Final Project of Linear Algebra Group 4

Topic: Markov matrix applied in solving population variation problem

1. Abstract:

Last month, the global population surpassed 8 billion, which gave us the motivation to make this report. Our project is to predict the future populations, just like The United Nations. But we use linear algebra methods and get quite accurate results.

2. Introduction:

Recently, as demographic trends continue to evolve, the Sustainable Development Goals will face new challenges. Because of the increasing birth rate, countries could face Insufficient social resources such as food and water shortages problems, while the decreasing birth rate countries could face labor shortage problems.

The United Nations has data that predicts future populations. The United Nations population estimates and projections data are used for many development indicators that use the United Nations system.

3. Materials and Methods:

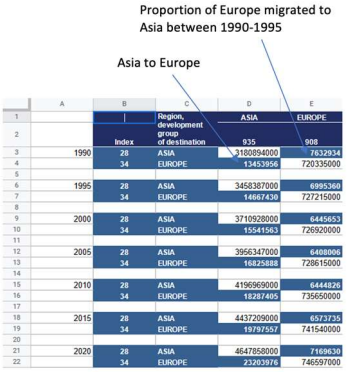
We hope to found a matrix that can predict population growth in this form: $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} X' \\ Y' \end{bmatrix}$

X, Y means the population of some area, X', Y' means population after one time of iteration. We hope that we can predict these two area's population growth by this transfer matrix. Obviously, b means the proportion of the population lived in Y that migrated to X (X' = aX + bY). For the same reason, c means the proportion of the population lived in X that migrated to Y.

We can get these two equation:

1. $b = P_{Y\to X} / Y$
2. $c = P_{X\to Y} / X$

For example, we can get b_{1990} = 7632934/720335000 and c_{1990} = 13453956/3180894000 with these two equations.



	A	B	C	D	E
1			Region, development group of destination	ASIA	EUROPE
2		Index		935	908
3	1990	28	ASIA	3180894000	7632934
4		34	EUROPE	13453956	720335000
5					
6	1995	28	ASIA	3458387000	6995360
7		34	EUROPE	14667430	727215000
8					
9	2000	28	ASIA	3710928000	6445653
10		34	EUROPE	15541563	726920000
11					
12	2005	28	ASIA	3956347000	6408006
13		34	EUROPE	16825888	728615000
14					
15	2010	28	ASIA	4196969000	6444826
16		34	EUROPE	18287405	735650000
17					
18	2015	28	ASIA	4437209000	6573735
19		34	EUROPE	19797557	741540000
20					
21	2020	28	ASIA	4647858000	7169630
22		34	EUROPE	23203976	746597000

Then, it's time to solve a, d.

This is the formula to express the population of some area :

$$X' = X\cdot PGR_X + P_{immigrate} - P_{emigrate}$$
$$\approx X\cdot PGR_X + P_{Y\to X} - P_{X\to Y}$$

when X, Y = Asia, Europe. *PGR: Population Growth Rate

We found that, for Asia, the population immigrating from other areas except Europe is approximately equal to the population emigrating to other areas except Europe, and vice versa.

In other words, $P_{emigrate,\ Europe} \approx P_{immigrate,\ Asia} \approx P_{Europe\to Asia}$, $P_{emigrate,\ Asia} \approx P_{immigrate,\ Europe} \approx P_{Asia\to Esurope}$.

Therefore, we choose Asia and Europe to be our target to conduct more in-depth research.

Then, we can rewrite X' = aX+b to X' = aX+$P_{Y\to X}$. After that, we can get a = (X'-$P_{Y\to X}$) / X.

Similarly, d = (Y'-$P_{X\to Y}$) / Y. So we can use these equations to generate transfer matrices every five years.

After that, we have to generate one transfer matrix to predict Asia and Europe's population. We choose the least square approximation method to find the next transfer matrix because we found that, in these six matrices, all of each elements' variation trend can be presented by an asymptote respectively. Thus, we use this method to find a transfer matrix for prediction. In the next page, we will prove that our algorithm is precise and reliable.

