Vector Space Model In Information Retrieval Latent Semantic Analysis

Chaitanya Kumar P-2020201012

November 2020

Abstract

Information retrieval is great technology behind web search services [1]. Largely three classic framework models have been used in the process of retrieving the information namely Boolean, vector space and Probabilistic models. In this presentation we in essence constrict to the Vector Space Model. Vector space model is one of the classical and widely applied retrieval models to evaluate the relevance of the web page. The main operation of retrieval is computing the cosine similarity between the query vector and set of documents vector and rankings were given accordingly. In this we present the different approaches of vector space and the problems and issues in using vector space model.

1 Introduction

Information retrieval systems are designed to help users to quickly find information on the web. Information retrieval(IR) deals with the representation, storage, organisation and access to information items. IR models attempts to capture the latent(hidden) semantic relationship between the data items .The field of information retrieval attained peak popularity in the recent years. In the process of information retrieval still two problems exit. First information retrieval process fetch some irrelevant documents together with relevant documents. Second search engines are not capable of fetching all the relevant documents. Vector space model in particular used to filter the relevant documents from the irrelevant documents. Vector space model became the baseline for many statistical techniques and probabilistic models and certain algorithms which are accounted for seeking information.

2 Latent Semantic Analysis

Latent Semantic Analysis(LSA) is a theory and method for **extracting** and **representing** the contextual-usage meaning of words by statistical computations applied to a large corpus of text. The underlying idea is that the aggregate of all the word contexts in which a given word does and does not appear provides a set of mutual constraints that largely determine the similarity of meaning of words and set of words to each other. It mimics the human word sorting and

category judgements. LSA represents the words used in it, and any set of these words- such as sentence, paragraph,or essay- either taken from corpus or new, as points in a very high dimensional semantic space(vector space of dimensions around 50 to 1500). LSA is based on singular value decomposition, a mathematical matrix decomposition technique.

`LSA and Latent Semantic Indexing are just classical variants of Vector Space Models(VSMs).`

A practical method for the characterisation of word meaning is that LSA produces *measures* of **word-word**, **word-passage**, **passage-passage** relations that are well correlated with several human cognitive phenomena involving association or semantic similarity[3]. LSA, as currently practiced , implements its representations of the meaning of words and passages from analysis of text alone. None of the knowledge comes from the physical world or by any means of feelings and intentions. LSA's knowledge is not perfect but it can offer a close approximation to peoples intentions and knowledge of the theories they wrote.

LSA in particular analyses the *relationship between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.* LSA assumes that words that are close in meaning will occur in similar pieces of text which is known as distributional hypothesis.

3 Vector Space Model of Text for LSA

The first step is to represent the text as matrix in which each **row** stands for a `unique words` and each **column** stands for a `text passage` or other context. Each `cell contains the frequency with which the word of its row` appears in the passage denoted by its column. Each cell frequency is weighted by a functions that expresses both the words importance in the particular passage and the degree to which the word type carries information. Next LSA applies singular value decomposition(SVD) to the matrix. This is a form of factor analysis. SVD is used to reduce the number of rows while preserving the similarity structure among columns. SVD is discussed in the following section in detail and also its applications.[4] [3]

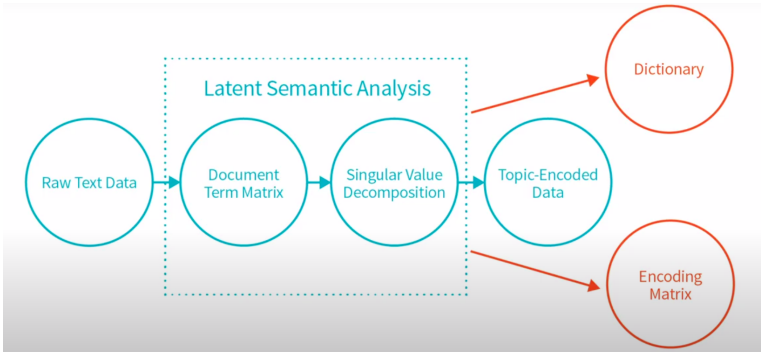


Figure 1: Basic model of Latent semantic analysis.

To fully understand the above we need to understand how the words are represented in vector space. Let us look at the basic model of Latent Semantic

analysis. Raw text is transformed into Document term matrix. On applying Singular value decomposition on the document term matrix we get the Encoding Matrix with frequencies as values of the words with reduced dimensionality(Usually important dimensions are preserved shifted left and the important words are passages are in a sense the document with higher numeric value is pushed to the top)[4]. The **Document Matrix** is `collection of words represented in the form of bag of words`.Normal methods of vector space representation uses **tf-idf** to show the relevance of a particular word but SVD uses a different approach to generate the numeric values. It is kind of similar to *Principal Component Analysis* since it also reduces the dimensions. We also get **dictionary** along with encoded matrix. Dictionary is *set of available words in all the documents altogether.*

Let us see the representation of a document in vector space. It is a point in Vector space with certain dimensions.The following image shows the document in vector space. As stated in the introduction document can be anything `sentence or passage or essay` .

