Categorização de texto utilizando SVM e Bag of Words

Caio Lima[1], Heron Sanches[1]

[1]Instituto de Matemática, Departamento de Ciência da Computação – Universidade Federal da Bahia (UFBA)
Av. Adhemar de Barros, s/n – Campus Ondina
CEP: 40170-110, Salvador – Bahia – Brasil

{caiolima, heronsanches}@dcc.ufba.br

Resumo. Este trabalho relata uma solução para classificar textos por assunto, a partir da taxa de ocorrência de palavras no texto, utilizando o extrator de características Bag of Words e o classificador SVM. Assim como uma análise dos resultados encontrados.

1. Introdução

Devido ao rápido crescimento de informações disponibilizadas na Internet, categorização de texto tem se tornado uma das técnicas mais frequentes para organização de textos.

Categorização de texto tem sido utilizada para classificar novas estórias, procurar informações interessantes na WEB e para guiar buscas de usuários dentro de um texto. O objetivo da categorização de texto é classificar documentos em categorias predefinidas. Cada documento pode está classificado em uma, múltiplas ou nenhuma categoria. Utilizando aprendizado de máquina, o objetivo é ensinar classificadores através de exemplos. [Joachims]

O primeiro passo para conseguir categorizar um texto utilizando um classificador, neste trabalho em específico, o SVM, é transformar um documento de texto, normalmente um conjunto de strings (palavras), em um formato no qual o classificador compreenda. Para este fim se utilizou o algoritmo "Bag of Words" [Scikit 2014], o qual capta as palavras (características) distintas do texto, contabiliza o quanto elas ocorrem e deixa essas informações num formato que possa ser entendido pelo SVM.

2. Explicação dos algoritmos Bag of Words e SVM

2.1 Bag of Words

Como um texto é uma sequência de caracteres (*raw data*), estes caracteres não podem ser usados diretamente no algoritmo SVM como parâmetro, é aí que o Bag of Words faz seu trabalho, converte o *raw data* num vetor de inteiros de tamanho fixo, que é o que o SVM aceita como parâmetro. O algoritmo *Bag of Words [Scikit] faz* tal conversão em três passos básicos:

1. Tokenização – para cada string distinta do texto é dado um identificador

único(token), os separadores considerados para a tokenização das strings são espaços em branco ou caracteres de pontuação.
2. Contagem – conta-se a ocorrência de um determinado token em cada documento.
3. Normalização – nesta fase as ocorrências de tokens são colocadas num vetor, estes valores podem ser ou não normalizados, no nosso caso eles não são, então o resultado é um vetor de inteiros.

Cada posição no vetor gerado é considerado como uma característica do documento analisado, este vetor é denominado de vetor de características, o qual será usado para compor o parâmetro no classificador SVM, cada vetor de característica representa um documento. O SVM, no nosso exemplo da *seção 3,* usa como parâmetro uma matriz, na qual cada linha desta matriz é considerada um documento (um vetor de características).

2.2 SVM

SVM (Support Vector Machine) é uma máquina de aprendizagem na qual é treinada a partir de exemplos inseridos na mesma. Na maioria das vezes esses exemplos são vetores de características (explanados na *seção 2.1*), nos quais são subconjuntos de R^n. Considerando a entrada (um exemplo a ser inserido no SVM) $X = (x_1, x_2, ... x_n)$, X pertence a um vetor de espaço n-dimensional R^n e $x_1, x_2, ... x_n$ são componentes do vetor X. X é assinado para classe positiva, se $f(X) \geq 0$, e para classe negativa se $f(X) < 0$. Neste caso a função f(X) é uma função de decisão. Cada vetor tem um atributo de marcação em $Y \in \{-1, +1\}$, i = 1...n, -1 pertencente a classe negativa e +1 a classe positiva. O SVM então mapeia X => Y. Como há somente 2 classes, o SVM neste caso será considerado como um classificador binário. [Guduru 2006]

Para explanar a respeito do SVM vamos utilizar um exemplo, como se segue. Imaginemos que se tem um vetor de características de duas dimensões (para esse exemplo, imaginemos a quantidade das palavras "teste" e "algoritmos" em um texto). Supondo que com essas características é possível classificar com uma boa precisão se um determinado documento é sobre o assunto "algoritmos de teste" ou não, e supondo o conjunto de dados para treino do SVM como se segue abaixo na *Tabela 1*.

Tabela 1 - Conjunto de treino para SVM

	teste	algoritmos	Classificação
Doc1	**1**	**1**	***não***
Doc2	**1**	**2**	***não***
Doc3	**1**	**3**	***não***
Doc4	**2**	**1**	***não***
Doc5	**3**	**4**	***não***
Doc6	**4**	**5**	***sim***

Doc7	**5**	**5**	***sim***
Doc8	**6**	**5**	***sim***
Doc9	**5**	**4**	***sim***
Doc10	**6**	**6**	***sim***

Se imaginarmos estes dados em um plano cartesiano, obteremos a imagem Figura 1, abaixo.

