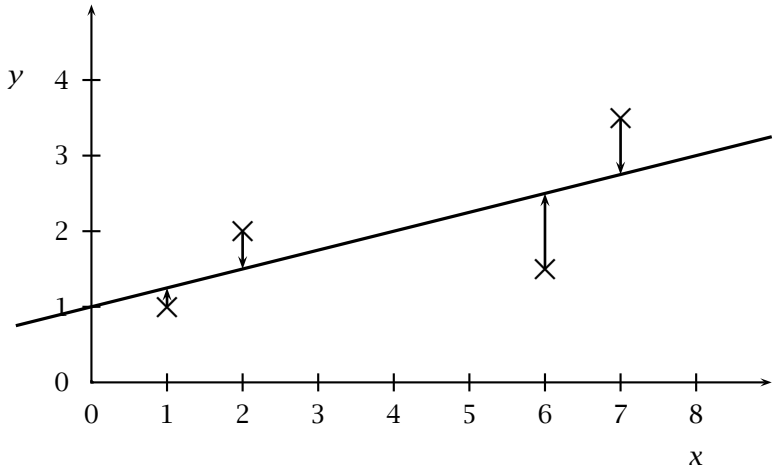

Figure 15.7 An example of linear regression. The line $y = 0.25x + 1$ is the best least-squares fit for the four points (1,1), (2,2), (6,1.5), (7,3.5). Arrows show which points on the line the original points are projected to.

$$\Leftrightarrow \quad -\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) + m\sum_{i=1}^{n}(x_i - \bar{x})^2 = 0$$

$$\Leftrightarrow \quad m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{15.10}$$

Figure 15.7 shows an example of a least square fit for the four points $(1,1)$, $(2,2)$, $(6,1.5)$, and $(7,3.5)$. We have: $\bar{x} = 4$, $\bar{y} = 2$,

$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{6.5}{26} = 0.25$$

and

$$b = \bar{y} - m\bar{x} = 2 - 0.25 \times 4 = 1$$

15.4.2 Singular Value Decomposition

As we have said, we can view Latent Semantic Indexing as a method of word co-occurrence analysis. Instead of using a simple word overlap measure like the cosine, we instead use a more sophisticated similarity measure that makes better similarity judgements based on word co-occurrence. Equivalently, we can view SVD as a method for dimensionality

reduction. The relation between these two viewpoints is that in the process of dimensionality reduction, co-occurring terms are mapped onto the same dimensions of the reduced space, thus increasing similarity in the representation of semantically similar documents.

Co-occurrence analysis and dimensionality reduction are two 'functional' ways of understanding LSI. We now look at the formal definition of LSI. LSI is the application of Singular Value Decomposition to term-by-document matrices in information retrieval. SVD takes a matrix A and represents it as $\hat{A}$ in a lower dimensional space such that the "distance" between the two matrices as measured by the 2-norm is minimized:

$$\Delta = \|A - \hat{A}\|_2 \tag{15.11}$$

The 2-norm for matrices is the equivalent of Euclidean distance for vectors. SVD is in fact very similar to fitting a line, a one-dimensional object, to a set of points, which exists in the two-dimensional plane. Figure 15.7 indicates with arrows which point on the one-dimensional line each of the original points corresponds to.

Just as the linear regression in figure 15.7 can be interpreted as projecting a two-dimensional space onto a one-dimensional line, so does SVD project an m-dimensional space onto a k-dimensional space where $k \ll m$. In our application (word-document matrices), m is the number of word types in the collection. Values of k that are frequently chosen are 100 and 150. The projection transforms a document's vector in m-dimensional word space into a vector in the k-dimensional reduced space.

One possible source of confusion is that equation (15.11) compares the original matrix and a lower-dimensional approximation. Shouldn't the second matrix have fewer rows and columns, which would make equation (15.11) ill-defined? The analogy with line fitting is again helpful here. The fitted line exists in two dimensions, but it is a one-dimensional object. The same is true for $\hat{A}$: it is a matrix of lower rank, that is, it could be represented in a lower-dimensional space by transforming the axes of the space. But for the particular axes chosen it has the same number of rows and columns as A.

