primeiros meses de 2010[8] e responsável por 7,7 trilhões de mensagens deste tipo somente em 2009[9].

A ausência de uma regulamentação jurídica, a alta vulnerabilidade dos sistemas computacionais e a falta de conhecimento dos usuários de e-mail são indicadas como os principais fatores que impulsionaram o Brasil a tornar-se um dos maiores disseminadores de spams do mundo.

Felizmente, existem diversos métodos para classificar mensagens como legítimas ou como spams, tais como filtros baseados em regras, listas negras, filtros colaborativos, técnicas que levam em consideração o domínio do remetente, dentre muitos outros. Entretanto, dentre todas essas abordagens, algoritmos de aprendizado de máquina vêm obtendo maior sucesso [8, 13]. Tais métodos incluem técnicas que são consideradas as melhores em categorização de textos, como algoritmos de indução de regras [11, 12], Rocchio [26, 39], algoritmo de compressão de dados [3], Boosting [10], aprendizado baseado em memória [7], máquinas de vetores de suporte (SVM) [1, 15, 20, 25, 30] e classificadores Bayesianos [4, 6, 21, 35, 38, 44].

Os classificadores Bayesianos vêm ganhando considerável destaque por serem muito populares em filtros comerciais e abertos (open-sources) [1, 24, 35, 44]. Isso se deve, provavelmente, a sua simplicidade de implementação, baixa complexidade computacional e acurácia, características que são comparáveis aos métodos de aprendizado mais elaborados [35]. Diversos servidores modernos de e-mail passaram a utilizá-lo. Além disso, filtros bastante consagrados, como OSBF-Lua (vencedor dos dois últimos campeonatos de filtros anti-spams), DSPAM, SpamAssassin, SpamBayes, Bogofilter, ASSP, dentre muitos outros, utilizam técnicas de classificação Bayesiana [24, 33, 43].

Apesar de todas as vantagens apresentadas, esses métodos possuem um ponto fraco bastante conhecido: a queda de desempenho conforme a dimensão do espaço de dados aumenta [8, 13, 48]. O espaço de características consiste de termos únicos extraídos das mensagens de e-mail, que podem aparecer em dezenas ou em centenas de milhares, mesmo para uma coleção de e-mails de tamanho moderado. Tal característica é proibitiva para a maioria dos métodos de aprendizado e, consequentemente, é altamente desejável que técnicas automáticas realizem a redução dos espaço dos dados sem sacrificar a acurácia dos classificadores [19].

Um fato pouco explorado na literatura de spams é que diversas técnicas de seleção de termos podem ser utilizadas para reduzir a dimensionalidade do espaço de dados antes da etapa de classificação. Da mesma forma, existem vários modelos diferentes de filtros Bayesianos que podem ser empregados na tarefa de filtragem de e-mails. Para preencher esse gap, neste artigo apresentamos um tutorial sobre oito técnicas de seleção de termos que podem ser empregadas em conjunto com sete modelos de filtro anti-spam Naïve Bayes.

O restante deste artigo está estruturado da seguinte maneira: a seção 2 apresenta detalhes sobre as técnicas mais conhecidas empregadas para a seleção de termos; diferentes modelos de filtros anti-spam Naïve Bayesianos estão descritos na seção 3; finalmente, a seção 4 oferece conclusões e linhas para trabalhos futuros.

2 Redução de dimensionalidade

Ao contrário do que ocorre em recuperação de textos, em categorização de texto, a alta dimensionalidade do espaço de termos ($\mathcal{T}$) pode ser problemática. Os algoritmos de recuperação de texto podem manipular altos

[6]Fonte: http://www.rockresearch.com/news_020305.php
[7]Fonte: http://www.ferris.com/2009/01/28/cost-of-spam-is-flattening-our-2009-predictions/.
[8]Fonte: http://www.pandasecurity.com/img/enc/Quarterly_Report_Pandalabs_Q1_2010.pdf
[9]Fonte: http://cisco.com/en/US/prod/vpndevc/annual_security_report.html.

valores de $|\mathcal{T}|$, o que não ocorre com a maioria dos métodos de aprendizado empregados para classificação. Consequentemente, antes da etapa de classificação, geralmente é aplicado um passo de redução de dimensionalidade, cujo efeito é reduzir o tamanho do vetor de espaço de $|\mathcal{T}|$ para $|\mathcal{T}'| \ll |\mathcal{T}|$. O conjunto $\mathcal{T}'$ é chamado de conjunto reduzido de termos [42].

A redução da dimensionalidade também é benéfica uma vez que tende a reduzir o superajustamento (overfitting) [42]. Classificadores que superajustem os dados são bons na reclassificação dos dados usados no treinamento, porém tendem a classificar incorretamente dados que ainda não foram vistos. Além disso, muitos classificadores apresentam baixo desempenho quando manipulam uma grande quantidade de atributos. Dessa forma, é recomendável um procedimento para reduzir o número de termos utilizados para representar as mensagens [18, 19].

Métodos empregados para reduzir a dimensionalidade do espaço de termos podem ser separados em duas classes distintas dependendo de se a tarefa é realizada localmente (por exemplo, para cada categoria individualmente) ou globalmente. Além disso, tais técnicas são compostas por seletores de termos ($\mathcal{T}' \subset \mathcal{T}$) ou extratores de termos (os termos de $\mathcal{T}'$ não são iguais aos termos de $\mathcal{T}$ e são obtidos por combinações ou transformações dos termos originais). O motivo de utilizar termos sintéticos é evitar problemas com polissemia, homônimos e sinônimos. Devido a essas características, técnicas de seleção de termos são geralmente aplicadas para reduzir a dimensionalidade do espaço de termos. De acordo com Yang and Pedersen [47], algumas dessas técnicas podem reduzir a dimensionalidade em até 100 vezes sem causar perda (ou até mesmo com um pequeno aumento) de eficácia.

2.1 Representação

Considerando que cada mensagem m é composta por um conjunto de termos (tokens) $m = t_1, \ldots, t_n$, sendo que cada termo t_k corresponde a uma palavra ("adulto"), um conjunto de palavras ("para ser removido") ou um único caractere ("$"), pode-se representar cada mensagem como um vetor $\vec{x} = \langle x_1, \ldots, x_n \rangle$, onde $x_1, \ldots, x_n$ são valores dos atributos $X_1, \ldots, X_n$ associados aos termos $t_1, \ldots, t_n$. No caso mais simples, cada termo representa uma única palavra e todos os atributos são Booleanos: $X_i = 1$ se a mensagem contém t_i, ou $X_i = 0$, em caso contrário.

