MATH 412 - Homework 2

Twymun Safford

February 26th, 2024

Contents

1 Homework 2 - Problem

A field has been completely devastated by fire. Two types of vegetation, grasses, and small shrubs will first begin to grow, but the small shrubs can take over an area only if preceded by the grasses. Figure 1 shows an example image. below, the transfer coefficient of 0.3 indicates that, by the end of the summer, 30% of the prior bare space in the field becomes occupied by grasses.

1. A field has been completely devastated by fire. Two types of vegetation, grasses and small shrubs will first begin to grow, but the small shrubs can take over an area only if preceded by the grasses. In the figure below, the transfer coefficient of 0.3 indicates that, by the end of the summer, 30% of the prior bare space in the field becomes occupied by grasses.

   (a) Find the transfer matrix $\mathbf{T}$.

   (b) Suppose $\mathbf{X} = \begin{pmatrix} 10 \\ 0 \\ 0 \end{pmatrix}$ and that the area is measured in acres. Use the recursion formula $\mathbf{X}_{n+1} = \mathbf{T}\mathbf{X}_n$, along with a calculator (no need for a computer), to determine the ground cover in each of the next 6 years.

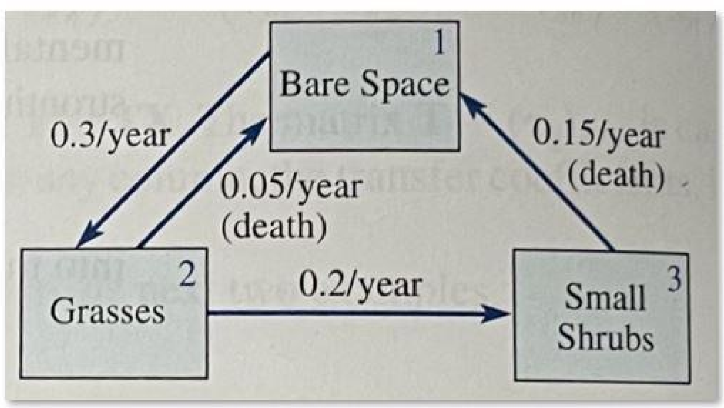

Figure 1: Example Image

1.1 Problem a

Find the transfer matrix T

1.1.1 Solution:

Based on Figure 1, we can create the following transfer matrix (tabular form):

	Bare Space	Grasses	Small Shrubs
Bare Space	0.7	0.05	0.15
Grasses	0.3	0.75	0
Small Shrubs	0	0.2	0.85

Table 1: Transfer Matrix in Tabular Form

Or, in matrix form:

$$T = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix}$$

The Matlab implementation of this is:

```
T = [0.7, 0.05, 0.15; 0.3, 0.75, 0; 0, 0.2, 0.85];
```

1.2 Problem b

Suppose x =

$$\begin{pmatrix} 10 \\ 0 \\ 0 \end{pmatrix}$$

and that area is measured in acres. Use the recursion formula $X_{n+1} = TX_n$, along with a calculator or a CAS, to determine the ground cover in each of the next 6 years.

1.2.1 Solution:

Since we are given that the formula $X_{n+1} = TX_n$ can be used to determine the ground cover in each of the next possible 6 years. We use:

$$X_0 = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

For time (t=0). For each year, we calculate the cover for each year with the help of Matlab. We observe that for year 1, for instance we get:

$$X_1 = TX_0 = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

And that for year 2, we get:

$$X_2 = TX_1 = T * (TX_0) = T^2 X_0 = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix}^2 \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

Thus, we see from our general form that we can calculate the coverage after n years have passed via:

$$X_n = TX_n = T^n X_0 = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix}^n \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

Where $n \in Z_{\geq 0}$. Thus, for each year:

1.2.2 Year 1:

$$X_1 = TX_0 = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 7 \\ 3 \\ 0 \end{bmatrix}$$

1.2.3 Year 2:

$$X_2 = TX_1 = T^2 X_0 = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix}^2 \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 5.0500 \\ 4.3500 \\ 0.6000 \end{bmatrix}$$

1.2.4 Year 3:

$$X_3 = TX_2 = T^3 X_0 = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix}^3 \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3.8425 \\ 4.7775 \\ 1.3800 \end{bmatrix}$$

1.2.5 Year 4:

$$X_4 = TX_3 = T^4 X_0 = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix}^4 \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3.1356 \\ 4.7359 \\ 2.1285 \end{bmatrix}$$

1.2.6 Year 5:

$$X_5 = TX_4 = T^5 X_0 = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix}^5 \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.7510 \\ 4.4926 \\ 2.7564 \end{bmatrix}$$

1.2.7 Year 6:

$$X_6 = TX_5 = T^6 X_0 = \begin{bmatrix} 0.7 & 0.05 & 0.15 \\ 0.3 & 0.75 & 0 \\ 0 & 0.2 & 0.85 \end{bmatrix}^6 \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.5638 \\ 4.1947 \\ 3.2415 \end{bmatrix}$$

Thus, we can see that after six years we should expect there to be approximately 2.5638 acres of bare space, 4.1947 acres of grasses, and 3.2415 acres of small shrubs in the area six years after the devastating fire.

