Vanessa Cristina Sabino

Categorização de Textos Usando Máquinas de Suporte Vetorial

Trabalho de Graduação apresentado ao Instituto de Matemática e Estatística da Universidade de São Paulo para obtenção de grau de Bacharelado em Matemática Aplicada e Computacional com Habilitação em Comunicação Científica

Orientadores:

Alair Pereira do Lago

Rosana de Lima Soares

UNIVERSIDADE DE SÃO PAULO
INSTUTUTO DE MATEMÁTICA E ESTATÍSTICA

São Paulo

Dezembro 2006

Resumo

Discorre sobre a dificuldade em gerenciar eficientemente o grande volume de conhecimento disponível atualmente devido aos avanços nos meios de comunicação e apresenta a categorização de textos como uma das abordagens para facilitar a recuperação de informação. É discutida a técnica de aprendizagem computacional conhecida como Máquinas de Suporte Vetorial, que apresenta bons resultados para a tarefa de classificação e está bem fundamentada em teorias matemáticas e estatísticas.

Sumário

1 Introdução

Um dos problemas mais difíceis que a comunidade científica, e a humanidade em geral, se depara hoje é encontrar uma forma de gerenciar eficientemente o conhecimento que a sociedade desenvolve diariamente, tanto em pesquisas científicas como no jornalismo tradicional.

A explosão de informação que enfrentamos hoje iniciou-se na época da Segunda Guerra Mundial, em que cientistas desenvolveram e compartilharam um corpo de conhecimento enorme. Um desses cientistas, Vannevar Bush, preocupado em automatizar processos repetitivos do pensamento humano, idealizou, em 1945, a máquina teórica chamada MEMEX, detalhada em seu artigo "As We May Think". O processo de funcionamento desta máquina deu origem à idéia de hipertexto, uma das bases da Internet, que "constitue de fato uma espécie de objetivação, de exteriorização, de virtualização do processo de leitura" (LEVY, 1996). Naquela época, Bush afirmou que "A somatória da experiência humana está sendo expandida numa velocidade prodigiosa, e os meios que usamos para achar nosso caminho no labirinto resultante até o item que importa no momento são os mesmos usados nos tempos dos veleiros" (BUSH, 1988). Porém, na mesma época estavam surgindo os primeiros computadores, que são o meio de maior capacidade jamais inventado, com alta eficiência de representação de palavras e números, o que torna possível armazenar e recuperar uma quantidade de informação muito além do que antes era possível. Janet Murrey, em seu livro Hamlet no Holodeck, faz a seguinte análise sobre a evolução da capacidade de armazenamento: "A memória humana foi estendida, com o meio digital, de uma unidade básica de disseminação portátil de 100 mil palavras (um livro médio, que ocupa cerca de um megabyte de espaço em sua versão completamente formatada), primeiro, para 65 milhões de palavras (um CD-ROM de 650 megabytes, o equivalente a 650 livros) e, agora, para 530 milhões de palavras (um videodisco digital de 5,3 gigabytes, equivalentes a 5300 livros), e daí para cima. Uma vez que passemos para os banco de dados globais da internet, acessíveis através de uma teia mundial de computadores interligados, os recursos crescem exponencialmente" (MURRAY, 2003).

O sucesso da Internet e de várias outras tecnologias da informação ampliam a sobrecarga de informações, sendo que uma das principais características dessa sobrecarga é a existência

de vários documentos que cobrem o mesmo tópico. Assim, tornam-se necessários modelos e técnicas mais eficientes para integrar a informação em contextos complexos, com o objetivo de ajudar as pessoas a encontrarem informações úteis em resposta a suas necessidades.

Segundo Cristina Ponte (2004), o jornal é apreciado pela sua função social de disseminar notícias, conhecimentos e algum entretenimento. O primeiro passo para realizar essa tarefa é a coleta de informação de diversas fontes. Mas uma vez que toda forma de representação está migrando para o formato eletrônico e todos os computadores do mundo são potencialmente acessíveis entre si, é possível conceber uma única e compreensível biblioteca digital de pinturas, filmes, livros, jornais, programas de televisão, etc. Porém, como apontado por Murrey, "a realidade é muito mais caótica e fragmentada: as informações veiculadas em rede são geralmente incompletas ou enganosas; as rotinas de busca são, com freqüência, intoleravelmente enfadonhas e frustrantes; e a informação que desejamos muitas vezes parece dolorosamente fora de alcance" (MURRAY, 2003).

Sistemas de busca convencionais utilizam como regra principal a ocorrência de determinada palavra dentro de um documento, e dessa forma seus resultados não representam conceitos mais gerais. Uma analogia interessante é feita por Lúcia Leão (1999), que explica como Hofstadter (1979), ao decorrer sobre as estruturas de rede, aponta algumas distinções quanto às propriedades locais e globais. Segundo ele, as propriedades locais exigem um observador próximo, que veja um vértice por vez. Já as propriedades globais solicitam uma visão "vasta", que não se limita a detalhes. A forma total da rede é uma propriedade global. Lúcia posteriormente identifica três problemas básicos ao fazer uma pesquisa na Web: o de "acessar" e conseguir encontrar a informação desejada; o de selecionar as mais relevantes; e, finalmente, o de organizar os dados de uma forma clara. Durante a pesquisa é necessário manter o foco central forte o bastante para não se perder, mas a intuição aguçada e viva o suficiente para ir tecendo uma trama complexa que enriquece a pesquisa.

Para uma interpretação semântica do conteúdo atual da Web, faz-se necessário o uso de outras técnicas além da busca. Uma solução é abstrair os conceitos em categorias e classificar os textos de acordo com elas. Joachims (2002) define a classificação de textos como o processo de agrupamento de documentos em diferentes categorias ou classes. Sebastiani (1999) associa a classificação de textos à atividade de rotular textos em linguagem natural com categorias temáticas de um conjunto pré-definido.

A classificação automática de textos começou a ser estudada na década de 60, mas somente tornou-se viável com o avanço de hardware e software. Durante a década de 80, a classificação automática de textos era realizada através da criação manual de regras de composição de textos,

processo que envolvia o conhecimento de especialistas na área de discurso que abrange os conceitos a serem descritos nas categorias. Somente a partir da década de 90 começou a ser utilizado o paradigma de aprendizagem computacional para categorização de textos. Dessa forma, são construídos classificadores automaticamente por um processo indutivo, observando as características de um conjunto de documentos previamente classificados sob cada categoria por um especialista no domínio.

Diferentemente de outras abordagens para classificação de textos, que se baseiam principalmente em evidências empíricas, o modelo usado por Máquinas de Suporte Vetorial explica quando e porque há boa performance do método, com base em propriedades estatísticas das tarefas de classificação de textos.

2 *Classificação de Textos*

2.1 Tarefa de aprendizado

A aprendizagem computacional, quando usada para classificação, visa atribuir a uma determinada informação o rótulo da classe a qual ela pertence (RUSSEL; NORVIG, 2004). Quando utilizada a aprendizagem supervisionada, um classificador é gerado a partir de um conjunto de dados de treinamento produzido por um especialista. O objetivo é que esse classificador seja capaz de prever a classe de instâncias quaisquer do domínio em que foi treinado.

Para desenvolver métodos efetivos e medir seus resultados é necessário definir a tarefa de aprendizado formalmente. Existem várias abordagens, mas a mais usada nos estudos existentes é a de aprendizagem indutiva.

O objetivo da classificação de texto indutiva é inferir uma regra de classficiação de uma amostra de treinamento cujos dados já estão previamente classificados, de forma que a regra criada classifique novos exemplos com alta acurácia. Formalmente, é dado ao algoritmo de aprendizagem A uma amostra de treinamendo S de n exemplos $(\vec{x_1}, y_1), \ldots, (\vec{x_n}, y_n)$ escolhidos de forma independente e identicamente distribuídos (i.i.d.) de acordo com uma distribuição de probabilidade fixada desconhecida $Pr(\vec{x}, y_1)$. Cada exemplo consiste no vetor $\vec{x}$, que descreve o documento de acordo com a representação que está sendo usada, e sua classe y, que depende de tipo de tarefa de classificação que será executada. O risco $R(h)$ mede a performance da regra de classificação h baseado em uma função perda $L(h(\vec{x}), y) \in \Re$, que mede o quanto a classificação estimada por uma regra de classificação $h(\vec{x})$ e a classe observada y são diferentes. A medida de performance correspondente $R(h)$ é a expectativa de perda em relação a $Pr(\vec{x}, y)$.

$$R(h) = \int L(h(\vec{x}), y) dPr(\vec{x}, y)$$

Como a medida de performance depende da distribuição desconhecida $Pr(\vec{x}, y)$, ela não pode ser calculada diretamente. A única informação que temos é a da amostra de treinamento S. Usando essa amostra, o algoritmo de aprendizagem A procura uma regra de classificação

$h_A = A(S)$ que minimiza o risco.

