CS395T: Large Scale Data-Mining

LSI Example

1 Data Set

Suppose we are give the following $\mathbf{d} = 9$ documents:

c1: *Human* machine *interface* for Lab ABC *computer* applications

c2: A *survey* of *user* opinion of *computer system response time*

c3: The *EPS user interface* management *system*

c4: *system* and *human system* engineering testing of EPS

c5: Relation of *user*-perceived *response time* to error measurement.

m1: The generation of random, binary, unordered *trees*

m2: The intersection *graph* of paths in *trees*

m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering

m4: *Graph minors*: A *survey*

2 Vector Space for data

The twelve italicized words are used to characterize each document in a $\mathbf{w} = 12$ dimensional *word vector space*. The i^{th} location of each document denotes the number of times the i^{th} word occured in the document. The 12 dimensional document vectors form the following $\mathbf{w} \times \mathbf{d}$ matrix denoted by $\mathbf{A}$.

Terms	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Table 1: **A**: Nine document vectors, each in a 12 dimensional word space

3 Normalized vs non-Normalized A

There are a number of normalizations that can be applied to **A**. The one that we consider in this example is to factor out the effect of the length of the document. In general, if a document is very long, the sum of elements in its vector will be large. In other words, the document will lie farther away from the origin of this word vector space. Since from query and search point of views, the document length is not important, we normalize each document vector to have a unit length. Let's denote **An** as the normalized **A**. One can think of all the documents in the first quadrant on a unit sphere in a **w** dimensional space.

4 Singular Vectors of A and An

Figure 1 shows the first two singular vectors of **A** and **An**. A couple of things to notice in Figure 1:

- All entries in the first singular vectors are positive.
- Some entries in the second singular vectors are negative.
- The dot product of the two singular vectors should be zero. This explains why some of the entries of the second singular vector MUST be negative.
- The components of the two singular vectors peak for different sets of documents.

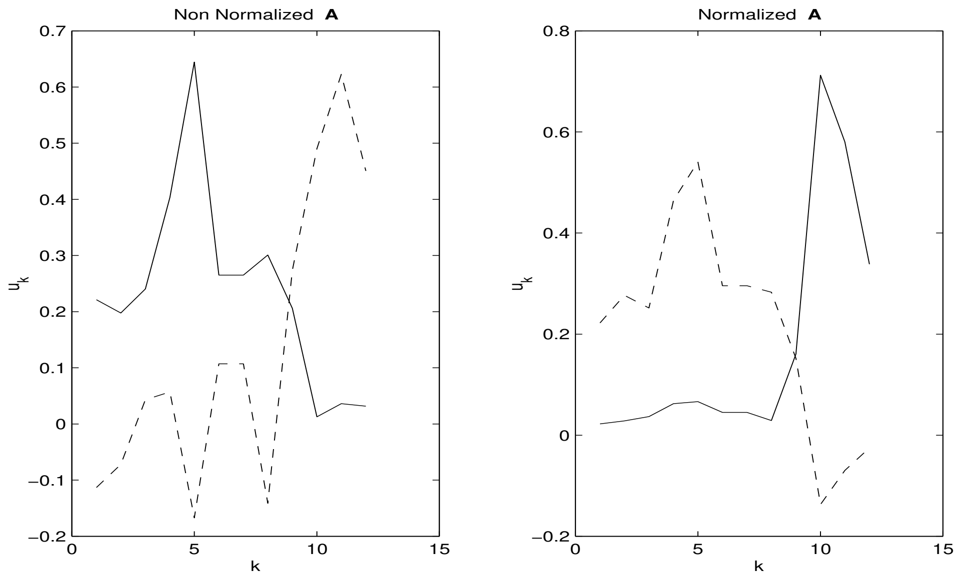

Figure 1: First two singular vectors for **A** and **An**

.

5 Projecting documents on 2-D space

The basis of Latent Semantic Indexing is that instead of projecting our documents or queries onto the **w** dimensional space (which will reduce to a word matching), we want to project on a smaller k-dimensional space that sort of represents k "concept vectors". The first k singular vectors obtained by the SVD of **A** (or **An**) represent this space.

For visualization purpose, let's choose $k = 2$ in our case. The two singular vectors shown in Figure 1 are used for this projection. Figure 2 shows the distribution of the 12 documents projected on the 2-D space obtained from **A** and Figure 3 shows the corresponding projections on the space **An**. The words projected on the same space yield the distribution shown in Figure 4.