The SVD projection is computed by decomposing the document-by-

$$T = \begin{array}{l|rrrrr} & \text{Dim. 1} & \text{Dim. 2} & \text{Dim. 3} & \text{Dim. 4} & \text{Dim. 5} \\ \hline \text{cosmonaut} & -0.44 & -0.30 & 0.57 & 0.58 & 0.25 \\ \text{astronaut} & -0.13 & -0.33 & -0.59 & 0.00 & 0.73 \\ \text{moon} & -0.48 & -0.51 & -0.37 & 0.00 & -0.61 \\ \text{car} & -0.70 & 0.35 & 0.15 & -0.58 & 0.16 \\ \text{truck} & -0.26 & 0.65 & -0.41 & 0.58 & -0.09 \end{array}$$

Figure 15.8 The matrix T of the SVD of the matrix in figure 15.5. Values are rounded.

$$S = \begin{pmatrix} 2.16 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.59 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.28 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.39 \end{pmatrix}$$

Figure 15.9 The matrix of singular values of the SVD of the matrix in figure 15.5. Values are rounded.

term matrix $A_{t\times d}$ into the product of three matrices,[1] $T_{t\times n}$, $S_{n\times n}$, and $D_{d\times n}$:

$$A_{t\times d} = T_{t\times n} S_{n\times n} (D_{d\times n})^{\mathrm{T}} \tag{15.12}$$

where $n = \min(t, d)$. We indicate dimensionality by subscripts: A has t rows and d columns, T has t rows and n columns and so on. D^{T} is the transpose of D, the matrix D rotated around its diagonal: $D_{ij} = (D^{\mathrm{T}})_{ji}$.

Examples of A, T, S, and D are given in figure 15.5 and figures 15.8 through 15.10. Figure 15.5 shows an example of A. A contains the document vectors with each column corresponding to one document. In other words, element a_{ij} of the matrix records how often term i occurs in document j. The counts should be appropriately weighted (as discussed in section 15.2). For simplicity of exposition, we have not applied weighting and assumed term frequencies of 1.

1. Technically, this is the definition of the so-called 'reduced SVD.' The full SVD takes the form $A_{t\times d} = T_{t\times t} S_{t\times d} (D_{d\times d})^{\mathrm{T}}$, where the extra rows or columns of S are zero vectors, and T and D are square orthogonal matrices (Trefethen and Bau 1997: 27).

$$D^{\mathrm{T}} = \begin{pmatrix} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \text{Dimension 1} & -0.75 & -0.28 & -0.20 & -0.45 & -0.33 & -0.12 \\ \text{Dimension 2} & -0.29 & -0.53 & -0.19 & 0.63 & 0.22 & 0.41 \\ \text{Dimension 3} & 0.28 & -0.75 & 0.45 & -0.20 & 0.12 & -0.33 \\ \text{Dimension 4} & 0.00 & 0.00 & 0.58 & 0.00 & -0.58 & 0.58 \\ \text{Dimension 5} & -0.53 & 0.29 & 0.63 & 0.19 & 0.41 & -0.22 \end{pmatrix}$$

Figure 15.10 The matrix D^{T} of the SVD of the matrix in figure 15.5. Values are rounded.

ORTHONORMAL

Figures 15.8 and 15.10 show T and D, respectively. These matrices have *orthonormal* columns. This means that the column vectors have unit length and are all orthogonal to each other. (If a matrix C has orthonormal columns, then $C^{\mathrm{T}}C = I$, where I is the diagonal matrix with a diagonal of 1's, and zeroes elsewhere. So we have $T^{\mathrm{T}}T = D^{\mathrm{T}}D = I$.)

