Improvements in Latent Semantic Analysis

Ryan Lloyd and Cheri Shakiban
Department of Mathematics
University of St. Thomas
St. Paul, Minnesota 55105-1079 USA

Received: April 17, 2004 Accepted: June 28, 2004

ABSTRACT

This paper proposes and examines modifications for the method of Latent Semantic Analysis (LSA). Several new local and global weight functions, along with normalization routines, are disclosed. Changes in the general structure of LSA are discussed. An application of LSA, in which the method is used to filter advertisements in e-mail, proves the worthiness of the advancements.

I. INTRODUCTION

Latent Semantic Analysis (or simply LSA) is a statistical method that describes the underlying structure of a text in statistical terms [1-8]. Knowing the basics of LSA is essential to understand the proposed modifications to the process. The core component of Latent Semantic Analysis is a two-dimensional matrix called a term-document matrix (or a TDM). The (i, j) cell of this matrix represents the frequency (an integer) of term i in document j. These matrices can become extremely large. To reduce the noise, we can exclude the one hundred or one thousand most commonly used words in the language of study. Words that only occur once or twice in all the documents are also removed.

We traditionally normalize the matrix by dividing each term frequency (the initial value in cell (i, j)) by the word count for document j. The normalized counts are inserted into the TDM over the original frequencies. We then apply a transformation that weighs each cell according to the term's importance in the document. The weighting step consists of the product of a local and a global weight function (abbreviated as (LWF, GWF)).

We let W(i, j) = L(i, j)·G(i), where i and j are the term and document addresses, respectively, in the TDM, L is the local weight function, G is the global weight function, and W is the weighted matrix that is being formed. The most common equations for L(i, j) and G(i) are given by Nakov et al., [4]. They prefer to have LWF equal 0 or 1. If LWF = 0, they let L(i, j) = tf(i, j), and if LWF = 1, they let L(i, j) = log2(tf(i,j)), where tf(i, j) indicates the term frequency of word i in document j. Therefore, they have to calculate logarithms of various cells. However, if a cell is zero, this will result in an error. The researchers also only allow the values of GWF to range from 0 to 5. These six global weight functions are listed below. The trivial case involves 0, where the cell in the new matrix is equivalent to the value provided by the local weight function.

$$GWF = 0 \rightarrow G(i) = 1$$

$$GWF = 1 \rightarrow G(i) = \frac{1}{\sqrt{\sum_{j=1}^{ndocs} (L(i,j))^2}}$$

$$GWF = 2 \rightarrow G(i) = gf(i) / df(i)$$

$$GWF = 3 \rightarrow G(i) = 1 + \log_2 (ndocs / df(i))$$

$$GWF = 4 \rightarrow G(i) = -\sum_{j=1}^{ndocs} \{p(i,j) \cdot \log_2 (p(i,j) + 1)\}$$

$$GWF = 5 \rightarrow G(i) = 1 + \frac{\sum_{j=1}^{ndocs} \{p(i,j) \cdot \log_2 (p(i,j) + 1)\}}{\log_2 (ndocs)}$$

Here, df(i) is the number of documents in which term i appears, and *ndocs* is

equivalent to the total number of documents used in the matrix. Lastly, p(i, j) is the conditional probability of document j, which equals tf(i, j)/gf(i), where gf(i) is the global frequency of term i, or the number of times that the word appears in all the documents under consideration.

II. NEW LOCAL WEIGHT FUNCTIONS (LWF)

In this paper, we will modify L(i, j) for LWF = 1 to be log2(tf(i,j) +1) to prevent the logarithm of zero from entering our calculation. Further, we will introduce new local weight functions as follows.