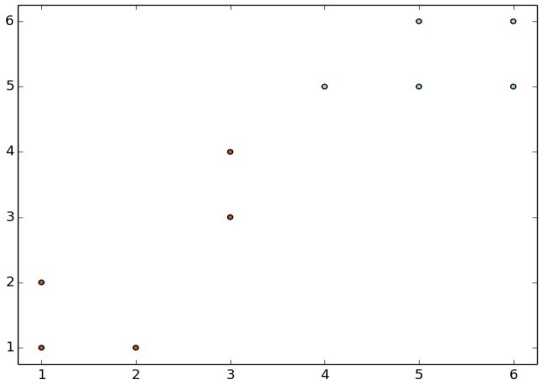

Figura 1 - Vetor de atributos em plano cartesiano

É importante notar que os pontos da classe "algoritmos de teste"(em azul) são separáveis em relação aos pontos em vermelho, e que existem diversos hiperplanos (no caso de um espaço vetorial de 2 dimensões, o hiperplano é uma linha) que separam essas classes. O objetivo do algoritmo de SVM é encontrar o hiperplano ótimo que separa estas classes. É importante perceber que, para encontrar o hiperplano é necessário treinar o SVM, ou seja, informar previamente cada dado e sua respectiva classe. Utilizando os dados da *Tabela 1* e treinando o SVM, obtemos o seguinte resultado, mostrado na *Figura 2* abaixo.

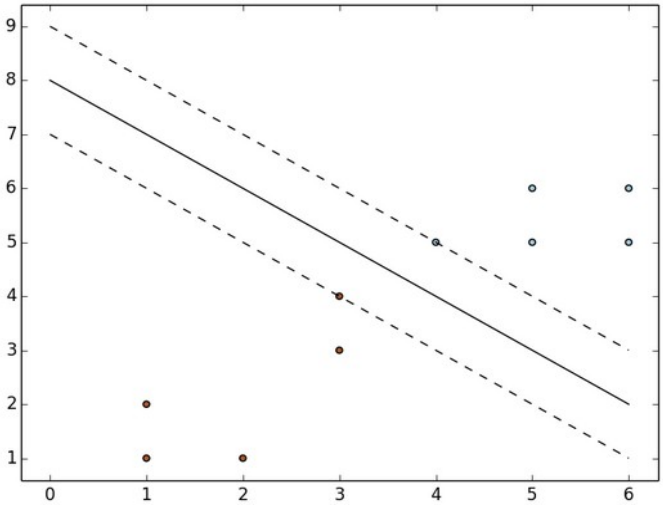

Figura 2 - Hiperplano de separação

Após o treinamento, ou seja, o hiperplano ótimo definido, é possível classificar um novo elemento, onde ele será da classe "sim" (entre os pontos da cor azul), caso esteja acima da fronteira ou "não" caso esteja abaixo da fronteira (entre os pontos da cor vermelha), como demonstra a *Figura 2*. Para entender melhor, vamos tentar classificar um documento cujo o vetor de característica é (1, 3). Ele está localizado com uma seta na *Figura 3* abaixo.

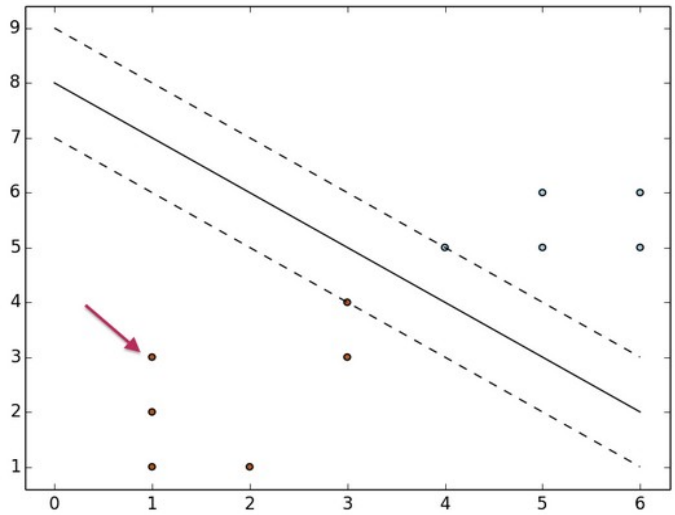

Figura 3 - Documento a ser classificado

Ao utilizar a regra descrita acima para a classificação deste documento, considera-se que o mesmo não é do assunto "algoritmos de teste".

3. SVM e Bag of Words para categorização de textos

Para este trabalho, utilizamos o algoritmo Bag of Words (já explanado na *seção 2.1*), e SVM para classificar se um documento é sobre o assunto do esporte hockey. Como no exemplo da *Figura 3*, utilizamos também um vetor de características onde cada componente representa a quantidade de vezes que a palavra X[i] ocorre no texto, sendo X (vetor de características) e i (uma posição com a quantidade em que a palavra ocorre). A grande diferença é que a quantidade de componentes será muito maior que 2.

