Analysis of the values in the LSI Term-Term Matrix

William Mill
Ursinus College
Math and Computer Science department
PO Box 1000
Collegeville, PA 19426

April Kontostathis
Ursinus College
Math and Computer Science department
PO Box 1000
Collegeville, PA 19426

ABSTRACT
Singular value decomposition (SVD), the process at the heart of Latent Semantic Indexing (LSI), is a computationally expensive procedure. In this paper we analyze the relationship between higher order term cooccurrence and the values produced by the LSI process. We show a strong correlation between the number of cooccurrence paths and the value produced in the LSI term-term matrix.

1. INTRODUCTION

Latent Semantic Indexing (LSI) [D90] is a search technique which has been applied to many information retrieval applications [D93] [D94] [D95] [S98] [ZH01]. It is helpful in avoiding the problems of synonymy and polysemy, which are so thorny in traditional vector-space retrieval.

One use of LSI is to create a matrix which assigns a similarity value to pairs of words. Even though a term pair may not appear in the same document together, they will frequently get a high LSI similarity value. This matrix is called the LSI term-term matrix.

In a paper presented at the Mathematical Foundations in Information Retrieval workshop in 2003, we proved that LSI incorporates term co-occurrence information [K03], and demonstrated a strong correlation between the LSI term-term matrix and LSI performance. In the current work, we extend these results, providing more detailed analyses of values in the LSI term-term matrix and their relationship with term co-occurrence.

2. LSI TERM-TERM MATRIX

The LSI term-term matrix is computed from the term-document matrix of a body of text. For the purposes of this paper, the term-document matrix is simply a matrix with all the terms in a collection on one axis and all the documents on the other. If a term *i* appears one or more times in a document *j*, then the value of the i,j^{th} element of the term-document matrix is 1. Otherwise, it is 0. Furthermore, the value of all the diagonal elements of the matrix is set to 0.

The first step in computing the LSI term-term matrix is to perform singular value decomposition on the term-document matrix. The decomposition of the term-

document matrix yields three matrices, T, S, and D, such that the value of TSD^T is the original term-document matrix. [D90]

The three resultant matrices are then truncated to k dimensions, where k is smaller than the rank of the term-document matrix. The value of the parameter k must be determined through trial and error, and is empirically based on retrieval performance. The LSI term-term matrix is equivalent to $T^kS^k(T^kS^k)^T$, where T^k and S^k are the T and S matrices after truncation to k values. [D90] The resultant matrix will have all the terms in the collection on both axes, and the i,j^{th} entry may be interpreted as the similarity of term *i* to term *j*.

3. COOCCURRENCE MATRICES

If a document contains both the terms *chip* and *wafer*, these two terms are said to cooccur, or to have a first order cooccurrence. Now assume that the terms *chip* and *silicon* cooccur in some document *a*, and that *wafer* and *silicon* cooccur in some other document *b*. The terms *chip* and *wafer* are then said to have a second order cooccurrence via the word *silicon*. If two terms have an *n* order cooccurrence, they are guaranteed to have an *n+1* order cooccurrence.

The cooccurrence matrices have all the terms in the collection on both axes. In the experimental results that follow, if a word *i* and a word *j* cooccur one or more times, then the first order cooccurrence matrix contains a 1 in the i,j^{th} element. Similarly, if *i* and *j* have any second order cooccurrences, the value of the i,j^{th} element of the second order cooccurrence matrix is 1. The value of all the diagonal elements of both these matrices is set to zero, since a term always cooccurs with itself.

The first order cooccurrence matrix may be calculated by multiplying the term-document matrix by its inverse, reducing its values to binary, and setting its diagonals to 0. The second order cooccurrence matrix may then be calculated similarly by squaring the first order cooccurrence matrix. Repeating this procedure will yield higher order coocurrence matrices.