4. Calculation & Results:

A. Verify our Algorithm:

We use the data from 1990 to 2015 to get the 4 matrix elements linear equation coefficients.

- m11: y = -0.007*t + 1.088, m12: y = -0.0004*t + 0.0107
- m21: y = 0.00003*t + 0.00417, m22: y = 0.0011*t + 0.9931

And we can generate the Markov matrix corresponding to 2015-2020, and then execute the matrix multiplication. Hence, we can use C program to do the calculation of area population in 2020.

Comparison	Data (2015)	Data (2020)	Calculation (2020)	Error (2020)
Asia	4437209000	4647858000	4647475396	0.01%
Europe	741540000	746597000	760619397	1.88%

We compare calculation results and data of area population in 2020, and we find the difference between calculations is below 2%. Therefore, we think our algorithm works so that we can further use matrix coefficients data from 1990 to 2020 to calculate the area population from 2025 to 2050.

B. Predict Future Area Population:

We use the data from 1990 to 2020 to get the 4 matrix elements linear equation coefficients.

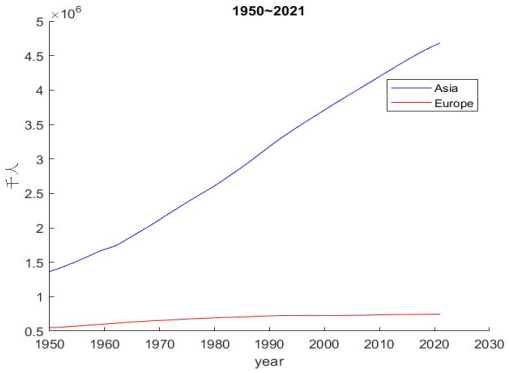
- m11: y = -0.007*t + 1.0881, m12: y = -0.0003*t + 0.0104
- m21: y = 0.00004*t + 0.00413, m22: y = 0.009*t + 0.9937

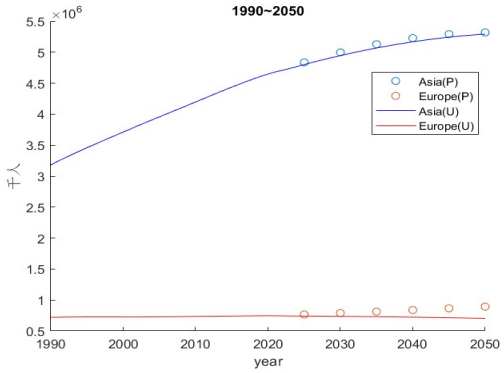
And then we use C program to generate the Markov matrix corresponding to 2020-2025 to 2045-2050. Finally, we can get the calculation result of area population from 2025 to 2050.

Year	2025	2030	2035	2040	2045	2050
Asia	4835786002	4997151486	5128657626	5227503725	5291483006	5319063477
Europe	767094053	789303686	813161642	838589998	865498614	893787094

C. Result

The difference between the predict data & the UN's data	Asia	Europe
2020	-0.00%	1.87%
2050	0.54%	26.92%





5. Discussion:

The first chart shows the value of the population of 2020 we predict by using 1990 to 2015's data is close to the real population. The second graph is to compare the data we predict and the UN predicts. The lines are the UN's , the circles are what we predict. We predict a little higher than the UN, the prediction of Asia is close.

We think that the UN's prediction considers more than the population change, maybe the climate change, or the fast of lower birth rate. We also use the data of migration between Asia and Europe, but the migration population between Asia and Europe is nearly the same percentage, we think it shall not affect the prediction.

6. Conclusion:

We use basic knowledge of linear algebra learned from the class with the help of C program and Matlab to predict the future trend of population between Asia and Europe. Furthermore, we hope that the result could be a credible reference for further research.

7. References:

World Population Prospects 2022, Online Edition. United Nations, Department of Economic and Social Affairs, Population Division (2022).