1.3 Problem c

What happens after a long time? Justify your answer by showing a mathematical proof. (Use diagonalization for matrix factorization.)

1.3.1 Solution:

We can use what we have called the "transition matrix" which represents the vegetation composition in the area. We can employ what is known as a Markov chain model to forecast in the long-term what the vegetation in the area looks like based on some initial trends.

The algorithm we can implement to solve this problem using Matlab uses the following steps:

Define a transition matrix based on the probabilities of the area losing/gaining different vegetation (given).

Define an initial state vector (x_0) with the proportion of land that is bare, made of grasses, and made of short shrubbery (given).

Use matrix multiplication to apply the transition matrix to the initial state vector to predict the distribution of vegetation.

Repeat the multiplication for subsequent elections to observe the long-term trends in voter distribution across parties.

We need to calculate the eigenvalues matrix (D) and eigenvectors matrix (Q) such that $QDQ^-1 = P$. We know that we desire the transfer matrix for the vegetation at an infinite amount of time. This requires us to find:

$$L = \lim_{n \to \infty} D^n$$

To find the eigenvalues, we need to solve the characteristic equation:

$$\det(T - \lambda I) = 0$$

where λ is the eigenvalue and I is the identity matrix. The matrix $T - \lambda I$ becomes:

$$T - \lambda I = \begin{pmatrix} 0.7-\lambda & 0.05 & 0.15 \\ 0.3 & 0.75-\lambda & 0 \\ 0 & 0.2 & 0.85-\lambda \end{pmatrix}$$

Now, let's calculate the determinant of $T - \lambda I$ and set it equal to zero to find the eigenvalues.

$$\det(T-\lambda I) = (0.7-\lambda)((0.75-\lambda)(0.85-\lambda)-0\cdot 0.2)-0.05(0.3(0.85-\lambda)-0\cdot 0)+0.15(0.3\cdot 0.2-(0.75-\lambda)\cdot 0)$$

Now, we set this polynomial equal to zero and solve for λ. We can use numerical methods or factorization to solve this cubic equation. We get for eigenvector and eigenvalue pairs:

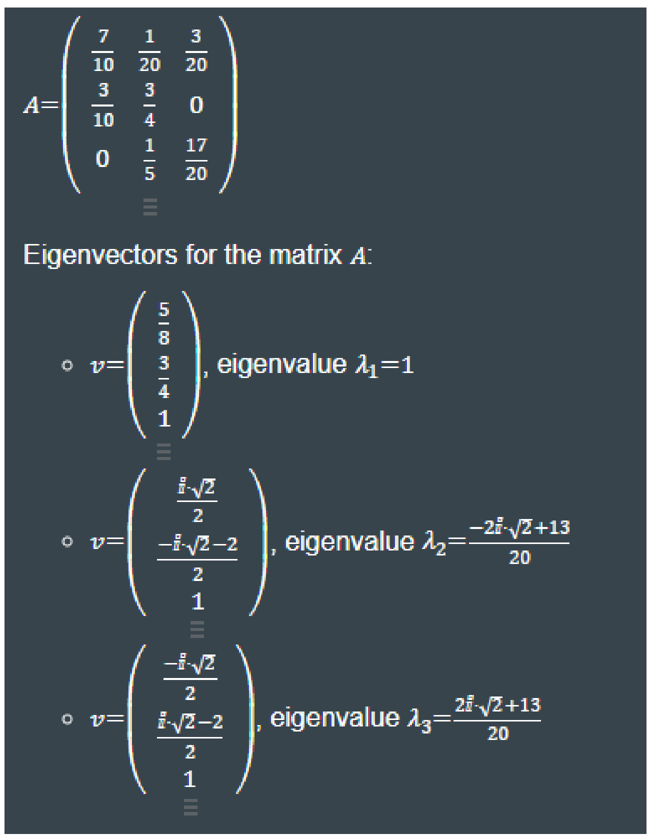

Figure 2: Eigenvalue-Eigenvector Pairs

We've calculated that D yields (in Matlab):

```
Eigenvalues:
   0.6500 - 0.1414i   0.6500 + 0.1414i   1.0000 + 0.0000i
```

