

Análise de Um Exemplo Funcional de Redes Bayesianas

Daniel Souto Neves
Leonardo Pliskievski
Gustavo Pliskevski

November 2019

1 Introdução

Para uma pessoa, devido à seus vários anos de experiência acumulados ao viver em sociedade, determinar quando a conotação de uma frase é positiva ou negativa é algo extremamente intuitivo, mas infelizmente para um computador isto já não é algo tão natural.

Uma forma de "ensinar" a máquina a fazer essa análise seria por meio de um algoritmo que aplica alguns conceitos de probabilidade condicional, como o teorema de Bayes e as Redes Bayesianas e, à partir de uma base de dados fornecida previamente, este programa consegue determinar pela probabilidade calculada a chance da frase digitada ser sobre algo bom ou ruim, e a este método dá-se o nome de um classificador de Bayes ingênuo.

2 Rede Bayesiana

Nos estudos probabilísticos, pode-se calcular a probabilidade condicional de um evento A sabendo que ocorreu um evento B ao multiplicar a probabilidade de A pela probabilidade de B dado A, e dividir pela probabilidade de B, e este cálculo é a definição do chamado teorema de Bayes.

Uma rede Bayesiana, por sua vez, é um grafo dirigido que apresenta todas as variáveis de uma situação e suas dependências condicionais. Um exemplo de uma rede Bayesiana, o qual foi implementado para este artigo, é o classificador de Bayes ingênuo no qual se aplica o teorema de Bayes para classificar uma informação à partir da aplicação do teorema de Bayes, presumindo que as variáveis na rede Bayesiana tenham uma forte independência entre si.

3 Implementação

3.1 Biblioteca

Para desenvolver o algoritmo foi utilizada como base a biblioteca chamada ‘textblob’, no qual é importada de dentro do pacote ‘classifiers’ o classificador ‘Naive Bayes’ - ‘NaiveBayesClassifier’

3.2 Problema

Classificar a conotação de frases baseadas em seu conteúdo, (Exemplos: a frase: ”Este carro é incrível” é classificada como positiva, já a frase ”Este carro é muito ruim” pode ser classificada com uma sensação negativa).

3.3 Como utilizar a biblioteca

Com base no objetivo de classificar os dados, a biblioteca oferece um ponto de entrada como lista que é possível ser inserido no seu construtor, esta lista pode conter um conjunto de dados com a finalidade de ser um ”train set” (conjunto de treinamento), este conjunto deve seguir a estrutura de um ou muitos objetos, cada um com seu respectivo conteúdo (um dado de entrada), e uma classificação para este dado previamente calculados com a finalidade de realizar uma entrada de dados para treinar o classificador. À partir disto, deve-se utilizar a biblioteca ‘TextBlob’ a fim de processar dados textuais, esta biblioteca providência uma API (Application Programming Interface) simples e intuitiva para realizar tarefas comuns de Processamento de Linguagem Natural (NLP), como análise de sentimentos, classificações, traduções e dentre outros.

4 Resultados

Após ser alimentado com uma frase digitada pelo usuário, o programa irá analisar, usando como referência a base de dados fornecida previamente pelo desenvolvedor, o programa irá determinar se essa frase é ‘positiva’ ou ‘negativa’, sendo que estas classificações são mutuamente exclusivas.

5 Conclusões

Podemos concluir à partir deste exemplo de um Naive Bayes classifier que quanto maior a base de dados do programa for, maior será a precisão do algoritmo para determinar a conotação da frase. Também é possível afirmar que cumprimentos maiores para a frase não alteram explicitamente o resultado final, visto que o programa analisa a frase palavra por palavra e, sendo assim, se a porcentagem de palavras positivas for maior que a porcentagem de palavras negativas a frase será positiva, e vice-versa.

6 Referências

<https://people.cs.umass.edu/~mccallum/courses/gm2011/02-bn-rep.pdf>
<http://www.eng.tau.ac.il/~bengal/BN.pdf>
<https://textblob.readthedocs.io/en/dev/>