Downloaded 09/05/14 to 144.92.179.191. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Chapter 2
Vectors and Matrices

We will assume that the basic notions of linear algebra are known to the reader. For completeness, some will be recapitulated here.

2.1 Matrix-Vector Multiplication

How basic operations in linear algebra are defined is important, since it influences one's mental images of the abstract notions. Sometimes one is led to thinking that the operations should be done in a certain order, when instead the *definition as such* imposes no ordering.[3] Let A be an $m \times n$ matrix. Consider the definition of matrix-vector multiplication:

$$y = Ax, \qquad y_i = \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, \ldots, m. \tag{2.1}$$

Symbolically one can illustrate the definition

$$\begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} = \begin{pmatrix} \leftarrow & - & - & \rightarrow \\ \leftarrow & - & - & \rightarrow \\ \leftarrow & - & - & \rightarrow \\ \leftarrow & - & - & \rightarrow \end{pmatrix} \begin{pmatrix} \uparrow \\ | \\ | \\ \downarrow \end{pmatrix}. \tag{2.2}$$

It is obvious that the computation of the different components of the vector y are completely independent of each other and can be done in any order. However, the definition may lead one to think that the matrix should be accessed rowwise, as illustrated in (2.2) and in the following MATLAB code:

[3] It is important to be aware that on modern computers, which invariably have memory hierarchies, the order in which operations are performed is often critical for the performance. However, we will not pursue this aspect here.

Downloaded 09/05/14 to 144.92.179.191. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

```
for i=1:m
  y(i)=0;
  for j=1:n
    y(i)=y(i)+A(i,j)*x(j);
  end
end
```

Alternatively, we can write the operation in the following way. Let $a_{\cdot j}$ be a column vector of A. Then we can write

$$y = Ax = \begin{pmatrix} a_{\cdot 1} & a_{\cdot 2} & \cdots & a_{\cdot n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \sum_{j=1}^{n} x_j a_{\cdot j}.$$

This can be illustrated symbolically:

$$\begin{pmatrix} \uparrow \\ | \\ | \\ \downarrow \end{pmatrix} = \begin{pmatrix} \uparrow & \uparrow & \uparrow & \uparrow \\ | & | & | & | \\ | & | & | & | \\ \downarrow & \downarrow & \downarrow & \downarrow \end{pmatrix} \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix}. \qquad (2.3)$$

Here the vectors are accessed columnwise. In MATLAB, this version can be written[4]

```
for i=1:m
    y(i)=0;
end
for j=1:n
    for i=1:m
        y(i)=y(i)+A(i,j)*x(j);
    end
end
```

or, equivalently, using the vector operations of MATLAB,

```
y(1:m)=0;
for j=1:n
    y(1:m)=y(1:m)+A(1:m,j)*x(j);
end
```

Thus the two ways of performing the matrix-vector multiplication correspond to changing the order of the loops in the code. This way of writing also emphasizes the view of the column vectors of A as *basis vectors* and the components of x as *coordinates* with respect to the basis.

[4] In the terminology of LAPACK [1] this is the SAXPY version of matrix-vector multiplication. SAXPY is an acronym from the Basic Linear Algebra Subroutine (BLAS) library.

Downloaded 09/05/14 to 144.92.179.191. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2.2 Matrix-Matrix Multiplication

Matrix multiplication can be done in several ways, each representing a different access pattern for the matrices. Let $A \in \mathbb{R}^{m\times k}$ and $B \in \mathbb{R}^{k\times n}$. The definition of matrix multiplication is

$$\mathbb{R}^{m\times n} \ni C = AB = (c_{ij}),$$

$$c_{ij} = \sum_{s=1}^{k} a_{is} b_{sj}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n. \tag{2.4}$$

In a comparison to the definition of matrix-vector multiplication (2.1), we see that in matrix multiplication *each column vector in B is multiplied by A.*

We can formulate (2.4) as a matrix multiplication code

```
for i=1:m
  for j=1:n
    for s=1:k
      C(i,j)=C(i,j)+A(i,s)*B(s,j)
    end
  end
end
```

This is an inner product version of matrix multiplication, which is emphasized in the following equivalent code:

```
for i=1:m
  for j=1:n
    C(i,j)=A(i,1:k)*B(1:k,j)
  end
end
```

