

# Matrices, Vector Spaces, and Information Retrieval

Steve Richards and Azuree Lovely

December 13, 2002

## Abstract

Classical methods of information storage and retrieval are inconsistent and lack the capability to handle the volume of information that comes with the advent of digital libraries and the internet. The goal of this paper is to show how linear algebra, in particular the vector space model, could be used to retrieve information more efficiently. Then the purpose of this paper is to outline the vector space model, to explain two methods of making the vector space model a more efficient system of information retrieval, and to introduce an example using the system.

## The need for Automated IR

In the past, documents were indexed by authors titles, abstracts, key words, and subject classifications. To retrieve any one of these documents involves searching through a card catalogue manually, which incorporates the opinions of the user. Then if an abstract or key word list were not provided, a professional indexer or cataloger could have written one incorporating personal opinion and thus inconsistency. But today,

- There are 60,000 new books printed annually in the United States.

The need for . . .

The Vector Space Model

An Example

Query comparison

Rank Reduction: . . .

Rank Reduction: . . .

Term-Term Comparison

Conclusion

Home Page

Title Page



Page 1 of 100

Go Back

Full Screen

Close

Quit

- The Library of Congress maintains a collection of more than 17 million books and receives 7000 new ones daily.
- There are currently 300 million web pages on the internet, with the average search engine acquiring pointers to about 10 million daily.

Automated IR can handle much larger databases without prejudice.

## The Vector Space Model

The vector space model begins with text documents. In order for each text document to be represented mathematically, each document is turned into a vector. A particular term associated with a given document is represented by a component in the vector for that document. Then, a database containing  $d$  documents and  $t$  terms is represented by a  $t \times d$  term-by-document matrix  $A$ . The  $d$  vectors representing the  $d$  documents are the columns of matrix  $A$ . The  $a_{ij}$  component of matrix  $A$  reflects the weighted frequency of the  $i^{th}$  term associated with the  $j^{th}$  document. Thus, the columns of matrix  $A$  are the document vectors and the rows of matrix  $A$  are the term vectors.

The purpose of using the vector space model is to find numerical and geometrical relationships between document vectors and a query vector in order to find the documents that are most relevant to the query. A query is a set of terms represented in a vector in much the same way as a document is represented. Query matching is finding those documents closest, geometrically, to the query vector in the vector space model. The document vectors closest to the query vector will represent the most relevant documents to the terms queried.

The need for . . .

The Vector Space Model

An Example

Query comparison

Rank Reduction: . . .

Rank Reduction: . . .

Term-Term Comparison

Conclusion

Home Page

Title Page

◀ ▶

◀ ▶

Page 2 of 100

Go Back

Full Screen

Close

Quit

The need for . . .
The Vector Space Model
<b>An Example</b>
Query comparison
Rank Reduction: . . .
Rank Reduction: . . .
Term-Term Comparison
Conclusion

[Home Page](#)

[Title Page](#)

◀▶

◀ ▶

Page 3 of 100

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## An Example

The document “The Chevy Automobile: a Mechanical Marvel” will be indexed by the terms “Chev(y, rolet)”, “Auto(mobile, motive)”, and “Mechanic(s, al)”, or more clearly, by the terms  $T_1$ ,  $T_2$ , and  $T_5$  below. The terms are identified by their stems, and any derivation of that root will be returned. This process is known as *Stemming*. Stemming is useful because it reduces the number of words that need to be maintained and, therefore; it decreases storage space. It might also be relevant to note here that very common, high frequency words such as *to* and *the* have been left off the list of example terms because they would hardly improve the term-by-document matrix. (A process known as *Stoplisting*). So, to return to the example, the above query vector would be

$$V = [1 \quad 1 \quad 0 \quad 0 \quad 1]^T.$$

Notice that non-zero entries are in the *1st*, *2nd*, and *5th* positions in the query vector. These positions correspond to the *1st*, *2nd*, and *5th* terms in the example term list. Also, the zero entries in the *3rd* and *4th* positions numerically show that the *3rd* and *4th* terms from the example term list are not contained in the example document. Graphically, this comparison would look like Figure 1, only in 5-space.