$$LWF = 2 \rightarrow L(i,j) = tf(i,j)^2$$

$$LWF = 3 \rightarrow L(i,j) = tf(i,j)^3$$

$$LWF = 4 \rightarrow L(i,j) = \sqrt{tf(i,j)}$$

$$LWF = 5 \rightarrow L(i,j) = \log_{(3/2)}\{tf(i,j)+1\}$$

The transformations corresponding to LWF = 2 and LWF = 3 allow the occurrence rates to grow exponentially. Previous local weight functions weigh the cells linearly (LWF = 0) or force the elements to increase less from frequency to frequency (LWF = 1). With LWF = 1, nearly all cells have the same values. For instance, an initial frequency of 10 translates to 3.4594 with LWF = 1, while the occurrence rate of 30 converts to just 4.9542. The word that is found 30 times in a document is clearly more important to the classification of the passage than the term with a frequency of 10, but the transformation does not reveal the difference. Rather than decreasing the importance of frequent words, the first two transformations listed above give greater significance to the words that are most common in the documents.

III. NEW GLOBAL WEIGHT FUNCTIONS (GWF)

Besides the local weight functions introduced earlier, we also suggest three new global weight functions:

$$GWF = 6 \rightarrow G(i) = df(i) / ndocs$$

$$GWF = 7 \rightarrow G(i) = \frac{\sum_{j=1}^{ndocs} L(i,j)}{ndocs}$$

$$GWF = 8 \rightarrow G(i) = 1 + \frac{\sum_{j=1}^{ndocs} \{p(i,j) \cdot \log_2(p(i,j)+1)\}}{ndocs}$$

The equation corresponding to GWF = 6 is equivalent to the number of documents that contain the i^{th} word divided by the total amount of documents used in the study. The formula gives a ratio of how important the word of interest is to the documents considered. The output of this equation should always be between zero and one, exclusive. All the terms listed must occur in at least one document, and none of the indexed terms should occur in all the documents provided.

The next equation computes the average frequency of the i^{th} word globally. Rather than using the original frequency values, this transformation relies on the output from a local weight function. If a word exists in a number of documents, and the frequencies of the word are large, then this function will provide a sizeable value for G(i). This equation is able to identify the keywords in a set of documents.

The global weight function denoted GWF = 8 is sometimes referred to as an entropy transformation (not to be confused with other entropy terms used in other fields). The numerator of the quotient matches that of GWF=5, but the denominator is simply the number of documents rather than the diminishing logarithm of the quantity of passages.

IV. NORMALIZATION EQUATIONS

A normalization transformation applied to the original TDM is an essential stage in determining the importance of a word in a document. The transformation weighs frequency cells according to how large their parenting documents are. There are a range of functions that can normalize the elements in the original matrix.

The first normalization equation (NORM = 1) reads as follows: B(i, j) = A(i, j)/nwords, where A is the TDM. Additionally, nwords equals the quantity of words in document j. Another normalization function (NORM = 2) is: B(i, j) = A(i, j)/(highest

frequency in A). This equation can be visualized as a line in a graph with the x-axis holding the original frequency amounts and the vertical scale containing the new frequencies. The line begins at the origin; if the initial cell equals zero, then the new element is also zero. The line extends to the point (highest frequency in A, 1) so that the word that is most common in the document receives a frequency value of one. NORM = 2 does not permit the sum of the elements in a column to always equal one. The third normalization routine (NORM = 3) assigns zero to a cell if the word never occurs, 0.5 if the word occurs rarely, and 1 when the word has an exceptionally high rate of occurrence. With NORM = 3, a word might receive the value of 0.5 if it consumes between zero and two percent of a document, and the term could be given 1 if more than two percent of the text is composed of the word. (For NORM = 0, B(i, j) = A(i, j).) All these equations ensure that the cells have values between zero and one and that words existing in short documents are not under represented.

V. SINGULAR VALUE DECOMPOSITION

The initial matrix and the matrix created after applying a transformation are usually littered with zeros and other low-frequency values that act as noise. The TDM is nearly upper-triangular, and an overwhelming majority of cells above the main diagonal are zero. If the researcher attempts to remove too much useless data, then he/she will also erase descriptive data and weaken the procedure's accuracy. Leaving too much noise in the frequency tables will also result in a loss of precision. A procedure must exist to reduce the noise in the frequency matrix while retaining precision.