It is immediately seen that the the loop variables can be permuted in $3! = 6$ different ways, and we can write a *generic matrix multiplication code*:

```
for ...
  for ...
    for ...
      C(i,j)=C(i,j)+A(i,s)*B(s,j)
    end
  end
end
```

A column-oriented (or SAXPY) version is given in

```
for j=1:n
  for s=1:k
    C(1:m,j)=C(1:m,j)+A(1:m,s)*B(s,j)
  end
end
```

Downloaded 09/05/14 to 144.92.179.191. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The matrix A is accessed by columns and B by scalars. This access pattern can be illustrated as

$$\begin{pmatrix} & \uparrow & \\ & | & \\ & | & \\ & \downarrow & \end{pmatrix} = \begin{pmatrix} \uparrow & \uparrow & \uparrow & \uparrow \\ | & | & | & | \\ | & | & | & | \\ \downarrow & \downarrow & \downarrow & \downarrow \end{pmatrix} \begin{pmatrix} & & \times & \\ & & \times & \\ & & \times & \\ & & \times & \end{pmatrix}$$

In another permutation we let the `s`-loop be the outermost:

```
for s=1:k
  for j=1:n
    C(1:m,j)=C(1:m,j)+A(1:m,s)*B(s,j)
  end
end
```

This can be illustrated as follows. Let $a_{\cdot k}$ denote the column vectors of A and let $b_{k\cdot}^T$ denote the row vectors of B. Then matrix multiplication can be written as

$$C = AB = \begin{pmatrix} a_{\cdot 1} & a_{\cdot 2} & \dots & a_{\cdot k} \end{pmatrix} \begin{pmatrix} b_{1\cdot}^T \\ b_{2\cdot}^T \\ \vdots \\ b_{k\cdot}^T \end{pmatrix} = \sum_{s=1}^{k} a_{\cdot s} b_{s\cdot}^T. \tag{2.5}$$

This is the *outer product* form of matrix multiplication. Remember that the outer product follows the standard definition of matrix multiplication: let x and y be column vectors in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively; then

$$\begin{aligned} xy^T &= \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \begin{pmatrix} y_1 & y_2 & \cdots & y_n \end{pmatrix} = \begin{pmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_n \\ x_2y_1 & x_2y_2 & \cdots & x_2y_n \\ \vdots & \vdots & & \vdots \\ x_my_1 & x_my_2 & \cdots & x_my_n \end{pmatrix} \\ &= \begin{pmatrix} y_1x & y_2x & \cdots & y_nx \end{pmatrix} = \begin{pmatrix} x_1y^T \\ x_2y^T \\ \vdots \\ x_my^T \end{pmatrix}. \end{aligned}$$

Writing the matrix $C = AB$ in the outer product form (2.5) can be considered as an *expansion* of C in terms of simple matrices $a_{\cdot s}b_{s\cdot}^T$. We will later see that such matrices have *rank* equal to 1.

Downloaded 09/05/14 to 144.92.179.191. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2.3 Inner Product and Vector Norms

In this section we will discuss briefly how to measure the "size" of a vector. The most common vector norms are

$$\begin{aligned} \| x \|_1 &= \sum_{i=1}^{n} |x_i|, \quad \text{1-norm,} \\ \| x \|_2 &= \sqrt{\sum_{i=1}^{n} x_i^2}, \quad \text{Euclidean norm (2-norm),} \\ \| x \|_\infty &= \max_{1 \le i \le n} |x_i|, \quad \text{max-norm.} \end{aligned}$$

The Euclidean vector norm is the generalization of the standard Euclidean distance in $\mathbb{R}^3$ to $\mathbb{R}^n$. All three norms defined here are special cases of the p-norm:

$$\| x \|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

Associated with the Euclidean vector norm is the *inner product* between two vectors x and y in $\mathbb{R}^n$, which is defined

$$(x, y) = x^T y.$$

Generally, a *vector norm* is a mapping $\mathbb{R}^n \to \mathbb{R}$ with the properties

$$\begin{aligned} \| x \| &\ge 0 \text{ for all } x, \\ \| x \| &= 0 \text{ if and only if } x = 0, \\ \| \alpha x \| &= |\alpha| \, \| x \|, \; \alpha \in \mathbb{R}, \\ \| x + y \| &\le \| x \| + \| y \|, \text{ the triangle inequality.} \end{aligned}$$