Alternativamente, os atributos podem ser valores inteiros obtidos a partir da frequência do termo (term frequency – TF), representando quantas vezes cada termo ocorre na mensagem. Esse tipo de representação oferece mais informação que a Booleana [35]. Uma terceira alternativa é associar cada atributo X_i a um TF normalizado, $x_i = \frac{\sharp t_i(m)}{|m|}$, onde $\sharp t_i(m)$ corresponde ao número de ocorrências do termo representado por X_i em m e $|m|$ representa o comprimento de m mensurado pelas ocorrências dos termos. TF normalizado leva em consideração a repetição dos termos em relação ao tamanho da mensagem. Ele é similar à pontuação TF–IDF (term frequency-inverse document frequency), comumente empregada em recuperação de informação, sendo que o componente IDF poderia ser adicionado para denotar termos que são comumente encontrados nas mensagens de treinamento.

2.2 Técnicas para seleção de termos

A seguir são apresentados os oito métodos mais utilizados para Redução do Espaço de Termos (RET) presentes na literatura. Probabilidades são interpretadas como eventos no espaço das mensagens (por exemplo, $P(\bar{t}_k, c_i)$ corresponde à probabilidade de, para uma mensagem aleatória m, o termo t_k não ocorrer em m e m pertencer à classe c_i) e são estimadas pela contagem das ocorrências no conjunto de treinamento Tr.

Como existem apenas duas categorias na filtragem de spams ($c = \{\texttt{spam}(\texttt{c}_\texttt{s}), \texttt{legitimo}(\texttt{c}_\texttt{l})\}$), algumas funções são especificamente "locais" em relação a uma dada categoria c_i. Para calcular o valor de um termo t_k num senso "global" e independente de categoria, tanto a soma $f_{sum}(t_k) = \sum_{i=1}^{|c|} f(t_k, c_i)$, a soma ponderada $f_{wsum}(t_k) = \sum_{i=1}^{|c|} P(c_i).f(t_k, c_i)$ ou o máximo $f_{max}(t_k) = max_{i=1}^{|c|} f(t_k, c_i)$ dos valores relativos a cada categoria específica $f(t_k, c_i)$ são comumente calculados. Essas funções tentam capturar a intuição de que os melhores termos para c_i são aqueles mais bem distribuídos nos conjuntos de amostras positivas e negativas de c_i.

2.2.1 Frequência de documentos (FD)

Trata-se de uma função simples, global e eficiente para RET. Ela é obtida pela frequência de mensagens com o termo t_k na base de treinamento, ou seja, somente os termos que aparecem em maior número de mensagens são selecionados. Nesse caso, supõe-se que os termos que aparecem raramente não são informativos para a predição da categoria nem influenciam no desempenho global. Dessa forma, a remoção dos termos raros reduz a dimensionalidade do espaço de termos e, possivelmente, melhora o desempenho do classificador, principalmente quando a maioria desses termos são ruídos inseridos pelos spammers. O valor FD de um termo t_k é calculado por

$$FD(t_k) = \frac{|Tr_{t_k}|}{|Tr|},$$

sendo $|Tr_{t_k}|$ a quantidade de mensagens que contêm o termo t_k na base de treinamento Tr e $|Tr|$ a quantidade total de mensagens processadas [47].

2.2.2 Fator de associação DIA (DIA)

O fator DIA de um termo t_k em uma classe c_i mede a probabilidade de encontrar mensagens da classe c_i dado o termo t_k. Essas probabilidades são obtidas pela frequência dos termos na base de treinamento Tr [22, 28],

$$DIA(t_k, c_i) = P(c_i|t_k).$$

As pontuações específicas de cada classe podem ser combinadas usando-se as funções f_{sum} ou f_{max} para calcular a representatividade do termo em escala global.

2.2.3 Ganho de informação (GI)

GI é frequentemente empregado em aprendizagem de máquina como critério de representatividade de termos [36]. Ele calcula o número de bits de informação obtido pela predição de categoria através do conhecimento da presença ou ausência de um termo em uma mensagem [47]. O GI de um termo t_k é calculado por

$$GI(t_k) = \sum_{c \in [c_i, \bar{c}_i]} \sum_{t \in [t_k, \bar{t}_k]} P(t, c).log \frac{P(t, c)}{P(t).P(c)}.$$

É importante destacar que GI é a técnica de seleção de termos mais utilizada por classificadores anti-spam. Alguns trabalhos relevantes que empregam GI com filtros anti-spam Naïve Bayesianos podem ser encontrados em Androuutsopoulos et al. [8], Metsis et al. [35] e [40].

2.2.4 Informação mútua (IM)

IM é um critério comumente utilizado em linguagem estatística para modelar associações entre palavras e aplicações relacionadas [47]. A informação mútua entre t_k e c_i é definida por

$$IM(t_k, c_i) = log \frac{P(t_k, c_i)}{P(t_k).P(c_i)}.$$

$IM(t_k, c_i)$ tem um valor natural igual a zero se t_k e c_i são independentes. Para medir a representatividade de um termo em um senso global, as pontuações específicas de cada categoria podem ser combinadas usando-se as funções f_{sum}, f_{wsum} ou f_{max}, como apresentado anteriormente.

Em alguns trabalhos, GI também é chamado de IM, causando uma certa confusão. Provavelmente, isso se deve ao fato de GI ser a média ponderada de $IM(t_k, c_i)$ e $IM(\bar{t}_k, c_i)$, na qual os pesos são dados pelas

probabilidades conjuntas $P(t_k, c_i)$ e $P(\bar{t}_k, c_i)$, respectivamente. Consequentemente, GI é também conhecido como IM média [42].

Existem duas diferenças fundamentais entre GI e IM: primeiro, GI usa informação relativa à ausência de termos, enquanto que IM a ignora e, segundo, GI normaliza a pontuação IM usando as probabilidades conjuntas, enquanto que IM utiliza uma pontuação não normalizada.