The Singular value decomposition (or SVD) is used to compress the semantic space. Once the singular value decomposition is applied to the matrix of transformed elements (obtained from the previous steps), one must choose the correct dimensionality of the central, diagonal matrix. Previous research suggests that the optimal number of singular values to retain is between 50 and 400 [5]. The first step in reducing the size of the middle matrix involves plotting the singular values (the vertical axis) against the indices for the singular values (the horizontal axis). The singular values must be plotted in descending order with the singular value at index one being the largest. The index of k at which the singular values level off and do not sink considerably from index to index reveals that k singular values must be retained. All the singular values after the k^{th} are set to zero. The later singular values are deemed too small to hold important information. When matrix U, the modified version of S, and matrix V are multiplied together, the matrix of best fit to the original frequency table is produced. The new, quieter matrix will have few zeros and will hold the vital information that describes the documents

The last step of LSA involves forming a correlation matrix, C, that describes how similar each document is to the other documents. C(i,j) gives the correlation between documents i and j with $0 \le C(i,j) \le 1$. The equation used to compute the correlation is $\cos(\square) = (a \cdot b) / (|a| * |b|)$. A quantity near one suggests that two texts are similar in word choice. Each C(i,j) is calculated by the cosine of the angle between two column vectors appearing in the matrix produced by SVD. Only the upper-triangular portion of the correlation matrix needs to be developed.

VI. AN EXAMPLE

The following example illustrates the steps of LSA. Suppose that the documents of interest are four short advertisements, which are listed here:

Advertisement #1: "Buy bargain goods here."

Advertisement #2: "Buy imported deals."

Advertisement #3: "Find best goods, best deals here."

Advertisement #4: "Imported goods are bargain. Are best goods bargain?"

The small term-document matrix below is for the ads. The documents contain nine unique words, and the number in each cell discloses the frequency of a word within a particular document.

	Ad #1	Ad #2	Ad #3	Ad #4
buy	1	1	0	0
bargain	1	0	0	2
goods	1	0	1	2
here	1	0	1	0
imported	0	1	0	1
deals	0	1	1	0
find	0	0	1	0
best	0	0	2	1
are	0	0	0	2

Next, the term-document matrix is normalized. Each cell in the frequency matrix receives a value between zero and one. The value of an element roughly equals the percent of the document that is composed of the word. The normalized TDM becomes

	Ad #1	Ad #2	Ad #3	Ad #4
buy	0.25	0.33	0	0
bargain	0.25	0	0	0.25
goods	0.25	0	0.17	0.25
here	0.25	0	0.17	0
imported	0	0.33	0	0.13
deals	0	0.33	0.17	0
find	0	0	0.17	0
best	0	0	0.33	0.13
are	0	0	0	0.25

After we apply the local and global transformation (LWF,GWF)= (0,2) on the normalized TDM, we get

	Ad #1	Ad #2	Ad #3	Ad #4
buy	0.25	0.33	0	0
bargain	0.38	0	0	0.38
goods	0.33	0	0.23	0.33
here	0.25	0	0.17	0
imported	0	0.33	0	0.13
deals	0	0.33	0.17	0
find	0	0	0.17	0
best	0	0	0.50	0.20
are	0	0	0	0.50

Once singular value decomposition is applied to the matrix of transformed elements and de-noising takes place, a correlation matrix is formed. The upper-right part of the matrix can then be reflected over the main diagonal to form the full correlation chart.

As an example, the correlation between advertisements #1 and #2 will be found. Recall that the columns for these two documents are:

	Advertisement #1	Advertisement #2
buy	0.25	0.33
bargain	0.38	0
goods	0.33	0
here	0.25	0
imported	0	0.33
deals	0	0.33
find	0	0
best	0	0
are	0	0

$$correlation = \cos(\theta) = \frac{a.b}{|a||b|} = \frac{0.0825}{(0.6151)(0.5716)} = 0.2346$$

As we can see, the correlation between the first and second ads is rather weak at 0.2346.