### Terms:

$T_1$  = auto(mobile, motive)

$T_2$  = Chev(y, rolet)

$T_3$  = Ford

$T_4$  = motor(s)

$T_5$  = mechanic(s, al)

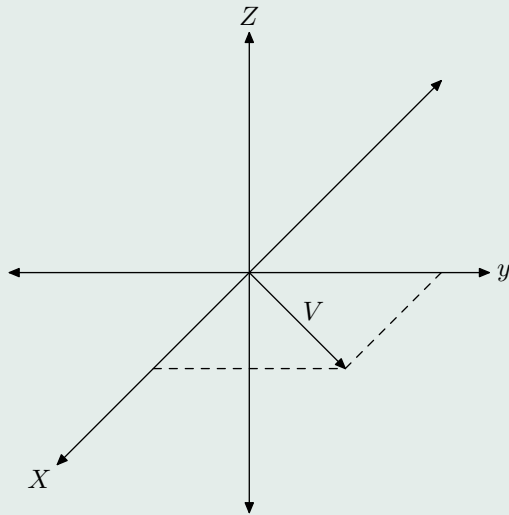


Figure 1: Query-vector comparison

*The need for . . .*

*The Vector Space Model*

*An Example*

*Query comparison*

*Rank Reduction: . . .*

*Rank Reduction: . . .*

*Term-Term Comparison*

*Conclusion*

*Home Page*

*Title Page*

◀◀ ▶▶

◀ ▶

*Page 4 of 100*

*Go Back*

*Full Screen*

*Close*

*Quit*

The need for . . .
The Vector Space Model
An Example
Query comparison
Rank Reduction: . . .
Rank Reduction: . . .
Term-Term Comparison
Conclusion

**Documents:**

- $D_1$  = A Generalization of the Automobile: A Mechanical Overview
- $D_2$  = Automobiles Inside and Out
- $D_3$  = The Ford Auto that rivaled Chevy's Chevelle
- $D_4$  = A Mechanical Comparison of the motors of Chevy and Ford.
- $D_5$  = A Mechanical Look at the motors in Chevy and Ford Automobiles

Now we describe our database by compiling the document vectors into the columns of a  $5 \times 5$  term-by-document matrix  $A$ .

$$A = \begin{pmatrix} T_1D_1 & T_1D_2 & T_1D_3 & T_1D_4 & T_1D_5 \\ T_2D_1 & T_2D_2 & T_2D_3 & T_2D_4 & T_2D_5 \\ T_3D_1 & T_3D_2 & T_3D_3 & T_3D_4 & T_3D_5 \\ T_4D_1 & T_4D_2 & T_4D_3 & T_4D_4 & T_4D_5 \\ T_5D_1 & T_5D_2 & T_5D_3 & T_5D_4 & T_5D_5 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

In order to weight each term in relevance to each document and also for query comparison, we normalize the matrix.

$$A = \begin{pmatrix} .7071 & 1 & .5774 & 0 & .4772 \\ 0 & 0 & .5774 & .5 & .4772 \\ 0 & 0 & .5774 & .5 & .4772 \\ 0 & 0 & 0 & .5 & .4772 \\ .7071 & 0 & 0 & .5 & .4772 \end{pmatrix} \tag{1}$$

## Query comparison

One way to compute the similarity between documents and query vectors is to find the cosine of the angle between them. The reason the cosine is useful is that it provides a

Home Page

Title Page

◀ ▶

◀ ▶

Page 5 of 100

Go Back

Full Screen

Close

Quit

numerical representation of the geometrical relationship. The cosine of zero is one. If the angle between two vectors were zero, the vectors would be parallel and, in other words, would be very similar. Therefore, the closer the value of the cosine between a document vector and a query vector gets to one, the higher the relevance is of that document to the query.

A query by a user will be represented as a vector in the same space. A user may query the database for Chevy motors, in which case the query vector would be  $q = [0 \ 1 \ 0 \ 1 \ 0]^T$ , which can also be normalized to

$$q = [0 \ .7071 \ 0 \ .7071 \ 0]^T.$$

The vectors in the database closest to that vector will be returned as relevant. This relevance is determined by the cosine of the angle between them,

$$\cos \theta = \frac{a_j^T q}{\|a_j\| \|q\|}, \quad (2)$$

where  $\|a_j^T\|$  is the Euclidian norm equal to  $\sqrt{a^T a}$ . This magnitude is equal to one because of the normalization of both matrix  $A$  and the query vector. Graphically this comparison would look like Figure 2, but in 5-space.