2.2.5 Estatística χ^2

A estatística χ^2 calcula a falta de independência entre o termo t_k e a classe c_i. Ela tem valor natural igual a zero se t_k e c_i são independentes. Pode-se calcular a estatística χ^2 de um termo t_k em uma classe c_i por

$$\chi^2(t_k) = \frac{|Tr|.[P(t_k, c_i).P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i).P(\bar{t}_k, c_i)]^2}{P(t_k).P(\bar{t}_k).P(c_i).P(\bar{c}_i)}.$$

2.2.6 Pontuação de relevância (PR)

PR mede a relação entre a presença do termo t_k na classe c_i e a ausência do mesmo na classe oposta $\bar{c}_i$ [29],

$$PR(t_k, c_i) = log\frac{P(t_k, c_i) + d}{P(\bar{t}_k, \bar{c}_i) + d},$$

sendo d um fator constante de amortecimento.

As funções f_{sum}, f_{wsum} ou f_{max} podem ser utilizadas para combinar as pontuações específicas de cada categoria.

2.2.7 Razão de chances (RC)

A RC foi proposta por Van Rijsbergen [45] para selecionar termos por realimentação de relevância. Trata-se de uma medida particularmente importante em estatística Bayesiana e regressão logística. Ela calcula a razão entre as chances de o termo aparecer em uma mensagem relevante em relação às chances de aparacer em uma mensagem não relevante. Em outras palavras, a RC é capaz de encontrar termos comumente presentes em mensagens que pertencem a uma certa classe [14]. A razão de chances entre t_k e c_i é dada por

$$RC(t_k, c_i) = \frac{P(t_k, c_i).(1 - P(t_k, \bar{c}_i))}{(1 - P(t_k, c_i)).P(t_k, \bar{c}_i)}.$$

Um RC igual a 1 indica que o termo t_k é representativo em ambas as classes c_i e $\bar{c}_i$. RC maior que 1 indica que t_k é mais representativo na classe c_i. Por outro lado, RC menor que 1 indica que t_k é menos representativo na classe c_i. Entretanto, o valor de RC sempre deve ser maior ou igual a zero. Quanto mais as chances de c_i se aproximarem de zero, o valor de RC também será próximo de zero. Quando as chances de $\bar{c}_i$ se aproximarem de zero, o valor de RC tenderá ao infinito.

Assim como em IM e PR, para calcular a representatividade de um termo em um senso global podem-se utilizar as funções f_{sum}, f_{wsum} ou f_{max}.

2.2.8 Coeficiente GSS (GSS)

O coeficiente GSS pode ser visto como uma variação simplificada da estatística χ^2, definido por [23]:

$$GSS(t_k) = P(t_k, c_i).P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i).P(\bar{t}_k, c_i).$$

Valores positivos correspondem a termos com indícios de pertinência, enquanto que valores negativos representam indícios de não pertinência. Isso significa que, quanto maior (menor) for o valor positivo (negativo), maior será o indício de pertinência (não pertinência) de t_k na classe c_i.

Para melhor conveniência, as equações matemáticas de todas as medidas apresentadas nesta seção estão sintetizadas na Tabela 1[10].

Tabela 1. Técnicas mais populares usadas para seleção de termos

Técnica	Denotação	Equação
Frequência de documento	$FD(t_k)$	$\frac{\lvert Tr_{t_k}\rvert}{\lvert Tr\rvert}$
Fator de associação DIA	$DIA(t_k, c_i)$	$P(c_i \lvert t_k)$
Ganho de informação	$GI(t_k)$	$\sum_{c\in[c_i,\bar{c}_i]}\sum_{t\in[t_k,\bar{t}_k]} P(t,c).log\frac{P(t,c)}{P(t).P(c)}$
Informação mútua	$IM(t_k, c_i)$	$log\frac{P(t_k,c_i)}{P(t_k).P(c_i)}$
Estatística χ^2	$\chi^2(t_k)$	$\frac{\lvert Tr\rvert.[P(t_k,c_i).P(\bar{t}_k,\bar{c}_i)-P(t_k,\bar{c}_i).P(\bar{t}_k,c_i)]^2}{P(t_k).P(\bar{t}_k).P(c_i).P(\bar{c}_i)}$
Pontuação de relevância	$PR(t_k, c_i)$	$log\frac{P(t_k,c_i)+d}{P(t_k,\bar{c}_i)+d}$
Razão de chances	$RC(t_k, c_i)$	$\frac{P(t_k,c_i).(1-P(t_k,\bar{c}_i))}{(1-P(t_k,c_i)).P(t_k,\bar{c}_i)}$
Coeficiente GSS	$GSS(t_k)$	$P(t_k, c_i).P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i).P(\bar{t}_k, c_i)$

2.3 Análise dos resultados da literatura

Resultados presentes na literatura indicam que, dentre as técnicas utilizadas para reduzir a dimensionalidade do espaço de termos empregadas em categorização de textos, $\{GI, \chi^2, FD\} > \{DIA, RC, PR, GSS\} >> IM$, sendo que "$>$" significa "oferece melhor desempenho que" [2, 5]. Entretanto, se levarmos em conta o desempenho médio obtido pelos classificadores, os resultados indicam que GI e χ^2 oferecem melhores resultados que FD [2, 4]. Por outro lado, o emprego da técnica IM não apresentou resultados satisfatórios [4, 5]. Para maiores detalhes, consulte-se Almeida et al. [5].

3 Filtros anti-spam Naïve Bayes

Os filtros probabilísticos são historicamente os primeiros a terem sido propostos e até hoje são amplamente utilizados [17]. Tais métodos são bastante empregados na classificação automática de spams devido a sua simplicidade e alto desempenho [48].

O primeiro programa que utilizou recursos da probabilidade Bayesiana para filtragem de spams foi o iFile, desenvolvido por Jason Rennie em 1996[11]. Todavia, o primeiro trabalho acadêmico a propor um filtro Bayesiano anti-spam foi apresentado por Sahami et al. [38]. A partir de então, diversas outras técnicas variantes têm sido propostas em uma grande quantidade de trabalhos de pesquisa e programas comercializados. Em 2002, os princípios da filtragem Bayesiana foram amplamente difundidos por Paul Graham[12].