With norms we can introduce the concepts of continuity and error in approximations of vectors. Let $\bar{x}$ be an approximation of the vector x. The for any given vector norm, we define the *absolute error*

$$\| \delta x \| = \| \bar{x} - x \|$$

and the *relative error* (assuming that $x \neq 0$)

$$\frac{\| \delta x \|}{\| x \|} = \frac{\| \bar{x} - x \|}{\| x \|}.$$

In a finite dimensional vector space all vector norms are equivalent in the sense that for any two norms $\| \cdot \|_\alpha$ and $\| \cdot \|_\beta$ there exist constants m and M such that

$$m \| x \|_\alpha \le \| x \|_\beta \le M \| x \|_\alpha, \tag{2.6}$$

Downloaded 09/05/14 to 144.92.179.191. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where m and M do not depend on x. For example, with $x \in \mathbb{R}^n$,

$$\| x \|_2 \leq \| x \|_1 \leq \sqrt{n} \, \| x \|_2.$$

This equivalence implies that if a sequence of vectors $(x_i)_{i=1}^{\infty}$ converges to x^* in one norm,

$$\lim_{i \to \infty} \| x_i - x^* \| = 0,$$

then it converges to the same limit in all norms.

In data mining applications it is common to use the *cosine of the angle* between two vectors as a distance measure:

$$\cos \theta(x, y) = \frac{x^T y}{\| x \|_2 \, \| y \|_2}.$$

With this measure two vectors are close if the cosine is close to one. Similarly, x and y are *orthogonal* if the angle between them is $\pi/2$, i.e., $x^T y = 0$.

2.4 Matrix Norms

For any vector norm we can define a corresponding *operator norm*. Let $\| \cdot \|$ be a vector norm. The corresponding *matrix norm* is defined as

$$\| A \| = \sup_{x \neq 0} \frac{\| Ax \|}{\| x \|}.$$

One can show that such a matrix norm satisfies (for $\alpha \in \mathbb{R}$)

$$\begin{aligned} \| A \| &\geq 0 \text{ for all } A, \\ \| A \| &= 0 \text{ if and only if } A = 0, \\ \| \alpha A \| &= |\alpha| \, \| A \|, \ \alpha \in \mathbb{R}, \\ \| A + B \| &\leq \| A \| + \| B \|, \text{ the triangle inequality.} \end{aligned}$$

For a matrix norm defined as above the following fundamental inequalities hold.

Proposition 2.1. *Let $\| \cdot \|$ denote a vector norm and the corresponding matrix norm. Then*

$$\begin{aligned} \| Ax \| &\leq \| A \| \, \| x \|, \\ \| AB \| &\leq \| A \| \, \| B \|. \end{aligned}$$

Proof. From the definition we have

$$\frac{\| Ax \|}{\| x \|} \leq \| A \|$$

Downloaded 09/05/14 to 144.92.179.191. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

for all $x \neq 0$, which gives the first inequality. The second is proved by using the first twice for $\| ABx \|$. □

One can show that the 2-*norm* satisfies

$$\| A \|_2 = \left(\max_{1 \le i \le n} \lambda_i(A^T A) \right)^{1/2},$$

i.e., the square root of the largest eigenvalue of the matrix $A^T A$. Thus it is a comparatively heavy computation to obtain $\| A \|_2$ for a given matrix (of medium or large dimensions). It is considerably easier to compute the *matrix infinity norm* (for $A \in \mathbb{R}^{m \times n}$),

$$\| A \|_\infty = \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}|,$$

and the *matrix* 1-*norm*

$$\| A \|_1 = \max_{1 \le j \le n} \sum_{i=1}^{m} |a_{ij}|.$$

In Section 6.1 we will see that the 2-norm of a matrix has an explicit expression in terms of the singular values of A.

Let $A \in \mathbb{R}^{m \times n}$. In some cases we will treat the matrix not as a linear operator but rather as a point in a space of dimension mn, i.e., $\mathbb{R}^{mn}$. Then we can use the *Frobenius* matrix norm, which is defined by

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}. \tag{2.7}$$

Sometimes it is practical to write the Frobenius norm in the equivalent form

$$\| A \|_F^2 = \operatorname{tr}(A^T A), \tag{2.8}$$

where the *trace* of a matrix $B \in \mathbb{R}^{n \times n}$ is the sum of its diagonal elements,

$$\operatorname{tr}(B) = \sum_{i=1}^{n} b_{ii}.$$

The Frobenius norm does not correspond to a vector norm, so it is not an operator norm in that sense. This norm has the advantage that it is easier to compute than the 2-norm. The Frobenius *matrix norm* is actually closely related to the Euclidean *vector norm* in the sense that it is the Euclidean vector norm on the (linear space) of matrices $\mathbb{R}^{m \times n}$, when the matrices are identified with elements in $\mathbb{R}^{mn}$.