Existem diversos tipos de classificação que podem ser tratados em um problema de aprendizado. O mais simples é a classificação binária, em que existem exatamente duas classes. A notação mais conveniente assume os valores $+1$ e -1, ou seja, $y \in \{-1, +1\}$. A função de perda mais comum é a perda 0/1, dada por:

$$L_{0/1}(h(\vec{x}), y) = \begin{cases} 0 & h(\vec{x}) = y \\ 1 & \text{caso contrário} \end{cases}$$

A medida de performance associada a essa função perda é a taxa de erro $Err(h)$, definida como a probabilidade de ocorrer uma previsão falsa em um exemplo escolhido aleatoriamente de acordo com $Pr(\vec{x}, y)$.

$$Err(h) = Pr(h(\vec{x}) \neq y | h) = \int L_{0/1}(h(\vec{x}), y) dPr(\vec{x}, y)$$

A taxa de erro trata todos os tipos de erro igualmente. Para dar maior importância a um determinado tipo de erro, podem ser usados fatores de custo:

$$L_{0/1}(h(\vec{x}), y) = \begin{cases} C_{+-} & h(\vec{x}) = +1 \text{ e } y = -1 \\ C_{-+} & h(\vec{x}) = -1 \text{ e } y = +1 \\ 0 & \text{caso contrário} \end{cases}$$

Na recuperação de informação a performance costuma ser medida por precisão (proporção de documentos recuperados que são relevantes) e revocação (proporção de documentos relevantes que foram recuperados) (BAEZA-YATES; RIBEIRO-NETO, 1999). Essas medidas são mais complicadas de serem implementadas em algoritmos de aprendizado, e portanto não são usadas diretamente, mas vale notar que uma taxa de erro nula implica em precisão e revocação perfeitos, porém uma taxa de erro baixa não implica necessariamente precisão e revocação altos.

A maioria dos algoritmos de aprendizado produzem regras de classificação $h_A(\vec{x})$ que não apenas dão uma classificação binária $+1$ ou -1, mas também um número real que está relacionado à probabilidade do exemplo $\vec{x}$ estar na classe dada. Estudos de Robertson (1977) e Platt (1999) mostram que o resultado de uma classificação feita através de SVMs produz curvas de precisão e revocação ótimas, sendo comparável às estimativas de outros métodos que aproximam $Pr(y = +1|\vec{x})$ diretamente.

Algumas tarefas de aprendizado envolvem mais do que duas classes. Assim, sem perda de

generalidade, $y \in \{1,\ldots,n\}$ e ainda podemos aplicar a perda 0/1, e portanto a taxa de erro e fatores de custo podem ser utilizados como medida de precisão. Apesar de existir uma abordagem de classificação em múltiplas classes através de SVMs, desenvolvida por Weston e Watkins (WETSON; WATKINS, 1998), ela não é computacionalmente eficiente. Então, problemas de múltiplas classes são divididos em n problemas binários, gerando regras de classificação $h^{(1)},\ldots,h^{(n)}$. Para classificar um novo exemplo $\vec{x}$, o resultado de cada $h^{(i)}(\vec{x})$ como uma estimativa de $Pr(y=i|\vec{x})$ é analisado, e é escolhida a classe em que $h^{(i)}(\vec{x})$ é maior, o que é justificado pela regra de Bayes, que afirma que a taxa de risco é minimizada quando o algoritmo destina cada exemplo à classe i em que $Pr(y=i|\vec{x})$ é maior. Portanto, se o algoritmo de aprendizado binário dá uma boa estimativa de $Pr(y=i|\vec{x})$, a taxa de erro resultante estará próxima da ótima. Existem alternativas para reduzir o número de problemas de classificação binário ao tratar problemas de múltiplas classes, como a classificação por pares, que resulta em $l(l-1)/2$ problemas, mas seu uso é menos freqüente.

Na maioria das tarefas de classificação de textos não há uma correspondência um-para-um entre classes e documentos. Para um número fixado de n categorias, cada documento pode estar em múltiplas, apenas uma, ou nenhuma categoria. Este problema pode ser modelado usando um rótulo de classe multivariado na forma de um vetor binário n-dimensional, isto é, $\vec{y} \in \{+1,-1\}^n$. Cada componente individual indica se o documento está ou não naquela categoria. Tratar esse y multivariado e contar os erros neste tipo de problema não é uma tarefa trivial, o que motiva a usar a mesma abordagem de dividir o problema em um conjunto de tarefas de classificação binárias, em que cada uma define se um documento deve ser destinado à certa categoria ou não. Uma categoria i é destinada a um documento $\vec{x}$ se a regra de classificação correspondente $h^{(1)}(\vec{x})$ resulta $+1$. Novamente, pela regra de Bayes, se assumirmos independência entre entre as categorias dado um documento, minimizar a taxa de erro em cada tarefa binária leva a um risco mínimo.

2.2 Representação do texto

A representação dos documentos através dos vetores $\vec{x}$ influencia o quanto o algoritmo de aprendizado consegue generalizar. Em geral, textos que já estão numa forma legível por computadores ainda não estão numa forma apropriada para o algoritmo de aprendizado, e portanto precisam ser transformados numa representação que sirva tanto para o algoritmo de aprendizado como para a tarefa de classificação.

Um problema fundamental ao lidar com linguagem natural é que o contexto tem uma in-

fluência substancial no significado de uma parte de um texto. Por exemplo, uma mesma palavra pode ter significados diferentes em sentenças diferentes (ex.: “banco”: instituição financeira ou mobília para sentar). Até mesmo uma sentença inteira pode ter significados diferentes, dependendo de quem está falando, da audiência e da situação. Diferentes abordagens de representação de texto para classificação podem considerar mais ou menos essas diferenças. As abordagens podem ser classificadas de acordo com o nível em que elas analisam o texto:

1. Sub-palavra: decomposição das palavras e sua morfologia
2. Palavra: palavras e informação léxica
3. Multi-palavras: frases e informação sintática
4. Semântico: significado do texto
5. Pragmático: significado do texto em relação ao contexto e situação (ex.: estrutura de diálogo)

Os blocos básicos em cada nível serão chamados de *termos de indexação*. Dessa forma, no nível da palavra, os termos de indexação referem-se a palavras, enquando no nível de multi-palavras, os termos de indexação podem ser frases ou sentenças inteiras.

Apesar dos benefícios para a lingüística computacional em estruturar o processamento de linguagem natural nessas categorias, elas não podem ser tratadas de forma independente, pois em cada nível existem ambigüidades que só podem ser resolvidas usando o próximo nível maior. Por exemplo, para identificar se uma palavra é um substantivo ou um verbo quando ambos assumem a mesma forma é necessário subir ao nível multi-palavras e verificar a informação sintática da frase em que a palavra se encontra.

De forma geral, quanto maior o nível, é possível capturar mais detalhes sobre o texto, mas também é maior a complexidade para produzir as representações automaticamente. O nível mais comum de representação de texto para tarefas de classificação é o da palavra, pois na maioria dos casos essas são unidades significativas de pouca ambigüidade, mesmo sem considerar o contexto, pois apesar de existirem palavras homógrafas, assume-se que elas têm pouco impacto na representação do documento como um todo. A principal vantagem desse nível é a simplicidade de implementação de um algoritmo que decompõe um texto em palavras, bastando, para a maioria das línguas, apenas separar nos espaços em branco e retirar a pontuação .

Em geral, assume-se que a ordem das palavras é irrelevante (ou de menor importância), e dessa forma apenas a freqüência da palavra em um documento é armazenada, enquanto toda

a estrutura do documento é ignorada. Essa representação costuma ser chamada de abordagem *bag-of-words*.

Os algoritmos de aprendizado computacional em geral precisam que cada exemplo esteja descrito em um vetor de dimensão fixa. Cada componente do vetor é o valor de um atributo do exemplo. Normalmente, cada palavra p é tratada como um desses atributos. O valor de um atributo para um documento d pode ser, por exemplo, o número de vezes que esse atributo ocorre no documento. Essa quantidade é chamada de *freqüência do termo* $TF(p,d)$ da palavra p no documento d. Apesar dessa abordagem acarretar em perda de informação sobre o documento, representações mais sofisticadas ainda não mostraram melhorias substanciais e consistentes. Pesquisas como a de Lewis (LEWIS, 1992) concluem que essa representação é a melhor para tarefas de recuperação de informação em geral, pois enquanto representações mais expressivas podem capturar mais do significado do documento, o aumento de sua complexidade piora a qualidade de modelos estatísticos que as utilizam.

A escolha do nível das palavras também é explicado por Whorf (WHORF, 1959), que afirma que as palavras são os elementos da linguagem em que a sintática e a semântica se encontram, pois são blocos sintáticos básicos que carregam seu próprio significado. O vocabulário de uma língua está em constante desenvolvimento e, intuitivamente, a composição e uso das palavras está em permanente otimização, de forma a codificarem uma informação ótima relativamente a tarefas que usam a língua. A língua parece adaptar-se à distribuição de tarefas, em particular através da introdução de novas palavras. Portanto, o vocabulário de uma língua reflete o que é visto como importante pela sociedade. A mesma noção de importância guia as tarefas de classificação de que as pessoas precisam. Em resumo, a hipótese pode ser descrita como a seguir: "O vocabulário de uma língua reflete a distribuição à priori das tarefas de classificação de texto: tarefas de classificação de texto para as quais o vocabulário contém palavras-chave indicativas são à priori mais prováveis" (JOACHIMS, 2002). O mesmo é aplicável a linguagens técnicas de assuntos específicos. Por exemplo, para um matemático, classificar artigos científicos de cálculo de acordo com seu tópico é uma tarefa de classificação razoável e a língua portuguesa possui palavras para diferenciar entre diversos tópicos. Logo, a abordagem *bag-of-words* provavelmente será bem sucedida. Porém, se usarmos a mesma abordagem para a língua aramaica, que não provê palavras específicas para tais assuntos e dessa forma requer construções mais complicadas, essa abordagem provavelmente não traria bons resultados. Mas ao mesmo tempo, é improvável que alguém que viveu na época em que o aramaico era utilizado teria necessidade desse tipo de classificação de textos.