Classificadores Bayesianos anti-spam tornaram-se mecanismos populares e diversos servidores modernos de e-mail passaram a utilizá-los. Filtros locais (por usuário) consagrados, como OSBF-Lua[13], DSPAM[14], Spa-

[10] A Tabela 1 apresenta todas as funções em termos de probabilidade subjetiva. Em alguns casos, como $\chi^2(t_k)$, isso é um pouco artificial, uma vez que essa função não é vista habitualmente em termos de probabilidade. As equações referem-se à forma "local" (específica por categoria) das funções.
[11] Fonte: http://people.csail.mit.edu/jrennie/ifile/
[12] Fonte: http://www.paulgraham.com/spam.html
[13] Vencedor do TREC's Spam Track 2006 e do CEAS 2008 Spam Filter Live Challenge. Consulte http://osbf-lua.luaforge.net/
[14] Fonte: http://dspam.nuclearelephant.com/index.shtml

mAssassin[15], SpamBayes[16], Bogofilter[17] e ASSP[18], são baseados em técnicas de classificação Bayesiana.

Dado um conjunto de mensagens $\mathcal{M} = \{m_1, m_2, \ldots, m_j, \ldots, m_{|\mathcal{M}|}\}$ e um conjunto de classes $\mathcal{C} = \{c_s = \texttt{spam}, c_l = \texttt{legitima}\}$, sendo que m_j é a j-ésima mensagem de $\mathcal{M}$ e $\mathcal{C}$ são as possíveis classes, a tarefa de classificação de spams consiste, basicamente, em uma função de categorização Booleana $\Phi(m_j, c_i) : \mathcal{M} \times \mathcal{C} \rightarrow \{\texttt{Verdadeiro}, \texttt{Falso}\}$. Quando $\Phi(m_j, c_i)$ é `Verdadeiro`, indica que a mensagem m_j pertence a categoria c_i; em caso contrário, m_j não pertence a c_i.

Na classificação de spams existem apenas duas possíveis categorias: `spam` e `legitima`. Cada mensagem $m_j \in \mathcal{M}$ pode ser associada a apenas uma delas, mas nunca a ambas. Dessa forma, podem-se utilizar uma função de categorização mais simplificada $\Phi_{\texttt{spam}}(m_j) : \mathcal{M} \rightarrow \{\texttt{Verdadeiro}, \texttt{Falso}\}$. Assim, uma mensagem é classificada como spam quando $\Phi_{\texttt{spam}}(m_j)$ for `Verdadeiro`, e legítima em caso contrário.

A aplicação dos algoritmos de aprendizagem supervisionada com ênfase em filtragem de spams consiste em duas etapas [46]:

1. *Treinamento*: um conjunto de mensagens rotuladas (Tr) deve ser fornecido como fonte de treinamento. Elas são previamente convertidas para uma representação que o algoritmo consiga compreender. A representação mais empregada para filtragem de spams é o modelo de espaço vetorial, sendo cada documento $m_j \in Tr$ transformado em um vetor real $\vec{x}_j \in \Re^{|\mathcal{V}|}$, no qual $\mathcal{V}$ é o vocabulário (conjunto de características) e as coordenadas de $\vec{x}_j$ representam o peso de cada característica de $\mathcal{V}$. Assim, podemos utilizar o algoritmo de aprendizado e os dados de treinamento para criar um classificador $\Phi_{\texttt{spam}}(\vec{x_j}) \rightarrow \{\texttt{Verdadeiro}, \texttt{Falso}\}$.
2. *Classificação*: o classificador $\Phi_{\texttt{spam}}(\vec{x_j})$ é aplicado na representação vetorial de uma mensagem nova $\vec{x}$ para produzir uma predição se ela é spam ou não.

O classificador Naïve Bayes (NB) proposto por Bernardo and Smith [9] e Duda and Hart [16] é o filtro mais simples derivado da teoria da decisão Bayesiana [9]. Do teorema de Bayes e da teoria da probabilidade total, sabe-se que a probabilidade de uma mensagem $\vec{x} = \langle x_1, \ldots, x_n \rangle$ pertencer à categoria $c_i \in \{c_s, c_l\}$ é:

$$P(c_i|\vec{x}) = \frac{P(c_i).P(\vec{x}|c_i)}{P(\vec{x})}.$$

Uma vez que o denominador não depende da classe, o filtro NB classifica cada mensagem na categoria que maximiza $P(c_i).P(\vec{x}|c_i)$. Isso é equivalente a classificar uma mensagem como spam (c_s) se

$$\frac{P(c_s).P(\vec{x}|c_s)}{P(c_s).P(\vec{x}|c_s) + P(c_l).P(\vec{x}|c_l)} > T,$$

com $T = 0,5$. Variando-se T, obtêm-se mais verdadeiros negativos e menos verdadeiros positivos ou vice-versa. A probabilidade $P(c_i)$ pode ser estimada pela frequência de documentos que pertencem à classe c_i no conjunto de treinamento Tr, enquanto que $P(\vec{x}|c_i)$ é praticamente impossível de ser estimada diretamente, pois isso requer que Tr contenha mensagens idênticas às que estão sendo classificadas. Entretanto, os classificadores NB fazem a suposição de que os termos de uma mensagem são condicionalmente independentes e que a ordem em que eles aparecem é irrelevante.

Apesar de essa suposição ser um tanto quanto simplista, diversos estudos indicam que o classificador NB é surpreendentemente eficaz na filtragem de spams [8, 35, 44, 48].

Embora esses filtros sejam bastante estudados e utilizados para a classificação de spams, um fato não muito contemplado pela literatura é a existência de várias formas distintas de estimar as probabilidades Bayesianas $P(\vec{x}|c_i)$, cada uma delas representando uma versão diferente de filtro anti-spam NB [1, 2, 4].

A seguir, são apresentadas sete versões distintas de classificadores anti-spam NB.

