An LSA Package for R

Fridolin Wild, *Vienna University of Economics and Business Administration, Austria*

***Abstract* — Latent semantic analysis (LSA) is an algorithm applied to approximate the meaning of texts, thereby exposing semantic structure to computation. LSA combines the classical vector-space model − well known in computational linguistics − with a singular value decomposition (SVD), a two-mode factor analysis. Thus, bag-of-words representations of texts can be mapped into a modified vector space that is assumed to reflect semantic structure. In this contribution the author describes the *lsa* package for the statistical language and environment *R* and illustrates its proper use through an example from the area of automated essay scoring.**

I. Concepts used in latent semantic analysis

When applying a latent semantic analysis (Deerwester et al., 1990), a process is executed that typically involves several (optional) steps and involves various data-types created as an output of these steps. To clarify which entities and processes are involved when performing an LSA, the following concepts shall be defined.

Term: The 'word' as it is written down in a document.

Corpus: The collection of documents containing texts that consist out of terms separated by punctuation marks.

Textmatrix: A representation of the document collection in matrix format: the cells contain the frequency, how often a particular term appears in a specific document. Terms are the rows, documents the columns. By transforming a corpus to this representation format, documents are treated as so-called bag of words, where the term order is neglected.

Latent-Semantic Space: When applying a singular-value decomposition (SVD) to a textmatrix, the matrix is resolved into the term-vector matrix T (constituting the left singular vectors), the document-vector matrix D (constituting the right singular vectors) being both orthonormal and the diagonal matrix S (Berry et al., 1995). These partial matrices are then truncated in order to reflect strong associations, eliminate noise, etc. The set of these three, truncated partial matrices T_k, S_k, and D_k is called 'latent-semantic space'. A latent-semantics space can be converted back to textmatrix format.

Folding In: To keep additional documents from changing the structure of a latent-semantic space, documents can be folded into the previously calculated space. Thereby, T_k and S_k of the space are re-used and combined with a textmatrix constructed over the new documents. See Wild and Stahl (2006) for more details.

Dimension: When truncating the partial matrices from the SVD, a particular number of the highest singular values are retained. This is called the 'dimensionality' of the latent-semantic space.

Distance / Similarity: Within a textmatrix, various methods can be applied to measure the distance (or, the other way round, similarity) between terms, documents, or terms and documents. One method is, e.g., to use the measure the cosine of the angle between two column-vectors in order to calculate the similarity between two documents. A high cosine value is equal to a small angle between the vectors, thus indicating high similarity.

Vocabulary: All terms used within a corpus form the vocabulary of this corpus. The vocabulary has a certain order to ensure that additional text matrices can be constructed that can be appended to an existing textmatrix.

II. The Process

A typical LSA process using the R package looks similar to the one depicted in Figure 1. First, a textmatrix is constructed with textmatrix() from the input corpus. The textmatrix can (but does not need to be) weighted using one of the various weighting schemes provided (see Wild (2005) for more details). Then, the singular-value decomposition is executed over this textmatrix and the resulting partial matrices are truncated and returned by lsa(). The number of dimension to keep can be set using various recommender routines (e.g., dimcalc_kaiser()). The resulting latent-semantic space can be converted back to textmatrix format using as.textmatrix().

Manuscript received October 9, 2001. (Write the date on which you submitted your paper for review.) This work was supported in part by the U.S. Department of Commerce under Grant BS123456 (sponsor and financial support acknowledgment goes here). Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title; short formulas that identify the elements are fine (e.g., "Nd–Fe–B"). Do not write "(Invited)" in the title. Full names of authors are preferred in the author field, but are not required. Put a space between authors' initials.

F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (corresponding author to provide phone: 303-555-5555; fax: 303-555-5555; e-mail: author@ boulder.nist.gov).

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar. colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).