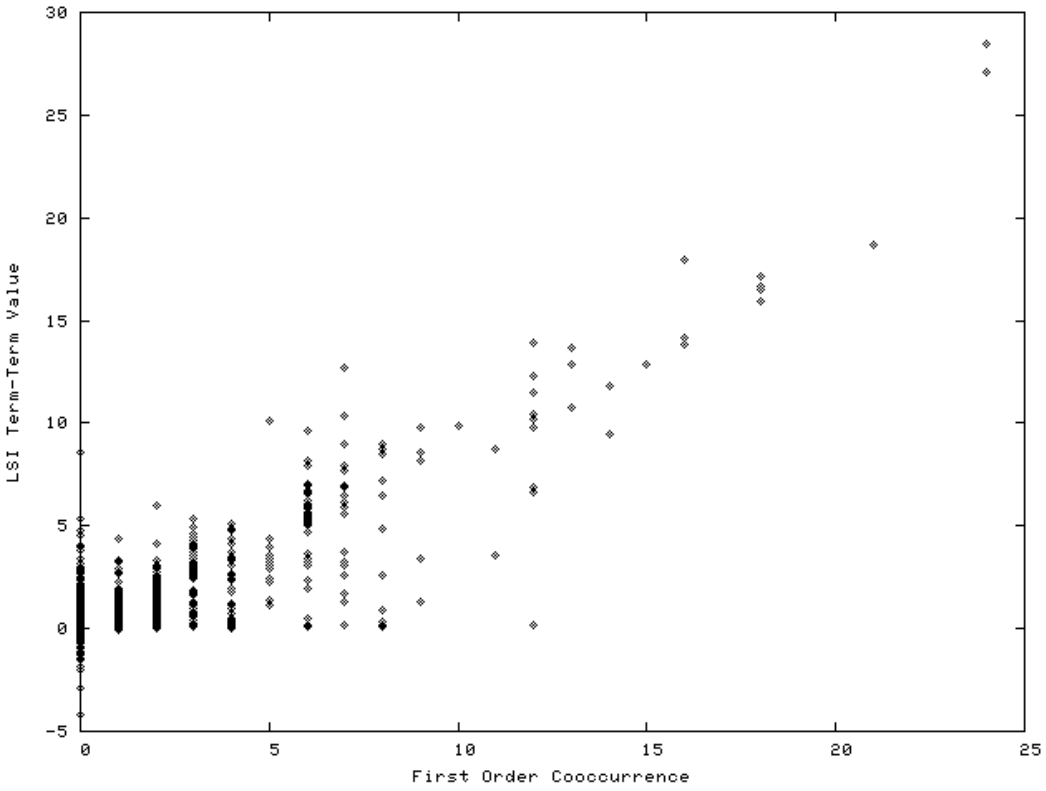
A *first order cooccurrence path* is the number of documents in which two terms cooccur. For example, if two terms *a* and *b* cooccur in 3 documents, then there are 3 first order cooccurrence paths between them. Similarly, a *second order cooccurrence path* between two terms *a* and *b* is a term *c* such that *a* cooccurs with *c* and *b* cooccurs with *c*. The number of unique terms *c* is the number of second order cooccurrence paths between *a* and *b*. Before a cooccurrence matrix is reduced to a binary matrix, the value of its i,j^{th} element is the number of coocurrence paths between terms *i* and *j*.

4. EXPERIMENTAL RESULTS

It was proven in a paper by Kontostathis and Pottenger [K03] that any two terms with an LSI term-term value greater than zero must have a cooccurrence path. Given this information, our hypothesis is that the LSI term-term value of a pair of terms is related to the number of first and second order cooccurrence paths of the pair.

The CRAN data set was chosen for initial testing of the hypothesis, because it had performed well with LSI in our previous research. The first tests compared the first