[15]Fonte: http://spamassassin.apache.org/
[16]Fonte: http://spambayes.sourceforge.net/
[17]Fonte: http://bogofilter.sourceforge.net/
[18]Fonte: http://assp.sourceforge.net/

3.1 NB básico

Conhece-se por NB básico o primeiro classificador anti-spam NB, proposto por Sahami et al.[38]. Seja $\mathcal{T}' = \{t_1, \ldots, t_n\}$ o conjunto de termos (tokens) extraídos das mensagens, cada mensagem m é representada por uma vetor binário $\vec{x} = \langle x_1, \ldots, x_n \rangle$, sendo que o valor de cada x_k indica se o termo t_k ocorreu ou não em m. As probabilidades $P(\vec{x}|c_i)$ são calculadas por

$$P(\vec{x}|c_i) = \prod_{k=1}^{n} P(t_k|c_i),$$

e o critério para classificar a mensagem como spam é dado por

$$\frac{P(c_s).\prod_{k=1}^{n} P(t_k|c_s)}{\sum_{c_i \in \{c_s, c_l\}} P(c_i).\prod_{k=1}^{n} P(t_k|c_i)} > T.$$

Assim, as probabilidades $P(t_k|c_i)$ são estimadas por

$$P(t_k|c_i) = \frac{|Tr_{t_k,c_i}|}{|Tr_{c_i}|},$$

sendo $|Tr_{t_k,c_i}|$ a quantidade de mensagens de treinamento da categoria c_i que contêm o termo t_k e $|Tr_{c_i}|$ o número total de mensagens de treinamento pertencentes à categoria c_i.

3.2 NB multinomial com frequência de termos

No classificador NB multinomial com frequência de termos (NB MN FT), cada mensagem é representada como um conjunto de termos $m = \{t_1, \ldots, t_n\}$, no qual cada t_k informa a quantidade de vezes que ele aparece em m. Neste caso, m pode ser representado por um vetor $\vec{x} = \langle x_1, \ldots, x_n \rangle$, tal que cada x_k corresponde ao número de ocorrências de t_k em m. Além disso, cada mensagem m de categoria c_i pode ser interpretada como o resultado da escolha independente de $|m|$ termos de $\mathcal{T}'$ como substitutos e probabilidade $P(t_k|c_i)$ para cada t_k [34]. Portanto, $P(\vec{x}|c_i)$ é a distribuição multinomial:

$$P(\vec{x}|c_i) = P(|m|).|m|!.\prod_{k=1}^{n} \frac{P(t_k|c_i)^{x_k}}{x_k!}.$$

Dessa forma, o critério para classificar uma mensagem como spam torna-se

$$\frac{P(c_s).\prod_{k=1}^{n} P(t_k|c_s)^{x_k}}{\sum_{c_i \in \{c_s, c_l\}} P(c_i).\prod_{k=1}^{n} P(t_k|c_i)^{x_k}} > T,$$

e as probabilidades $P(t_k|c_i)$ são calculadas pela estimativa Laplaciana

$$P(t_k|c_i) = \frac{1 + N_{t_k,c_i}}{n + N_{c_i}},$$

sendo N_{t_k,c_i} correspondente ao número de ocorrências do termo t_k nas mensagens de treinamento da categoria c_i e $N_{c_i} = \sum_{k=i}^{n} N_{t_k,c_i}$.

3.3 NB multinomial Booleano

O classificador NB multinomial Booleano (NB MN Bool) é similar ao multinomial com frequência de termos, incluindo a estimativa de $P(t_k|c_i)$, exceto pelo fato de que cada atributo x_k é Booleano. Note-se que esses métodos não levam em conta a ausência de termos ($x_k = 0$) nas mensagens. Schneider [41] demonstrou

que o classificador NB multinomial Booleano costuma ter desempenho superior ao multinomial com frequência de termos.

Isso se deve ao fato de que o classificador NB multinomial com frequência de termos é equivalente a versão do classificador NB com atributos modelados seguindo uma distribuição de Poisson em cada categoria, assumindo que o comprimento de cada mensagem é independente da categoria. Dessa forma, o classificador NB multinomial pode obter desempenho melhor usando atributos Booleanos, caso as frequências dos termos não sigam uma distribuição de Poisson.

3.4 NB multivariado Bernoulli

Seja $\mathcal{T}' = \{t_1, \ldots, t_n\}$ o conjunto de termos extraídos das mensagens, o classificador NB multivariado Bernoulli (NB MV Bern) representa cada mensagem m em termos de presença e ausência de cada termo. Dessa forma, m pode ser representada como um vetor binário $\vec{x} = \langle x_1, \ldots, x_n \rangle$, em que cada x_k informa se t_k está presente ou não em m. Além disso, cada mensagem m de categoria c_i é vista como o resultado de n tentativas de Bernoulli, no qual cada tentativa decide se t_k aparece ou não em m. A probabilidade de obter um resultado positivo em uma tentativa k é dada por $P(t_k|c_i)$. Dessa forma, as probabilidades $P(\vec{x}|c_i)$ são calculadas por

$$P(\vec{x}|c_i) = \prod_{k=1}^{n} P(t_k|c_i)^{x_k}.(1 - P(t_k|c_i))^{(1-x_k)}.$$

O critério para classificar uma mensagem como spam torna-se:

$$\frac{P(c_s).\prod_{k=1}^{n} P(t_k|c_s)^{x_k}.(1 - P(t_k|c_s))^{(1-x_k)}}{\sum_{c_i \in \{c_s, c_l\}} P(c_i).\prod_{k=1}^{n} P(t_k|c_i)^{x_k}.(1 - P(t_k|c_i))^{(1-x_k)}} > T,$$

e as probabilidades $P(t_k|c_i)$ são calculadas pela estimativa Laplaciana

$$P(t_k|c_i) = \frac{1 + |Tr_{t_k,c_i}|}{2 + |Tr_{c_i}|},$$

sendo $|Tr_{t_k,c_i}|$ a quantidade de mensagens de treinamento que pertencem à categoria c_i e que possuem o termo t_k e $|Tr_{c_i}|$ o número total de mensagens de treinamento da categoria c_i. Para maiores informações a respeito desse classificador, consulte-se Losada e Azzopardi [32].

3.5 NB Booleano

Denomina-se NB Booleano o classificador similar ao NB multivariado Bernoulli, com a diferença de que ele não leva em consideração a ausência de termos. Portanto, as probabilidades $P(\vec{x}|c_i)$ são calculadas somente por

$$P(\vec{x}|c_i) = \prod_{k=1}^{n} P(t_k|c_i),$$

e o critério para classificar uma mensagem como spam torna-se:

$$\frac{P(c_s).\prod_{k=1}^{n} P(t_k|c_s)}{\sum_{c_i \in \{c_s, c_l\}} P(c_i).\prod_{k=1}^{n} P(t_k|c_i)} > T.$$

As probabilidades $P(t_k|c_i)$ são estimadas da mesma forma do classificador NB multivariado Bernoulli.