Once the cosines between the angles have been computed, a threshold must be set for the minimum acceptable value for  $\cos \theta$  of those documents returned to the user. As mentioned previously, as the minimum acceptable value for the cosine approaches one, the similarities between the vectors increase. A stringent minimum value might be a value of 0.9. However, ours being a small example, a minimum value of 0.5 will suffice.

The cosine of the angles between the document vectors in the example database and the example query vector are 0, 0, 0.4083, 0.7071, and 0.6324. This query, with the minimum acceptable cosine value set at 0.5 would return the fourth and fifth documents. Thus far, comparisons between terms and documents have just begun. What follows are

Home Page

Title Page

◀ ▶

◀ ▶

Page 6 of 100

Go Back

Full Screen

Close

Quit

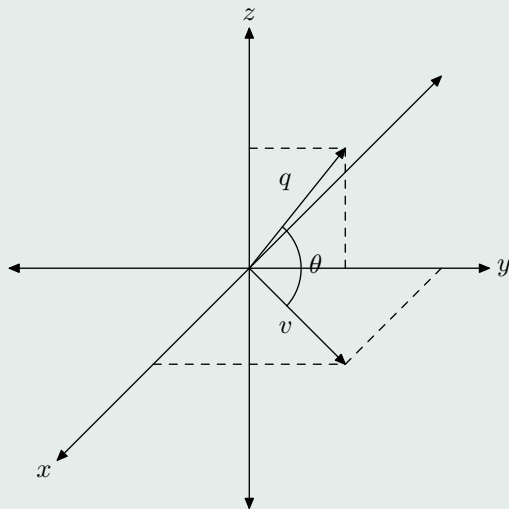


Figure 2: Query-vector comparison

- The need for . . .
- The Vector Space Model
- An Example
- Query comparison
- Rank Reduction: . . .
- Rank Reduction: . . .
- Term-Term Comparison
- Conclusion

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 7 of 100

Go Back

Full Screen

Close

Quit

- The need for . . .
- The Vector Space Model
- An Example
- Query comparison
- Rank Reduction: . . .
- Rank Reduction: . . .
- Term-Term Comparison
- Conclusion

two methods of information retrieval, that also use the vector space model, that should increase the expectancy of relevant document returns.

## Rank Reduction: Using $QR$ Factorization

To make our system more efficient in handling mass amounts of information, the first step is to remove excess information, contained in the column space of  $A$ , that adds no new insight to the database. We can do this by identifying and ignoring dependencies in the columns of  $A$ . Reducing the rank of our term-document matrix can accomplish this, and one method for doing so is  $QR$  Factorization. Where

$$\begin{aligned}
 A &= QR \\
 R &= t \times d \text{ upper triangular} \\
 Q &= t \times t \text{ orthogonal.}
 \end{aligned}$$

The relationship  $A = QR$  says that the columns of  $A$  are linear combinations of the columns of  $Q$ . Therefore the columns of  $Q$  form a basis for the column space of  $A$ .

Returning to our example, the factors would be

$$Q = \left( \begin{array}{cccc|c}
 .7071 & .7071 & 0 & 0 & 0 \\
 0 & 0 & .7071 & 0 & 0 \\
 0 & 0 & .7071 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 \\
 .7071 & -.7071 & 0 & 0 & 0
 \end{array} \right) \quad (3)$$

[Home Page](#)  
[Title Page](#)  
◀◀
▶▶  
◀
▶  
Page 8 of 100  
[Go Back](#)  
[Full Screen](#)  
Close  
Quit

and

$$R = \begin{pmatrix} 1 & .7071 & .4083 & .3536 & .6324 \\ 0 & .7071 & .4083 & -.3536 & 0 \\ 0 & 0 & .8166 & .7071 & .6324 \\ 0 & 0 & 0 & .5 & .4472 \\ \hline 0 & 0 & 0 & 0 & 0. \end{pmatrix}. \quad (4)$$

The block multiplication  $A = [Q_A \ Q_A^\perp] [R_A \ 0]^T$  specifies that  $Q_A^\perp$  does not contribute to the column space of  $A$ .  $Q_A r_j$  may be substituted into the cosine formula for query comparison.

