A Note on Latent Semantic Analysis

Yoav Goldberg
yoav.goldberg@gmail.com

January 18, 2014

The purpose of this document is to explain why LSA works – specifically, why (and when) is it (mathematically) justified to use the similarity between term vectors or document vectors.

All of the material here appears in the highly cited paper "Indexing by Latent Semantic Analysis" [1] (as well as other publications introducing the LSA and LSI methods). However, it is unfortunately not discussed much in NLP publications that use it.

1 Intro to Latent Semantic Analysis

1.1 The Vector Space Model

The departure point of LSA (Latent Semantic Analysis / Indexing), is the vector-space model.

We have a corpus of d documents over a vocabulary of v words. We arrange the corpus in a matrix C of dimensions $v \times d$, where C_{ij} is the amount of association between word i and document j. The amount of association is either the count, or a function based on the count such as PMI, TF-IDF and so on. While choosing the association measure is important for obtaining good performance, it is not important to this explanation. Similarly, the documents can be generalized to any context a word appears in (e.g. same sentence, k preceding and following words, syntactic relations, and so on).

Each row in C is associated with a word and each column is associated with a document. Each word vector reflects the contexts the word appears in, and each document vector reflects the words that appear in it.

Based on the intuition that words appearing in similar documents (contexts) are similar, we can measure the similarity between words by measuring the similarity between their corresponding vectors (matrix rows). Similarly, we can measure the similarity between documents using the similarity between their corresponding document vectors (matrix columns). One common way of measuring similarity is the cosine similarity measure: $sim_{cos}(x, y) = \frac{\langle x\ ,\ y \rangle}{\|x\|\|y\|}$.

1.2 LSA (Dimensionality Reduction)

One problem with the vector space model is that of data sparsity – some entries in the matrix C may be incorrect because we did not observe enough data points. LSA (latent semantic analysis / indexing) is a way of "smoothing" the matrix: based on robust patterns in the data, some of the counts are "fixed". This has the effect, for example, of adding words to contexts that they were not seen with, if other words in this context seem to co-locate with each other.

Another effect of LDA is representing each word (or document) as a dense k-dimensional vector instead of a sparse d-dimensional (or v dimensional) one, where $k \ll v$ and $k \ll d$ (typical choices are $50 < k < 300$). One can then compute similarities based on the dense k-dim vectors instead of the sparse high-dimensional ones. The purpose of this document is to explain WHY the similarities in the low-dimensional representation are equivalent to similarities in the high-dimensional space.

2 The Mathematics of LSA

2.1 The SVD

LSA builds upon the mathematical technique of singular value decomposition (SVD). Using SVD, the matrix C is decomposed to a multiplication of three matrices:

$$C = U_0 \Sigma_0 V_0$$

where U_0 is $v \times v$, Σ_0 is diagonal $v \times d$ and V_0 is $d \times d$.

Matrices U_0 and V_0 are orthonormal (meaning their rows are both unit-length and orthogonal to each other). The diagonal of Σ_0 contain the singular values of C, in decreasing order.

We then keep the k largest values of Σ_0, zeroing the rest. This zeros the corresponding rows and columns of U_0 and V_0 as they will contain only zeroes after the multiplication with the modified Σ_0. After deleting the zero rows and columns from all matrices, we are left with matrices U $(v \times k)$, Σ $(k \times k)$ and V $(k \times d)$.

The product

$$C' = U \Sigma V$$

is a $(v \times d)$ matrix of rank k. The *Eckart-Young Theorem* states that C' is the best rank-k approximation of C, in the sense that:

$$C' = \arg\min_{M} \|M - C\|_2 \text{ s.t. } M \text{ is rank-}k$$

C' can be thought of as a smoothed version of C in the sense that it uses only the k most influential directions in the data. Empirically, the matrix C' can produce more robust similarities (when compared to human judgement on how similar should the words / documents be) than the matrix C.

Because the SVD is unique, applying SVD to C' will reconstruct $U\Sigma V$ (the resulting matrices will be $v \times v$, $v \times d$, $d \times d$ but are equal to the matrices defined above after removal of zero-valued rows and columns).

2.2 Not constructing C' explicitly

When using LSA in practice, the matrix C' is never constructed explicitly. Instead, similarity is computed based on word- and document-vectors.

In order to compute similarities between words, we consider the word vectors:

$$W = U\Sigma$$

This is a $v \times k$ matrix, in which row W_i correspond to word i in the vocabulary. The similarity of words i and j can be computed as $sim(w_i, w_j)$.

In order to compute similarities between documents, we consider the document vectors:

$$D = \Sigma V$$

This is a $k \times d$ matrix, in which column $D_{,i}$ correspond to document i in the corpus. The similarity of documents i and j can be computed as $sim(D_{,i}, D_{,j})$.

2.3 Justification for using W instead of C'

Why can we use W instead of C' for computing word similarities? We assume for now that similarities are equivalent to dot products $sim(x, y) = \langle x\ ,\ y \rangle$. We will show that $\langle C'_i\ ,\ C'_j \rangle = \langle W_i\ ,\ W_j \rangle$.

Consider the similarity matrix $S^{C'} = C'C'^T$, in which $S^{C'}_{ij} = \langle C'_i\ ,\ C'_j \rangle$. Similarly for $S^W = WW^T$, we have $S^W_{ij} = \langle W_i\ ,\ W_j \rangle$. Remember that $VV^T = I$ because V is orthonormal. Now:

$$\begin{aligned} S^{C'} &= C'C'^T = (U\Sigma V)(U\Sigma V)^T = (U\Sigma V)(V^T\Sigma^T U^T) \\ &= (U\Sigma)(VV^T)(\Sigma^T U^T) = (U\Sigma)(U\Sigma)^T = WW^T = S^W \end{aligned}$$

The argument for using D instead of C' is parallel.

Note that the multiplication WD *does not* produce C'. Note also that many works compute similarities based on either U or V directly, without multiplying by Σ. This is wrong. Similarly, some define $W = U\sqrt{\Sigma}$ and $D = \sqrt{\Sigma}V$. Under this definition $WD = C'$ but the rows (columns) of W (D) again cannot be used instead of the rows (columns) of C'.

References

[1] Scott Deerwester, Susan Dumais, Georg Furnas, Thomas Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.