3.6 NB Gauss multivariado

O classificador NB Gauss multivariado (NB Gauss MV) utiliza atributos com valores reais, assumindo que cada atributo segue uma distribuição Gaussiana $g(x_k; \mu_{k,c_i}, \sigma_{k,c_i})$ para cada categoria c_i, tal que os valores de μ_{k,c_i} e σ_{k,c_i} de cada distribuição são estimados do conjunto de treinamento Tr.

As probabilidades $P(\vec{x}|c_i)$ são calculadas por

$$P(\vec{x}|c_i) = \prod_{k=1}^{n} g(x_k; \mu_{k,c_i}, \sigma_{k,c_i}),$$

e o critério para classificar uma mensagem como spam torna-se

$$\frac{P(c_s).\prod_{k=1}^{n} g(x_k; \mu_{k,c_s}, \sigma_{k,c_s})}{\sum_{c_i \in \{c_s,c_l\}} P(c_i).\prod_{k=1}^{n} g(x_k; \mu_{k,c_i}, \sigma_{k,c_i})} > T.$$

3.7 Bayes flexível

O classificador Bayes flexível é similar ao NB Gauss multivariado. Entretanto, ao invés de usar uma única distribuição normal para cada atributo X_k por categoria c_i, o filtro Bayes flexível representa as probabilidades $P(\vec{x}|c_i)$ como médias de L_{k,c_i} distribuições normais com diferentes valores para μ_{k,c_i}, mas com o mesmo valor para σ_{k,c_i}

$$P(x_k|c_i) = \frac{1}{L_{k,c_i}} \sum_{l=1}^{L_{k,c_i}} g(x_k; \mu_{k,c_i,l}, \sigma_{c_i}),$$

sendo que L_{k,c_i} representa a quantidade de valores diferentes que o atributo X_k assume nas mensagens de categoria c_i do conjunto de treinamento Tr. Cada um desses valores é usado como $\mu_{k,c_i,l}$ de uma distribuição normal da categoria c_i. No entanto, todas as distribuições de uma categoria c_i são calculadas com o mesmo $\sigma_{c_i} = \frac{1}{\sqrt{|Tr_{c_i}|}}$.

Assim, a distribuição de cada categoria torna-se cada vez mais "limitada" conforme as mensagens de treinamento de cada categoria são acumuladas. Pela média de muitas distribuições normais, o filtro Bayes flexível consegue aproximar as distribuições verdadeiras dos atributos melhor que o classificador NB Gauss multivariado, quando a suposição de que eles seguem uma distribuição normal é violada. Para obter maiores detalhes sobre esse método de classificação, consulte-se Androutsopoulos et al. [8] e John e Langley [27].

A Tabela 2 sumariza as sete versões de classificadores anti-spam NB apresentadas nesta seção[19].

3.8 Análise dos resultados da literatura

Resultados presentes na literatura indicam que o desempenho dos classificadores Naïve Bayes é altamente sensível à qualidade dos termos selecionados pelas técnicas de seleção de termos (TSTs), bem como ao número de termos selecionados $|\mathcal{T}'|$. Geralmente, os classificadores pioram o desempenho quando o conjunto completo de termos $|\mathcal{T}|$ é utilizado, com exceção da técnica IM. Portanto, não empregar IM para reduzir a dimensão do espaço de termos é mais favorável à classificação. Entretanto, existe um intervalo entre 30% – 60% de $|\mathcal{T}|$ que oferece melhor desempenho para os demais métodos de seleção. Mesmo um conjunto de termos reduzido, com apenas 10% – 30% de $|\mathcal{T}|$, oferece resultados superiores ao conjunto completo. Dessa forma, empiricamente foi possível comprovar que o emprego de técnicas de seleção de termos na filtragem de spams, além de reduzir a dimensionalidade, aumenta a capacidade de predição dos classificadores Naïve Bayesianos [4, 5, 6, 7, 8, 35].

Com respeito ao desempenho dos classificadores apresentados, a análise dos resultados presentes na literatura indica que {NB Booleano, NB Básico, Bayes flexível} $>$ {NB Booleano MN, NB Bernoulli MV, NB Gauss MV, NB MN com frequência de termos} para filtragem automática de spams [35]. Os filtros NB Booleano e NB Básico vêm obtendo mais sucesso para a maiorias das bases de e-mails [2, 4].

Adicionalmente, vários resultados experimentais comprovaram que os filtros que utilizam atributos Booleanos são mais eficientes que aqueles que empregam frequência de termos, apesar de a frequência de termos oferecer mais informações [4, 35, 40, 41].

[19] As complexidades computacionais estão de acordo com Metsis et al. [35]. Em relação ao custo de classificação, a complexidade do Bayes flexível é $O(n.|Tr|)$, porque ele precisa somar L_k distribuições.

Tabela 2. Diferentes versões de filtros anti-spam Naive Bayes

Classificador NB	$P(\vec{x}\|c_i)$	Complexidade Treinamento	Complexidade Classificação
NB Básico	$\prod_{k=1}^{n} P(t_k\|c_i)$	$O(n.\|Tr\|)$	$O(n)$
NB MN FT	$\prod_{k=1}^{n} P(t_k\|c_i)^{x_k}$	$O(n.\|Tr\|)$	$O(n)$
NB MN Booleano	$\prod_{k=1}^{n} P(t_k\|c_i)^{x_k}$	$O(n.\|Tr\|)$	$O(n)$
NB MV Bernoulli	$\prod_{k=1}^{n} P(t_k\|c_i)^{x_k}.(1-P(t_k\|c_i))^{(1-x_k)}$	$O(n.\|Tr\|)$	$O(n)$
NB Booleano	$\prod_{k=1}^{n} P(t_k\|c_i)$	$O(n.\|Tr\|)$	$O(n)$
NB Gauss MV	$\prod_{k=1}^{n} g(x_k;\mu_{k,c_i},\sigma_{k,c_i})$	$O(n.\|Tr\|)$	$O(n)$
Bayes Flexível	$\prod_{k=1}^{n} \frac{1}{L_{k,c_i}} \sum_{l=1}^{L_{k,c_i}} g(x_k;\mu_{k,c_i,l},\sigma_{c_i})$	$O(n.\|Tr\|)$	$O(n.\|Tr\|)$

Metsis et al. [35] mostraram que o classificador Bayes flexível é menos sensível à variação do parâmetro T que os demais filtros analisados. Em seus experimentos, utilizaram diferentes filtros Bayesianos para classificar a mesma base de dados, variando o valor de T de zero a um. Os resultados obtidos indicam que filtro Bayes flexível é capaz de obter um alto grau de reconhecimento de spams mesmo em situações nas quais se requer um alto grau de reconhecimento de mensagens legítimas (altos valores para T). Os demais classificadores tendem a obter melhor reconhecimento de spams conforme T diminui [5].