$$\cos \theta_j = \frac{(Q_A r_j)^T q}{\|Q_A r_j\| \|q\|}$$

However,

$$\|Q_A r_j\| = \sqrt{(Q_A r_j)^T Q_A r_j} = \sqrt{r^T Q_A^T Q_A r_j} = \sqrt{r_j^T r_j} = \|r_j\|.$$

So,

$$\cos \theta_j = \frac{r_j^T (Q_A^T q)}{\|r_j\| \|q\|}. \quad (5)$$

The cosines of the angles between the example query vector using this new cosine formula are the same as before 0, 0, .4083, .7071, and .6324 proving that no important information was lost in the factorization.

To reduce the rank of  $R$ , we first block out the matrix  $R$ , as in Equation 6.

$$\left( \begin{array}{ccc|cc} 1 & .7071 & .4083 & .3536 & .6324 \\ 0 & .7071 & .4083 & -.3536 & 0 \\ 0 & 0 & .8166 & .7071 & .6324 \\ \hline 0 & 0 & 0 & .5 & .4472 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right) = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} = \hat{R} \quad (6)$$

Home Page

Title Page



Page 9 of 100

Go Back

Full Screen

Close

Quit

- The need for . . .
- The Vector Space Model
- An Example
- Query comparison
- Rank Reduction: . . .
- Rank Reduction: . . .
- Term-Term Comparison
- Conclusion

Because  $R_{22}$  is a very small part of matrix  $R$ , (about 30% as shown below), if  $R_{22}$  is set to the zero matrix, then the rank of the matrix  $R$  will decrease from 4 to 3 as will the rank of matrix  $A$  by extension.

$$\frac{\|R_{22}\|}{\|R\|} = .3000$$

Thus, because setting  $R_{22}$  equal to zero only produces a 30% change in matrix  $R$  and therefore also in the matrix  $A$ , the new reduced rank matrix  $\hat{R}$  could be a good approximation to  $R$ .

By  $\hat{A} = Q\hat{R}$ , the new matrix  $\hat{A}$  is

$$\hat{A} = \begin{pmatrix} .7071 & 1 & .5774 & 0 & .4472 \\ 0 & 0 & .5774 & .5 & .4472 \\ 0 & 0 & .5774 & .5 & .4472 \\ .7071 & 0 & 0 & .5 & .4472. \end{pmatrix} \quad (7)$$

Calculating  $\cos\theta$  between the query and the new matrix  $\hat{A}$  returns values of 0, 0, 0.4083, 0.4083, and 0.3953. Therefore the change in  $A$  was too large. Sometimes this may be the case, which is why we need to find a better means of obtaining a low rank approximation to matrix  $A$ .

## Rank Reduction: Using Singular Value Decomposition

$QR$  Factorization identifies dependencies in the column space of matrix  $A$ , removing excess information from the system. However, dependencies in the row space must also be addressed. SVD is one method used for identifying and removing those dependencies, and also for producing a low rank approximation to  $A$ . SVD can also be used to compare terms to terms in the database thus dealing with problems such as synonymy: multiple

Home Page

Title Page

◀ ▶

◀ ▶

Page 10 of 100

Go Back

Full Screen

Close

Quit

The need for . . .
The Vector Space Model
An Example
Query comparison
Rank Reduction: . . .
<b>Rank Reduction: . . .</b>
Term-Term Comparison
Conclusion

words with the same definition, and polysemy: words with multiple definitions. Term-term comparison shall be addressed shortly. For now, it can be seen that the singular value decomposition factors matrix  $A$  into

$$\begin{aligned}
 A &= U\Sigma V^T \\
 U &= t \times t \text{ orthogonal} \\
 \Sigma &= t \times d \text{ diagonal} \\
 V &= d \times d \text{ orthogonal.}
 \end{aligned}$$

The matrix  $U$  contains the column space of  $A$ , the matrix  $V$  contains the row space of  $A$ , and  $\Sigma$  contains the singular values of matrix  $A$ .

