Vector space model

From Wikipedia, the free encyclopedia

Jump to: navigation, search

Vector space model (or *term vector model*) is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, such as, for example, index terms. It is used in information filtering, information retrieval, indexing and relevancy rankings. Its first use was in the SMART Information Retrieval System.

Contents

[hide]

[edit] Definitions

Documents and queries are represented as vectors.

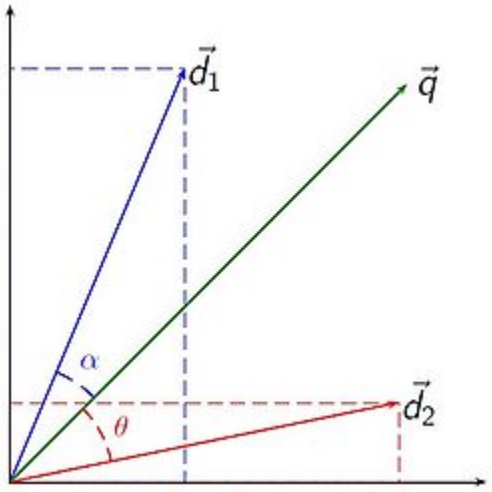
$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$
$$q = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$$

Each dimension corresponds to a separate term. If a term occurs in the document, its value in the vector is non-zero. Several different ways of computing these values, also known as (term) weights, have been developed. One of the best known schemes is tf-idf weighting (see the example below).

The definition of *term* depends on the application. Typically terms are single words, keywords, or longer phrases. If the words are chosen to be the terms, the dimensionality of the vector is the number of words in the vocabulary (the number of distinct words occurring in the corpus).

Vector operations can be used to compare documents with queries.

[edit] Applications



Relevancy rankings of documents in a keyword search can be calculated, using the assumptions of document similarities theory, by comparing the deviation of angles between each document vector and the original query vector where the query is represented as same kind of vector as the documents.

In practice, it is easier to calculate the cosine of the angle between the vectors instead of the angle:

A cosine value of zero means that the query and document vector are orthogonal and have no match (i.e. the query term does not exist in the document being considered). See cosine similarity for further information.

[edit] Example: tf-idf weights

In the classic vector space model proposed by Salton, Wong and Yang [1] the term specific weights in the document vectors are products of local and global parameters. The model is known as term frequency-inverse document frequency model. The weight vector for document *d* is , where

and

- tf_t is term frequency of term *t* in document *d* (a local parameter)

- is inverse document frequency (a global parameter). $|D|$ is the total number of documents in the document set; is the number of documents containing the term t.

Using the cosine the similarity between document d_j and query q can be calculated as:

In a simpler Term Count Model the term specific weights do not include the global parameter. Instead the weights are just the counts of term occurrences: $w_{t,d} = \mathrm{tf}_t$.

[edit] Advantages

The vector space model has the following advantages over the Standard Boolean model:

1. Simple model based on linear algebra
2. Term weights not binary
3. Allows computing a continuous degree of similarity between queries and documents
4. allows ranking documents according to their possible relevance
5. Allows partial matching

[edit] Limitations

The vector space model has the following limitations:

1. Long documents are poorly represented because they have poor similarity values (a small scalar product and a large dimensionality)
2. Search keywords must precisely match document terms; word substrings might result in a "false positive match"
3. Semantic sensitivity; documents with similar context but different term vocabulary won't be associated, resulting in a "false negative match".
4. The order in which the terms appear in the document is lost in the vector space representation.
5. Assumes terms are independent
6. Weighting is intuitive but not very formal

[edit] Models based on and extending the vector space model

Models based on and extending the vector space model include:

- Generalized vector space model
- (enhanced) Topic-based Vector Space Model
- Latent semantic analysis
- Latent semantic indexing
- DSIR model
- Term Discrimination
- Rocchio Classification

[edit] Software that implements the vector space model

The following software packages may be of interest to those wishing to experiment with vector models and implement search services based upon them.

[edit] Free open source software

- Apache Lucene. Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java.
- SemanticVectors. Semantic Vector indexes, created by applying a Random Projection algorithm (similar to Latent semantic analysis) to term-document matrices created using Apache Lucene.

[edit] Further reading

- G. Salton, A. Wong, and C. S. Yang (1975), "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, nr. 11, pages 613–620. *(The article in which the vector space model was first presented)*
- Description of the vector space model
- Description of the classic vector space model by Dr E Garcia

[edit] See also

- Inverted index
- Compound term processing

[edit] References

1. ^ G. Salton , A. Wong , C. S. Yang, A vector space model for automatic indexing, Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975