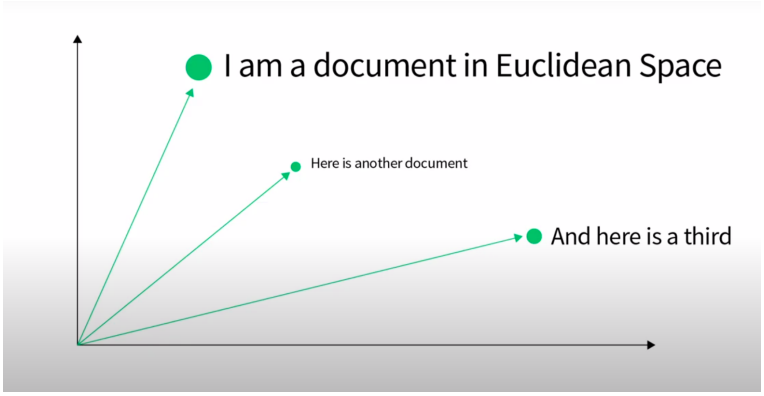
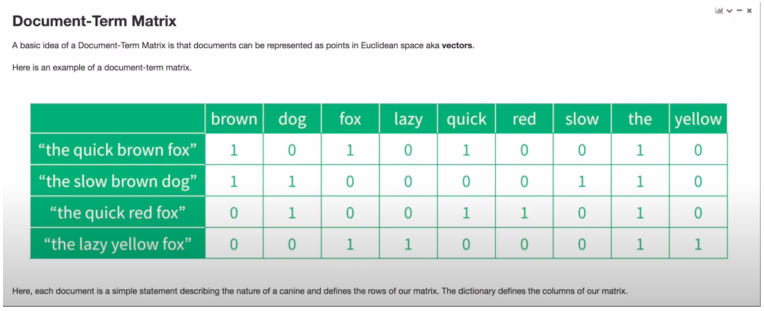


Figure 2: Document representation in vector space.

The document term matrix representation is shown below in figure 3. As stated each cell contains the frequency of the word in the document. `Usually bag of words technique is used`.



	brown	dog	fox	lazy	quick	red	slow	the	yellow
“the quick brown fox”	1	0	1	0	1	0	0	1	0
“the slow brown dog”	1	1	0	0	0	0	1	1	0
“the quick red fox”	0	1	0	0	1	1	0	1	0
“the lazy yellow fox”	0	0	1	1	0	0	0	1	1

Figure 3: Document Matrix representation.

4 Singular Value Decomposition

Singular Value Decomposition is a factorisation of a real or complex matrix that generalises the eigen decomposition of a square normal matrix to any $m \times n$ through an extension called polar decomposition.[5]

Let A be an $m \times n$ matrix with singular values $\lambda 1 \geq \lambda 2 \geq \cdots \geq \lambda n \geq 0$. Let r denote the number of nonzero singular values of A, or simply the rank of A.

Notation: A singular value decomposition is a factorization

$$A = U \sum V^T$$

where:

- U is an $m \times m$ orthogonal matrix.
- V is an $n \times n$ orthogonal matrix.
- $\sum$ is an $m \times n$ matrix whose i^{th} diagonal entry equals the i^{th} singular value λ_i for i=1,.....,r. All other entries of $\sum$ are zero.

The number of non zero values are equal to the rank of A. The columns of **U** and columns of **V** are called the **left-singular vectors** and **right-singular vectors** of **A**.

The SVD is not unique. It is always possible to choose the decomposition so that the singular values $\sum_{ii}$ are in descending order. The procedure to find the SVD is given in here Some important aspects of calculating SVD are:[5]

- The left-singular vectors of M are a set of ortho normal eigen vectors of AA*.
- The right-singular vectors of M are a set of ortho normal eigen vectors of A*A.
- The non-negative singular values of M (found on the diagonal entries of $\sum$ are the square roots of the non-negative eigenvalues of both A*A and AA*.

In simpler terms SVD is simply a rectangular matrix is decomposed into product of three other matrices as mentioned with certain properties.LSA using SVD efficiently reduce the rows while preserving the similarity of the structure among the columns. But still the dimensionality of the vector space is quite large enough to produce the meaningful approximation.[3] Thus finding the optimal dimensionality of the final representation is the important component. It is similar to Principal component analysis.