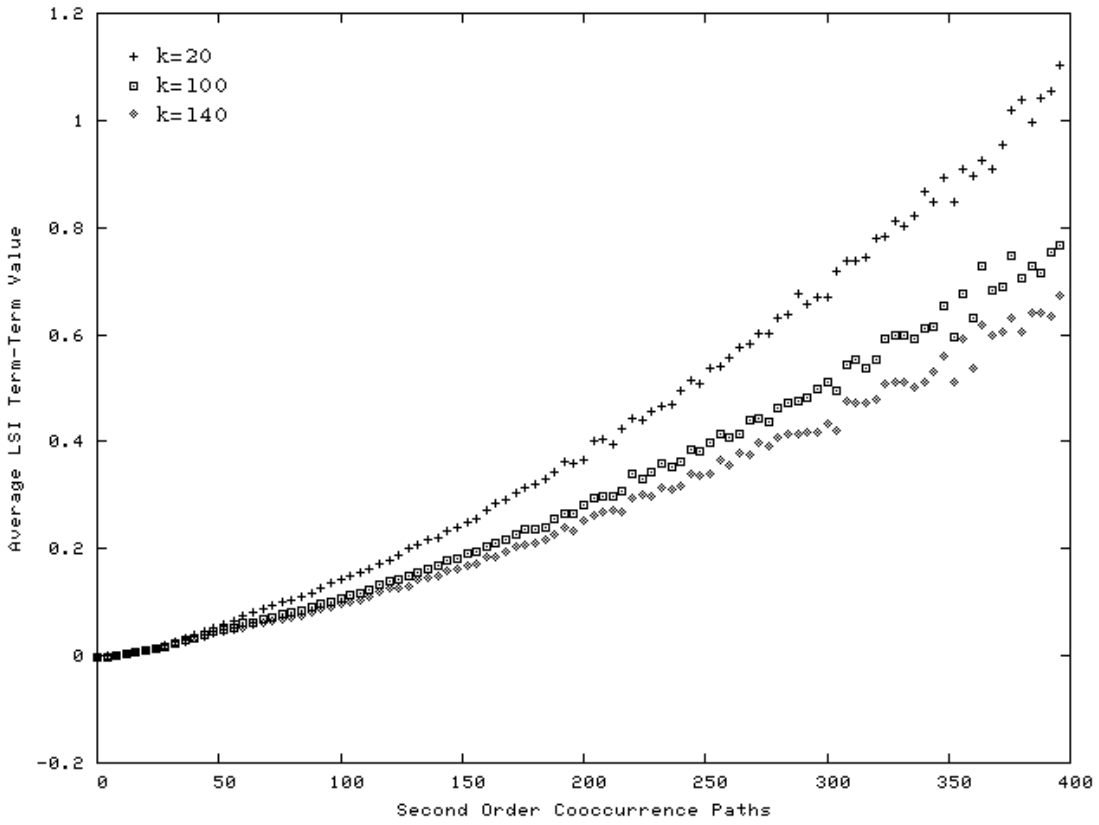
order cooccurrence paths of a term pair to the LSI term-term value of the term pair. Figure 1 shows the results of this comparison on the first 20 documents of CRAN, where k=20. The first order cooccurrence matrix has yet to be reduced to a binary matrix, so the graph shows the number of first order cooccurrence paths. Only the first 10000 term pairs have been graphed, to make it more readable.



High first order cooccurrence, as expected, correlates well with high LSI values. As you can see in figure 1, there is an almost linear rise in the LSI value as cooccurrence increases. This is unsurprising, as the LSI term-term matrix itself is an approximation of the first order cooccurrence matrix. [BDO95]

The first immediately interesting information provided by figure 1 are the results at 0. The LSI values for term pairs which do not occur in the same document range from almost -5 to just over 5. Next we will examine these term pairs, to see if it is possible to determine why there is such a variation of LSI values in pairs with no cooccurrence.

Figure 2 details the case where the first order cooccurrence of a term pair is zero by examining the second order cooccurrence of that term pair. The graph shows the average LSI value of a term pair with no first order cooccurrence, sorted by its second order cooccurrence. The data for figure 2 is again from the CRAN data set, and represents all term pairs from the first 1000 documents where first order cooccurrence is equal to zero, with a range of k from 20 to 140.