2.3 Seleção de características

A seleção de características é uma etapa de pré-processamento da representação dos textos que tem como objetivo eliminar atributos irrelevantes ou inapropriados. Uma das principais vantagens desse processo é reduzir o risco de *overfitting*. Outra motivação é diminuir o número de dimensões do espaço de características em que será executado o algoritmo de aprendizado, o que pode aumentar a eficiência computacional em tempo e/ou espaço.

Existem duas abordagems básicas para fazer seleção de características: seleção de subconjuntos de características e construção de características. No primeiro caso, a nova representação consiste em um subconjunto dos atributos originais, enquanto no segundo, novas características são introduzidas através da combinação de características originais.

2.3.1 Seleção de subconjuntos de características

A aplicação mais comum desta abordagem é o de eliminação de *stopwords*, que são palavras que ocorrem muito no texto mas contêm pouco significado, tais como artigos e conjunções, que são irrelevantes para a tarefa de classificação. Outra abordagem é a chamada *document frequency thresholding*, que elimina todas as palavras que aparecem menos do que m vezes no documento, reduzindo dramaticamente o número de características mesmo para valores pequenos de m. Esta abordagem está baseada na conjuntura de Apté e Damerau (1994), que afirma que estimativas de parâmetros para termos de baixa freqüência não são confiáveis o suficiente para contribuir com informação útil.

Outras abordagens mais avançadas analisam a classificação dos dados de treinamento para eliminar atributos irrelevantes. Dessas, a mais usada é a de *ganho de informação*, apresentada por Cover e Thomas (1991). Ela mede a redução de entropia ao considerar duas variáveis aleatórias Y e W juntas ao invés de individualmente.

$$I(Y,W) = H(Y) - H(Y|W) = \sum_{y\in\{-1,+1\}} \sum_{w\in\{0,1\}} Pr(y,w) \frac{Pr(y,w)}{Pr(y)Pr(w)}$$

Neste caso, Y indica a classe atribuída a um documento e W se uma palavra em particular aparece no documento. A entropia $H(X)$ mede o grau de incerteza da variável aleatória X como o número esperado de bits necessários para codificar X. $I(Y,W)$ descreve a informação que a palavra W contribui para a codificação do rótulo de classe Y independentemente de outras palavras do documento. As probabilidades podem ser estimadas a partir da amostra de treinamento através de estimadores de máxima verossimilhança. As características selecionadas são os ter-

mos com o maior ganho de informação empírico. Outras abordagens deste tipo são a razão de chances e testes χ^2.

2.3.2 Construção de características

Através destes métodos a redução do número de dimensões é realizada introduzindo novas características que representam a maior parte da informação original ao mesmo tempo que minimizam o número de atributos. As técnicas mais usadas são:

- *stemming*: faz uma análise morfológica da palavra e armazena apenas o prefixo, assumindo que diferentes palavras que possuem o mesmo prefixo são equivalentes em relação à tarefa de classificação. Por exemplo, "computação", "computabilidade" e "computador" são projetadas no atributo "comput";

- *tesauros*: semelhante ao anterior, mas usando uma abordagem semântica. Um tesauro contém informações de diversos tipos de relações entre palavras, tais como sinônimos, que são agrupados em classes de equivalência, e relações do tipo "mais geral" ou "mais específico";

- *indexação semântica latente*: é uma forma especial de análise de componente principal linear aplicada a textos, que produz um mapeamento dos vetores de características em um sub-espaço de menor dimensão usando decomposição de valor singular. É calculada uma transformação ortogonal do sistema de coordenadas, em que os novos valores correspondem às novas características, tentando fazer palavras relacionadas serem agrupadas no mesmo componente principal;

- *clusterização de termos*: termos semânticamente similares são agrupados em um cluster, que torna-se uma nova característica. Os clusters são gerados através de algortimos de aprendizado não-supervisionados que usam meta-atributos da palavra. Esses meta-atributos costumam ser os próprios documentos que contêm a palavra, esperando-se que a distância entre os vetores desses meta-atributos reflita similaridade semântica através de padrões de co-ocorrência.

2.4 Ponderação de termos

A ponderação de termos ajusta a influência relativa de atributos e geralmente consiste em três componentes (SALTON; BUCKLEY, 1988):

- *componente de documento*: captura estatísticas sobre um termo em particular em um documento em particular. Normalmente, é medido através da *freqüência de termo* $FT(p_i, d_j)$, definida como o número de vezes em que a palavra p_i ocorre no documento d_j;

- *componente de coleção*: usada para atribuir peso menor aos termos que ocorrem em muitos documentos. Sua estatística básica é a *freqüência de documento* $FD(p_i)$, isto é, o número de documentos em que a palavra p_i ocorre pelo menos uma vez. Se a freqüência de documento é alta, o peso do termo é reduzido;

- *componente de normalização*: como documentos podem ter tamanhos diferentes, é feito um ajuste do peso de forma a possibilitar que documentos pequenos e grandes possam ser comparados na mesma escala.

2.5 Propriedades das tarefas de classificação de textos

As propriedades a seguir servem de motivação para o modelo de aprendizado que será desenvolvido adiante.

- *alta dimensão do espaço de características*: Se cada palavra que ocorre nos documentos de treinamento for usada como uma característica, problemas de classificação com alguns milhares de exemplos podem levar a 10.000 ou mais dimensões. Por exemplo, na coleção de dados da Reuters, que contém 9.603 documentos, ocorrem 27.658 palavras distintas. A Lei de Heaps (1978) afirma que o número de palavras distintas V é relacionado ao número total de palavras s de um documento por $V = ks^{\beta}$, onde k e β dependem do tipo de documento e s é suficientemente grande. Em geral, k varia entre 10 e 100 e β entre 0,4 e 0,6 (BAEZA-YATES; RIBEIRO-NETO, 1999). Tratando uma coleção de documentos como sua concatenação em um grande corpo de texto ajuda a analisar o número de dimensões em uma tarefa de classificação. Por exemplo, para $k = 15$ e $\beta = 0,5$ em uma coleção de 10.000 documentos que contêm em média 50 palavras, teríamos aproximadamente 35.000 dimensões. Esse cálculo reflete os resultados de observações experimentais.

- *vetores esparsos*: Apesar de haver um número grande de características, cada documento contém um número reduzido de palavras distintas. Os documentos da base Reuters contém em média 152 palavras, sendo 74 distintas. Considerando o número total de palavras dessa base, observa-se que os vetores de documento são bastante esparsos.

- *uso de termos heterogêneos*: no caso de categorização de textos há muitas características relevantes, ou seja, não existe um conjunto pequeno de palavras que seja suficiente para descrever todos os documentos em relação a uma tarefa de classificação. Portanto, ocorre muita perda de informação caso sejam utilizadas técnicas de seleção de características mais agressivas. Porém, vale a relação mais fraca de "semelhança em família", que afirma que apesar de não existir um grupo definido de características comuns a todos os membros, documentos relacionados partilham algumas palavras-chave. Também é importante lembrar que a linguagem natural permite a expressão de um mesmo conteúdo de diversas formas, como por exemplo no caso de sinônimos.

- *alto nível de redundância*: a maioria dos documentos contêm mais de uma palavra que indica a sua classe. Mesmo removendo as melhores características, as palavras restantes, na maioria das vezes, ainda retêm bastante informação para descrever o documento. Isso significa que vetores de documento são redundantes com respeito à tarefa de classificação. Muitas palavras têm uma distribuição similar em relação à tarefa de aprendizagem e podem ser tratadas como sinônimos para o propósito de classificação (JOACHIMS, 2002).

- *distribuição de freqüência de palavras*: a freqüência de ocorrência de palavras em linguagem natural se comporta de maneira estável. A Lei de Zipf (1949) afirma que se as palavras forem ordenadas pela freqüência dos termos, a n-ésima palavra mais freqüente ocorre $\frac{1}{n}$ vezes a freqüência das palavras mais freqüente. Isso significa que existe um pequeno número de palavras que aparecem com muita freqüência, enquanto a maioria das palavras ocorrem raramente. Estudos experimentais mais recentes indicam que as distribuições Mandelbrot aproximam melhor a distribuição de palavras, o que implica na seguinte fórmula que é chamada de Lei de Zipf generalizada:

$$FT_i = \frac{c}{(k+r)^{\phi}}$$

3 *Medidas de Performance*

Nesta seção serão discutidas algumas medidas de performance usadas na classificação de textos, usando a notação dada na tabela de contingência abaixo:

	rótulo $y = +1$	rótulo $y = -1$
previsão $h(\vec{x} = +1)$	f_{++}	f_{+-}
previsão $h(\vec{x} = -1)$	f_{-+}	f_{--}

As células da diagonal principal contam quantas vezes a previsão está correta, enquanto a diagonal secundária mostra a freqüência de erros de previsão.

3.1 Taxa de erro e custo assimétrico

A taxa de erro é definida como a probabilidade da regra de classificação h prever a classe errada, e é calculada através da fórmula:

$$Err_{teste}(h) = \frac{f_{+-} + f_{-+}}{f_{++} + f_{+-} + f_{-+} + f_{--}}$$

Porém, como na classificação de textos, em geral, o número de exemplos negativos é muito maior do que os positivos, um classificador que sempre retorna $h(\vec{x}) = -1$ tem uma taxa de erro baixa, o que mostra que essa não é uma medida de performance significativa. Para a maioria das aplicações, prever um exemplo positivo corretamente é mais importante do que um negativo, e é possível incorporar isto na medida de performance usando uma matriz de custo ou utilidade. Os elementos da matriz custo são multiplicados pelas entradas correspondentes da tabela de contingência formando uma função custo linear.

3.2 Precisão e revocação

As medidas de precisão e revocação fornecem resultados cuja interpretação é mais intuitiva.