The documents projected on the 2-D space obtained from the normalized **A** matrix are shown in Figure 5 and the words projected on the same space are shown in Figure 6.

6 Rank-k approximations

The value of k was chosen to be 2 for visualization purposes and also because we knew that the documents fell into one of two "concepts" only. In general this is not known and higher values of k are required. As $k \to \mathbf{w}$, better and better approximation of **A** is obtained. For any k, $\mathbf{A}_k = U_k \Sigma_k V_k^T$ is the "best" rank k approximation of **A**, where best is defined in terms of the norm of $\mathbf{A}$-$\mathbf{A}_k$.

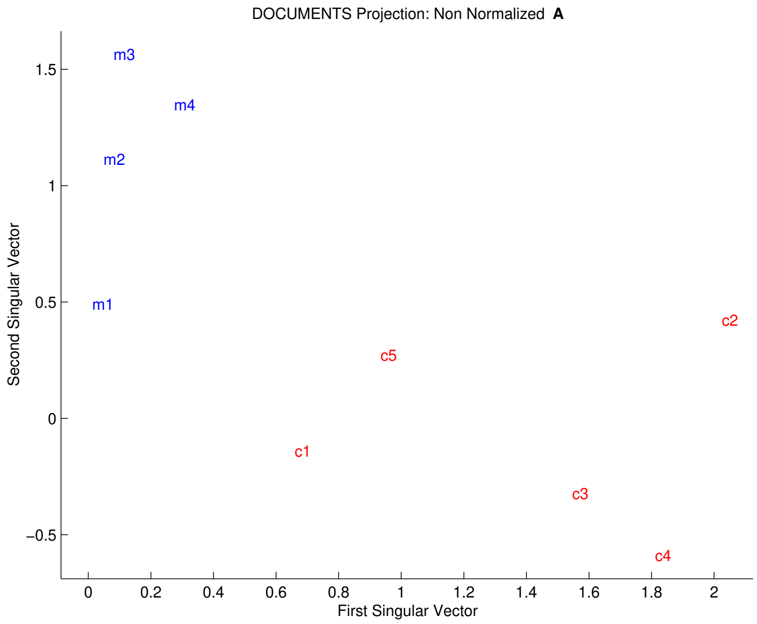

Figure 2: Documents projected on 2-D space from **A**

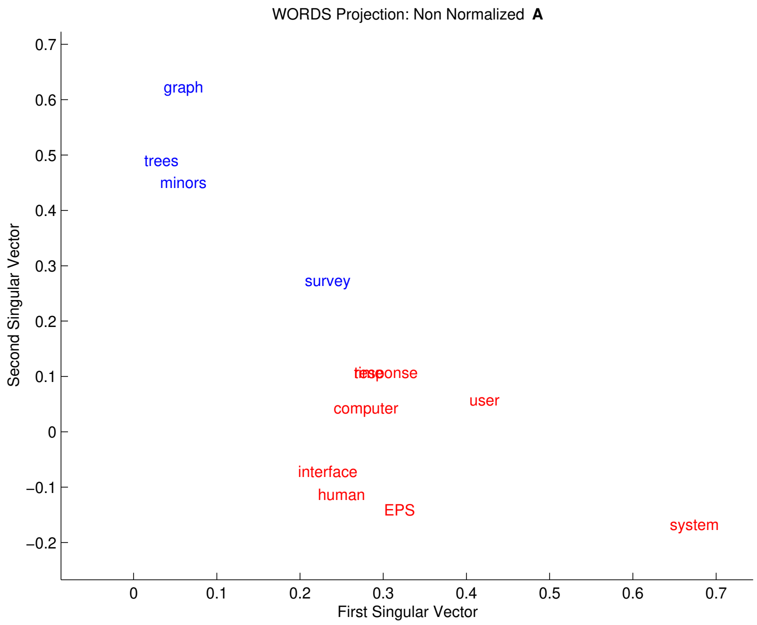

Figure 3: Words projected on 2-D space from **A**

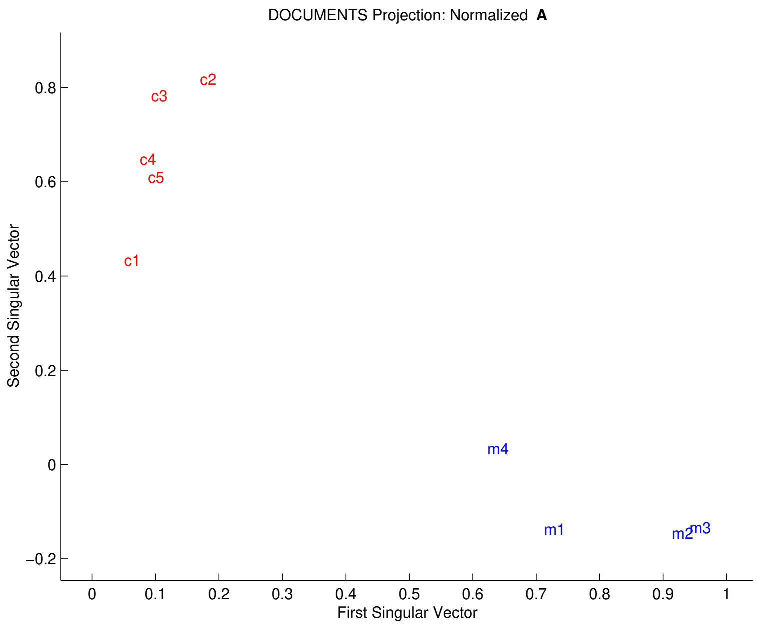

Figure 4: Documents projected on 2-D space from **An**

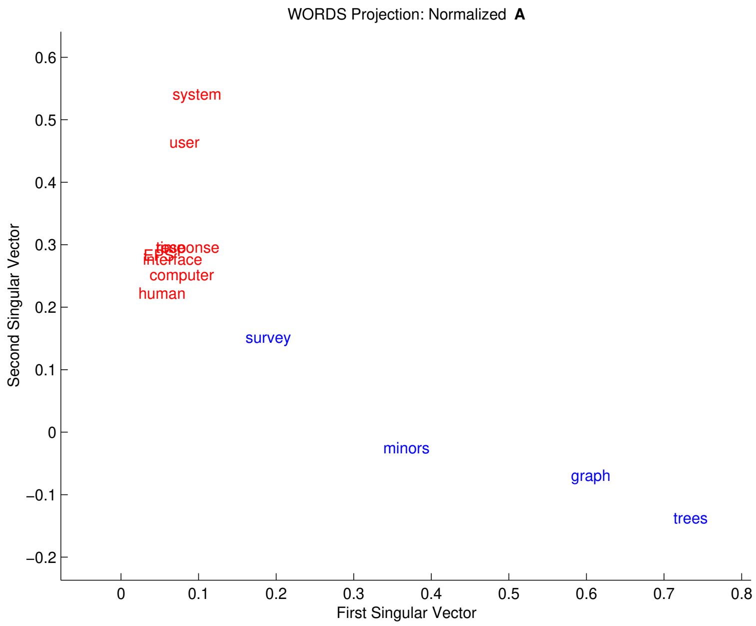

