

单词向量空间^[1]

NLP

单词向量空间^[1]

序

基本想法

基本假设

严格定义

TF-IDF为什么表示重要程度？

相似度

缺点

序

文本数据挖掘的一个核心问题是对文本的语义进行表示，进行文本相似度计算。

最简单方法利用向量空间模型VSM（也就是单词向量空间模型WVSM）。

基本想法

给定一个文本,用一个向量表示文本的语义,向量每一维对应一个单词,其数值为单词出现在文本中的频数或权值。

基本假设

- 文本中所有单词的出现情况表示了文本的语义内容。
- 文本集合中每个文本都表示一个向量，存在一个向量空间
- 向量空间的度量，即内积或标准化内积表示语义相似度

严格定义

给定一个含有 n 个文本的集合：

$$D = \{d_1, d_2, \dots, d_n\}$$

以及所有文本中出现的 m 个单词的集合:

$$W = \{w_1, w_2, \dots, w_n\}$$

将单词在文本中出现的数据用 单词-文本矩阵 表示：

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

x_{ij} 表示单词 w_i 在文本 d_i 中的频数或权值。经常为一个稀疏矩阵。（想一想）

具体权值计算用单词频率-逆文本频率（TF-IDF）：

$$TFIDF_{ij} = \frac{tf_{ij}}{tf_{\cdot j}} \log \frac{df}{df_i}, i = 1, 2, \dots, m; j = 1, 2, \dots, n$$

tf_{ij} 是单词 w_i 出现在文本 d_i 中的频率数。

$tf_{\cdot j}$ 是文本 d_i 中出现所有单词的频率之和

df_i 含有单词 w_i 的文本数。

df 文本集合 D 的全部文本数。

TF-IDF 为什么表示重要程度？

频数越高，重要程度越高。出现文本数越少，越能体现特点。

两者相乘表示综合重要程度

相似度

单词-文本矩阵第 j 列向量 x_j ：

$$x_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{bmatrix}, j = 1, 2, \dots, n$$

表示文本 d_j 中各个单词的重要程度,故矩阵可以重写为：

$$X = [x_1 x_2 \dots x_n]$$

两个单词的内积或者标准化内积（余弦）表示对应文本之间的语义相似度。

因此 d_i 与 d_j 之间的相似度为：

$$x_i * x_j \text{ or } \frac{x_i * x_j}{\|x_i\| \|x_j\|}$$

缺点

内积相似度未必能够准确表达两个文本的语义相似度，因为自然语言的单词具有一词多义性和多词一意性。

[1] 本文内容全部来自《统计学习方法（第二版）》，有微量修改和整理。 [↩](#)