Uma implementação do algoritmo Naïve Bayes para classificação de texto

Giancarlo Lucca, Igor A. Pereira, André Prisco, Eduardo N. Borges

Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Rio Grande – RS – Brasil

{giancarlo.lucca, igor.pereira, andre.prisco, eduardoborges}@furg.br

Abstract. *This paper present a text classification tool based on the Naïve Bayes algorithm. We have described some basic concepts about textual classification in the Information Retrieval area, the algorithm chosen, an example of use and the architecture of our tool.*

Resumo. *Este artigo apresenta uma ferramenta para classificação de texto baseada no algoritmo Naïve Bayes. São descritos alguns conceitos básicos sobre classificação textual na área Recuperação de Informações, o algoritmo escolhido, um exemplo de utilização e a arquitetura da ferramenta.*

1. Introdução

Uma das técnicas de mineração de dados amplamente utilizada é a classificação de dados. A classificação consiste no processo de encontrar, através de aprendizado de máquina, um modelo ou função que descreva diferentes classes de dados [Han e Kamber 2006]. O objetivo da classificação é rotular, automaticamente, novas instâncias da base de dados com uma determinada classe aplicando o modelo ou função "aprendidos". Este modelo é baseado no valor dos atributos das instâncias de treinamento. Diversos classificadores foram propostos nos últimos anos. Alguns utilizam árvores de decisão para rotular registros. CART [Breiman et al. 1984] e C4.5 [Quinlan 1993] são exemplos bem conhecidos. Outros algoritmos se baseiam em redes neurais artificiais, modelos probabilísticos (bayesianos) ou em regras [Mitchell 1997].

A classificação pode ser especializada na categorização textual, que consiste na organização de documentos em tópicos preestabelecidos. Esta categorização tem diversas aplicações na área de Recuperação de Informação, tais como detecção de SPAM, organização automática de e-mails, identificação de páginas com conteúdo adulto e detecção de expressões multipalavras [Manning et al. 2008].

Dado um conjunto de j classes $\mathrm{C} = \{\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_\mathrm{j}\}$ e outro de i documentos $\mathrm{D} = \{\mathrm{d}_1, \mathrm{d}_2, \ldots, \mathrm{d}_\mathrm{i}\}$ descritos por um espaço multidimensional X composto pelas palavras ou termos que aparecem em toda a coleção, o algoritmo aprende uma função de classificação $f: X \rightarrow C$ que mapeia os documentos nas classes.

2. O Algoritmo Naïve Bayes

Implementado por ferramentas como MALLET, Apache Mahout e NLTK[1], Naïve Bayes computa a probabilidade $P(c|d)$ de um documento pertencer a uma determinada classe a partir da probabilidade a priori $P(c)$ de um documento ser desta classe e das

[1] http://mallet.cs.umass.edu, http://mahout.apache.org e http://nltk.org, respectivamente.

probabilidades condicionais $P(t_k|c)$ de cada termo t_k ocorrer em um documento da mesma classe. O objetivo do algoritmo é encontrar a melhor classe C_{map} para um documento maximizando a probabilidade a posteriori conforme Equação 1, onde n_d é o número de termos no documento *d*.

$$C_{map} = \arg\max_{c \in C} P(c|d) = \arg\max_{c \in C} P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \quad (1)$$

Para evitar o *underflow* de ponto flutuante, o produto das probabilidades é substituído pela soma dos logaritmos das probabilidades, resultando no algoritmo apresentado na Figura 1. A função de treinamento extrai o vocabulário da coleção de documentos *D*. A seguir é calculado um vetor de probabilidades a priori dividindo o número de documentos de cada classe pelo tamanho da coleção. Ao final do treinamento é estimada uma matriz de probabilidades condicionais através da frequência relativa dos termos nos documentos que pertencem a uma determinada classe. Para evitar que essa estimativa seja nula para uma combinação de termo e classe que não ocorra na coleção de treinamento, é usada uma suavização de Laplace [Manning et al. 2008] que incrementa cada contagem. A função de classificação recebe como parâmetros, além do documento de teste, o conjunto de classes, o vocabulário e as probabilidades estimadas no treinamento. Para cada classe, a probabilidade a posteriori é calculada somando o logaritmo da probabilidade a priori com os logaritmos das probabilidades condicionais de cada termo do documento de teste. O documento então é rotulado com a classe c que obtiver a maior probabilidade a posteriori.

```
TRAINMULTINOMIALNB(C, D)
 1  V ← EXTRACTVOCABULARY(D)
 2  N ← COUNTDOCS(D)
 3  for each c ∈ C
 4  do N_c ← COUNTDOCSINCLASS(D, c)
 5     prior[c] ← N_c/N
 6     text_c ← CONCATENATETEXTOFALLDOCSINCLASS(D, c)
 7     for each t ∈ V
 8     do T_ct ← COUNTTOKENSOFTERM(text_c, t)
 9     for each t ∈ V
10     do condprob[t][c] ← (T_ct+1) / Σ_t'(T_ct'+1)
11  return V, prior, condprob

APPLYMULTINOMIALNB(C, V, prior, condprob, d)
1  W ← EXTRACTTOKENSFROMDOC(V, d)
2  for each c ∈ C
3  do score[c] ← log prior[c]
4     for each t ∈ W
5     do score[c] += log condprob[t][c]
6  return arg max_{c∈C} score[c]
```