2.5 Linear Independence: Bases

Given a set of vectors $(v_j)_{j=1}^n$ in $\mathbb{R}^m$, $m \geq n$, consider the set of linear combinations

$$\operatorname{span}(v_1, v_2, \ldots, v_n) = \left\{ y \mid y = \sum_{j=1}^{n} \alpha_j v_j \right\}$$

for arbitrary coefficients α_j. The vectors $(v_j)_{j=1}^n$ are called *linearly independent* when

$$\textstyle\sum_{j=1}^n \alpha_j v_j = 0 \ \text{ if and only if } \ \alpha_j = 0 \text{ for } j = 1, 2, \ldots, n.$$

A set of m linearly independent vectors in $\mathbb{R}^m$ is called a *basis* in $\mathbb{R}^m$: any vector in $\mathbb{R}^m$ can be expressed as a linear combination of the basis vectors.

Proposition 2.2. *Assume that the vectors $(v_j)_{j=1}^n$ are linearly dependent. Then some v_k can be written as linear combinations of the rest, $v_k = \sum_{j \neq k} \beta_j v_j$.*

Proof. There exist coefficients α_j with some $\alpha_k \neq 0$ such that

$$\sum_{j=1}^{n} \alpha_j v_j = 0.$$

Take an $\alpha_k \neq 0$ and write

$$\alpha_k v_k = \sum_{j \neq k} -\alpha_j v_j,$$

which is the same as

$$v_k = \sum_{j \neq k} \beta_j v_j$$

with $\beta_j = -\alpha_j / \alpha_k$. □

If we have a set of linearly dependent vectors, then we can keep a linearly independent subset and express the rest in terms of the linearly independent ones. Thus we can consider the number of linearly independent vectors as a measure of the information contents of the set and compress the set accordingly: take the linearly independent vectors as representatives (basis vectors) for the set, and compute the coordinates of the rest in terms of the basis. However, in real applications we seldom have *exactly linearly dependent vectors* but rather *almost linearly dependent vectors*. It turns out that for such a *data reduction procedure* to be practical and numerically stable, we need the basis vectors to be not only linearly independent but orthogonal. We will come back to this in Chapter 4.

Downloaded 09/05/14 to 144.92.179.191. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 09/05/14 to 144.92.179.191. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2.6 The Rank of a Matrix

The *rank* of a matrix is defined as the maximum number of linearly independent column vectors. It is a standard result in linear algebra that the number of linearly independent column vectors is equal to the number of linearly independent row vectors.

We will see later that any matrix can be represented as an expansion of rank-1 matrices.

Proposition 2.3. *An outer product matrix xy^T, where x and y are vectors in $\mathbb{R}^n$, has rank* 1.

Proof.

$$xy^T = \begin{pmatrix} y_1x & y_2x & \cdots & y_nx \end{pmatrix} = \begin{pmatrix} x_1y^T \\ x_2y^T \\ \vdots \\ x_ny^T \end{pmatrix}.$$

Thus, all the columns (rows) of xy^T are linearly dependent. $\square$

A square matrix $A \in \mathbb{R}^{n\times n}$ with rank n is called *nonsingular* and has an *inverse* A^{-1} satisfying

$$AA^{-1} = A^{-1}A = I.$$

If we multiply linearly independent vectors by a nonsingular matrix, then the vectors remain linearly independent.

Proposition 2.4. *Assume that the vectors $v_1, \dots, v_p$ are linearly independent. Then for any nonsingular matrix T, the vectors $Tv_1, \dots, Tv_p$ are linearly independent.*

Proof. Obviously $\sum_{j=1}^{p} \alpha_j v_j = 0$ if and only if $\sum_{j=1}^{p} \alpha_j T v_j = 0$ (since we can multiply any of the equations by T or T^{-1}). Therefore the statement follows. $\square$