Figure 5: Words projected on 2-D space from **An**

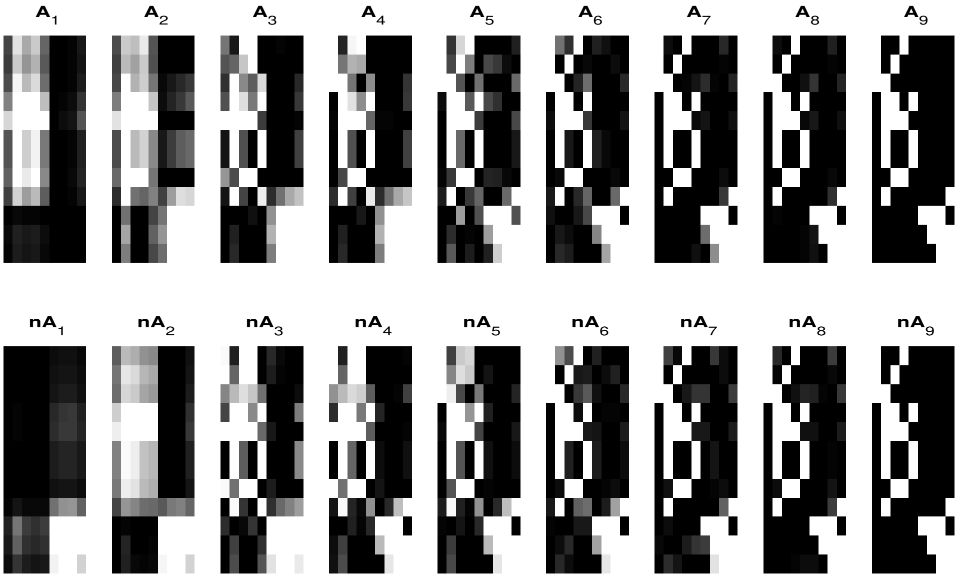

Figure 6 shows the successive rank k approximations of $\mathbf{A}$. White denotes high value and black denotes low. As k increases from 1 to 9, better and better approximations are obtained.

$$10 \times \mathbf{A}_1 = \begin{bmatrix} 0.01 & 0.04 & 0.02 & 0.02 & 0.02 & 0.16 & 0.21 & 0.21 & 0.14 \\ 0.01 & 0.05 & 0.03 & 0.02 & 0.03 & 0.20 & 0.26 & 0.27 & 0.18 \\ 0.02 & 0.06 & 0.03 & 0.03 & 0.03 & 0.26 & 0.34 & 0.35 & 0.23 \\ 0.03 & 0.11 & 0.06 & 0.05 & 0.06 & 0.45 & 0.57 & 0.59 & 0.39 \\ 0.03 & 0.11 & 0.06 & 0.05 & 0.06 & 0.47 & 0.61 & 0.63 & 0.41 \\ 0.02 & 0.08 & 0.04 & 0.03 & 0.04 & 0.32 & 0.41 & 0.42 & 0.28 \\ 0.02 & 0.08 & 0.04 & 0.03 & 0.04 & 0.32 & 0.41 & 0.42 & 0.28 \\ 0.01 & 0.05 & 0.03 & 0.02 & 0.03 & 0.21 & 0.26 & 0.27 & 0.18 \\ 0.08 & 0.27 & 0.15 & 0.12 & 0.14 & 1.14 & 1.47 & 1.51 & 1.00 \\ 0.36 & 1.21 & 0.66 & 0.54 & 0.63 & 5.07 & 6.51 & 6.70 & 4.44 \\ 0.29 & 0.99 & 0.54 & 0.44 & 0.51 & 4.13 & 5.30 & 5.46 & 3.61 \\ 0.17 & 0.58 & 0.32 & 0.25 & 0.30 & 2.41 & 3.09 & 3.18 & 2.11 \end{bmatrix} \tag{1}$$