A precisão de uma regra de classificação h é a probabilidade de que um documento classificado como $h(\vec{x}) = 1$ está de fato classificado corretamente, ou seja, $Prec(h) = Pr(y = 1|h(\vec{x}) = 1, h)$. Sua fórmula de cálculo é:

$$Prec_{teste}(h) = \frac{f_{++}}{f_{++} + f_{+-}}$$

Já a revocação é definida como a probabilidade de um documento com rótulo $y = 1$ ser classificado corretamente, ou seja, $Rev(h) = Pr(h(\vec{x}) = 1|y = 1, h)$, e é calculado por

$$Rev_{teste}(h) = \frac{f_{++}}{f_{++} + f_{-+}}$$

3.3 Medida F_β

A medida F_β é uma média harmônica ponderada entre precisão e revocação:

$$F_\beta(h) = \frac{(1+\beta^2)Prec(h)Rev(h)}{\beta^2 Prec(h) + Rev(h)}$$

Se o parâmetro $\beta = 1$, é dado peso igual para precisão e revocação.

A partir da tabela de contingência, a medida F_β é calculada por:

$$F_\beta(h) = \frac{(1+\beta^2)f_{++}}{(1+\beta^2)f_{++} + f_{+-} + \beta^2 f_{-+}}$$

3.4 Média micro e macro

Muitas vezes é útil calcular a performance média de um algoritmo de aprendizado sobre múltiplos conjuntos de treinamento e teste ou múltiplas tarefas de classificação, como por exemplo no caso em que há várias classes e há interesse em avaliar o resultado geral em todas as classes e não apenas uma. Assim, é procurada a média do resultado de m tarefas binárias para obter um único valor que indique a performance.

A *média macro* consiste no cálculo da média aritimética das medidas de performance de cada um dos m experimentos. Por exemplo, no caso da medida F_1, temos

$$F_1^{macro} = \frac{1}{m}\sum_{i=1}^{m} F_1(h_i)$$

Já a *média micro* faz a média de cada célula da tabela de contingência, obtendo uma tabela

de contingência média com elementos f_{++}^{med},f_{+-}^{med}, f_{-+}^{med} e f_{--}^{med}. No caso da medida F_1 isso implica

$$F_1^{micro} = \frac{2f_{++}^{med}}{2f_{++}^{med} + f_{+-}^{med} + f_{-+}^{med}}$$

4 *Máquinas de Suporte Vetorial*

As Máquinas de Suporte Vetorial (*Support Vector Machines*) são uma técnica de aprendizagem computacional que tem mostrado desempenho superior para algumas tarefas, tais como categorização de textos, processamento de imagens e bioinformática.

As principais vantagens das Máquinas de Suporte Vetorial, segundo Smola, Schölkopf e Müller (1999), são:

- *Boa capacidade de generalização*: a capacidade de generalização de um classificador é medida por sua eficiência na classificação de dados que não pertençam ao conjunto utilizado em seu treinamento. Quando o preditor se torna muito especializado no conjunto de treinamento, chamamos de *overfitting*. Como será visto adiante, a teoria de aprendizado estatístico utilizada por SVMs, apresentada por Vapnik e Chernovenkis, apresenta limites estatísticos para o erro de classificação na população de dados;
- *Robustez em grandes dimensões*: diferentemente das técnicas para gerar classificadores mais comums, as SVMs não causam *overfitting* quando usadas em objetos de grandes dimensões;
- *Convexidade da função objetivo*: para encontrar a solução ótima do classificador é usada uma função quadrática, em que não há presença de vários míminos locais, e sim apenas um mínimo global, o que permite encontrar com mais facilidade o valor ótimo;
- *Teoria bem definida*: a técnica das SVMs está bem fundamentada em teorias da matemática e estatística.

4.1 Limites do Risco Funcional

A teoria de aprendizado estatístico visa estabelecer condições matemáticas que permitam a escolha de um classificador f, dentro do conjunto F de todos os classificadores possíveis para aquele conjunto de treinamento S, que seja capaz de classificar dados daquele tipo da forma

mais correta possível. Supõe-se que os dados do domínio em que irá atuar o classificador são independentes e identicamente distribuídos de acordo com uma distribuição de probabilidade P.

É comum que a escolha do classificador seja feita de forma a minimizar o erro de classificação apenas dentro do conjunto de treinamento, ou seja, o risco empírico. O desempenho de generalização de um classificador pode ser medido por seu risco funcional, que é definido como a probabilidade de que f cometa erro na classificação de um novo exemplo gerado segundo P.

Seja S um conjunto de treinamento em que cada exemplo $\vec{x}_i$ pertence ao espaço $\Re^m$ e os rótulos correspondentes y_i assumem valores -1 ou $+1$. A partir de um processo de indução, o objetivo é encontrar uma função $g : \Re^m \rightarrow \{-1,+1\}$ capaz de predizer a classe de novos pontos $(\vec{x},y)$ de forma precisa. Para realizar essa tarefa é usada uma função sinal composta com uma função $f(\vec{x})$ que define uma fronteira de separação entre os dados.

Os limites no risco funcional para funções sinal relacionam o número de exemplos de treinamento, o risco empírico obtido neste conjunto e a complexidade do espaço de hipóteses, que é medida através do conceito de dimensão de Vapnik-Chervonenkis.

4.1.1 Dimensão VC

Dado um conjunto de funções sinal G, sua dimensão VC é definida como o tamanho do maior conjunto de pontos que pode ser particionado arbitrariamente pelas funções contidas em G (SMOLA; SCHÖLKOPF; MÜLLER, 1999).

Seja $\Delta_G(S)$ o número de dicotomias que o algoritmo de aprendizado tem capacidade de induzir sobre S. Diz-se que S é fragmentado por G se $\Delta_G(S) = 2^{|S|}$, onde $|.|$ representa a cardinalidade de um conjunto. A dimensão VC de um conjunto de dicotomias G é então definida como a cardinalidade do maior conjunto S que é fragmentado por G, ou seja, o maior N tal que $\Delta_G(S) = 2^N$ em que $N = |S|$.

4.1.2 Limites baseados na dimensão VC

Teorema 1 (Limite Superior) *Seja G um conjunto de funções de decisão mapeando $\Re^m$ a $\{-1,+1\}$ com dimensão VC h. Para qualquer distribuição de probabilidade P em $\Re^m \times \{-1,+1\}$, com probabilidade de ao menos $1-\delta$ sobre n exemplos e para qualquer hipótese g*

em G o risco funcional é limitado por

$$R(g) \leq R_{emp}(g) + \sqrt{\frac{c}{n}\left(h + ln\left(\frac{1}{\delta}\right)\right)}$$

em que c é uma constante universal. Se $\widehat{g} \in G$ *minimiza o risco empírico, então com probabilidade* $1-\delta$

$$R(\widehat{g}) \leq \inf_{g' \in G} R_{emp}(g) + \sqrt{\frac{c}{n}\left(h + ln\left(\frac{1}{\delta}\right)\right)}$$

Desse teorema, observa-se que quanto menor a dimensão VC de uma função, maior sua capacidade de generalização.

Como os limites apresentados dizem respeito a uma clase de funções G, e não simplesmente a escolhas de funções particulares g, introduz-se uma estrutura em G e realiza-se a minimização dos limites sobre essa estrutura. Este princípio é denominado Minimização do Risco Estrutural (SMOLA; SCHÖLKOPF, 2002).

4.2 Conceito de margem

A margem de um classificador é definida como a menor distância entre os exemplos do conjunto de treinamento e o hiperplano utilizado na separação desses dados em classes.

Teorema 2 *Seja* $X_0 \subset \Re^m$ *o conjunto de entradas com norma menor que* $R > 0$ *(*$\| \vec{x}_i \| \leq R$*, para todo* $\vec{x}_i \in X_0$*) e* F *o conjunto de funções lineares definidas em* X_0 *e satisfazendo* $\| f(\vec{x}) \| \geq \rho$*, em que* ρ *é a margem do classificador*

$$F = \{\vec{x} \rightarrow \vec{w} \cdot \vec{x} \mid \ \| \vec{w} \| \leq 1, \vec{x} \in X_0\}$$

Considerando G o conjunto de funções sinal obtidas a partir de $G = sgn(F)$ *e h a dimensão VC de G, tem-se o resultado*

$$h \leq \left\{\frac{R^2}{\rho^2}, m\right\} + 1$$

Portanto, a dimensão VC de um conjunto pode ser ainda menor ao considerarmos a margem ρ. Neste teorema também observa-se que quanto maior a margem de um classificador, menor sua dimensão VC.

Teorema 3 *Definindo a margem* ρ *de um classificador* f *como*

$$\rho = min_i y_i f(\vec{x}_i),$$

seja o erro marginal de f ($R_\rho(f)$) a proporção de exemplos de treinamento que tem margem menor que ρ.

$$R_\rho(f) = \frac{1}{n}\sum_{i=1}^{n} |y_i f(\vec{x}_i) < \rho|$$

Seja G o conjunto de funções $g(x) = sgn(f(\vec{x})) = sgn(\vec{w}\cdot\vec{x})$ com $\| \vec{w} \| \leq \Lambda$ e $\| \vec{x} \| \leq R$, para algum R, $\Lambda > 0$. Seja $\rho > 0$. Para todas distribuições P gerando os dados, com probabilidade de ao menos $1-\delta$ sobre n exemplos, e para qualquer $\rho > 0$ e $\delta \in (0,1)$, a probabilidade de um ponto de teste amostrado independentemente segundo P ser classificado incorretamente é limitado superiormente por

$$R_\rho(g) + \sqrt{\frac{c}{n}\left(\frac{R^2\Lambda^2}{\rho^2}\ln^2 n + \ln\left(\frac{1}{\rho}\right)\right)}$$

em que c é uma constante universal.