Figura 1. Pseudocódigo do algoritmo Naïve Bayes [Manning et al. 2008].

A Tabela 1 apresenta um exemplo de classificação textual executado na ferramenta para validação da implementação. Os documentos representam um conjunto de manchetes, extraídas do jornal Diário Catarinense, pré-processadas e rotuladas com uma das seguintes classes: cultura, esportes e policial. Para facilitar a visualização, apenas algumas palavras-chave (termos) foram apresentadas. Uma vez aprendida a função de classificação dos documentos de treinamento, o modelo é aplicado sobre o documento de teste.

Para estimar a classe do documento 9 são calculadas as probabilidades a priori $P(Cultura) = 3/8$, $P(Esportes) = 3/8$, $P(Polícia) = 2/8$. Após, o algoritmo calcula as

probabilidades condicionais da seguinte forma. As frequências 27, 24 e 18 representam o número de termos por classe no conjunto de treinamento e 67 é o tamanho do vocabulário.

$$P(clubes|Cultura) = P(catarinenses|Cultura) = \frac{0+1}{27+67} = \frac{1}{94}$$

$$P(cultural|Cultura) = \frac{1+1}{27+67} = \frac{2}{94}$$

$$P(clubes|Esportes) = P(catarinenses|Esportes) = \frac{1+1}{24+67} = \frac{2}{91}$$

$$P(cultural|Esportes) = \frac{0+1}{24+67} = \frac{1}{91}$$

$$P(clubes|Polícia) = P(catarinenses|Polícia) = P(cultural|Polícia) = \frac{0+1}{18+67} = \frac{1}{85}$$

Tabela 1. Termos que compõe cada documento distribuídos por classe.

Classe	d	Termos contidos nos documentos
Cultura	1	brinquedos criativos estimulam encantam criancas pais summer balneario
	2	prefeitura camboriu abre inscricoes concurso rainha princesas camboriu festa ru
	3	concurso cultural vai batizar filhotes zoo beto carrero world
Esportes	4	desafio estrelas pilotos famosos estarao em penha
	5	apos entrega laudos clubes catarinenses esperam liberacao estadios
	6	presidente confederacao brasileira de tenis acreditamos nesta nova geracao
Polícia	7	pais chamam policia encontrar drogas escondidas quarto filho anos timbo
	8	prefeita luzia recebe visita comandante batalhão policia militar
?	9	clubes catarinenses clubes cultural

Por fim, são calculadas as probabilidades a posteriori para cada classe.

$$P(Cultura|d_9) = log\big(P(Cultura)\big) + 2\,log\big(P(clubes|Cultura)\big) + log\big(P(catarinenses|Cultura)\big) + log\big(P(cultural|Cultura)\big)$$

$$P(Cultura|d_9) = log\left(\frac{3}{8}\right) + 2\,log\left(\frac{1}{94}\right) + log\left(\frac{1}{94}\right) + log\left(\frac{2}{94}\right) \cong -8{,}02$$

Seguindo o mesmo raciocínio para as demais classes, as probabilidades a posteriori são $P(Esportes|d_9) \cong -7{,}36$ *e* $P(Polícia|d_9) \cong -8{,}32$. Assim, o documento de teste é rotulado com a classe Esportes porque obteve-se a maior estimativa.