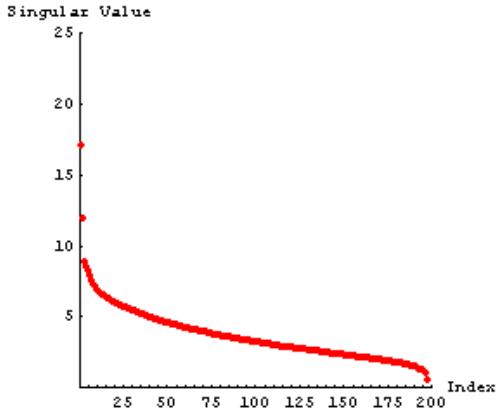
VII. AN APPLICATION

To discover whether or not Latent Semantic Analysis could be used to defeat SPAM, an experiment using 260 e-mails was performed. SPAM is the junk mail that accumulates each day in the inboxes of most e-mail users. 130 of the messages were legitimate, while the other 130 e-mails were illegitimate. In addition, 40 new e-mails were used as the test documents. The first step of analysis involved the construction of the TDM for the 260 e-mails. The TDM was saturated with zeros and had the characteristic upper-triangular form. A Java program created the frequency matrix with the help of efficient vectors.

The optimum transformation combination featuring NORM = 3, LWF = 2, and GWF = 8 was applied. Previous logical and empirical testing concluded that this combination best reveals the similarities between like documents and highlights the dissimilarities between texts that are truly different. Singular value decomposition was also utilized to eliminate noise from the term-document matrix. The rank of the central matrix remained at 75 for the SVD.

The plots of some of the singular values (the vertical axis) versus their ranks

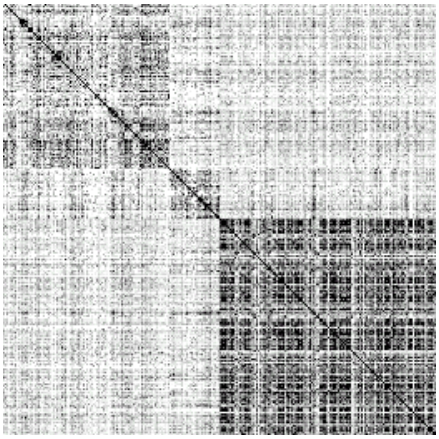
(the horizontal axis) appear below. A simple Mathematica program read in the transformed elements from text files and then outputted the various stages of the term-document matrices and the graphs of singular values.



For this particular transformation, the points begin with a singular value of about 17 but quickly drop. However, the points never seem to level off. They decrease in a linear fashion from the index of 50 to 190. The singular values then dive towards zero. Since the singular values do not change drastically after the index value of 75, the rank of the central matrix resulting from SVD was still around 75.

The smallest singular values (those above the 75th index) were all set to zero in the diagonal matrix. Multiplying the original matrix U, the simplified form of S, and the initial version of V then produced the matrix of best fit.

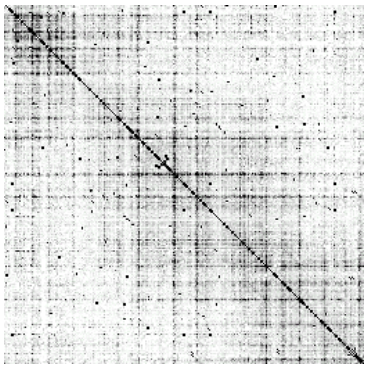
The correlation matrices for the transformations are displayed next. A Java application built the matrices and stored them to text files. To accelerate the process of building the correlation tables, only the upper-triangular portions of the matrices were calculated. A program written in OpenGL processed the correlation matrices and saved a bitmap picture file for each matrix. The gray level of a pixel in an image represents the correlation between two documents. Darker areas are indicative of tighter relationships between e-mails. Some researchers use four or five shades of gray when constructing a correlation matrix. Each gray level is associated with a certain correlation interval, with white representing relations that are under 0.5. These traditional intervals do not accurately reveal the correlations between documents. In this project, almost every possible correlation value received its own shade of gray with the help of the following equation: (intensity of white) = (1 – correlation value). Thus, if two e-mails are perfectly related, the correlation value shall be 1, and the intensity of white light will be zero (the pixel will be black).