Por esse teorema, nota-se que fixando R e Λ, o termo de maior importância torna-se ρ. Deve-se buscar, portanto, o hiperplano que tenha margem ρ alta e cometa poucos erros marginais, minimizando-se assim o erro sobre os dados de teste e de treinamento, respectivamente. O hiperplano ótimo, que procura maximizar a margem de separação entre os dados, também possui duas propriedades interessantes: robustez em relação aos padrões e robustez em relação aos parâmetros (SMOLA; SCHÖLKOPF, 2002), que ditam, respectivamente, que uma pequena perturbação nos dados longe da borda e dos parâmetros de f não afetam a classificação dos dados.

4.3 SVMs lineares com margens rígidas

Quando o conjunto de treinamento é linearmente separável, isto é, é possível separar os padrões das classes diferentes por pelo menos um hiperplano, podemos utilizar os denominados SVMs de margens rígidas.

Um classificador linear pode ser definido pela equação $\vec{w}\cdot\vec{x}+b=0$, onde $\vec{w}\cdot\vec{x}$ é o produto escalar entre os vetores $\vec{w}$ e $\vec{x}$, $\vec{w}$ é o vetor normal ao hiperplano e b é um termo compensador. O par $(\vec{w},b)$ é determinado durante o treinamento do classificador. A divisão que este hiperplano faz no espaço de entradas leva à função:

$$\begin{cases} y_i = +1 & \text{se} \quad \vec{w}\cdot\vec{x}_i + b > 0 \\ y_i = -1 & \text{se} \quad \vec{w}\cdot\vec{x}_i + b < 0 \end{cases}$$

Uma função sinal $g(\vec{x}) = sgn(f(\vec{x})) = sgn(\vec{w} \cdot \vec{x}_i + b)$ pode ser então aplicada sobre essa função, levando à classificação $+1$ se $f(\vec{x}) > 0$ e -1 se $f(\vec{x}) < 0$. Logo, um conjunto de treinamento é linearmente separável se é possível determinar pelo menos um par $(\vec{w}, b)$ tal que a função $g(\vec{x})$ consiga classificar corretamente todos os exemplos contidos neste grupo.

As condições utilizadas ao escolher um classificador para aplicação da Teoria de Aprendizado Estatístico são as discutidas anteriormente, de menor risco empírico e que também pertença a uma família F com dimensão VC pequena. No caso de conjuntos linearmente separáveis, o risco empírico é zero para pelo menos um par $(\vec{w}, b)$ definido anteriormente. Já em relação a dimensão VC, utiliza-se o resultado que relaciona o risco funcional de uma função, a margem ρ de separação entre os dados de treinamento e o hiperplano separador. Smola, Schölkopf e Müller (1999) fornecem a seguinte definição de margem:

Definição 1 *Seja f uma hipótese utilizada para classificação de entradas na forma $(\vec{x}_i, y_i)$ em que y_i representa a classe do padrão $\vec{x}_i$. Então a equação a seguir define a margem com a qual o padrão $\vec{x}_i$ é classificado. A margem γ de um classificador é dada por:*

$$\rho_f(\vec{x}_i, y_i) = y f(\vec{x}_i)$$

$$\rho = min(y_i f(\vec{x}_i))$$

Portanto, entre os classificadores que minimizam o risco empírico, deve-se escolher aquele que possui a maior margem ρ'. O hiperplano correspondente a essa margem é o hiperplano ótimo.

Para obter o hiperplano ótimo, no caso de um conjunto de treinamento linearmente separável, o primeiro passo é obter a representação canônica do hiperplano, que é feita reescalando $\vec{w}$ e b de forma que os pontos mais próximos do hiperplano separador satisfaçam $|\vec{w} \cdot \vec{x}_i + b| = 1$. Dessa forma não há pontos entre $\vec{w} \cdot \vec{x}_i + b = 0$ e $\vec{w} \cdot \vec{x}_i + b = \pm 1$. Deste fato vem o nome SVMs de margens rígidas, já que ρ é sempre maior que a distância entre os hiperplanos $\vec{w} \cdot \vec{x}_i + b = 0$ e $|\vec{w} \cdot \vec{x}_i + b = 1|$. Após essa transformação, os classificadores lineares são caracterizados pela seguinte desigualdade:

$$\begin{cases} \vec{w} \cdot \vec{x}_i + b \geq +1 & \text{se} \quad y_i = +1 \\ \vec{w} \cdot \vec{x}_i + b \leq -1 & \text{se} \quad y_i = -1 \\ i = 1, \ldots, n \end{cases}$$

Sejam $\vec{x_1}$ e $\vec{x_2}$ pontos sobre as retas $\vec{w} \cdot \vec{x} + b = -1$ e $\vec{w} \cdot \vec{x} + b = +1$, respectivamente, tal que uma reta perpendicular a $\vec{w} \cdot \vec{x}_i + b = 0$ intercepte ambos os pontos. Então temos que:

$$\begin{cases} \vec{w}\cdot\vec{x_1}+b=-1 \\ \vec{w}\cdot\vec{x_2}+b=+1 \end{cases} \Longrightarrow \vec{w}\cdot(\vec{x_2}-\vec{x_1})=2$$

E pela ortogonalidade entre o hiperplano separador e $\vec{w}$ e $\vec{x_2}-\vec{x_1}$, temos que esses vetores são paralelos entre si, o que nos dá a equação:

$$|\vec{w}\cdot(\vec{x_2}-\vec{x_1})| = \| \vec{w} \| \times \| \vec{x_2}-\vec{x_1} \|$$

E substituindo na equação anterior obtemos:

$$\| \vec{x_2}-\vec{x_1} \| = \frac{2}{\| \vec{w} \|}$$

que nos dá a distância entre os hiperplanos $\vec{w}\cdot\vec{x_1}+b=-1$ e $\vec{w}\cdot\vec{x_2}+b=+1$. De forma análoga, temos que a distância entre os hiperplanos $\vec{w}\cdot\vec{x}+b=0$ e $\vec{w}\cdot\vec{x}+b=1$ ou $\vec{w}\cdot\vec{x}+b=-1$ é dada por $\frac{1}{\|w\|}$.

Como a margem é sempre maior que essa distância, a minimização de $\| w \|$ leva à maximização da margem. O vetor peso $\vec{w}$ e a constante b que resolvem o problema de otimização abaixo descrevem o hiperplano de margem máxima.

$$\begin{aligned} \textit{minimizar}: &\quad \| \vec{w} \|^2 \\ \textit{sujeito a}: &\quad y_i(\vec{w}\cdot\vec{x_i}) \geq 1 \quad \text{para } i=1,\ldots,n \end{aligned}$$

Este problema de otimização quadrática é resolvido com o auxílio de uma função Lagrangiana:

$$L(\vec{w},b,\alpha) = \frac{1}{2} \| \vec{w} \|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\vec{w}\cdot\vec{x_i}+b)-1)$$

em que os α_i são denomindados *multiplicadores de Lagrange*. Nessa forma o problema torna-se de minimização da função acima em relação a $\vec{w}$ e b e maximização dos α_i. Os pontos de mínimo são obtidos igualando a zero as derivadas da função em relação a $\vec{x}$ e b, o que leva às equações:

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i$$

E substituindo estas equações na função Lagrangiana é obtido o seguinte problema dual de otimização:

$$\begin{aligned} maximizar: \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \\ sujeito\ a: \quad & \begin{cases} \alpha_i \geq 0,\ i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \end{aligned}$$

Dessa forma, temos o seguinte algoritmo para determinação do hiperplano ótimo para conjuntos linearmente separáveis (VERT, 2001):

1. Para cada conjunto de treinamento linearmente separável $S = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$
2. Seja $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*)$ a solução do seguinte problema de otimização com restrições:
3. Maximixar $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$
4. Sob as restrições: $\begin{cases} \alpha_i \geq 0,\ i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$
5. O par $(\vec{w}^*, b^*)$ apresentado a seguir define o hiperplano ótimo
6. $\vec{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \vec{x}_i$
7. $b^* = -\frac{1}{2}[\max_{i|y_i=-1} (\vec{w}^* \cdot \vec{x}_i) + \min_{i|y_i=+1} (\vec{w}^* \cdot \vec{x}_i)]$

A solução é esparsa, pois, conforme demonstrado por Cristianini e Shawe-Taylor (2000), α_i^* assume valores positivos para exemplos de treinamento que estão a uma distância do hiperplano ótimo exatamente igual à margem (chamados vetores de suporte) e zero para todos os outros. Conseqüentemente, o hiperplano ótimo é determinado unicamente pelos vetores de suporte.

4.4 SVMs lineares com margens suaves

Em casos em que a natureza do problema não é linear ou há muito ruído nos dados, em geral, os conjuntos não são linearmente separáveis. Então, faz-se necessária a suavização das margens, admitindo alguns erros de classificação dos exemplos.