We can view SVD as a method for rotating the axes of the n-dimensional space such that the first axis runs along the direction of largest variation among the documents, the second dimension runs along the direction with the second largest variation and so forth. The matrices T and D represent terms and documents in this new space. For example, the first row of T corresponds to the first row of A, and the first column of D^T corresponds to the first column of A.

The diagonal matrix S contains the singular values of A in descending order (as in figure 15.9). The i^{th} singular value indicates the amount of variation along the i^{th} axis. By restricting the matrices T, S, and D to their first $k < n$ columns one obtains the matrices $T_{t\times k}$, $S_{k\times k}$, and $(D_{d\times k})^{\mathrm{T}}$. Their product $\hat{A}$ is the best least squares approximation of A by a matrix of rank k in the sense defined in equation (15.11). One can also prove that SVD is 'almost' unique, that is, there is only one possible decomposition of a given matrix.[2] See Golub and van Loan (1989) for an extensive treatment of SVD including a proof of the optimality property.

That SVD finds the optimal projection to a low-dimensional space is the

2. For any given SVD solution, you can get additional non-identical ones by flipping signs in corresponding left and right singular vectors of T and D, and, if there are two or more identical singular values, then the subspace determined by the corresponding singular vectors is unique, but can be described by any appropriate orthonormal basis vectors. But, apart from these cases, SVD is unique.

	d_1	d_2	d_3	d_4	d_5	d_6
Dimension 1	−1.62	−0.60	−0.44	−0.97	−0.70	−0.26
Dimension 2	−0.46	−0.84	−0.30	1.00	0.35	0.65

Figure 15.11 The matrix $B_{2\times d} = S_{2\times 2} D^{\mathrm{T}}{}_{2\times d}$ of documents after rescaling with singular values and reduction to two dimensions. Values are rounded.

	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1.00					
d_2	0.78	1.00				
d_3	0.95	0.94	1.00			
d_4	0.47	−0.18	0.17	1.00		
d_5	0.74	0.16	0.49	0.94	1.00	
d_6	0.10	−0.54	−0.22	0.93	0.75	1.00

Table 15.9 The matrix of document correlations $E^{\mathrm{T}}E$ where E is B with length-normalized columns. For example, the normalized correlation coefficient of d_3 and d_2 (when represented as in figure 15.11) is 0.88. Values are rounded.

key property for exploiting word co-occurrence patterns. SVD represents terms and documents in the lower dimensional space as well as possible. In the process, some words that have similar co-occurrence patterns are projected (or collapsed) onto the same dimension. As a consequence, the similarity metric will make topically similar documents and queries come out as similar even if different words are used for describing the topic. If we restrict the matrix in figure 15.8 to the first two dimensions, we end up with two groups of terms: space exploration terms (*cosmonaut*, *astronaut*, and *moon*) which have negative values on the second dimension and automobile terms (*car* and *truck*) which have positive values on the second dimension. The second dimension directly reflects the different co-occurrence patterns of these two groups: space exploration terms only co-occur with other space exploration terms, automobile terms only co-occur with other automobile terms (with one exception: the occurrence of *car* in d_1). In some cases, we will be misled by such co-occurrence patterns and wrongly infer semantic similarity. However, in most cases co-occurrence is a valid indicator of topical relatedness.

These term similarities have a direct impact on document similarity. Let us assume a reduction to two dimensions. After rescaling with the singular values, we get the matrix $B = S_{2\times 2} D^{\mathrm{T}}{}_{2\times d}$ shown in figure 15.11,

where $S_{2\times2}$ is S restricted to two dimensions (with the diagonal elements 2.16, 1.59). Matrix B is a reduced dimensionality representation of the documents in the original matrix A, and is what was shown in figure 15.6.

Table 15.9 shows the similarities between documents when they are represented in this new space. Not surprisingly, there is high similarity between d_1 and d_2 (0.78) and d_4, d_5, and d_6 (0.94, 0.93, 0.75). These document similarities are about the same in the original space (i.e., when we compute correlations for the original document vectors in figure 15.5). The key change is that d_2 and d_3, whose similarity is 0.00 in the original space, are now highly similar (0.94). Although d_2 and d_3 have no common terms, they are now recognized as being topically similar because of the co-occurrence patterns in the corpus.