How do we calculate the orthogonal Matrices U and V^T : Since we assume the both are orthogonal they follow the property that $UU^T = I$ and similarly $VV^T = I$ Therefore Matrix $AA^T = (U \sum V^T)(V \sum^T U^T)$ and we know that since `V` is a orthogonal matrix the equation reduces to

$$AA^T = (U \sum \sum^T U^T)$$

Now the the singular values becomes squared as opposed to previous since

$$\sum^{T} \textit{ is same as } \sum$$

Let the matrix formed by AA^T be `M`. Then the eigen values of `M` are the square of the singular values and the eigen vectors are given by Orthogonal Matrix `U`.

$$\therefore M\mu = \lambda\mu$$

$$\textit{since we already have } \lambda \textit{ values we can substitute}$$

$$\textit{the values and obtain eigen vectors of } AA^T \textit{i.e,. } U$$

$$\textit{Similarly we can calculate the vector } V^T \textit{ for } Matrix A^T A$$

5 Example Of LSA:

The below is a small example that gives the flavour of the analysis and demonstrates what the technique accomplishes[3]. This example uses a text passage the titles of nine technical memorandum, five about human computer interaction, and four about mathematical graph theory. Both concepts are disjoint. Now, We will submit this **Document Matrix** to the SVD and in

Example of text data: Titles of Some Technical Memos

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

$\{X\} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Figure 4: Document Matrix representation of words from text corpus.

return it will give us Orthogonal matrices **W,P** and `Decomposed matrix` **X** which contains the numerical values associated with the words in the document.
First two dimensions in the figure 6 Part b were the most important dimensions. The choice of dimensions depends on the person.The higher positive values are important as the higher negative values.c2, c3, c4, c5 are examples of one category m1,m2,m3,m4 are of another category . As we can see the titles of c1,c2,c3... are similar so the numerical values are also similar and the sample applies for m1,m2,m3,m4 . The negative values clearly indicate that there is no Inter correlation among them.
If we take a word **human**, the word-word relation is closer to first category of words and the words **minor** is closer second category.

$$\{X\} = \{W\}\{S\}\{P\}'$$

$\{W\} =$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

$\{S\} =$

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

$\{P\} =$

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Figure 5: Orthogonal Matrices.

$\{\hat{x}\} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

(a) Matrix X

Correlations in two dimensional space:

c2	0.91							
c3	1.00	0.91						
c4	1.00	0.88	1.00					
c5	0.85	0.99	0.85	0.81				
m1	-0.85	-0.56	-0.85	-0.88	-0.45			
m2	-0.85	-0.56	-0.85	-0.88	-0.44	1.00		
m3	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	
m4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00

(b) Final SVD with two important dimensions

Figure 6: 2 Figures side by side

6 LSA In Information retrieval and Query similarity:

The way to express their commonalty is assume a scenario that a person is expressing his thoughts in terms of words which has certain semantic meaning. The text is then represented as query .Large corpus of text is represented as a document Matrix in the document database(Key is either title or some keywords of the document)[3]. Then it is subjected through SVD generating reduced dimensionality vector usually with top topics of 50 to 400 dimensions. A query is represented as **pseudo document** a weighted average of the vectors of the words it contains.(A document vector in the SVD solution is also a weighted average of the vectors of words it contains.)
Let Q be the mini query vector.The transformed vector or pseudo document vector Q^* is of low dimensional space and is describes as [2]

$$Q^* = \sum^{-1} U_r^T Q$$

$$\sum^{-1} \; is \; the \; inverse \; matrix \; of \; \sum$$

.
Similarity between Q and document d_i is given by cosine value between Q^* and the column vector $V^T(:, d_i)$

$$sim(Q, d_i) = cosine(Q^*, V^T(:, d_i)) = \frac{\sum_{t_j} Q^*(t_j) V^T(t_j, d_i)}{\sqrt{\sum_{t_j} (Q^*(t_j))^2} \sqrt{\sum_{t_j} (V^T(t_j, d_i))^2}}$$

7 Conclusion

References

[1] Sanjay Kumar Dwivedi Jitendra Nath Singh. *Analysis of Vector Space Model in Information Retrieval.* 2000.

[2] Wei Wang Mingxi Zhang, Pohan Li. *An index-based algorithm for fast online query processing of latent semantic analysis.*

[3] Darrell Laham Thomas K Landauer, Peter W. Foltz. *An Introduction to Latent Semantic Analysis.* 1998.

[4] wikipedia article. *Latent semantic Analysis.*

[5] wikipedia article. *Singular Value Decomposition.*