3. Implementação

O código-fonte foi desenvolvido na linguagem Java e está coberto principalmente pela classe *NaiveBayes* descrita pelos atributos e métodos apresentados na Figura 2. A maioria dos métodos é análoga às funções apresentadas na Figura 2. Necessitam explicações adicionais o atributo *vetClasses* que armazena o conjunto de classes e *listaTextos* que armazena a coleção. O método *buscaTextos* é um procedimento que percorre o sistema de arquivos investigando os diretórios configurados como classes, analisando os documentos contidos nesses diretórios e chamando a função *extractVOCABULARY* internamente. *imprimeInfos* é um método utilizado como teste que exibe informações como número de classes, textos, vocabulário, etc. *indiceVocabulario* recebe um termo e retorna o índice equivalente no vocabulário.

A Figura 3 apresenta a interface gráfica da ferramenta que destaca os documentos de treinamento, o número de documentos por classe, a probabilidade a posteriori do documento de teste pertencer a cada classe (*Score*) e as palavras que compõem o vocabulário. Por fim, a classe predita é apresentada.

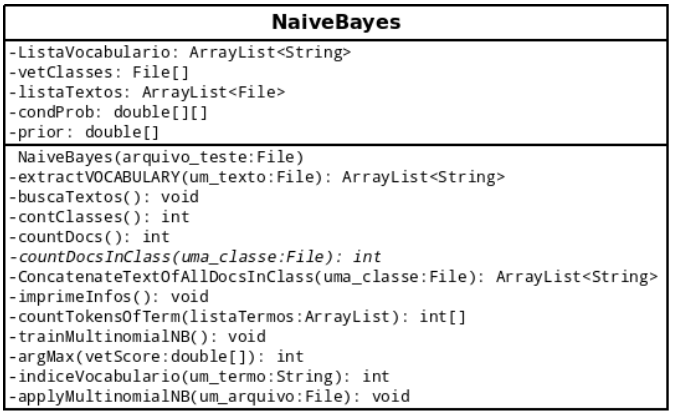

Figura 2. Classe representando o algoritmo Naïve Bayes.

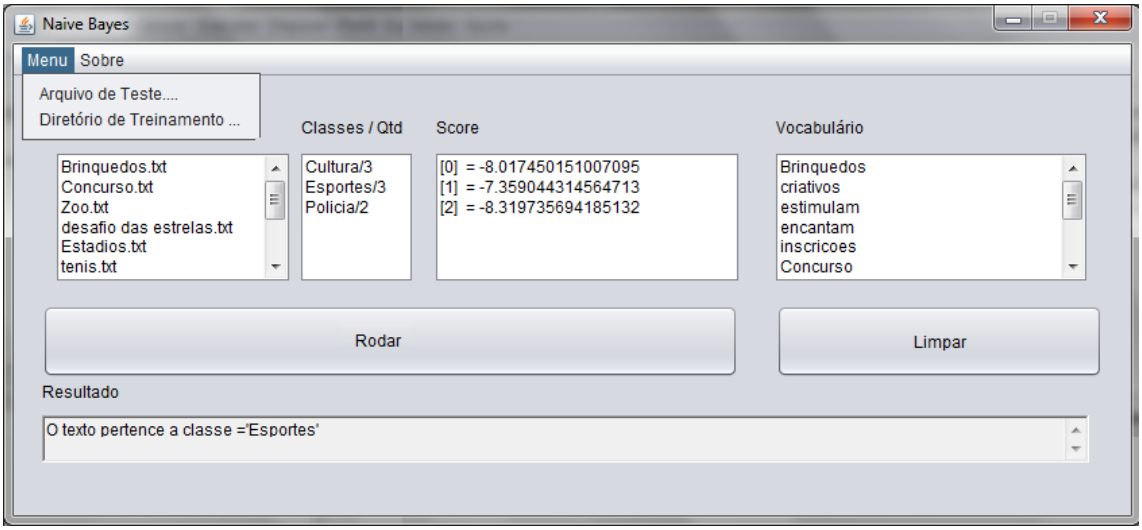

Figura 3. Interface gráfica da ferramenta proposta.

4. Conclusões e trabalhos futuros

A ferramenta desenvolvida será utilizada para apoiar o ensino da disciplina Sistemas de Informações Avançados oferecida pelo Centro de Ciências Computacionais da FURG. Ainda é necessário codificar determinadas melhorias como os recursos de interface. Pretende-se diferenciar a ferramenta proposta de outras implementações evidenciando os cálculos dos passos internos do algoritmo quando o usuário seleciona determinada linha do pseudocódigo, facilitando o aprendizado dos estudantes.

Referências

Breiman, L.; Friedman, J.; Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks.

Han, J. and Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2nd ed.

Manning, C. D.; Raghavan, P. and Schutze H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers.