Model the Vector Space of the Documents (Latent Semantic Analysis)

Latent Semantic Analysis (LSA):

Latent Semantic Analysis, or LSA, is an approach to discovering the hidden "topics" that exist across a set of documents. The basic assumption is words that co-occur in the same document are similar, and documents containing the same or similar words are similar: thus, simply by relying on the distributional properties of words in the dataset (which words occur where), you can infer what a document is "about".

There are two main steps to this process: building a term-document matrix, then reducing the dimensionality of that matrix.

Step 1: Build a term-document matrix

The first step to the LSA is constructing a term-document matrix based on TF-IDF. The logic of considering TF-IDF is that terms that are very frequent in a specific document but occur very rarely throughout the rest of a corpus, are probably more "representative" of that document's meaning.

Step 2: Use singular-value decomposition (SVD) to discover latent topics

Obtaining a tf-idf matrix is a great first step for document representation.

But one problem with tf-idf vectors is that they are sparse. If we assume there are 10,000 words that occur across all of our documents, each document will likely only contain a small subset of those words; this means that the vector representation for each document will consist mostly of 0s. This may still prove informative, but it misses valuable relationships between words: by insisting that every word corresponds to distinct elements of a vector, we fail to capture the fact that "fruit" and "vegetable" are both examples of food. Thus, if Document 1 only mentions "fruit", and Document 2 only mentions "vegetable", those documents would not be identified as similar (whereas if Document 3 mentions both "fruit" and "vegetable", it will be identified as similar to both Document 1 and Document 2). In other words, our term-document vectors don't capture the fact that different words can still be related (e.g. synonyms).

LSA attempts to remedy this problem by performing dimensionality reduction on the term-document matrix: given an M x N matrix (M=#terms, N=#documents), LSA constructs a low-rank approximation k x N matrix (k=#topics, N=#documents). Put more simply, we want to turn our sparse term-document matrix into a denser matrix, ideally one where each of our k columns represents a semantic topic.

Dr. Abu Nowshed Chy
Lecturer, Dept. of CSE, CU

There are a few well-known approaches to dimensionality reduction. LSA uses singular value decomposition (SVD). In brief, SVD gives us a matrix with k dimensions (where k=#topics), based on the idea that terms with similar meanings will co-occur with other terms and in similar documents (and documents with similar meanings will contain similar terms).

Advantages:

1) Easy to implement, understand and use. There are many practical and scalable implementations available thus reliable.
2) Noise removal and robustness by dimension reduction. The noise is a data which could be certain less important terms.
3) Unsupervised approach.
4) *Performance:* LSA is capable of assuring decent results, much better than the plain vector space model. It works well on a dataset with diverse topics.
5) *Synonymy:* LSA can handle Synonymy (same meaning of multiple words) problems to some extent (depends on dataset though)
6) *Runtime:* Since it only involves decomposing your term-document matrix, it is faster, compared to other dimensionality reduction models.
7) It is not sensitive to starting conditions (like a neural network), so consistent.
8) Since it is a popular method, will get good community support.
9) Apply it on new data is easier and faster compared to other methods.
10) Empirical study shows it outperforms naïve vector space model.

Disadvantages:

1) LSA vectors require large storage. There are many advances in electronic storage media, but still the loss of sparseness due to large data is the more serious implication.
2) LSA performs relatively well for long documents due to the small number of context vectors used to describe each document. However, due to the large size of data, it requires an additional storage space and computing time which reduces the efficiency of LSA.
3) Since it is a distributional model, so not an efficient representation, when compared against state-of-the-art methods (e.g. deep neural networks).
4) Representation is dense, so hard to index based on individual dimensions.
5) It is a linear model, so not the best solution to handle non-linear dependencies
6) The latent topic dimension can not be chosen to arbitrary numbers. It depends on the rank of the matrix, so can't go beyond that.
7) Deciding on the number of topics is based on heuristics and needs some expertise. The conventional solution is to sort the cumulative singular values in descending order and finding a cut (say x% of total value).
8) Ad hoc selection of the number of dimensions, model selection.
9) No probabilistic model of term occurrences.
10) Problem of polysemy (multiple meanings for the same word) is not addressed.

