VECTOR SPACE MODEL

Francois Role - francois.role@parisdescartes.fr

September 2017

1 Introduction

Some important text mining activities: * Look for a set of documents relevant to a given query (information retrieval) * categorize a set of documents w.r.t. predefined classes (classification) * Partition a set of documents (clustering)

In all these cases, it is useful to represent documents as vectors.

2 Document Vectors

Given the following three documents (each line corresponds to a distinct document):

```
aaa bbb ccc aaa eee aaa ggg.

ggg fff eee.

ddd aaa ggg aaa eee eee.
```

Let's build a so-called "document-term" matrix D which represents these documents:

$$\begin{bmatrix} 3 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 2 & 0 & 0 & 1 & 2 & 0 & 1 \end{bmatrix}$$

In the above matrix, each row is a vector that represents a document and you can notice that each row has seven components since there are seven distinct words $w_1, ..., w_7$

Note that the order of the components does not reflect the order in which words appear in the documents, thus justifying the use of the phrase **bag of word** to describe this kind of representation. For example the word "fff" corresponds to column 6 while it appears as the fifth word in document 1 and as the third word in document 2.

D is an example of a **co-occurrence matrix**: for a document **d**, the component c_i of each row vector has as value the number of times term w_i appears in **d**.

Exercise

(a) assuming that a collection of documents is represented as a list of chain write a program to build the document-term matrix D for the above-mentioned documents.

3 Vector Similarity

3.0.1 Limitations of the Euclidean Distance

Suppose we have the following vocabulary of 4 words:

word 0 funny, word 1 cat, word 2 nuclear, word 3 energy

and suppose we have the following vector representation for 3 documents $\begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 50 \\ 20 \\ 0 \\ 0 \end{pmatrix}$, and $\begin{pmatrix} 0 \\ 0 \\ 2 \\ 3 \end{pmatrix}$

According to the euclidean distance, the first document, which is about *funny cats* is closer to document 3, which is about *nuclear energy* than to the second document which is also about cats!

Clearly, euclidean distance is not appropriate in this case. One way of overcoming the problem is to divide each coordinate by the sample standard deviation. In our case, it allows to find that the first document is closer to the second than the third but very narrowly.Try it! We need a measure that more clearly brings closer documents talking about the same topic.

3.0.2 Cosine Similarity

The best solution is to abstract from the magnitude problems by considering the dierction of the vectors. The similarity between two vectors u and v is computed as:

$$\cos \widehat{u, v} = \frac{u \cdot v}{\|u\| \|v\|}$$

However, before that, we have to weight the vectors. Else, frequent but not necessarily informative words would have too great a weight (think of words such as "the","and","for", "thing", etc.)

4 Weighting the Vectors

The basic idea is: "a term is important for a document if it frequently occurs in this document and rarely elsewhere"

4.1 Basic TF-IDF formulas

How frequent is term i in the collection (**inverse document frequency**) :

$$idf_i = \ln \frac{n}{df_i}$$

How many times does term i occur in document j :

$$tf_{i,j}$$

Multiply to estimate how term i characterizes document j : $w_{i,j} = tf_{i,j} * idf_i$

4.2 Some Variants

- $tf * (idf + 1) = tf + tf * idf$ instead of $tf * idf$ so that terms that occur in all documents of a training set, will not be ignored
- $df_i + 1$ instead of df_i to prevent 0 division (idf smoothing)
- $1 + log(tf)$ instead of tf (sublinear tf scaling)

5 Normalizing the Vectors

Normalize a vector u by dividing each of its components by its norm. For example, using L2 norm, each component of u has to be divided by $\|u\|_2 = \left(\sum_{i=1}^{m} |u_i|^2 \right)^{1/2}$

An added benefit of normalization is that it will simplify the computation of the cosinus between two vectors.

Exercise

(b) Weight the three document vectors (row vectors) in matrix D using TF-IDF weighting.
(c) Normalize the weighted vectors.

6 Computing Vector Similarities between Queries and Documents

We have represented documents as weighted, normalized vectors. Why not do the same for the query? A query is a small document after all. We could then compute the similarity between a document d and a query q as follows:

$$\cos \widehat{q, d} = q \cdot d$$

Exercise

(d) Compute the angle in degrees between vectors $\begin{pmatrix} 3 \\ 8 \end{pmatrix}$ and $\begin{pmatrix} 2 \\ 7 \end{pmatrix}$.
(e) Compute the cosine similarity between the query "aaa eee" and each of the three sample documents. Form the query vector and use the idf_i values computed from the the corpus to weight this vector.
(f) Deduce a ranked list of documents (from the most similar to the query to the last similar).