An Application of MATLAB in practical problems

XIE Huiyang
College of Sciences
Beijing Forestry University
Beijing, China
xhyang@bjfu.edu.cn

Bi Qiuxiang
Fixed-Income Department
GF securities CO., LTD
Guangzhou, China
gfbqx@tom.com

A*bstract*—This article gives an example of using Matlab to solve practical problems

Keywords-Linear Algebra; Matlab

I. INTRODUCTION

Matlab(Matrix Laboratory) is a commercial mathematical software developed by MathWorks for numerical computation. One of the nice features of Matlab is its ease of computations with vectors and matrices. Matrix operations, numerical analysis, signal processing, graphics, and user interface creation can be easily manipulated in Matlab. It also provides many implementations of algorithms and interface to other programming language. With help of additional toolboxes and package, Matlab even supports symbolic computation, graphics manipulation and model-base designing for dynamic and embedded systems.

Matlab is widely used in all areas of science and technology in education and research at universities. Many courses of mathematics and engineering are using Matlab for education on campus.

Linear algebra is a branch of mathematics, which is central to modern mathematics and incredibly useful in the modern world. It plays important roles in the researches of mathematics, mechanics, physics and other subjects, and it is widely applied to solve the real world problems. For most non-linear problems, the usual method is to approximate them by linear problems. People are only really good at solving linear problems by now due to the development and widespread use of the computer in areas that apply mathematics. By using Matlab, the complicated calculations could be avoided. With Matlab, we can utilize linear algebra into real life to solve practical problems. In the past two decades, the applications of linear algebra to real world problems have mushroomed. In this paper, some examples on solving linear problems with Matlab are illuminated and how to use Matlab in practice is discussed.

II. APPLICATION IN LINEAR ALGEBRA OPERATIONS

Linear algebra operations such as Matrix multiplication, finding inverse of a matrix, solving a system of linear equations and diagonalizing quadratic forms would lead to boring and complicated calculations. However, these operations can be performed very simply using matlab.

Sample 1.

Let $A=\begin{pmatrix} 1 & 2 & -1 \\ 0 & 1 & 2 \\ -3 & 6 & 4 \end{pmatrix}, B=\begin{pmatrix} -1 & 0 & 1 \\ 0 & 2 & 2 \\ 3 & 5 & 1 \end{pmatrix}$,

calculate AB, $A^{-1}B$.

```
>> A = [1 2 -1; 0 1 2; -3 6 4]；
>> B = [-1 0 1; 0 2 2; 3 5 1]；
>> A * B
```

Result of AB returned:

```
ans =
     -4      -1       4
      6      12       4
     15      32      13
>> inv(A) * B
```

Result of $A^{-1}B$ returned:

```
ans =
     -1.0000      0.1304      1.3478
      0           0.3478      0.2609
      0           0.8261      0.8696
```

(The numbers could be expressed as fractions either.)

Sample 2.

Find the solution of equations

$$\begin{cases} x_1+2x_2+2x_3+x_4=0 \\ 2x_1+x_2-2x_3-2x_4=0 \\ x_1-x_2-4x_3-3x_4=0 \end{cases}.$$

```
>> A = [1   2   2   1; 2   1   -2   -2; 1   -1   -4   -3];
```

```
>> format rat   // Let the solution outputs in the rational form.
>> B = null(A, 'r')   // Get the rational basis of space A.
B =
     2            5/3
     -2           -4/3
     1            0
     0            1
```

The column vectors in B form the basic set of

solutions. Apply the following commands to get the general solution.

```
>> syms k1 k2    // define the symbols k1 and k2
>> X = k1 * B(: , 1) + k2 * B(: , 2)   // write the general solution
>> pretty(X)                              // make X
more pretty
```

We usually reduce coefficient matrix into row echelon form to use general solution to solve linear equations. Command "rref" would provide row echelon form matrix in Matlab.

```
>> B = rref(A)
B =
    -5/3      1          0          -2
     4/3      0          1           2
     0        0          0           0
```

To observe intermediate data, sometimes, the user-defined functions are needed, showing the information expected. In sample 3, reducing matrix to its row echelon form is implemented by user-defined M function.

Sample 3.

Let $A=\begin{pmatrix}1 & 4 & 11 & 5\\ 2 & 8 & 6 & 7\\ 3 & 2 & 12 & 8\\ 7 & 15 & 66 & 0\end{pmatrix}$, get the row echelon form of A.

We will program two M-functions for the intermediate data. Given the matrix A and row index i and column index j, the M-function named zeroelement.m is going to reduce A by making $a_{ij}=0$, when $a_{ii}\neq 0$ and $i<j$.

```
function   [A]= zeroelement(A, i, j)
if   i>j | i==j
    error('i is not less than j')
else
    A(j, :)=A(j, :)-A(i, :)*(A(j, i)/A(i, i));
end
```

The other function named swaparray.m will swap the i-th row and j-th row in A.

```
function   [A] = swaparray(A, i, j)
if   i>n|j>n
    error('out of range')
else
    temp1=A(i, :);
    temp2=A(j, :);
    A(i, :)=temp2;
    A(j, :)=temp1;
end
```

Invoking these functions repeatedly, we'll finally get the row echelon form of matrix A.

```
>> A=[1 4 11 5; 2 8 6 7; 3 2 12 8; 1 15 66 0];
>> zeroelement(1, 2)
>> zeroelement(1, 3)
>> zeroelement(1, 4)
>> wamparray(2, 3)
>> zeroelement(2, 3)
>> zeroelement(2, 4)
>> zeroelement(3, 4)
```

Each step implemented, the result will be displayed to help users observe. You can attach a semicolon at the end of the row to omit the intermediate output.