Como conjunto de dados de treino e teste utilizamos um subconjunto do dataset Reuters-21578 Text Categorization Collection [Reuters 2007], onde utilizamos 80 exemplos de texto em inglês sobre os assuntos de hockey(40 exemplos, sendo 20 para treino e 20 para teste), religião e política (20 exemlos de cada, sendo 10 para treino e 10 para teste).

4. Análise dos resultados encontrados

4.1 Métrica Utilizada

Para a avaliação dos resultados obtidos da *seção 3*, utilizamos a curva ROC, um método gráfico para avaliação, organização e seleção de sistemas de diagnóstico e/ou predição (no nosso caso de classificadores) [Prati] e, a taxa de acurácia na classificação. O método gráfico da curva ROC é usado em grande escala pelos pesquisadores de

aprendizagem de máquina, principalmente pela característica de apresentar a taxa de “true positives” e “false positives” de uma classificação.

4.2 Resultado Encontrado

Como resultado da avaliação, tivemos a seguinte curva ROC (*Figura 4*) e a taxa de acurácia na classificação de 77.5% de sucesso na classificação (ou seja, 31 dos 40 documentos foram classificados corretamente), 10% foram falsos negativos e 12,5% falsos positivos.

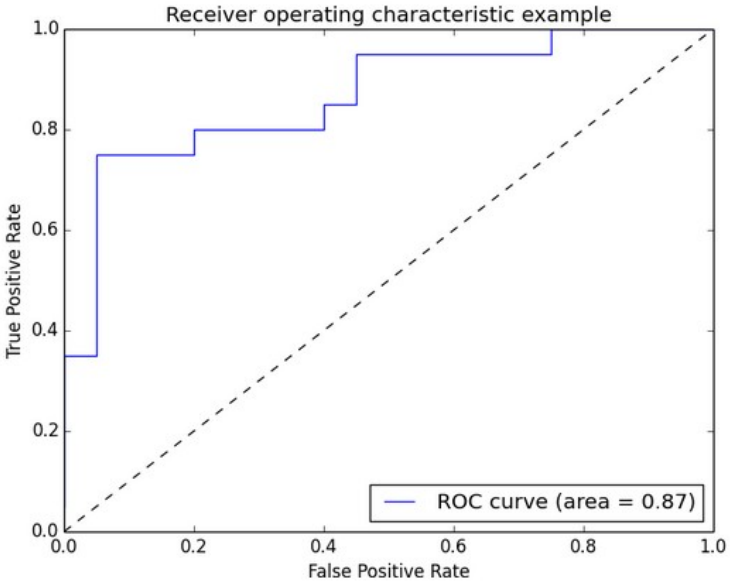

Figura 4 - Curva ROC da aplicação proposta

A taxa de precisão classificação foi um valor razoável, porém o valor que preocupa um pouco é a taxa de falsos positivos de 12,5%. Em um sistema para classificação de textos por assunto isto é uma taxa alta e causaria muita insatisfação dos usuários, visto que muitos textos serão classificados como do assunto, não sendo. A curva ROC acima também serve como argumento para esta análise, mostrando que a taxa de falso positivos é alta, mesmo com uma taxa de verdadeiro positivo também alta.

4.3 Relevância da aplicação e dos resultados encontrados

A aplicação proposta mostra, de forma muito simples, como é possível ter uma classificação por assunto de documentos em estruturas predefinidas. Isto é uma tecnologia já dominada por grandes empresas como Google, que até mesmo oferecem serviços relacionados a este tópico aos seus usuários. Além disso, o mesmo pode ser utilizado em aplicações para identificação de conteúdo pejorativo, ameaçador ou inconveniente (como é o caso dos filtros anti-spam).

5. Conclusão

Com os resultados obtidos a partir da aplicação proposta e de acordo com a metodologia aplicado para a avaliação da aplicação, concluímos que o SVM se configura como um

algoritmo rápido e simples de ser utilizado para classificação de documentos por assunto.

Os resultados obtidos são influenciados por diversos fatores além dos apresentados, principalmente pelo fato do vetor de característica utilizado não passar por nenhum pré-processamento antes de sua classificação e treino, etapa esta que pode aumentar significativamente a precisão do classificador e que não está dentro do escopo desse trabalho.

Referências

Scikit Learn. (2014) “Machine Learning In Python – Feature Extraction”,

http://scikit-learn.org/stable/modules/feature_extraction.html#the-bag-of-words representation, Julho.

Joachims, T. “Text Categorization With Support Vector Machines: Learning With Many Relevant,

http://www.cs.cornell.edu/People/tj/publications/joachims_98a.ps.gz, Julho.

Reuters-21578. (2007) “Text Categorization Collection”,

http://www.cs.umb.edu/~smimarog/textmining/datasets/, Julho.

Prati, R., Batista, G. e Monard, M. “Curvas Roc Para Avaliação De Classificadores”,

http://www.ime.unicamp.br/~wanderson/Aulas/Aula11/artigo_curva_ROC.pdf, Julho.

Guduru, N. (2006) “Text Mining With Support Vector Machines And Non-Negative Matrix Factorization Algorithms”,

http://homepage.cs.uri.edu/faculty/hamel/pubs/theses/Thesis-Neelima-Guduru.pdf, Julho.