4 Conclusões e perspectivas futuras

Neste trabalho, apresentamos um tutorial sobre diversos métodos seletores de termos empregados para reduzir a dimensionalidade do espaço de atributos na tarefa de filtragem automática de spams realizada por classificadores Naïve Bayesianos. Adicionalmente, revisamos sete versões diferentes de filtros Naïve Bayes encontrados na literatura.

Em relação às técnicas empregadas para seleção de termos, resultados presentes na literatura indicam que Ganho de informação, Estatística χ^2 e Frequência de documentos são mais agressivas na remoção de termos sem causar perda de acurácia. Fator de associação DIA, Pontuação de relevância, Razão de chances e Coeficiente GSS também oferecem ganho de desempenho. Por outro lado, o uso de Informação mútua obteve resultados insatisfatórios e frequentemente degrada a performance do classificador. Os resultados também indicam que Ganho de informação e Estatística χ^2 apresentam os melhores desempenhos médios por serem menos sensíveis à variação de $|\mathcal{T}'|$.

Dentre todos os classificadores apresentados, NB Booleano e NB Básico vêm obtendo melhor desempenho para a maioria das bases de dados. A análise dos resultados presentes na literatura também é conclusiva no sentido de que o emprego de atributos Booleanos é melhor que por frequência de termos. Além disso, também verificamos que o desempenho dos classificadores Naïve Bayesianos é altamente sensível à qualidade e ao número de termos selecionados na etapa de treinamento.

Trabalhos futuros devem levar em consideração que a filtragem de spams é um problema coevolucionário, pois enquanto o filtro tenta evoluir a sua capacidade de predição, os spammers tentam evoluir os spams com o intuito de que as suas mensagens não sejam identificadas pelos classificadores. Dessa forma, um método eficiente deve possuir uma maneira eficaz de ajustar as suas regras de forma a detectar mudanças nas características dos

spams. Nesse sentido, filtros colaborativos [31] podem ser empregados para auxiliar o classificador, acelerando o processo de adaptação de regras. Além disso, os spammers geralmente inserem uma grande quantidade de ruídos nas mensagens com o intuito de dificultar a estimação das probabilidades. Assim, os filtros deveriam adotar técnicas mais flexíveis para comparar os termos na etapa de classificação. Métodos baseados em lógica fuzzy [37] poderiam ser utilizados para permitir tal flexibilização.

Agradecimentos

Os autores são gratos à Capes, à Fapesp e ao CNPq pelo apoio financeiro.

Referências

[1] ALMEIDA, T.; YAMAKAMI, A. Content-Based Spam Filtering. In: 23rd IEEE International Joint Conference on Neural Networks. *Proceedings*, p. 1-7, 2010.

[2] ALMEIDA, T.; YAMAKAMI, A.; ALMEIDA, J. Evaluation of Approaches for Dimensionality Reduction Applied with Naive Bayes Anti-Spam Filters. In: 8th IEEE International Conference on Machine Learning and Applications. *Proceedings*, p. 517-522, 2009.

[3] ALMEIDA, T.; YAMAKAMI, A.; ALMEIDA, J. Filtering Spams using the Minimum Description Length Principle. In: 25th ACM Symposium On Applied Computing. *Proceedings*, p. 1856-1860, 2010.

[4] ALMEIDA, T.; YAMAKAMI, A.; ALMEIDA, J. Probabilistic Anti-Spam Filtering with Dimensionality Reduction. In: 25th ACM Symposium On Applied Computing. *Proceedings*, p. 1804-1808, 2010.

[5] ALMEIDA, T.; ALMEIDA, J.; YAMAKAMI, A. Spam Filtering: How the Dimensionality Reduction Affects the Accuracy of Naive Bayes Classifiers . *Journal of Internet Services and Applications*, 1(3), 183-200, 2011.

[6] ANDROUTSOPOULOS, I. et al. An Evalutation of Naive Bayesian Anti-Spam Filtering. In: 11st European Conference on Machine Learning. *Proceedings*, p. 9-17, 2000.

[7] ANDROUTSOPOULOS, I. et al. Learning to Filter Spam E-mail: A Comparison of a Naive Bayesian and a Memory-Based Approach. In: 4th European Conference on Principles and Practice of Knowledge Discovery in Databases. *Proceedings*, p. 1-13, 2000.

[8] ANDROUTSOPOULOS, I.; PALIOURAS, G.; MICHELAKIS, E. *Learning to Filter Unsolicited Commercial E-Mail*. Technical Report 2004/2, National Centre for Scientific Research "Demokritos", 2004.

[9] BERNARDO, J.; SMITH, A. *Bayesian Theory*. Wiley & Sons. 1994.

[10] CARRERAS, X.; MARQUEZ, L. Boosting Trees for Anti-Spam Email Filtering. In: 4th International Conference on Recent Advances in Natural Language Processing. *Proceedings*, p. 58-64, 2001.

[11] COHEN, W. Fast Effective Rule Induction. In: *Proceedings of 12nd International Conference on Machine Learning*. *Proceedings*, p. 115-123, 1995.

[12] COHEN, W. Learning Rules that Classify E-mail. In: AAAI Spring Symposium on Machine Learning in Information Access. *Proceedings*, p. 18-25, 1996.

[13] CORMACK, G. Email Spam Filtering: A Systematic Review. *Foundations and Trends in Information Retrieval*, 1(4), 335–455. 2008.