Sample Problem Formulation:

Compute the full SVD for the following matrix:

$$\begin{vmatrix} 4 & 0 \\ 3 & -5 \end{vmatrix}$$

Solution:

Let, $A = \begin{vmatrix} 4 & 0 \\ 3 & -5 \end{vmatrix}$

Step 1: Compute its transpose A^T and AA^T

$$A^T = \begin{vmatrix} 4 & 3 \\ 0 & -5 \end{vmatrix}$$

$$A^T A = \begin{vmatrix} 4 & 3 \\ 0 & -5 \end{vmatrix} \begin{vmatrix} 4 & 0 \\ 3 & -5 \end{vmatrix} = \begin{vmatrix} 25 & -15 \\ -15 & 25 \end{vmatrix}$$

Step 2: Determine the eigenvalues of $A^T A$ and sort these in descending order, in the absolute sense. Square roots these to obtain the singular values of A.

$$A^T A - \lambda \mathrm{I} = \begin{vmatrix} 25 & -15 \\ -15 & 25 \end{vmatrix} - \lambda \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$

$$= \begin{vmatrix} 25 & -15 \\ -15 & 25 \end{vmatrix} - \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix}$$

$$= \begin{vmatrix} 25-\lambda & -15 \\ -15 & 25-\lambda \end{vmatrix} \text{ .... (1)}$$

Now, $|A^T A - \lambda \mathrm{I}| = 0$

$$(25-\lambda)(25-\lambda) - (-15)(-15) = 0$$

$$625 - 25\lambda - 25\lambda + \lambda^2 - 225 = 0$$

$$\lambda^2 - 50\lambda + 400 = 0$$

$$(\lambda - 40)(\lambda - 10) = 0$$

Sort the values of λ in descending order, $(40 > 10)$.

Therefore, the eigenvalues are $\lambda_1 = 40$ and $\lambda_2 = 10$

And singual values are $s_1 = \sqrt{40} = 6.3245$ and $s_2 = \sqrt{10} = 3.1622$

Dr. Abu Nowshed Chy
Lecturer, Dept. of CSE, CU

Step 3: Construct diagonal matrix S by placing singular values in descending order along its diagonal.

$$S = \begin{vmatrix} 6.3245 & 0 \\ 0 & 3.1622 \end{vmatrix}$$

Compute its inverse, S^{-1}.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{\underbrace{ad-bc}_{\text{determinant}}} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$S^{-1} = \frac{1}{(6.3245 \times 3.1622) - 0} \begin{vmatrix} 3.1622 & 0 \\ 0 & 6.3245 \end{vmatrix}$$

$$= \frac{1}{19.99} \begin{vmatrix} 3.1622 & 0 \\ 0 & 6.3245 \end{vmatrix}$$

$$= \begin{vmatrix} 0.1581 & 0 \\ 0 & 0.3162 \end{vmatrix}$$

Step 4: Use the ordered eigenvalues from Step 2 and compute the eigenvectors of $A^T A$. Place these eigenvectors along the columns of V and compute its transpose, V^T.

For, $\lambda = 40$, we get:

$$(A^T A - \lambda \text{I}) E_1 = 0$$

$$(A^T A - \lambda \text{I}) \begin{vmatrix} x \\ y \end{vmatrix} = 0 \qquad \left(\text{Let, } E_1 = \begin{vmatrix} x \\ y \end{vmatrix}\right)$$

$$\begin{vmatrix} 25 - \lambda & -15 \\ -15 & 25 - \lambda \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = 0 \quad \text{(Using Eqn. (1))}$$

$$\begin{vmatrix} 25 - 40 & -15 \\ -15 & 25 - 40 \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = 0$$

$$\begin{vmatrix} -15 & -15 \\ -15 & -15 \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = 0$$

$$\begin{vmatrix} -1 & -1 \\ -1 & -1 \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = 0$$