A adaptação das SVMs lineares com margens rígidas é feita através de variáveis de relaxamento ξ, que suavizam as restrições na determinação do hiperplano ótimo, permitindo a ocorrência de alguns casos de erro de classificação. Elas medem onde se encontram os exemplos $(\vec{x}_i, y_i)$ em relação aos hiperplanos $\vec{w} \cdot \vec{x} + b = \pm 1$ nos casos em que a classificação está incorreta, e são definidas pelas seguintes equações (VERT, 2001):

$$\text{Para } y_i = +1 \quad \xi_i(\vec{w}, b) = \begin{cases} 0 & \text{se } \vec{w} \cdot \vec{x}_i \geq 1 \\ 1 - \vec{w} \cdot \vec{x}_i + b & \text{se } \vec{w} \cdot \vec{x}_i < 1 \end{cases}$$

$$\text{Para } y_i = -1 \quad \xi_i(\vec{w}, b) = \begin{cases} 0 & \text{se } \vec{w} \cdot \vec{x}_i \leq -1 \\ 1 + \vec{w} \cdot \vec{x}_i + b & \text{se } \vec{w} \cdot \vec{x}_i > -1 \end{cases}$$

Para obter o menor número possível de erros de treinamento, as variáveis de relaxamento ξ devem ter valor mínimo, e para maximizar a margem de separação entre as classes procura-se a minimização de $\| \vec{w} \|$, como no caso anterior. Esses dois valores a serem minimizados podem ser combinados na seguinte equação (CAMPBELL; KRISTIN, 2000):

$$\varepsilon(\vec{w}, b) = \| \vec{w} \|^2 + C \sum_{i=1}^{n} \xi_i(\vec{w}, b)$$

em que C é uma constante que impõe um peso diferente para o treinamento em relação à generalização e deve ser determinada empiricamente. Como as funções $\xi_i(\vec{w}, b)$ não são diferenciáveis em $\vec{w}$ e b, é usada uma variável auxiliar ζ tal que $\zeta_i \geq \xi_i(\vec{w}, b)$ e aplicando as equações que definem $\xi_i(\vec{w}, b)$ obtemos o seguinte problema de minimização (SMOLA; SCHÖLKOPF; MÜLLER, 1999):

$$\begin{aligned} \textit{minimizar}: \quad & \| \vec{w} \|^2 + C \sum_{i=1}^{n} \zeta_i \\ \textit{sujeito a}: \quad & \begin{cases} \zeta_i \geq 0 \\ y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \zeta_i \end{cases} \end{aligned}$$

Esse problema é resolvido pelo seu dual, usando os mesmos passos apresentados anteri-

ormente para SVMs de margem rígida, e aparecem as condições denominadas Karush-Kuhn-Tucker, necessárias para que o conjunto seja ótimo:

(i) $\alpha_i = 0 \Rightarrow y_i f(\vec{x}_i) \geq 1$ e $\zeta_i = 0$

(ii) $0 < \alpha_i < C \Rightarrow y_i f(\vec{x}_i) = 1$ e $\zeta_i = 0$

(iii) $\alpha_i = C \Rightarrow y_i f(\vec{x}_i) < 1$ e $\zeta_i \geq 0$

Os casos (ii) e (iii), em que os multiplicadores de Lagrange possuem valor estritamente positivo, correspondem aos vetores de suporte. Em (ii), tem-se a representação de um vetor de suporte sobre a margem, e em (iii), um vetor de suporte entre as margens. Para os demais padrões, o valor do multiplicador de Lagrange associado é nulo (i) (LORENA, 2003).

A classificação de um novo exemplo é dada por:

$$sgn(\sum_{x_i \in SV} \alpha_i^* y_i \vec{x}_i \cdot \vec{x} + b^*)$$

4.5 SVMs não lineares

A utilização de classificadores lineares é limitada, pois em muitos casos não é possível dividir satisfatoriamente os dados de treinamento através de um hiperplano. Para generalizar SVMs lineares de forma a lidar com tais situações são definidas funções reais $\Phi_1, \ldots, \Phi_M$, que podem ser não lineares, que mapeiam o conjunto de treinamento S para o *espaço de características* de forma a tornar o conjunto de treinamento linearmente separável neste espaço.

$$\vec{x}_i \ (i = 1, \ldots, n) \mapsto \vec{\Phi}(\vec{x}_i) = (\Phi_1(\vec{x}_i), \ldots, \Phi_M(\vec{x}_i))$$

$$\Rightarrow \vec{\Phi}(S) = \{(\vec{\Phi}(\vec{x}_1), y_1), \ldots, (\vec{\Phi}(\vec{x}_n), y_n)\}$$

Encontrada essa função, o algoritmo para determinação do hiperplano ótimo é o mesmo do caso anterior, substituindo as ocorrências de $\vec{x}_i$ por $\vec{\Phi}(\vec{x}_i)$.

Na aplicação de SVMs não é necessário aplicar as funções Φ diretamente, bastando saber calcular o produto interno $\vec{\Phi}(\vec{x}_i) \cdot \vec{\Phi}(\vec{x}_j)$, o que é feito através de uma função *Kernel*:

$$K(x, z) = \Phi(x) \cdot \Phi(z)$$

Segundo o Teorema de Mercer, os kerneis devem ser matrizes positivas semi-definidas para qualquer subconjunto finito de S.

A tabela a seguir apresenta os principais kerneis utilizados nas SVMs (HAYKIN, 1999):

Tipo de Kernel	**Função** $K(\vec{x}_i, \vec{x}_j)$	**Comentários**
Polinomial	$(\vec{x}_i \cdot \vec{x}_j + 1)^p$	A potência p deve ser especificada pelo usuário
Gaussiano	$e^{(-\frac{1}{2\sigma^2}\|\vec{x}_i - \vec{x}_j\|^2)}$	A ampliture σ^2 é especificada pelo usuário
Sigmoidal	$tanh(\beta_0(\vec{x}_i \cdot \vec{x}_j) + \beta_1)$	Utilizado somente para alguns valores de β_0 e β_1

4.6 SVMs incrementais

As técnicas tradicionais de SVMs requerem que seja feito um novo treinamento a partir do zero sempre que há uma alteração no conjunto de treinamento. A reutilização de resultados anteriores, proposta pela técnica de SVMs incrementais, torna os aprendizados sucessivos mais rápidos e também pode reduzir o custo de armazenamento descartando exemplos antigos.

O método iterativo proposto por Xiao, Wang e Zhang (2000) é realizado da seguinte forma: primeiro, o classificador antigo é utilizado no novo conjunto de exemplos incremental, e aqueles que forem classificados incorretamente são combinados ao conjunto de vetores de suporte atual para construir um novo conjunto de treinamento, enquanto os outros exemplos formam um novo conjunto de testes. A seguir, um novo classificador é treinado no novo conjunto de treinamento, e o novo conjunto de testes é utilizado para repetir a operação anterior. O processo continua até que todos os pontos sejam classificados corretamente.

Algumas medidas são tomadas para reduzir o custo de armazenagem e acelerar a convergência: os exemplos que nunca são selecionados como vetores de suporte são descartados gradualmente usando o esquema LRU, e exemplos que aparecem freqüentemente no conjunto de vetores de suporte são introduzidos de forma otimizada ao conjunto de treinamento. Maiores detalhes sobre o algoritmo α-ISVM de aprendizagem SVM incremental e seus resultados empíricos podem ser encontrados em (XIAO; WANG; ZHANG, 2000).

5 *Modelo de Aprendizado Estatístico para Classificação de Textos através de SVMs*

A seguir será descrito o modelo de aprendizado estatístico para classificação de textos criado por Joachims (2002), que é baseado nas propriedades vistas na seção 2.5 e faz uma conexão com a taxa de erro esperada de SVMs.

5.1 Passo 1: Limitando o erro esperado baseado na margem

A importância deste passo é a garantia de uma boa generalização, conseguida através da combinação de uma margem grande com um erro de treinamento pequeno.

A teoria de aprendizado estatístico desenvolvida por Vapnik apresentou vários limites sobre o erro esperado, em particular o do teorema a seguir, aplicável a todos os SVMs de margem suave, que conecta o erro esperado com o número de vetores de suporte:

Teorema 4 (Limite no Erro Esperado de SVMs de Margens Suaves) *O erro esperado* $\varepsilon(Err^n(h_{SVM}))$ *de um SVM de margem suave baseado em n exemplos de treinamento com* $c \leq K(\vec{x}_i,\vec{x}_j) \leq c+R^2$ *para alguma constante c é limitado por*

$$\varepsilon(Err^n(h_{SVM})) \leq \frac{\rho\varepsilon(\frac{R^2}{\delta^2}) + \rho C'\varepsilon\Big(\sum_{i=1}^{n+1}\xi_i\Big)}{n+1}$$

com $C' = CR^2$ *se* $C \geq \frac{1}{\rho R^2}$*, e* $C' = CR^2+1$ *caso contrário. Para hiperplanos imparciais* $\rho = 1$*, e para hiperplanos estáveis* $\rho = 2$*. As expectativas à direita são para conjuntos de treinamento de tamanho* $n+1$.

Esse limite mostra que as quantidades mais importantes são a margem δ, a perda de treinamento ξ e a quantidade R associada ao tamanho dos vetores de documento, que atua como uma

constante para escalar a margem δ.

5.2 Passo 2: Conceitos TCat homogêneos como um modelo de tarefas de classificação de texto

Não é possível deduzir diretamente se a margem de uma tarefa de classificação de textos será alta, pois essa propriedade só é observável a partir do momento em que os dados de treinamento são processados pela SVM. É possível mostrar que as propriedades identificadas na seção 2.5 levam a uma margem alta, o que explica porque as SVMs possuem boa performance em tarefas de classificação apesar da alta dimensionalidade.

Definição 2 (Conceitos TCat Homogêneos) *O conceito TCat*

$$TCat([p_1 : n_1 : f_1], \ldots, [p_s : n_s : f_s])$$

descreve uma tarefa de classificação binária com s conjuntos disjuntos de características. O i-ésimo conjunto inclui f_i características. Cada exemplo positivo contém p_i ocorrências de características do conjunto respectivo, e cada exemplo negativo contém n_i ocorrências. Uma mesma característica pode ocorrer múltiplas vezes em um documento.