[14] Cunningham, P. et al. A Case-Based Approach to Spam Filtering than Can Track Concept Drift. In: 5th International Conference on Case Based Reasoning. *Proceedings*, p. 115-123, 2003.

[15] DRUCKER, H.; WU, D.; VAPNIK, V. Support Vector Machines for Spam Categorization. *IEEE Transactions on Neural Networks*, 10(5), 1048–1054. 1999.

[16] DUDA, R.; HART, P. *Bayes Decision Theory*, chapter 2, p. 10-43. John Wiley & Sons. 1973.

[17] DUNNING, T. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1), 61-74. 1993

[18] FORMAN, G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, 3, 1289-1305. 2003.

[19] FORMAN, G.; KIRSHENBAUM, E. Extremely Fast Text Feature Extraction for Classification and Indexing. In: 17th ACM Conference on Information and Knowledge Management. *Proceedings*, p. 1221-1230, 2008.

[20] FORMAN, G.; SCHOLZ, M.; and RAJARAM, S. Feature Shaping for Linear SVM Classifiers. In: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. *Proceedings*, pages 299–308, 2009.

[21] FRIEDMAN, N.; GEIGER, D.; and GOLDSZMIDT, M. Bayesian Network Classifiers. *Machine Learning*, 29(3), 131-163. 1997.

[22] FUHR, N.; BUCKLEY, C. A Probabilistic Learning Approach for Document Indexing. *ACM Transactions on Information Systems*, 9(3), 223–248. 1991.

[23] GALAVOTTI, L.; SEBASTIANI, F.; SIMI, M. Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization. In: 4th European Conference on Research and Advanced Technology for Digital Libraries. *Proceedings*, p. 59-68, 2000.

[24] GUZELLA, T.; CAMINHAS, W. A Review of Machine Learning Approaches to Spam Filtering. *Expert Systems with Applications*, 36(7), 10206-10222. 2009.

[25] HIDALGO, J. Evaluating Cost-Sensitive Unsolicited Bulk Email Categorization. In: 17th ACM Symposium on Applied Computing. *Proceedings*, p. 615-620, 2002.

[26] JOACHIMS, T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In: 14th International Conference on Machine Learning. *Proceedings*, p. 143-151, 1997

[27] JOHN, G.; LANGLEY, P. Estimating Continuous Distributions in Bayesian Classifiers. In: 11st International Conference on Uncertainty in Artificial Intelligence. *Proceedings*, p. 338-345, 1995.

[28] JOHN, G.; KOHAVI, R.; and PFLEGER, K. Irrelevant Features and the Subset Selection Problem. In: International Conference on Machine Learning. *Proceedings*, p. 121-129, 1994.

[29] KIRA, K.; RENDELL, L. A Practical Approach to Feature Selection. In: 9th International Workshop on Machine Learning. *Proceedings*, p. 249-256, 1992.

[30] KOLCZ, A.; ALSPECTOR, J. SVM-based Filtering of E-mail Spam with Content-Specific Misclassification Costs. In: 1st International Conference on Data Mining. *Proceedings*, p. 1-14, 2001.

[31] LEMIRE, D. Scale and Translation Invariant Collaborative Filtering Systems. *Information Retrieval*, 8(1), 129-150. 2005.

[32] LOSADA, D.; AZZOPARDI, L. Assessing Multivariate Bernoulli Models for Information Retrieval. *ACM Transactions on Information Systems*, 26(3), 1-46. 2008.

[33] MARSONO, M.; EL-KHARASHI, N.; GEBALI, F. Targeting Spam Control on Middleboxes: Spam Detection Based on Layer-3 E-mail Content Classification. *Computer Networks*, 53(6), 835-848. 2009.

[34] McCALLUM, A.; NIGAM, K. A Comparison of Event Models for Naive Bayes Text Classication. In: 15th AAAI Workshop on Learning for Text Categorization. *Proceedings*, p. 41-48, 1998.

[35] METSIS, V.; ANDROUTSOPOULOS, I.; PALIOURAS, G. Spam Filtering with Naive Bayes - Which Naive Bayes? In: 3rd International Conference on Email and Anti-Spam. *Proceedings*, p. 1-5, 2006.

[36] MITCHELL, T. *Machine Learning*. McCraw-Hill. 1997.

[37] PEDRYCZ, W.; GOMIDE, F. *Fuzzy Systems Engineering: Toward Human-Cenric Computing*. IEEE/Wiley Interscience. 2007.

[38] SAHAMI, M. et al. A Bayesian Approach to Filtering Junk E-mail. In: 15th National Conference on Artificial Intelligence. *Proceedings*, p. 55-62, 1998.

[39] SCHAPIRE, R.; SINGER, Y.; SINGHAL, A. Boosting and Rocchio Applied to Text Filtering. In: 21st Annual International Conference on Information Retrieval. *Proceedings*, p. 215-223, 1998.

[40] SCHNEIDER, K. A Comparison of Event Models for Naive Bayes Anti-spam E-mail Filteriring. In: 10th Conference of the European Chapter of the Association for Computational Linguistics. *Proceedings*, p. 307-314, 2003.

[41] SCHNEIDER, K. On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification. In: 4th International Conference on Advances in Natural Language Processing. *Proceedings*, p. 474-485, 2004.

[42] SEBASTIANI, F. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1), 1-47. 2002.

[43] SEEWALD, A. An Evaluation of Naive Bayes Variants in Content-based Learning for Spam Filtering. *Intelligent Data Analysis*, 11(5), 497-524. 2007.

[44] SONG, Y.; KOLCZ, A.; GILEZ, C. Better Naive Bayes Classification for High-precision Spam Detection. *Software – Practice and Experience*, 39(11), 1003-1024. 2009.

[45] VAN RIJSBERGEN, C. *Information Retrieval*.2 edition. Butterworths, London. 1979.

[46] VAPNIK, V. N. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, USA. 1995.

[47] YANG, Y.; PEDERSEN, J. A Comparative Study on Feature Selection in Text Categorization. In: 14th International Conference on Machine Learning. *Proceedings*, p. 412-420, 1997.

[48] ZHANG, L.; ZHU, J.; YAO, T. An Evaluation of Statistical Spam Filtering Techniques. *ACM Transactions on Asian Language Information Processing*, 3(4), 243-269. 2004.