Figure 3: Eigenvalues - Matrix

Repetitive multiplication by the same matrix eventually aligns the resulting product along the dominant eigenvector of the matrix i.e. the eigenvector corresponding to the dominant eigenvalue. The real eigenvalue with the largest magnitude for the transfer matrix is $\lambda = 1$.
Therefore, the eigenvector that corresponds with this is:

$$v_1 = \begin{bmatrix} \frac{5}{8} \\ \frac{3}{4} \\ 1 \end{bmatrix}$$

Thus, we get for L:

```
L = 3×3
       0     0     0
       0     0     0
       0     0     1

Limit as n approaches infinity for eigenvalues matrix:
     0     0     0
     0     0     0
     0     0     1
```

Figure 4: Limit-L

Such that now we yield:

$$QDQ^{-1} = P_\infty$$

Using Matlab, we calculate the distribution matrix at infinity:

```
Pinf = 3x3 complex
     0.2632 + 0.0000i   0.2632 + 0.0000i ...
     0.3158 + 0.0000i   0.3158 + 0.0000i
     0.4211 + 0.0000i   0.4211 + 0.0000i

Pinf:

   0.2632 + 0.0000i   0.2632 + 0.0000i   0.2632 - 0.0000i
   0.3158 + 0.0000i   0.3158 + 0.0000i   0.3158 - 0.0000i
   0.4211 + 0.0000i   0.4211 + 0.0000i   0.4211 - 0.0000i
```

Figure 5: Distribution matrix at infinity

The vegetation ratios end up being:

$$X_\infty = P_\infty X_0 = \begin{bmatrix} 2.6316 \\ 3.1579 \\ 4.2105 \end{bmatrix}$$

The final matrix P_∞ describes the long-term probabilities of vegetations entering into/leaving the barren landscape and indicate the equilibrium vegetation distributions the system approaches (10 acres of land) as time tends to infinity. This implies that for the system, there's likely to be a larger ratio of vegetation to barren land in the acres under examination.

See the subsection for problem c codes for the full implementation and solving process.

2 References

Deisenroth, M. P., Faisal, A. A., 38; Ong, C. S. (2020). Mathematics for Machine Learning. Cambridge: Cambridge University Press.

3 Appendix - Matlab Code

Listed below are the Matlab code scripts that were used to calculate the results for the problem.

3.1 Problem a code - Transfer matrix

```

% Homework 2 - Problem 1a
% Student Name: Twymun Safford
% Class Name: MATH-412
% Creates the transfer matrix for a problem.

% Metadata
studentName = 'Twymun Safford';
className = 'MATH-412';
assignmentName = 'Homework 2 - Problem 1a';

% Generate and save transfer matrix
transfer_matrix = [0.7, 0.05, 0.15; 0.3, 0.75, 0; 0, 0.2, 0.85];
save('transfer_matrix.mat', 'transfer_matrix');

% Generate log file
logFileName = 'homework_2a_log.txt';
logFileID = fopen(logFileName, 'a'); % Open log file in append mode

% Get current date and time
currentDateTime = datetime('now', 'Format', 'yyyy-MM-dd HH:mm:ss');

% Write execution log
fprintf(logFileID, 'Last execution: %s\n', char(currentDateTime));
fprintf(logFileID, 'Student Name: %s\n', studentName);
fprintf(logFileID, 'Class Name: %s\n', className);
fprintf(logFileID, 'Assignment Name: %s\n\n', assignmentName);

% Close log file
fclose(logFileID);
```