Joachims usa um conceito TCat como uma tarefa hipotética de classificação de textos, definido como

$$\begin{aligned}\text{TCat(}\ &[20:20:100], [4:1:200], [1:4:200], [5:5:600],\\ &[9:1:3000], [1:9:3000], [10:10:4000]\ \text{)}\end{aligned}$$

e mostra como essas 11100 palavras apresentam as propriedades de espaço de entrada de alta dimensão, vetor de documento esparso, alto nível de redundância, uso heretogêneo de termos e a Lei de Zipf. O hiperplano classificador

$$h(\vec{x}) = \vec{w} \cdot \vec{x} + b = \sum_{i=1}^{11100} w_i x_i + b$$

com $b = 0$ e

$$w_i = \begin{cases} +0.23 & \text{para as 200 palavras de média freqüência indicando POS} \\ -0.23 & \text{para as 200 palavras de média freqüência indicando NEG} \\ +0.04 & \text{para as 3000 palavras de baixa freqüência indicando POS} \\ -0.04 & \text{para as 3000 palavras de baixa freqüência indicando NEG} \\ 0 & \text{para todas as outras palavras} \end{cases}$$

possui uma margem δ de pelo menos $\sqrt{1/30,15}$ para o exemplo definido por Joachims.

Em seguida, Joachims mostra como exemplos reais das coleções Reuters, WebKB e Ohsumed também podem ser modelados como conceitos TCat. Como exemplo, cria um TCat com as estatísticas médias de um documento da WebKB da categoria "course", resultando em

$$\text{TCat}(\ [77:29:98],[4:21:52],\ [16:2:431],[1:12:341],$$
$$[9:1:5045],[1:21:24276],[169:191:8116]\ )$$

5.3 Passo 3: Capacidade de aprendizagem de conceitos TCat

Este último passo conecta os conceitos TCat ao limite de generalização de uma SVM.

Lema 1 (Limite inferior da margem de conceitos TCat livres de ruído) *Para um conceito* $TCat([p_1 : n_1 : f_1],\ldots,[p_s : n_s : f_s])$*, existe sempre um hiperplano passando através da origem que tem margem* δ *limitada por*

$$\delta^2 \geq \frac{ac-b^2}{a+2b+c} \qquad com \qquad a=\sum_{i=1}^{s}\frac{p_i^2}{f_i} \quad b=\sum_{i=1}^{s}\frac{p_i n_i}{f_i} \quad c=\sum_{i=1}^{s}\frac{n_i^2}{f_i}$$

Este lema mostra que qualquer conjunto de documentos consistente com os conceitos TCat especificados é separável linearmente com uma certa margem mínima.

Lema 2 (Distância Euclidiana dos Vetores de Documento) *Se as freqüências de termos rankeadas* FT_r *em um documento com* l *termos têm a forma da Lei de Zipf generalizada*

$$TF_r = \frac{c}{(r+k)^{\phi}}$$

baseado em seu rank de freqüência r, então o quadrado da distância euclidiana do vetor de documento $\vec{x}$ *de freqüências de termos é limitado por*

$$\| \vec{x} \| \leq \sqrt{\sum_{r=1}^{d}\left(\frac{c}{(r+k)^{\phi}}\right)^2} \qquad com\ d\ tal\ que \qquad \sum_{r=1}^{d}\frac{c}{(r+k)^{\phi}} = l$$

Esse lema mostra como, devido a Lei de Zipf, a distância euclidiana é menor do que l, pois a maioria dos termos não se repete muito freqüentemente e o número de termos distintos d é

alto. Isso leva a um valor baixo de R^2 no limite na performance de generalização esperada. Combinando esses dois lemas com o teorema anterior obtemos:

Teorema 5 (Capacidade de Aprendizagem de Conceitos TCat) *Para conceitos*

$$TCat([p_1 : n_1 : f_1], \ldots, [p_s : n_s : f_s])$$

e documentos com l termos distribuídos de acordo com a Lei de Zipf generalizada

$$TF_r = \frac{c}{(r+k)^{\phi}},$$

o erro de generalização esperado de uma SVM após treinamento em n exemplos é limitado por

$$\varepsilon(Err^n(h_{SVM})) \leq \rho \frac{R^2}{n+1} \frac{ac-b^2}{a+2b+c} \qquad com \qquad \begin{aligned} a &= \sum_{i=1}^{s} \frac{p_i^2}{f_i} \\ b &= \sum_{i=1}^{s} \frac{p_i n_i}{f_i} \\ c &= \sum_{i=1}^{s} \frac{n_i^2}{f_i} \\ R^2 &= \sum_{i=1}^{s} \left(\frac{c}{(r+k)^{\phi}} \right)^2 \end{aligned}$$

a não ser que $\forall_{i=1}^{s} : p_i = n_i$. *d é escolhido tal que* $\sum_{r=1}^{d} \frac{c}{(r+k)^{\phi}} = l$. *Para SVMs imparciais* ρ *é igual a 1, e para SVMs parciais* ρ *é igual a 2.*

Esse último teorema demonstra que a categorização de textos realizada através da técnica de SVMs apresenta bons resultados em relação a erros de classificação, pois a margem esperada é alta, devido a propriedades dos conceitos TCat que servem para modelar esse tipo de problema. O modelo teórico foi comparado com resultados experimentais por Joachims. Ainda que o modelo não dê uma previsão precisa da performance exata observada em cada conjunto de testes, os resultados obtidos mostram que são comparáveis os resultados de quais tarefas de classificação são mais difíceis, como pode ser visto na tabela abaixo, e estes resultados validam que os conceitos TCat podem formalizar propriedades chave de tarefas de classificação relevantes para a capacidade de aprendizagem das SVMs.

	modelo	experimento
WebKB “course”	11,2%	4,4%
Reuters “earn”	1,5%	1,3%
Ohsumed “pathology”	94,5%	23,1%

É importante ressaltar que foram usadas algumas hipóteses para criar este modelo que não são totalmente válidas na prática. Em primeiro lugar, assumiu-se que cada documento segue à risca a Lei de Zipf generalizada, negligenciando a variância que ocorre especialmente em documentos menores. Em particular, isso implica que todos os documentos são do mesmo tamanho. Além disso, o modelo fixa o número de ocorrências de cada conjunto de palavras no modelo TCat. A violação disso é tratada como uma forma de ruído. Este fator poderia ser tratado adicionando mais parâmetros, porém isso aumentaria significativamente a complexidade do modelo.

6 *Métodos Convencionais de Classificação*

A seguir serão listados outros métodos utilizados para classificação de texto identificados por Joachims (2002). Uma comparação entre os métodos mais importantes em relação a tempo de treinamento, velocidade de classificação e acurácia na classificação pode ser encontrada em (DUMAIS et al., 1998).

6.1 Classificador Naive Bayes

Este classificador utiliza um modelo probabilístico de texto para estimar $Pr(y|d)$, que é a probabilidade de um documento d estar na classe y. No modelo *multinomial mixture*, assume-se que palavras ocorrem de forma independente dentro do documento de uma dada classe e que todos os documentos atribuídos a uma categoria são gerados de acordo com o componente do modelo associado a essa categoria.

A regra de Bayes afirma que para alcançar o melhor resultado de classificação o documento d deve ser atribuído à classe $y \in \{-1,+1\}$ com maior $Pr(y|d)$.

$$Pr(y|d) = \frac{Pr(d|y,l') \cdot Pr(y|l')}{\sum_{y' \in \{-1,+1\}} Pr(d|y',l') \cdot Pr(y'|l')}$$

onde $Pr(d|y,l')$ é a probabilidade de observar um documento d na classe y dado seu tamanho l' e $Pr(y|l')$ é a probabilidade à priori de que um documento de tamanho l' esteja na classe y.

Se assumirmos que a categoria de um documento não depende do seu tamanho, ou seja, $Pr(y|l') = Pr(y)$, podemos estimar $Pr(y)$ a partir da fração de documentos de treinamento que estão atribuídos à classe y:

$$\widehat{Pr}(y) = \frac{|y|}{\sum_{y' \in \{-1,+1\}} |y'|} = \frac{|y|}{|D|}$$

onde $|y|$ denota o número de documentos de treinamento na classe y e $|D|$ o número total de documentos.

Já $Pr(d|y,l')$ pode ser estimado usando a hipótese do modelo unigrama, que implica que a ocorrência de uma palavra só depende da classe do documento, mas que ocorre de forma independente de outras palavras do documento e não depende do tamanho do documento:

$$Pr(d|y,l') \approx \prod_{i=1}^{|d|} Pr(w_i|y)$$

onde w_i varia sobre a seqüência de palavras em um documento d que são consideradas características e $|d|$ é o número de palavras em um documento d. A estimativa de $Pr(d|y)$ é reduzida a estimar cada $Pr(w_i|y)$ independentemente. Uma estimativa Bayesiana é usada para $Pr(w_i|y)$:

$$\widehat{Pr}(w_i|y) = \frac{1 + TF(w_i,y)}{|F| + \sum_{w' \in |F|} TF(w',y)}$$

onde $TF(w,y)$ é o número total de vezes que a palavra w ocorre dentro de documentos da classe y. Esse estimador assume que a observação de cada palavra é à priori igualmente provável.