Notice that we get the same similarity as in the original space (that is, zero similarity) if we compute similarity in the transformed space without any dimensionality reduction. Using the full vectors from figure 15.10 and rescaling them with the appropriate singular values we get:

$$-0.28 \times -0.20 \times 2.16^2 + -0.53 \times -0.19 \times 1.59^2 +$$
$$-0.75 \times 0.45 \times 1.28^2 + 0.00 \times 0.58 \times 1.00^2 + 0.29 \times 0.63 \times 0.39^2 \approx 0.00$$

(If you actually compute this expression, you will find that the answer is not quite zero, but this is only because of rounding errors. But this is as good a point as any to observe that many matrix computations are quite sensitive to rounding errors.)

We have computed document similarity in the reduced space using the product of S and D^T. The correctness of this procedure can be seen by looking at A^TA, which is the matrix of all document correlations for the original space:

$$A^TA = (TSD^T)^TTSD^T = DS^TT^TTSD^T = DS^TSD^T = (SD^T)^T(SD^T) = B^TB \tag{15.13}$$

Because T has orthonormal columns, we have $T^TT = I$. Furthermore, since S is diagonal, $S = S^T$. Term similarities are computed analogously since one observes that the term correlations are given by:

$$AA^T = TSD^T(TSD^T)^T = TSD^TDS^TT^T = (TS)(TS)^T \tag{15.14}$$

One remaining problem for a practical application is how to fold queries and new documents into the reduced space. The SVD computation only gives us reduced representations for the document vectors in matrix A. We do not want to do a completely new SVD every time a new

query is launched. In addition, in order to handle large corpora efficiently we may want to do SVD for only a sample of the documents (for example a third or a fourth). The remaining documents would then be folded in.

The equation for folding documents into the space can again be derived from the basic SVD equation:

$$A = TSD^{\mathrm{T}} \tag{15.15}$$
$$\Leftrightarrow \quad T^{\mathrm{T}}A = T^{\mathrm{T}}TSD^{\mathrm{T}}$$
$$\Leftrightarrow \quad T^{\mathrm{T}}A = SD^{\mathrm{T}}$$

So we just multiply the query or document vector with the transpose of the term matrix T (after it has been truncated to the desired dimensionality). For example, for a query vector $\vec{q}$ and a reduction to dimensionality k, the query representation in the reduced space is ${T_{t\times k}}^{\mathrm{T}}\vec{q}$.

15.4.3 Latent Semantic Indexing in IR

LATENT SEMANTIC INDEXING

The application of SVD to information retrieval was originally proposed by a group of researchers at Bellcore (Deerwester et al. 1990) and called *Latent Semantic Indexing* (LSI) in this context. LSI has been compared to standard vector space search on several document collections. It was found that LSI performs better than vector space search in many cases, especially for high-recall searches (Deerwester et al. 1990; Dumais 1995). LSI's strength in high-recall searches is not surprising since a method that takes co-occurrence into account is expected to achieve higher recall. On the other hand, due to the noise added by spurious co-occurrence data one sometimes finds a decrease in precision.

The appropriateness of LSI also depends on the document collection. Recall the example of the vocabulary problem in figure 15.8. In a heterogeneous collection, documents may use different words to refer to the same topic like *HCI* and *user interface* in the figure. Here, LSI can help identify the underlying semantic similarity between seemingly dissimilar documents. However, in a collection with homogeneous vocabulary, LSI is less likely to be useful.

The application of SVD to information retrieval is called *Latent Semantic Indexing* because the document representations in the original term space are transformed to representations in a new reduced space. The dimensions in the reduced space are linear combinations of the original dimensions (this is so since matrix multiplications as in equation (15.16)