Dr. Abu Nowshed Chy
Lecturer, Dept. of CSE, CU

Therefore, we will get:

$$-x - y = 0 \dots \dots \dots (2)$$
$$-x - y = 0 \dots \dots \dots (3)$$

From eqn (2) and (3) we will get,

$$y = -x$$

Now,

$$E_1 = \begin{vmatrix} x \\ y \end{vmatrix} = \begin{vmatrix} x \\ -x \end{vmatrix}$$

Dividing by its length, $L = \sqrt{x^2 + (-x)^2} = x\sqrt{2}$

$$E_1 = \begin{vmatrix} \frac{x}{x\sqrt{2}} \\ \frac{-x}{x\sqrt{2}} \end{vmatrix} = \begin{vmatrix} 0.7071 \\ -0.7071 \end{vmatrix}$$

Therefore, the eigenvector is, $E_1 = \begin{vmatrix} 0.7071 \\ -0.7071 \end{vmatrix}$

Similarly, for $\lambda = 10$, we get:

$$(A^T A - \lambda \mathrm{I}) E_2 = 0$$

$$(A^T A - \lambda \mathrm{I}) \begin{vmatrix} x \\ y \end{vmatrix} = 0 \qquad \left(\text{Let, } E_2 = \begin{vmatrix} x \\ y \end{vmatrix}\right)$$

$$\begin{vmatrix} 25-\lambda & -15 \\ -15 & 25-\lambda \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = 0$$

$$\begin{vmatrix} 25-10 & -15 \\ -15 & 25-10 \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = 0$$

$$\begin{vmatrix} 15 & -15 \\ -15 & 15 \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = 0$$

$$\begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = 0$$

Therefore, we will get:

$$x - y = 0 \dots \dots \dots (4)$$
$$-x + y = 0 \dots \dots \dots (5)$$

Dr. Abu Nowshed Chy
Lecturer, Dept. of CSE, CU

From eqn (4) and 5), we will get,

$$y = x$$

Now,

$$E_2 = \begin{vmatrix} x \\ y \end{vmatrix} = \begin{vmatrix} x \\ x \end{vmatrix}$$

Dividing by its length, $L = \sqrt{x^2 + (x)^2} = x\sqrt{2}$

$$E_2 = \begin{vmatrix} \frac{x}{x\sqrt{2}} \\ \frac{x}{x\sqrt{2}} \end{vmatrix} = \begin{vmatrix} 0.7071 \\ 0.7071 \end{vmatrix}$$

Therefore, the eigenvector is, $E_2 = \begin{vmatrix} 0.7071 \\ 0.7071 \end{vmatrix}$

Now, placing these eigenvectors along the columns of

$$V = |E_1 \quad E_2| = \begin{vmatrix} 0.707 & 0.707 \\ -0.707 & 0.707 \end{vmatrix}$$

$$V^T = \begin{vmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{vmatrix}$$

Step 5: Compute U as $U = AVS^{-1}$. Then, compute the full SVD using $A = USV^T$.

$$U = AVS^{-1} = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix} \begin{bmatrix} 0.1581 & 0 \\ 0 & 0.3162 \end{bmatrix}$$

$$U = AVS^{-1} = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \begin{bmatrix} 0.1118 & 0.2236 \\ -0.1118 & 0.2236 \end{bmatrix}$$

$$U = AVS^{-1} = \begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{bmatrix}$$

$$A = USV^T = \begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{bmatrix} \begin{bmatrix} 6.3245 & 0 \\ 0 & 3.1622 \end{bmatrix} \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}$$

$$A = USV^T = \begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{bmatrix} \begin{bmatrix} 4.4721 & -4.4721 \\ 2.2360 & 2.2360 \end{bmatrix}$$

$$A = USV^T = \begin{bmatrix} 3.9998 & 0 \\ 2.9999 & -4.9997 \end{bmatrix} \approx \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix}$$

------ o ------

Applications of SVD: Self Study

Dr. Abu Nowshed Chy
Lecturer, Dept. of CSE, CU