Applications of Matrices and Linear Algebra

1 Curves and Surfaces Through Given Points

In this section and the ones that follow, we describe a number of applications of matrices and linear algebra. Most of these examples come from the text [1]. Other examples are also available on the web site.

1.1 Two Points on a Line

We are given two points in the plane, say (x_1, y_1) and (x_2, y_2), and we want to find the equation of the line determined by these two points. Let the equation of the line be

$$c_1x + c_2y + c_3 = 0,$$

where c_1, c_2, and c_3 are to be determined. We write

$$x_1c_1 + y_1c_2 + 1c_3 = 0,$$

$$x_2c_1 + y_2c_2 + 1c_3 = 0,$$

and

$$xc_1 + yc_2 + 1c_3 = 0.$$

Since there is to be a non-trivial solution, the matrix

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x & y & 1 \end{bmatrix} \tag{1.1}$$

must have determinant equal to zero. Therefore

$$x(y_1 - y_2) - y(x_1 - x_2) + (x_1y_2 - x_2y_1) = 0,$$

and we can choose

$$c_1 = (y_1 - y_2),$$

$$c_2 = (x_2 - x_1),$$

and

$$c_3 = (x_1y_2 - x_2y_1).$$

1.2 A Circle Through Three Given Points

We are given the three non-collinear points (x_1, y_1), (x_2, y_2), and (x_3, y_3) and want to find the circle containing these points. Any circle in the plane can be described by an equation of the form

$$(x-a)^2 + (y-b)^2 = r^2.$$

Rewriting this as

$$(x^2 + y^2) - 2ax - 2by + a^2 + b^2 - r^2 = 0,$$

we see that the points (x, y) on the circle must satisfy the equation

$$c_1(x^2 + y^2) + c_2 x + c_3 y + c_4 = 0,$$

for some choice of the unknowns c_1, c_2, c_3, and c_4. We write

$$c_1(x_1^2 + y_1^2) + c_2 x_1 + c_3 y_1 + c_4 = 0,$$

$$c_1(x_2^2 + y_2^2) + c_2 x_2 + c_3 y_2 + c_4 = 0,$$

$$c_1(x_3^2 + y_3^2) + c_2 x_3 + c_3 y_3 + c_4 = 0,$$

and

$$c_1(x^2 + y^2) + c_2 x + c_3 y + c_4 = 0.$$

Once again, since there is to be a non-trivial solution, the determinant of the matrix

$$A = \begin{bmatrix} x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \\ x^2 + y^2 & x & y & 1 \end{bmatrix} \tag{1.2}$$

must be zero, and writing this out will give us an equation of the circle in the variables x and y.

1.3 Additional Examples

The same approach can be used to determine a conic through five points, a plane through three points, and a sphere through four points.

2 Allocation Problems

We have n different jobs to assign to n different people. For $i = 1, ..., n$ and $j = 1, ..., n$ the quantity C_{ij} is the cost of having person i do job j. The n by n matrix C with these entries is the *cost matrix*. An *assignment* is a selection of n entries of C so that no two are in the same column or the same row; that is, everybody gets one job. Our goal is to find an assignment that minimizes the total cost.

We know that there are $n!$ ways to make assignments, so one solution would be to determine the cost of each of these assignments and selection the cheapest. But for large n this is impractical. We want an algorithm that will solve the problem with less calculation. The algorithm we present here, discovered by two Hungarian mathematicians in the 1930's, is called The Hungarian Method.

To illustrate, suppose there are three people and three jobs, and the cost matrix is

$$C = \begin{bmatrix} 53 & 96 & 37 \\ 47 & 87 & 41 \\ 60 & 92 & 36 \end{bmatrix}. \tag{2.3}$$

The algorithm is as follows:

- **Step 1:** Subtract the minimum of each row from all the entries of that row. This is equivalent to saying that each person charges a certain amount just for participating, even before any assignments are made, and we must pay these costs in any case. Subtracting these necessary costs, which do not depend on the ultimate assignments, cannot change the optimal solutions.

  The new matrix is then

$$\begin{bmatrix} 16 & 59 & 0 \\ 6 & 46 & 0 \\ 24 & 56 & 0 \end{bmatrix}. \tag{2.4}$$

- **Step 2:** Subtract each column minimum from the entries of its column. This is equivalent to saying that each job has a minimum cost that we must pay, regardless of who performs it. As before, subtracting these necessary costs does not change the optimal solutions.

  The matrix becomes

$$\begin{bmatrix} 10 & 13 & 0 \\ 0 & 0 & 0 \\ 18 & 10 & 0 \end{bmatrix}. \tag{2.5}$$

- **Step 3:** Draw a line through the smallest number of rows and columns that results in all zeros being covered by a line; here I have put in boldface the entries covered by a line. The matrix becomes

$$\begin{bmatrix} 10 & 13 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 18 & 10 & \mathbf{0} \end{bmatrix}. \tag{2.6}$$

  We have used a total of two lines, one row and one column.

What we want is a set of n zeros with the property that each row contains one of them, and each column contains one of them; we shall call such a set an *optimal set of zeros*. Such a set of zeros will provide the desired cheapest assignments. The next two steps help us decide whether or not such a set of zeros currently exists.

- **Step 4:** If the number of lines just drawn is n we have finished. In our example, we are not finished. Proving that needing n lines to cover all the zeros means that there is an optimal set of zeros is difficult. It is easy to show that if there exists an optimal set of zeros, then n or more lines are necessary to cover all the zeros; just notice that any line can contain at most one member of an optimal set of zeros.