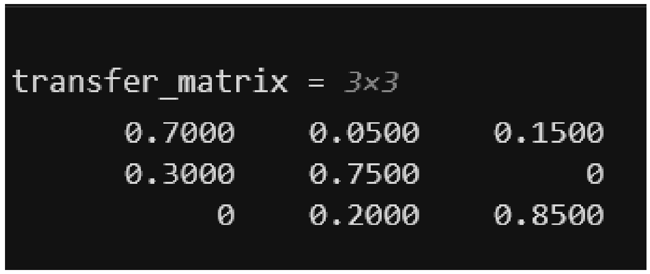

Figure 6: Homework 2a - matrix

3.2 Problem b - Vegetation distribution at each year

```
%{
Homework 2 - Problem b
Student Name: Twymun Safford
Class Name: MATH-412
Calculates the grass, shrub, and bare space
ratios for 10 acres of land.
%}

% Metadata
studentName = 'Twymun Safford';
className = 'MATH-412';
assignmentName = 'Homework 2 - Problem 1b';

% Generate and save transfer matrix
transfer_matrix = [0.7, 0.05, 0.15; 0.3, 0.75, 0; 0, 0.2,
    0.85];
disp(transfer_matrix)
save('transfer_matrix.mat', 'transfer_matrix');

%Generate matrix for initial coverage conditions
x_0 = [10 0 0]';
disp(x_0)

% Iterate from 0 to 6 for each year
for n = 0:6
    disp(['Vegetation vector for year: ', num2str(n)]);
    result_matrix = transfer_matrix^n * x_0;
    disp(result_matrix)
    filename = sprintf('vegetation_column_year_%d.mat', n);
    save(filename, 'result_matrix');
end

% Generate log file
logFileName = 'homework_2b_log.txt';
```

```
logFileID = fopen(logFileName, 'a'); % Open log file in append mode

% Get current date and time
currentDateTime = datetime('now', 'Format', 'yyyy-MM-dd HH:mm:ss');

% Write execution log
fprintf(logFileID, 'Last execution: %s\n', char(currentDateTime));
fprintf(logFileID, 'Student Name: %s\n', studentName);
fprintf(logFileID, 'Class Name: %s\n', className);
fprintf(logFileID, 'Assignment Name: %s\n\n', assignmentName);

% Close log file
fclose(logFileID);
```

3.3 Problem c - Finding distribution matrix and vegetation for long times

```
%{
Homework 2 - Problem c
Student Name: Twymun Safford
Class Name: MATH-412
Calculates the characteristic vegetation for an area that has
suffered through a catastrophic wildfire.
%}

% Metadata
studentName = 'Twymun Safford';
className = 'MATH-412';
assignmentName = 'Homework 2 - Problem 1b';

% Generate and save transfer matrix
transfer_matrix = [0.7, 0.05, 0.15; 0.3, 0.75, 0; 0, 0.2, 0.85];
disp(transfer_matrix)
save('transfer_matrix.mat', 'transfer_matrix');

%Generate matrix for initial coverage conditions
x_0 = [10 0 0]';
disp(x_0)

%Display what happens for longer periods of time
disp('After 10 years:');
x_10 = transfer_matrix^10 * x_0
```

```
disp(x_10);

disp('After 1000 years:');
x_1000 = transfer_matrix^1000 * x_0
disp(x_1000);

%Display what happens for longer periods of time
disp('After 100000 years:');
x_100000 = transfer_matrix^100000 * x_0
disp(x_100000);

% Find Matrix Q and Diagonal Matrix D
[Q,D] = eig(transfer_matrix)

% Display eigenvalues and eigenvectors
disp('Eigenvalues:');
disp(diag(D)');
disp('Eigenvectors:');
disp(Q);

% Save eigenvalues to mat file
save('eigenvalues.mat', 'D');

% Save eigenvectors to mat file
save('eigenvectors.mat', 'Q');

% Find L (Limit of D^n when n tends to infinity)
L = D^inf
disp("Limit as n approaches infinity for eigenvalues matrix
    :")
disp(L)
save('L.mat', 'L');

% Calculate Pinf
Pinf = Q*L*inv(Q)
disp("Pinf:")
disp(Pinf)
save('Pinf.mat', 'Pinf');

% Find Pinf * x0
Pinf_x0 = Pinf*x_0
disp("Vegetation after 'infinite' amount of time:")
disp(Pinf_x0)
save('Pinf_x0.mat', 'Pinf_x0');

% Random vector in R^4
y = [6,0,0]'
disp("X_1 - another vector:")
disp(y)
Pinf_y = Pinf*y
```

```
disp(Pinf_y)
save('Pinf_y.mat', 'Pinf_y');

% Generate log file
logFileName = 'homework_2c_log.txt';
logFileID = fopen(logFileName, 'a'); % Open log file in
    append mode

% Get current date and time
currentDateTime = datetime('now', 'Format', 'yyyy-MM-dd HH:
    mm:ss');

% Write execution log
fprintf(logFileID, 'Last execution: %s\n', char(
    currentDateTime));
fprintf(logFileID, 'Student Name: %s\n', studentName);
fprintf(logFileID, 'Class Name: %s\n', className);
fprintf(logFileID, 'Assignment Name: %s\n\n', assignmentName
    );

% Close log file
fclose(logFileID);
```