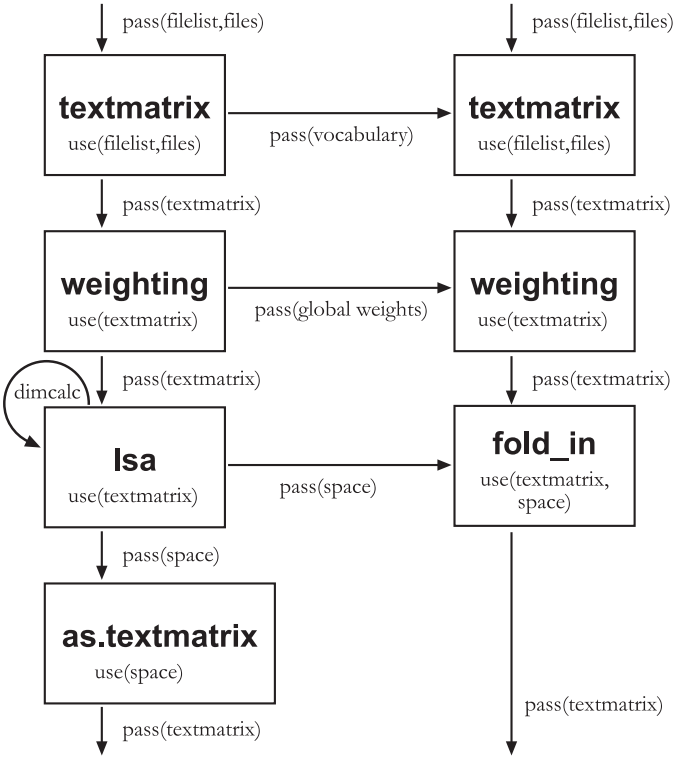

Figure 1. Overall Workflow.

In case that additional documents are to be folded into the existing latent-semantic space, again a new textmatrix is constructed using textmatrix() re-using the vocabulary of the first. Again the resulting textmatrix can be weighted (eventually re-using the global weights of the first textmatrix). Using fold_in(), the resulting textmatrix can be mapped into the existing latent-semantic space, thereby re-using the truncated left-sided and the diagonal partial matrices of the SVD. In this case, the result is directly a textmatrix.

III. Sanitizing Corpora with the Package

Looking more closely at the textmatrix routine, it can be seen that several text sanitizing and pre-processing steps are embedded in the textmatrix generation routines: the routine included means to convert the terms to lower case, simple routines for stripping XML tags, automatic removal of punctuation marks and some other special characters, and trimming of multiple white spaces. Furthermore, stop words can be filtered (by providing stop word lists) or a controlled vocabulary can be deployed. Furthermore, frequency filters can be applied to delete terms below or above a certain frequency threshold (within a document or within the corpus) or outside a certain term-length range. Terms consisting purely of numbers can be removed automatically. Also all terms can be reduced to their word stems (using Porter's snowball stemmer).

The package is open-source and available via CRAN, the Comprehensive R Archive Network.

IV. Demonstration

The following code example illustrates how LSA may be used to automatically score free-text essays in an educational assessment setting.

```
library("lsa")

# load training texts
trm = textmatrix("trainingtexts/")
trm = lw_bintf(trm) * gw_idf(trm) #
weighting
space = lsa(trm) # create LSA space

# fold-in test and gold standard essays
tem = textmatrix("essays/",
vocabulary=rownames(trm))
tem = lw_bintf(tem) * gw_idf(tem) #
weighting
tem_red = fold_in(tem, space)

# score essay against gold standard
cor(tem_red[,"gold.txt"],
tem_red[,"E1.txt"]) # 0.7
```

Listing 1. Essay Scoring Example.

References

[1] DEERWESTER, S., DUMAIS, S., FURNAS, G., LANDAUER, T., HARSHMAN, R. (1990): Indexing by Latent Semantic Analysis. JASIS, 41, 391--407.

[2] BERRY, M., DUMAIS, S. and O'BRIEN, G. (1995): Using Linear Algebra for Intelligent Infor-mation Retrieval. SIAM Review, 37, 573—595.

[3] WILD, F. (2005): lsa: Latent Semantic Analysis. R package version 0.57.

[4] WILD, F., STAHL, C. (2006): Investigating Unstructured Texts with Latent Semantic Analysis. In: Lenz, Decker (Eds.): Advances in Data Analysis, Springer, 2007