- **Step 5:** If, as in our example, the number of lines drawn is fewer than n, determine the smallest entry not yet covered by a line (not boldface, here). It is 10 in our example. Then subtract this number from all the uncovered entries and add it to all the entries covered by both a vertical and horizontal line. This step is equivalent to, first, subtracting the smallest entry from every row not completely covered by a line, and, second, adding this smallest entry to every column covered by a line. Since adding or subtracting a fixed amount from any row or column does not change the optimal solutions, we can return then to Step 3.

  Our matrix becomes

$$\begin{bmatrix} 0 & 3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 10 \\ 8 & 0 & \mathbf{0} \end{bmatrix}. \tag{2.7}$$

  Now return to Step 3.

In our example, when we return to Step 3 we find that we need three lines now and so we are finished. The optimal allocation is to assign the second person to the first job, the third person to the second job, and the first person to the third job. Generally, finding an optimal set of zeros for larger cost matrices, even when we know such a set must exist, is not a simple matter; there are computer algorithms to perform this task, however.

Exercise 2.1 *Apply this algorithm to the cost matrix*

$$C = \begin{bmatrix} 90 & 75 & 75 & 80 \\ 35 & 85 & 55 & 65 \\ 125 & 95 & 90 & 105 \\ 45 & 110 & 95 & 115 \end{bmatrix}. \tag{2.8}$$

3 Graph Theory

A *directed graph* is a set of symbols $\{P_1, P_2, ..., P_n\}$ called the *vertices* and a set of ordered pairs (P_i, P_j) for $P_i \neq P_j$, called the *edges* of the directed graph. Write $P_i \to P_j$ if and only if (P_i, P_j) is an edge. We represent this directed graph using the matrix M with $M_{ij} = 1$ if $P_i \to P_j$ and $M_{ij} = 0$ otherwise. See the download "Motivating Matrix Operations" for a discussion of *influence graphs* and *dominance-directed graphs*.

A subset of vertices is called a *clique* if and only if it has at least three members, $P_i \to P_j$ and $P_j \to P_i$ for each pair of vertices in the subset, and we cannot add a vertex to the subset without violating the second condition. Define a matrix S so that $S_{ij} = 1$ if and only if $P_i \to P_j$ and $P_j \to P_i$; otherwise $S_{ij} = 0$. Then the vertex P_i is a member of a clique if and only if $(S^3)_{ii} \neq 0$.

4 Game Theory

The use of a pay-off matrix in game theory provides a good application of linear algebra; see the pdf on the web site.

5 Markov Chains

Let $\{1, 2, ..., k\}$ be *states*. For $i = 1, ..., k$ and $j = 1, ..., k$ let $P_{ij} \geq 0$ be the probability of going from state j to state i in one step. The matrix P with entries P_{ij} is called the *transition matrix*. We begin with a column vector $x^0 = (x_1^0, ..., x_k^0)^T$, where x_i^0 is the probability that we start in state i. For $n = 1, 2, ...$ the vector x^n is the vector whose entry x_i^n is the probability of being in state i after n steps, given the initial probability vector x^0. We then have

$$x^{n+1} = Px^n.$$

We are often interested in the limiting behavior of the x^n.

For example, let

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{5.9}$$

and let $x^0 = [1\ \ 0]^T$. Then for n even we have $x^n = [1\ \ 0]^T$ and for n odd we have $x^n = [0\ \ 1]^T$.

We say that P is *regular* if, for some n, the matrix P^n has only positive entries. A basic theorem in Markov Chains is the following:

Theorem 5.1 *If P is regular, then there is a probability vector $q = (q_1, ..., q_k)$, with all entries positive, such that, as $n \to \infty$, $P^n_{ij} \to q_i$ for each j. Let Q be the matrix with $Q_{ij} = q_i$, for each i and j.*

So the limiting probability of going from j to i is independent of j. For any probability (column) vector x we have $Qx = q$. Also $Qq = q$, so q is an eigenvector of Q associated with eigenvalue $\lambda = 1$.

6 Hill Codes

Simple substitution codes are easily broken by frequency analysis. Here we consider a code involving matrices.

Number the letters of the alphabet from $A = 1$ to $Y = 25$ and $Z = 0$. To encode the sentence "I am hiding" , we first group the letters in pairs, as "ia mh id in gg". Then replace each letter by its number, to get

$$9\ 1, 13\ 8, 9\ 4, 9\ 14, 7\ 7,$$

and view each of these pairs of numbers as a two-by one vector. Select an encoding matrix A; in this case we use a two-by-two matrix

$$P = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}. \tag{6.10}$$

Now multiply each of the two-by-one vectors by A; in our example we get $[11\ \ 3]^T$, which is KC, $[29\ \ 24] = [3\ \ 24]^T$, which is CX, and so on. This is a Hill-2 cipher. To decode, we need A to be invertible modulo 26, which means that its determinant, modulo 26 must have an inverse, modulo 26, which will happen if and only if this determinant modulo 26 is not divisible by 2 or 13.

Exercise 6.1 *Find a decoding procedure for this Hill code by calculating a modulo-26 inverse for the matrix A.*

References

[1] Anton, H., and Rorres, C. (1986) *Elementary Linear Algebra and Applications*, Wiley and Sons, New York.

[2] Gale, D. (1960) *The Theory of Linear Economic Models*. New York: McGraw-Hill.