Laboratório Nº 6: Similaridade do Cosseno

Extração Automática de Informação
2021/2022

Prof. Joaquim Filipe
Eng. Filipe Mariano

Objetivos

- Utilizar a similaridade do cosseno para determinar a que classe pertence um documento

1. Introdução

Uma abordagem comum utilizada para combinar documentos semelhantes baseia-se na contagem do número máximo de palavras comuns entre os documentos (distância euclidiana). Contudo, essa abordagem apresenta uma falha, que é à medida que o documemnto aumenta, o número de palavras comuns tende a aumentar mesmo que os documentos falem sobre tópicos diferentes. Neste sentido, a métrica de similaridade do cosseno é mais adequada para a identificação da similaridade entre documentos.

2. Métricas de Similaridade entre Documentos

No que diz respeito a um espaço vetor-modelo, criado a partir de uma matriz termo-documento, existem várias métricas que representam um valor de distância entre vetores(ou pontos) e, em alguns casos, um valor de similaridade. É de notar que nos casos em que a medida retorna um valor de distância, é possível converter esse valor para um valor de similaridade. Tipicamente, esta conversão é feita pela operação contrária, isto é, `similaridade = (1-distância)`, o que significa que quanto mais próximos os vetores(ou pontos) estejam no espaço, maior a similaridade entre os objetos a classificar.

Existem diversas medidas para o cálculo da similaridade sendo que duas das mais conhecidas são: a Distância Euclidiana e a Similaridade do Cosseno.

2.1. Distância Euclidiana

Esta medida calcula um valor de distância, não normalizado, entre dois pontos no espaço. Esta distância é equivalente ao comprimento do segmento de reta que une os respetivos pontos. É dada pela fórmula:

$$D(a,b) = \sqrt{\sum (a_i - b_i)^2}$$

A distância euclidiana tem em conta a magnitude do vetor. Em termos práticos, isto significa que, no exemplo do contexto de classificação de documentos por palavras, o valor da distância é influenciado pelo peso da palavra no documento. Esta medida é tipicamente usada em contextos em que os objetos a comparar tenham tamanhos equivalentes e os atributos dos mesmos tenham pesos equivalentes.

2.2. Similaridade do Cosseno

A similaridade do cosseno é uma métrica usada para medir a similaridade de documentos, independentemente do seu tamanho. Do ponto de vista matemático, mede o cosseno do ângulo entre dois vetores projetados num espaço multidimensional. Esta análise torna-se vantajosa porque mesmo que os dois documentos estejam distantes da distância euclidiana (devido ao tamanho do documento), é provável que possam na mesma ser similares. Quanto menor o ângulo entre vetores, maior será a similaridade do cosseno.

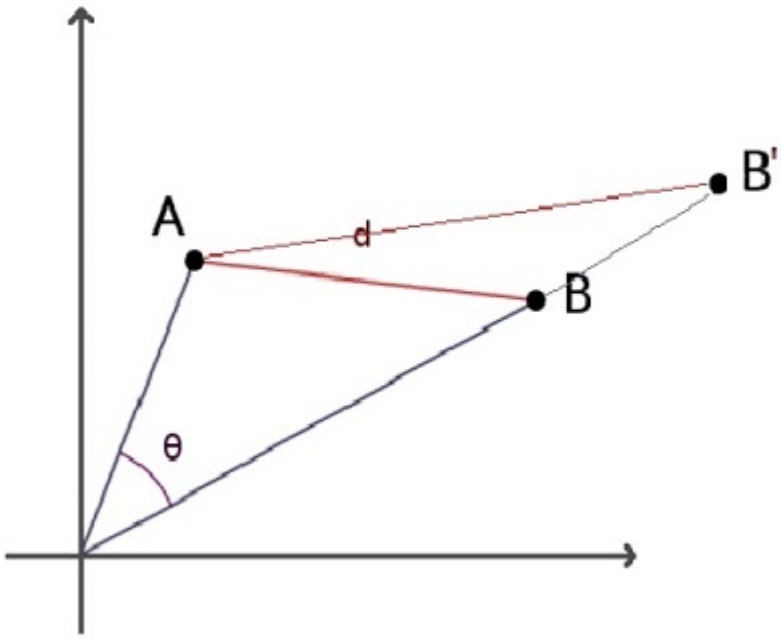
$$C(a,b) = \frac{A \cdot B}{||A||||B||}$$

ou

$$C(a,b) = \frac{\sum(a_i * b_i)}{\sqrt{\sum(a_i)^2} * \sqrt{\sum(b_i)^2}}$$

2.3. Diferença entre Distância Euclidiana e Similaridade do Cosseno

A figura abaixo demonstra a diferença entre as medidas da distância euclidiana e similaridade do cosseno num espaço bidimensional, onde d equivale ao segmento de reta traçado entre os pontos A e B e Θ igual ao valor do cosseno do ângulo entre os vetores.



Se considerarmos a distância entre A e B', observamos que, apesar de o peso dos valores ter aumentado, o cosseno do ângulo entre os vetores é constante enquanto o comprimento do segmento de reta usado no cálculo da distância euclidiana aumenta e por consequência a similaridade entre A e B diminui.

3. Exercícios

1. No módulo `train.js` exporte uma propriedade `classVectors` que após o pré-processamento dos textos do conjunto de treino e do cálculo dos diversos vetores ficará preenchido com um objeto em que cada classe terá o seu vetor de médias do TFIDF e IDF, assim como a bag of words.

Possível Exemplo:

```
[
    { label: "happy", bagofwords: [...], tfidf: [...], idf: [...] },
    { label: "not happy", bagofwords: [...], tfidf: [...], idf: [...] }
]
```

2. Crie um módulo `classifier.js` que exporte uma função `cosineSimilarity` que recebe um texto e o `classVectors` definido no exercício anterior.
a. Em primeiro lugar deverá ser feito todo o pré-processamento anteriormente programado para o texto recebido como argumento.
b. Calcular o `tf-idf` para os termos encontrados no texto e que existem na bag of words calculada durante o treino, para cada classe (`happy` e `not happy`). Ou seja, com a bag of words existente e os valores encontrados para cada classe irá se tentar criar 2 arrays com o mesmo número de elementos da bag of words de cada classe com os respetivos valores de tf-idf encontrados para cada termo, nesse texto recebido como argumento.

Nota: Irá precisar dos valores de IDF, que estão no objeto criado no exercício 1, para calcular o `tf-idf` do termo.

c. Escrever uma função `calculateCosineSimilarity` que recebe 2 vetores e faz os cálculo (ver secção 2.2) para obter o valor de similaridade do cosseno desses 2 vetores. Utilizar essa função na função `cosineSimilarity` para calcular a similaridade do cosseno para a classe `happy` e para a classe `not happy`. A classe cujo resultado da similaridade for maior será a que deverá ser retornada pela função, com o respetivo valor de similaridade.

3. Com a nova função exportada pelo módulo `classifier.js` testar a função `cosineSimilarity` para uma amostra de 50 textos existentes na base de dados marcados como `happy` e outros 50 marcados como `not happy`. Deverá utilizar textos diferentes daqueles que utilizou para o conjunto de treino.