Three dark squares appear in this matrix. All the visual correlation matrices take on the same checkerboard pattern. The dark-colored boxes indicate that moderately strong correlations exist among all the e-mails in the same class. The black area in the lower-right corner means that the junk messages are highly similar. The remaining lighter-shaded parts signify that good and bad electronic communications are weakly correlated. The blocks are naturally occurring and would not be evident if the classes of messages were mixed. Observe that the points along the main diagonals are solid black due to the flawless relationship between a message and itself.

The smaller square in the graph's center belongs to the category of good e-mails. The first 100 good messages came from the researcher's inbox, while the last 30 good messages were harvested from an associate's inbox. The central square shows the strong correlation among these 30 good e-mails, which are not highly similar to messages in the other two groups. Building a TDM from e-mails is a personalized process. Fortunately, the second subclass of good messages is weakly associated with the advertisements.

The correlation matrix produced after applying SVD is displayed next. Closed categories within the plot are difficult to identify. The lower-right and upper-left regions are still dark, but the darkness is spread out along the main diagonal and grows lighter as the points move away from the main diagonal.



The computer was able to correctly label 15 of the 20 personal messages and all 20 of the advertisements in the test set. Thus, the final precision of the e-mail filtering algorithm was 87.5%.

Current e-mail filters utilizing LSA require about 1,000 messages to sufficiently learn about SPAM. Any artificially intelligent e-mail filter must be taught the difference between good and bad messages by humans. In the filtering application, a message enters the e-mail application and the user decides whether it is good or bad. The computer then adds that message to an enormous term-document matrix. Ideally, the computer should have to read two or three good and bad e-mails to accurately sort the mail in the future. The filtering system could improve if it has access to a pre-built TDM filled with words that commonly occur in SPAM and personal letters. Giving the computer previous data on e-mails will speed up the learning process and make the judgment of new messages more precise.

Instead of using a single TDM to hold the data on the good and bad e-mails, two term-document matrices could be employed. One matrix is for the positive messages, while the second table is reserved for the negative communications. First, the new e-mail in question is temporarily placed in the table for the quality messages. Its average correlation with the other good emails is calculated. The message is then provisionally inserted into the matrix for the bad e-mails, and its average relationship with the known advertisements is determined. The highest average correlation indicates which group the new communication belongs in. The information in the e-mail is permanently added to the corresponding term-document matrix, enabling the computer to better identify SPAM.

Although Latent Semantic Analysis is not the most efficient way of filtering SPAM e-mails, this example shows that LSA has potential in the field of artificial intelligence and could become the language component of an artificially intelligent computer that is learning how to communicate.

REFERENCES

1. T. Hofmann, “Probabilistic Latent Semantic Indexing”, *Proceedings of the 22nd Annual International SIGIR Conference on Research and Development in Information Retrieval*, (University of California Press, San Diego, 1999).
2. T.K. Landauer, L. Darrell and P. Foltz. “Learning Human-like Knowledge by Singular Value Decomposition: A Progress Report,” (University of Colorado Press, Boulder, 1998).
3. J.I. Maletic., and N. Valluri. “Automatic Software Clustering Via Latent Semantic Analysis,” (University of Memphis Press, Memphis, 1999).
4. P.I. Nakov, A. Popova, and P. Mateev. “Weight Functions Impact on LSA Performance,” (Sofia University Press, Sofia, 2001).
5. P.I. Nakov, “Latent Semantic Analysis of Textual Data,” (Sofia University Press, Sofia. 2000).
6. Terzieva, Senia Petrova, P.I. Nakov, and S. Handjieva. “Investigating the Degree of Adequacy of the Relations in the Concept Structure of Students Using the Method of Latent Semantic Analysis,” (Bulgarian Computer Science Conference, Sofia, 2001).
7. R. Zhang, and A.I. Rudnicky. “Improve Latent Semantic Analysis Based Language Model by Integrating Multiple Level Knowledge,” (Carnegie Mellon University Press, Pittsburgh, 1998).