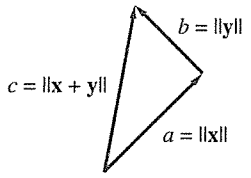

Figure 5.1.4.

In many applications, the cosine of the angle between two nonzero vectors is used as a measure of how closely the directions of the vectors match up. If $\cos\theta$ is near 1, then the angle between the vectors is small and hence the vectors are in nearly the same direction. A cosine value near zero would indicate that the angle between the vectors is nearly a right angle.

APPLICATION I Information Retrieval Revisited

In Section 4 of Chapter 1, we considered the problem of searching a database for documents that contain certain key words. If there are m possible key search words and a total of n documents in the collection, then the database can be represented by an $m \times n$ matrix A. Each column of A represents a document in the database. The entries of the jth column correspond to the relative frequencies of the key words in the jth document.

Refined search techniques must deal with vocabulary disparities and the complexities of language. Two of the main problems are *polysemy* (words having multiple meanings) and *synonymy* (multiple words having the same meaning). On the one hand, some of the words that you are searching for may have multiple meanings and could appear in contexts that are completely irrelevant to your particular search. For example, the word *calculus* would occur frequently in both mathematical papers and in dentistry papers. On the other hand, most words have synonyms, and it is possible that many of the documents may use the synonyms rather than the specified search words. For example, you could search for an article on rabies using the key word *dogs*; however, the author of the article may have preferred to use the word *canines* throughout the paper. To handle these problems, we need a technique to find the documents that best match the list of search words without necessarily matching every word on the list. We want to pick out the column vectors of the database matrix that most closely match a given search vector. To do this, we use the cosine of the angle between two vectors as a measure of how closely the vectors match up.

In practice, both m and n are quite large, as there are many possible key words and many documents to search. For simplicity, let us consider an example where $m = 10$ and $n = 8$. Suppose that a Web site has eight modules for learning linear algebra and each module is located on a separate Web page. Our list of possible search words consists of

determinants, eigenvalues, linear, matrices, numerical,
orthogonality, spaces, systems, transformations, vector

(This list of key words was compiled from the chapter headings for this book.) Table 1 shows the frequencies of the key words in each of the modules. The (2, 6) entry of the table is 5, which indicates that the key word *eigenvalues* appears five times in the sixth module.

Table 1 Frequency of Key Words

	Modules							
Key words	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
determinants	0	6	3	0	1	0	1	1
eigenvalues	0	0	0	0	0	5	3	2
linear	5	4	4	5	4	0	3	3
matrices	6	5	3	3	4	4	3	2
numerical	0	0	0	0	3	0	4	3
orthogonality	0	0	0	0	4	6	0	2
spaces	0	0	5	2	3	3	0	1
systems	5	3	3	2	4	2	1	1
transformations	0	0	0	5	1	3	1	0
vector	0	4	4	3	4	1	0	3

The database matrix is formed by scaling each column of the table so that all column vectors are unit vectors. Thus, if A is the matrix corresponding to Table 1, then the columns of the database matrix Q are determined by setting

$$\mathbf{q}_j = \frac{1}{\|\mathbf{a}_j\|}\mathbf{a}_j \quad j = 1, \ldots, 8$$

To do a search for the key words *orthogonality*, *spaces*, and *vector*, we form a search vector $\mathbf{x}$ whose entries are all 0 except for the three rows corresponding to the search words. To obtain a unit search vector, we put $\frac{1}{\sqrt{3}}$ in each of the rows corresponding to the search words. For this example, the database matrix Q and search vector $\mathbf{x}$ (with entries rounded to three decimal places) are given by

$$Q = \begin{bmatrix} 0.000 & 0.594 & 0.327 & 0.000 & 0.100 & 0.000 & 0.147 & 0.154 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.500 & 0.442 & 0.309 \\ 0.539 & 0.396 & 0.436 & 0.574 & 0.400 & 0.000 & 0.442 & 0.463 \\ 0.647 & 0.495 & 0.327 & 0.344 & 0.400 & 0.400 & 0.442 & 0.309 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.300 & 0.000 & 0.590 & 0.463 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.400 & 0.600 & 0.000 & 0.309 \\ 0.000 & 0.000 & 0.546 & 0.229 & 0.300 & 0.300 & 0.000 & 0.154 \\ 0.539 & 0.297 & 0.327 & 0.229 & 0.400 & 0.200 & 0.147 & 0.154 \\ 0.000 & 0.000 & 0.000 & 0.574 & 0.100 & 0.300 & 0.147 & 0.000 \\ 0.000 & 0.396 & 0.436 & 0.344 & 0.400 & 0.100 & 0.000 & 0.463 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.577 \\ 0.577 \\ 0.000 \\ 0.000 \\ 0.577 \end{bmatrix}$$

If we set $\mathbf{y} = Q^T\mathbf{x}$, then

$$y_i = \mathbf{q}_i^T\mathbf{x} = \cos\theta_i$$

where θ_i is the angle between the unit vectors $\mathbf{x}$ and $\mathbf{q}_i$. For our example,

$$\mathbf{y} = (0.000,\ 0.229,\ 0.567,\ 0.331,\ 0.635,\ 0.577,\ 0.000,\ 0.535)^T$$

Since $y_5 = 0.635$ is the entry of $\mathbf{y}$ that is closest to 1, the direction of the search vector $\mathbf{x}$ is closest to the direction of $\mathbf{q}_5$ and hence module 5 is the one that best matches

our search criteria. The next-best matches come from modules 6 ($y_6 = 0.577$) and 3 ($y_3 = 0.567$). If a document doesn't contain any of the search words, then the corresponding column vector of the database matrix will be orthogonal to the search vector. Note that modules 1 and 7 do not have any of the three search words, and consequently

$$y_1 = \mathbf{q}_1^T\mathbf{x} = 0 \quad \text{and} \quad y_7 = \mathbf{q}_7^T\mathbf{x} = 0$$

This example illustrates some of the basic ideas behind database searches. Using modern matrix techniques, we can improve the search process significantly. We can speed up searches and at the same time correct for errors due to polysemy and synonymy. These advanced techniques are referred to as *latent semantic indexing* (LSI) and depend on a matrix factorization, the *singular value decomposition*, which we will discuss in Section 5 of Chapter 6.

There are many other important applications involving angles between vectors. In particular, statisticians use the cosine of the angle between two vectors as a measure of how closely the two vectors are correlated.

APPLICATION 2 Statistics—Correlation and Covariance Matrices

Suppose that we wanted to compare how closely exam scores for a class correlate with scores on homework assignments. As an example, we consider the total scores on assignments and tests of a mathematics class at the University of Massachusetts Dartmouth. The total scores for homework assignments during the semester for the class are given in the second column of Table 2. The third column represents the total scores for the two exams given during the semester, and the last column contains the scores on the final exam. In each case, a perfect score would be 200 points. The last row of the table summarizes the class averages.

Table 2 Math Scores Fall 1996

	Scores		
Student	**Assignments**	**Exams**	**Final**
S1	198	200	196
S2	160	165	165
S3	158	158	133
S4	150	165	91
S5	175	182	151
S6	134	135	101
S7	152	136	80
Average	161	163	131

We would like to measure how student performance compares between each set of exam or assignment scores. To see how closely the two sets of scores are correlated and allow for any differences in difficulty, we need to adjust the scores so that each test has a mean of 0. If, in each column, we subtract the average score from each of the