We can now reduce the rank of  $A$  to  $A_k = U_k \Sigma_k V_k^T$  by setting all but the  $k$  largest singular values of  $A$  equal to zero. Returning to our previous example,

$$\Sigma = \left( \begin{array}{ccc|cc}
 1.7873 & 0 & 0 & 0 & 0 \\
 0 & 1.0925 & 0 & 0 & 0 \\
 0 & 0 & .7276 & 0 & 0 \\
 \hline
 0 & 0 & 0 & .2874 & 0 \\
 0 & 0 & 0 & 0 & 0
 \end{array} \right) = \begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{pmatrix} = \Sigma_k. \quad (8)$$

Setting  $\Sigma_{22}$  and ,therefore, the lowest singular values of  $A$  equal to zero produces only a 13% change in  $A$ .

$$\frac{\|\Sigma_{22}\|}{\|\Sigma\|} = \frac{\|A_k\|}{\|A\|} = .1256$$

Comparing this to the 30% change in  $A$  produced by  $QR$  Factorization, it can be

Home Page

Title Page

◀ ▶

◀ ▶

Page 11 of 100

Go Back

Full Screen

Close

Quit

The need for . . .
The Vector Space Model
An Example
Query comparison
Rank Reduction: . . .
Rank Reduction: . . .
<b>Term-Term Comparison</b>
Conclusion

seen that SVD is a better approximation to  $A$ . Returning to our example,

$$\hat{A} = \begin{pmatrix} .7293 & .9761 & .6013 & -.0070 & .4302 \\ -.0303 & .0326 & .5447 & .5096 & .4704 \\ -.0303 & .0326 & .5447 & .5096 & .4704 \\ .1250 & -.1346 & .1349 & .4603 & .3515 \\ .6558 & .0552 & -.0553 & .5163 & .4865. \end{pmatrix}. \quad (9)$$

The new cosine formula becomes

$$\cos \theta_j = \frac{A_{kj}^T q}{\|A_{kj}\| \|q\|}. \quad (10)$$

The cosines of the angles between the example query vector and this new approximation to  $A$  are 0.1098,  $-0.0721$ , 0.4805, 0.6858, and 0.5811. Since the fourth and fifth documents are returned we have a successful reduced rank approximation to  $A$ . Also, it should be noted here that although the new cosines for the first three comparisons still do not retrieve documents in this small example, the new cosines found with SVD would hold more importance with respect to document retrieval as the database being queried grew larger and larger because rather than just an answer of zero for several comparisons, some comparisons would begin to stand out more than others.

## Term-Term Comparison

Now that the singular value decomposition is a tool in the tool box, so to speak, methods for information retrieval become even more advanced when terms are compared to terms. Recall that  $QR$  factorization permits only comparisons of terms with documents. That was improved on with the SVD because SVD allows not only the comparisons between terms and documents, but also comparisons between terms. This next step using the SVD follows because SVD provides information about the row space of matrix  $A$ .

Home Page

Title Page



Page 12 of 100

Go Back

Full Screen

Close

Quit

<a href="#">The need for . . .</a>
<a href="#">The Vector Space Model</a>
<a href="#">An Example</a>
<a href="#">Query comparison</a>
<a href="#">Rank Reduction: . . .</a>
<a href="#">Rank Reduction: . . .</a>
<b><a href="#">Term-Term Comparison</a></b>
<a href="#">Conclusion</a>

Term-term comparisons are powerful because, as mentioned briefly under SVD, the comparisons can fight problems such as synonymy and polysemy. In order to better make this point, an example is in order. This example shall address the problem of polysemy: a single word with multiple meanings.

To begin the example, another set of terms and documents must be set up.

**The  $t = 7$  terms:**

- $T_1 = \text{Ford(ing)}$
- $T_2 = \text{auto(mobile, motive)}$
- $T_3 = \text{mechanic(s, al)}$
- $T_4 = \text{engine(s)}$
- $T_5 = \text{President}$
- $T_6 = \text{Gerald}$
- $T_7 = \text{river(s)}$

**The  $d = 5$  document titles:**

- $D_1 = \text{The Mechanical Simplicity of the Engine in a Ford Automobile}$
- $D_2 = \text{Fording Rivers on the Oregon Trail}$
- $D_3 = \text{How to Improve the Horsepower of Your Ford Engine}$
- $D_4 = \text{The Biography of President Gerald R. Ford}$
- $D_5 = \text{Preventative Measures that will Keep Your Ford Engine in Good Condition}$

Notice that the titles of the documents all contain the single polysemous query term *Ford*. Notice also that the titles reflect three of the ways the key word *Ford* might be used. Term-term comparisons can help a user to focus only on the documents that use the meaning of the key word that was intended.