$$10 \times \mathbf{A}_2 = \begin{bmatrix} 0.97 & 1.85 & 1.76 & 1.46 & 1.37 & -0.14 & -0.12 & -0.09 & 0.21 \\ 1.21 & 2.31 & 2.19 & 1.81 & 1.71 & -0.18 & -0.14 & -0.10 & 0.27 \\ 1.11 & 2.12 & 2.01 & 1.66 & 1.57 & -0.08 & -0.03 & 0.01 & 0.31 \\ 2.04 & 3.90 & 3.69 & 3.06 & 2.89 & -0.19 & -0.11 & -0.03 & 0.54 \\ 2.38 & 4.53 & 4.29 & 3.55 & 3.36 & -0.27 & -0.18 & -0.10 & 0.59 \\ 1.30 & 2.49 & 2.36 & 1.95 & 1.84 & -0.08 & -0.02 & 0.03 & 0.38 \\ 1.30 & 2.49 & 2.36 & 1.95 & 1.84 & -0.08 & -0.02 & 0.03 & 0.38 \\ 1.24 & 2.36 & 2.24 & 1.85 & 1.75 & -0.18 & -0.15 & -0.11 & 0.27 \\ 0.74 & 1.52 & 1.34 & 1.11 & 1.07 & 0.94 & 1.25 & 1.31 & 1.05 \\ -0.23 & 0.09 & -0.41 & -0.35 & -0.21 & 5.26 & 6.71 & 6.89 & 4.39 \\ -0.01 & 0.42 & 0.00 & -0.01 & 0.09 & 4.23 & 5.40 & 5.56 & 3.59 \\ 0.06 & 0.37 & 0.12 & 0.09 & 0.15 & 2.44 & 3.13 & 3.22 & 2.10 \end{bmatrix} \tag{2}$$