Processamento de texto

Inteligência na Web e Big Data

Fabricio Olivetti de França e Thiago Ferreira Covões
folivetti@ufabc.edu.br, thiago.covoes@ufabc.edu.br

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC

UFABC

CMCC
Centro de Matemática, Computação e Cognição
Center for Mathematics, Computation and Cognition

Representação Textual

Mineração de Textos

- Documentos de textos:
  - Não possuem representação vetorial
  - Interdependência dos atributos
  - Tamanho variável

Textos como conjuntos

- Se representarmos os documentos de textos como o conjunto de suas palavras:

D1 = ”Estou assistindo a uma aula de Big Data, mas tudo que aprendi foi Haskell durante a aula!”

D2 = ”Hoje aprendi Haskell na aula, será que o que aprendi será útil na minha vida?”

Textos como conjuntos

- Se representarmos os documentos de textos como o conjunto de suas palavras:

$$D1 = \{\texttt{Estou, assistindo, a, uma, aula, de, Big, Data, mas, tudo, que, aprendi,foi, Haskell, durante, aula!}\}$$

$$D2 = \{\texttt{Hoje, aprendi, Haskell, na, aula, será, que, o, útil, minha, vida?}\}$$

Textos como conjuntos

- Calculando a similaridade de Jaccard, temos:

$$D1 \cap D2 = \{\texttt{que, aprendi, Haskell}\}$$

$$D1 \cup D2 = \{\texttt{Estou, assistindo, a, uma, aula, de, Big, Data, mas, tudo, que, aprendi, foi, Haskell, durante, aula!, Hoje, na, aula, será, o, útil, minha, vida?}\}$$

$$J(D1, D2) = \frac{3}{24} = 0.125$$

- Essa representação é conhecida como Bag-of-Words.

Normalização do Texto

- Pode ser interessante padronizar a forma do texto para termos serem considerados como um elemento único do conjunto independente de como é escrito.
- Por exemplo: Estou, estou, esTou, aula!, aula, aula?, útil, util.

$$D1 = \{\texttt{estou, assistindo, a, uma, aula, de, big, data, mas, tudo, que, aprendi, foi, haskell, durante, aula}\}$$

$$D2 = \{\texttt{hoje, aprendi, haskell, na, aula, sera, que, o, util, minha, vida}\}$$

$$J(D1, D2) = \frac{4}{23} = 0.17$$

Eliminação de atributos irrelevantes

- Podemos eliminar palavras que não apresentam significado sozinhas:

$$D1 = \{\texttt{estou, assistindo, aula, big, data, tudo, aprendi, haskell, durante, aula}\}$$

$$D2 = \{\texttt{hoje, aprendi, haskell, aula, sera, util, minha, vida}\}$$

$$J(D1, D2) = \frac{4}{14} = 0.28$$

Term-Frequency

- Se um termo aparece repetidas vezes em um documento, isso significa que ele pode ter uma importância maior do que os outros termos.
- No nosso exemplo, a repetição do termo Haskell indica um dos temas dos nossos documentos.
- A informação de frequência pode ser importante para a representação de nossos documentos. Podemos fazer então:

$$fn(t, d) = \frac{f(t, d)}{|d|},$$

- com $f(t, d)$ sendo a frequência do termo t no documento d e $|d|$ a quantidade de termos no documento d.

Term-Frequency

- A ideia para computar os vetores TF é primeiro representar cada documento como uma lista $(token, 1.0)$ e, em seguida:
  - Ordenar essa lista pelo token
  - Agrupar os itens com mesmo token
  - Somar os valores em cada grupo
  - Dividir os valores pelo número de tokens

Inverse Document Frequency

- Algumas palavras aparecem com uma frequência muito superior as demais, como: e, que, ou, etc.
- Essas palavras não costumam apresentar um significado discriminatório e, portanto, podem ter um peso menor. Para isso podemos multiplicar o TF por:

$$idf(t) = \log \frac{|D|}{|\{d \in D : t \in D\}|}$$

Exercício

Dada as tabelas abaixo, calcule a similaridade de cosseno entre os documentos 1 e 2 com a representação tf e tf-idf:

$$s_{\cos}(\mathbf{x}y) = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$

	A	B	C	D
D1	30	5	0	1
D2	60	0	4	10
D3	10	2	0	0
D4	40	0	4	8

Exercício

tf	A	B	C	D
D1	0.83	0.14	0.00	0.03
D2	0.81	0.00	0.05	0.13
D3	0.83	0.17	0.00	0.00
D4	0.77	0.00	0.08	0.15

Exercício

$$idf = [0.00, 0.69, 0.69, 0.29]$$

(base e)

tf-idf	A	B	C	D
D1	0.00	0.10	0.00	0.01
D2	0.00	0.00	0.04	0.04
D3	0.00	0.115	0.00	0.00
D4	0.00	0.00	0.05	0.04

$$s_{tf} = [0.97]$$

$$s_{tfidf} = [0.06]$$

Padronizando e Normalizando os Atributos

Padronização

- Muitos algoritmos de Aprendizado de Máquina supõem que os valores dos atributos seguem $N(0, 1)$.
- Se um atributo não segue esse padrão, pode dominar a função-objetivo e se tornar importante demais.

Padronização

- Dado uma matriz de dados X, podemos padronizar os valores de cada um de seus elementos como:

$$\hat{X}_{i,j} = \frac{X_{i,j} - \bar{X}_{i,j}}{\sigma_j}$$

Escalonamento

- Algoritmos baseados em métodos de gradiente tendem a se beneficiar quando os atributos estão entre $[0, 1]$.

$$\hat{X}_{i,j} = \frac{X_{i,j} - \min X_{:,j}}{\max X_{:,j} - \min X_{:,j}}$$

Normalização

- Finalmente podemos normalizar cada amostra da base utilizando a normalização de vetores:

$$\hat{X}_{i,j} = \frac{X_{i,j}}{\|X_i\|_p}$$

- Esta transformação tem uma propriedade pois relacionada a similaridade do coseno com a distância euclidiana, uma vez que $\|\mathbf{x}\| = 1$

Exercício

Calcule o vetor padronizado, normalizado e escalonado de $[1, 3, 2, 5, 4, 6, 3]$

$$v_p = [-1.4, -0.25, -0.83, 0.91, 0.33, 1.50, -0.25]$$
$$v_n = [0.1, 0.3, 0.2, 0.5, 0.4, 0.6, 0.3]$$
$$v_e = [0, 0.4, 0.2, 0.8, 0.6, 1, 0.4]$$