Following the same process as was used earlier in the paper, the new  $7 \times 5$  term-by-

[Home Page](#)

[Title Page](#)

◀▶

◀▶

Page 13 of 100

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

The need for . . .
The Vector Space Model
An Example
Query comparison
Rank Reduction: . . .
Rank Reduction: . . .
<b>Term-Term Comparison</b>
Conclusion

[Home Page](#)

[Title Page](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

Page 14 of 100

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

document matrix  $G$  with unit columns becomes

$$G = \begin{pmatrix} 0.5 & 0.7071 & 0.7071 & 0.5774 & 0.7071 \\ 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.7071 & 0 & 0.7071 \\ 0 & 0 & 0 & 0.5774 & 0 \\ 0 & 0 & 0 & 0.5774 & 0 \\ 0 & 0.7071 & 0 & 0 & 0 \end{pmatrix}. \quad (11)$$

The next step is to compute the cosines of the angles  $w_{ij}$  between all pairs of term vectors  $i$  and  $j$  such that

$$\cos w_{ij} = \frac{(e_i^T G)(G^T e_j)}{\|G^T e_i\| \|G^T e_j\|}, \quad (12)$$

where  $e_i$  denotes the  $i^{\text{th}}$  canonical vector of dimension  $t$ , or in other words, the  $i^{\text{th}}$  column of the  $t \times t$  identity matrix.

The cosines are listed in the matrix  $C$  where  $C_{ij} = \cos w_{ij}$ .

$$C = \begin{pmatrix} 1 & 0.3464 & 0.3464 & 0.7746 & 0.4 & 0.4 & 0.4899 \\ & 1 & 1 & 0.4472 & 0 & 0 & 0 \\ & & 1 & 0.4472 & 0 & 0 & 0 \\ & & & 1 & 0 & 0 & 0 \\ & & & & 1 & 1 & 0 \\ & & & & & 1 & 0 \\ & & & & & & 1 \end{pmatrix} \quad (13)$$

In order to better present the relationships between the term vectors, only the entries in the top half of the symmetric matrix  $C$  are shown.

Notice how the vectors in matrix  $C$  divide into three geometrically different groups that correspond to the different usages of the key word *Ford*. The first group corresponds

The need for . . .
The Vector Space Model
An Example
Query comparison
Rank Reduction: . . .
Rank Reduction: . . .
Term-Term Comparison
Conclusion

to terms two through four: auto, mechanic, and engine. The second group corresponds to terms five and six: President and Gerald. The third and last group corresponds to only the seventh term: river.

The matrix  $C$  has been partitioned to help show the three different geometric groupings. Now that the different usages have been separated, a user could be prompted to choose the correct meaning of the word they were looking for.

This has been an example of a process called *Clustering*: the grouping of terms according to related content. The implementation of Clustering through term-term comparisons greatly advances methods of information retrieval because information excess can be greatly depleted.

## Conclusion

We have seen how we can apply a vector space model to terms and documents in a database. We've seen how  $QR$  Factorization removed dependencies in the column space of a matrix, but could not, in this case, reduce the rank of matrix  $A$  without losing information important to the database. SVD, on the other hand, not only removed the dependencies in the column space and also from the row space of matrix  $A$ , thereby successfully reducing the rank of the original matrix  $A$ , but it also allowed term-term comparisons due to the information given by this method about the row space of the matrix  $A$ . With these new tools, the vector space model can be used effectively and can efficiently retrieve information.

## References

- [1] Arnold, Dave *Consultation*
- [2] Berry, Michael W. *Matrices, Vector Spaces, and Information Retrieval*

[Home Page](#)

[Title Page](#)



Page 15 of 100

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

[3] Berry, Michael W. *Understanding Search Engines*

[4] Strang, Gilbert *Introduction to Linear Algebra*

[The need for . . .](#)

[The Vector Space Model](#)

[An Example](#)

[Query comparison](#)

[Rank Reduction: . . .](#)

[Rank Reduction: . . .](#)

[Term-Term Comparison](#)

[Conclusion](#)

[Home Page](#)

[Title Page](#)



Page 16 of 100

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)