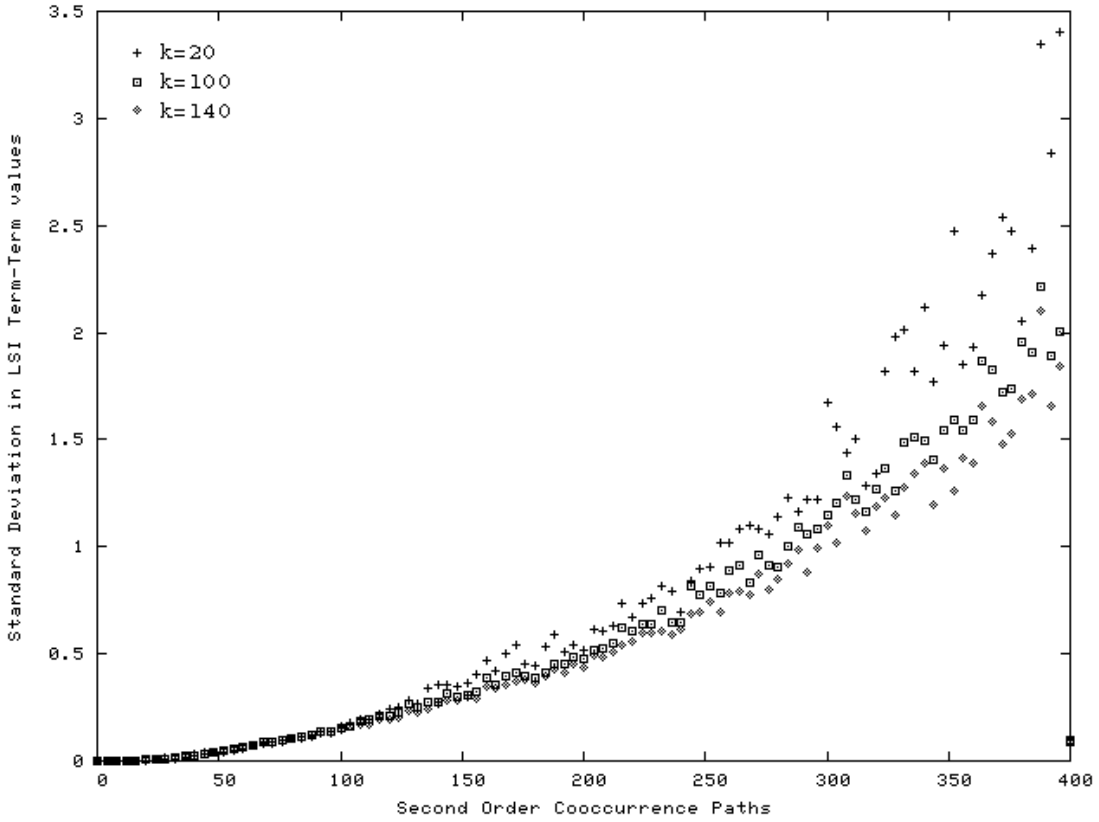


The first thing to notice about this graph is the generally exponential rise of the average LSI values as the number of second order cooccurrence paths rise. It is clear that a high number of second order paths correlates with a high LSI value.

The next thing to notice is that as the k value increases, so does the correlation of cooccurrence paths and LSI value. This is to be expected, because at higher k values, the LSI term-term relationship matrix is a better approximation of the first order cooccurrence matrix. Thus, LSI values will be related more closely to first order cooccurrences, and the second order cooccurrence will have less effect on the LSI value.

5. WORK IN PROGRESS

Unfortunately, the second order cooccurrence value of these term pairs is not very helpful in estimating the LSI value of the pair. As the number of second order paths increases, so does the standard deviation of the LSI values of pairs in that range. Figure 3 demonstrates that the standard deviation can get quite large.



The data that has been gathered so far has only been gathered from the CRAN data set. Further data sets will need to be tested to see if they follow the same pattern as CRAN. It will also be necessary to test higher order cooccurrences to see if they will be of use in approximating LSI.

Before the final draft of this paper, we plan to analyze another data set, to see if it follows the same patterns as the CRAN data set. Furthermore, we plan to test the correlation of the third-order cooccurrence data with LSI values on CRAN.

6. CONCLUSION

Due to the promising results, we feel that the relationship of the cooccurrence matrices to the LSI term-term matrix merits further study. When higher orders of cooccurrence and data from different sets are taken into account, a clearer picture of this relationship should emerge.

7. REFERENCES

[BDO95] Berry, M. W., S. T. Dumais, and G. W. O'Brien. 1995. Using linear algebra for intelligent information retrieval. SIAM Review, Volume 37, No. 4. pp. 573-595.

[D90] Deerwester, Scott, et al. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science,* 41(6): 391-407.

[D93] Dumais, S. T. 1993. LSI meets TREC: A status report. The First

Text REtrieval Conference (TREC1), D. Harman (Ed.), National Institute of Standards and Technology Special Publication 500-207, pp. 137-152.

[D94] Dumais, S. T. 1994. Latent Semantic Indexing (LSI) and TREC-2. In: D. Harman (Ed.), The Second Text REtrieval Conference (TREC2), National Institute of Standards and Technology Special Publication 500-215, pp. 105-116

[D95] Dumais, S. T. 1995. Using LSI for information filtering: TREC-3 experiments. In: D. Harman (Ed.), The Third Text REtrieval Conference (TREC3) National Institute of Standards and Technology Special Publication.

[K03] Kontostathis, April and William M. Pottenger. 2003. A framework for understanding LSI Performance. Proceedings of ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval-ACMSIGIR MF/IR 2003.

[S98] Schütze, H. 1998. Automatic Word Sense Disambiguation. Computational Linguistics, Volume 24, No. 1.

[ZH01] Zelikovitz, S. and H. Hirsh. 2001. Using LSI for Text Classification in the Presence of Background Text. Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management.