III. Application in Solving Practical Problems

Sample 1. There're five cities with its own airport and some one-way airlines between the cities. As figure i shows, a directed edge from node A to node B represents an airline from city A to city B.

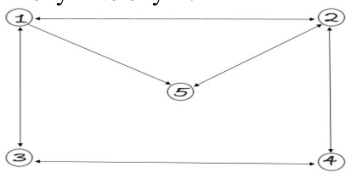

Figure i

The cities and the airlines could be represented by an adjacency matrix $A=(a_{ij})$. If there's an airline from city i to city j, let $a_{ij}=1$, otherwise $a_{ij}=0$. Figure I can be represented as matrix A.

$$A=\begin{bmatrix}0 & 1 & 1 & 0 & 1\\ 1 & 0 & 0 & 1 & 1\\ 1 & 0 & 0 & 1 & 0\\ 0 & 1 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\end{bmatrix}$$

In A^2, aij is the number of route from city i to city j though exact one transfer. Enter the following commands in Matlab.

$A=[0,1,1,0,1;1,0,0,1,1;1,0,0,1,0;0,1,1,0,0;0,1,0,0,0]$

$A1=A*A$

```
A1 =

    2    1    0    2    1

    0    3    2    0    1

    0    2    2    0    1

    2    0    0    2    1

    1    0    0    1    1
```

Let's take an element in A1 for verification. A1[2][3] = 2 means there're two different strategies to reach city 3 from city 2 though a transfer. Enumerating all the paths, we can find exactly two ways which are 2→4→3, 2→1→3.

The matrix keeps the number of routes between each two cities though at most one transfer could be simply described as $B = A + A^2$. Enter the command $B = A + A1$ in Matlab.

```
B =

    2    2    1    2    2
    1    3    2    1    2
    1    2    2    1    1
    2    1    1    2    1
    1    1    0    1    1
```

Rest can be deduced by analogy.

Sample 2.

In autosomal genetic, individual inherits one gene for each gene pair from its parents to form its own unique gene pairs. The inherited gene is selected randomly. If a parent has gene pair Aa, the descendant will have the same probabilities to inherit gene A or gene a. For a parent with gene pair aa and another with gene pair Aa, the descendant can be with gene pair Aa or aa in the same probability. Enumerating all the parents gene pairs, the probabilities of each case are listed below.

		Parents’ gene pairs					
		$AA-A$	$AA-Aa$	$AA-aa$	$Aa-Aa$	$Aa-aa$	$aa-aa$
Descendant’s gene pair	AA	1	1/2	0	1/4	0	0
	Aa	0	1/2	1	1/2	1/2	0
	aa	0	0	0	1/4	1/2	1

There’re some crops consist of three genotypes AA, Aa, aa based on a certain distribution. All the crops are pollinated by the crops with AA genotype. Let a_n, b_n, c_n be the portion of crops with genotype AA, Aa, aa in n-th generation. $(n = 0,1,2)$. a_0, b_0, c_0 are the original distribution and satisfying $a_0 + b_0 + c_0 = 1$. The n-th generation distribution satisfies the following equations.

$$\begin{cases} a_n = a_{n-1} + \frac{1}{2} b_{n-1} \\ b_n = c_{n-1} + \frac{1}{2} b_{n-1} \\ c_n = 0 \end{cases}$$

Let

$$X^{(n)} = \begin{pmatrix} a_n \\ b_n \\ c_n \end{pmatrix}, M = \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$X^{(n)} = MX^{(n-1)} \qquad X^{(n)} = M^n X^{(0)}$$

There’re three eigenvalues to matrix M, so M can be diagonalized which means there is a diagonal matrix D and a invertible matrix P satisfy $M = PDP^{-1}$ .So $X^{(n)} = PD^n P^{-1} X^{(0)}$ should be satisfied. The diagonal elements in D are the eigenvalues of M and P is the eigenvector correspondingly.

First of all, we’ll calculate the eigenvalues and eigenvector of M.

```
>> A = [1 0.5 0;0 0.5 1;0 0 0];
>> [P, D] = eig(A);   // eigenvalues and eigenvector of matrix A
```

$$P = \begin{pmatrix} 1 & -0.7071 & 0.4082 \\ 0 & 0.7071 & -0.8165 \\ 0 & 0 & 0.4082 \end{pmatrix}, D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

Combined with the results returned, the distributions at each generation can be calculated by $X^{(n)} = PD^n P^{-1} X^{(0)}$, with $X^{(0)} = \begin{pmatrix} a_0 \\ b_0 \\ c_0 \end{pmatrix}$ and $a_0 + b_0 + c_0 = 1$. The multiplication of matrix in Matlab is helpful here.

Summary

Linear algebra is an efficient tool to help do scientific calculations on immense amounts of data, especially when it is combined with computers. This article gives some examples to demonstrate the advantages of using Matlab in linear algebra. Teaching linear algebra with Matlab not only enhances students’ learning interest, but also improves their ability of applying theories to solve practical problems.

Acknowledgment

Funded projects: (863) "large number DNA decomposition computer model research project (Code: 2009AA01Z413)

References

[1] Chen Huaichen, Gong Jiemin, and MATLAB introductory linear algebra practice (Second Edition), publishing house of electronics industry, 2009

[2] Zhou Jianxing, will Tsu Take, MATLAB Xingming, from entry to the master, the people post and Telecommunications Press, 2008