A regra de classificação resultante fica:

$$\begin{aligned} h_{BAYES}(d) &= argmax_{y \in \{-1,+1\}} \frac{Pr(y) \cdot \prod_{i=1}^{|d|} Pr(w_i|y)}{\sum_{y' \in \{-1,+1\}} Pr(y') \cdot \prod_{i=1}^{|d|} Pr(w_i|y')} \\ &= argmax_{y \in \{-1,+1\}} \frac{Pr(y) \cdot \prod_{w \in X} Pr(w|y)^{TF(w,d)}}{\sum_{y' \in \{-1,+1\}} Pr(y') \cdot \prod_{w \in X} Pr(w|y')^{TF(w,d)}} \end{aligned}$$

6.2 Algortimo Rocchio

Esse classificador é baseado no algorítimo de relevância-retroalimentação proposto por Rocchio (1971) para o modelo de recuperação no espaço de vetores.

A componente linear da regra de classificação é calculada por:

$$\vec{w} = \frac{1}{|i : y_i = +1|} \sum_{i:y_i=+1} \vec{x}_i - \beta \frac{1}{j : y_j = -1} \sum_{|j:y_j=-1|} \vec{x}_j$$

É necessário que elementos negativos do vetor w sejam substituídos por 0. β é um parâmetro que ajusta o impacto relativo de exemplos de treinamento positivos e negativos, cujo valor depende da tarefa e é essencial para uma boa performance do classificador.

Usando o cosseno do ângulo entre $\vec{w}$ e $\vec{x}$ como medida de similaridade e $\beta = 1$, Rocchio mostra que para $\vec{w}$ como calculado a similaridade média dos exemplos de treinamento positivos com $\vec{w}$ menos a similaridade dos exemplos de treinamento negativos com $\vec{w}$ é maximizada:

$$\frac{1}{|i : y_i = +1|} \sum_{i:y_i=+1} cos(\vec{w}, \vec{x}_i) - \frac{1}{|j : y_j = -1|} \sum_{j:y_j=-1} cos(\vec{w}, \vec{x}_i) \rightarrow max$$

6.3 *k*-nearest neighbours

O classificador *k-nearest neighbours* (*k*-NN) é baseado na hipótese de que exemplos localizados próximos um dos outros, de acordo com uma métrica de similaridade, provavelmente pertencem a uma mesma classe. Ele também é derivado da regra de Bayes e usa o cosseno como métrica de similaridade. $knn(\vec{x})$ denota os índices dos k documentos que possuem os maiores cossenos com o documento para classificar $\vec{x}$.

$$h_{knn}\vec{x} = sign\left(\frac{\sum_{i \in knn(\vec{x})} y_i cos(\vec{x}\vec{x}_i)}{\sum_{i \in knn(\vec{x})} y_i cos(\vec{x}\vec{x}_i)} \right)$$

6.4 Outros métodos

Outros métodos citados por Joachims (2002) bastante usados para classificação de textos são:

- *Classificador de Árvore de Decisão*: O C4.5 é o algoritmo mais popular de árvore de decisão e mostrou bons resultados em diversos problemas. Ele retorna um nível de confiança ao classificar novos exemplos, que é usado para calcular tabelas de precisão e revocação;

- *Rede Bayesiana*: Um dos problemas do classificador naive Bayes é a hipótese de independência condicional. Usando modelos de rede Bayesianas mais gerais é possível superar essa limitação e pesquisas mostraram que a construção automática de redes Bayesianas com dependência limitada pode melhorar a performance de previsão.

- *Regressão Logística*: Esta é uma forma de estimar a probabilidade $Pr(y|\vec{x})$ que usa uma abordagem discriminativa ao invés de um modelo generativo. É encontrado um hiperplano no espaço de características que maximiza a verossimilhança condicional nos dados de treinamento;

- *Redes Neurais*: Este método está relacionado a regressão logística mas utiliza modelos mais complexos do que os lineares. Como as redes neurais estão muito sujeitas a *overfitting*, é necessário fazer uma seleção de características;

- *Regressão Polinomial*: A forma geral da regressão linear, também já foi usada em trabalhos pioneiros de classificação te textos;

- *Algoritimos de Boosting*: O mais conhecido algoritmo de boosting é o AdaBoost, que combina iterativamente múltiplas hipóteses base (por exemplo árvores de decisão) usando um modelo linear. Boosting também pode ser interpretado com maximização de margem, porém ao invés de usar a norma L_2 como nos SVMs, os algoritmos de Boosting usam a norma L_1. Com uma função de perda modificada, Boosting pode ser formulado como um problema de otimização semelhante ao de SVMs;

- *Aprendizagem de Regras*: Esta abordagem foca em boas estratégias de busca e representações compactas. Um exemplo é busca genética. A vantagem é maior interpretabilidade que, por exemplo, modelos lineares;

- *Aprendizagem de Regras Relacional*: É uma representação mais poderosa que usa predicados relacionais para expressar relações entre atributos, como por exemplo a ordenação de palavras;

- *Aprendizagem Ativa*: Esta é uma modificação do modelo de aprendizagem indutiva em que são requisitados os rótulos de exemplos particulares, reduzindo o número de exemplos necessários para treinar o classificador.

7 Conclusão

A categorização de textos automática, ou seja, a designação de classes a textos em linguagem natural de acordo com seu conteúdo, é um componente importante em várias tarefas que lidam com gerenciamento de informação, tais como ordenação em tempo-real de e-mails ou arquivos em hierarquias de pastas, identificação de tópicos para dar suporte a operações de processamento de tópicos específicos, busca e/ou navegação estruturada, ou encontrar documentos que combinam com certos interesses (DUMAIS et al., 1998).

Métodos de aprendizado indutivos para construção de classificadores apresentam como principal vantagem a facilidade em criá-los, dependendo apenas de informações simples de serem obtidas, tais como exemplos para treinamento, o que possibilita também maior customização para categorias específicas de interesse para indivíduos.

A técnica de Máquinas de Suporte Vetorial está bem fundamentada em teorias estatísticas. Modelos aplicados à tarefa de classificação de textos demonstram que, apesar da alta dimensionalidade, esta técnica garante bons resultados devido a características dos textos de linguagem natural que acarretam em margem elevada ao realizar a tarefa de aprendizado.

Referências Bibliográficas

APTÉ, C.; DAMERAU, F. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, v. 12, n. 3, p. 233–251, 1994.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. [S.l.]: Addison Wesley, 1999.

BUSH, V. As we may think. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 17–34, 1988.

CAMPBELL, C.; KRISTIN, P. Support vector machines: Hype or hallelujah? *SIGKDD Explorations*, v. 2, n. 2, p. 1–13, 2000.

COVER, T.; THOMAS, J. *Elements of Information Theory*. [S.l.]: Wiley, 1991.

CRISTIANINI, N.; SHAWE-TAYLOR, J. *An Introduction to Support Vector Machines and other kernel-based learning methods*. [S.l.]: Cambridge University Press, 2000.

DUMAIS, S. et al. Inductive learning algorithms and representations for text categorization. In: *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*. New York, NY, USA: ACM Press, 1998. p. 148–155. ISBN 1-58113-061-9.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. [S.l.]: Prentice Hall, 1999.

HEAPS, H. *Information Retrieval: Computational and Theoretical Aspects*. [S.l.]: Academic Press, 1978.

HOFSTADTER, D. *Gödel, Escher, Bach: An eternal golden braid*. [S.l.]: Basic Books, 1979.

JOACHIMS, T. *Learning to Classify Text Using Support Vector Machines*. [S.l.]: Kluwer Academic Publishers, 2002.

LEÃO, L. *O Labirinto da Hipermídia*. [S.l.]: Iluminuras, 1999.

LEVY, P. *O Que É Virtual*. [S.l.]: Editora 34, 1996.

LEWIS, D. *Representation and Learning in Information Retrieval*. Tese (Doutorado) — University of Massachusetts, 1992.

LORENA, A. *Introdução às Máquinas de Vetores de Suporte (Support Vector Machines)*. [S.l.], 2003.

MURRAY, J. *Hamlet no Holodeck*. [S.l.]: Editora UNESP, 2003.

PLATT, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *Advances in Large Margin Classifiers*. [S.l.]: MIT Press, 1999.

PONTE, C. *Leitura das Notícias*. [S.l.]: Livros Horizonte, 2004.

ROBERTSON, S. The probability ranking principle in ir. *Journal of Documentation*, v. 33, n. 4, p. 294–304, 1977.

ROCCHIO, J. Relevance feedback in information retrieval. *The SMART Retrieval System: Experiments in Automatic Document Processing*, p. 313–323, 1971.

RUSSEL, S.; NORVIG, P. *Inteligência Artificial*. [S.l.]: Editora Campus, 2004.

SALTON, G.; BUCKLEY, C. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, v. 24, n. 5, p. 513–523, 1988.

SEBASTIANI, F. *Machine Learning in Automated Text Categorization*. [S.l.], 1999.

SMOLA, A.; SCHÖLKOPF, B. Support vector machines and kernel algorithms. In: *The Handbook of Brain Theory and Neural Networks*. [S.l.]: MIT Press, 2002.

SMOLA, A.; SCHÖLKOPF, B.; MÜLLER, K. Kernel principal component analysis. In: *Advances in Kernel Methods—Support Vector Learning*. [S.l.]: MIT Press, 1999. p. 327–352.

Text Categorization Using Adaptive Context Trees.

WETSON, J.; WATKINS, C. *Multi-Class Support Vector Machines*. [S.l.], 1998.

WHORF, B. *Language, Thought and Reality*. [S.l.]: Whiley, 1959.

XIAO, R.; WANG, J.; ZHANG, F. An approach to incremental svm learning algorithm. In: *Tools with Artificial Intelligence*. [S.l.: s.n.], 2000.

ZIPF, G. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. [S.l.]: Addison Wesley, 1949.