8. Appendix
    A. Author Contributions
        - 楊芊華：30%
          Collecting useful official information
          Data pre-processing
          Determine the topic of the report
          Addressing gaps in the topic to make it more complete
          Confirm that the whole group has reached a consensus
          Help individual team members keep up with progress
          Monitor task completion
          Assist in the distribution of tasks and assignments

        - 彭鈞偉：20%
          Transfer matrix model construction
          Data collection
          Mathematical proof
          Algorithm
          Precision testing

        - 陳觀宇：20%
          Data Analysis (Calculate least square coefficient)
          Writing C Program for Calculation (Verify the Algorithm & Predict Future Area Population)
          Calculation part for report

        - 蔡柏紹：14%
          use matlab to draw charts
          analyze the data
          Result and Discussion part for report

        - 林泉龍：8%
          perfecting any mistypes, grammars, translation in the report and ppt
          Abstract and Introduction part for report

        - 潘彥誠：8%
          gather information, attribution, presentation and conclusion part for report

B. Code:

Markov Matrix.c

```
#include <stdio.h>
#include <string.h>
#define N 2                                  // set markov matrix size is N*N
#define MAX_YEAR_INTERVAL 100    // set maximum calculation year interval(5 years per interval)
#define MAX_DATA (int) (1e6)             // set maximum output result string size

typedef struct
{
   double m[N][N];

} matrix;

matrix Markov;                                                  // declare Markov Matirx
double coefficient_a, coefficient_b;                    // least square equation coefficient(linear)
double markov_data[MAX_YEAR_INTERVAL][N][N];   // markov matrix of the corresponding year
char area_name[N][MAX_DATA];     // area name
double people[N];                        // area initial population
double new_people[N];                // population after once matrix multiplication
int data_start_year;                     // least square equation coefficient data start year
int data_end_year;                       // least square equation coefficient data end year
int data_year_interval;                // least square equation coefficient data year interval(5 years/interval)
int cal_year;                            // target year we want to calculate
int cal_year_interval;                // year interval between data end year and target year(5 years/interval)
double result[MAX_YEAR_INTERVAL][N];
// calculation result (year vs. area population)
char year_data[MAX_YEAR_INTERVAL][MAX_DATA];
// output year string (from data end year to target year)
char population_data[MAX_YEAR_INTERVAL][N][MAX_DATA];
// output population result string (year vs. area population)
void input_data()
{
   //scanf input from input_data.txt
   freopen("input_data.txt", "r", stdin);

   //input data_start_year & data_end_year & cal_year and calculate data_year_interval & cal_year_interval
   scanf("%d%d%d", &data_start_year, &data_end_year, &cal_year);
   data_year_interval = (data_end_year-data_start_year)/5;
   cal_year_interval = (cal_year-data_end_year)/5;

   for(int row = 0; row < N; row++)
   {
      for(int col = 0; col < N; col++)
      {
         //input least square equation coefficients
         scanf("%lf", &coefficient_a);
         scanf("%lf", &coefficient_b);
         for(int t=1; t<=cal_year_interval; t++)
         {
            //use linear equation coefficient to calculate the matrix
            markov_data[t][row][col] = coefficient_a*(t + data_year_interval) + coefficient_b;
         }
      }
   }
   //input area name string & area initial population
   for(int i = 0; i < N; i++)
   {
```

```
        scanf("%s", area_name[i]);
        scanf("%lf", &people[i]);
    }
}
void Calculate_People(int cal_year_interval)
{
    for(int t = 0 ; t <= cal_year_interval ; t++)
    {
        //t=0 means data end year
        if(t == 0)
        {
            for(int i = 0 ; i < N; i++)
            {
                //area initial population
                result[0][i] = people[i];
            }
        }
        else
        {
            //load markov matrix of the corresponding year from markov_data
            for(int row = 0; row < N; row++)
            {
                for(int col = 0; col < N; col++)
                {
                    Markov.m[row][col] = markov_data[t][row][col];
                }
            }

            //implement markov matrix multiplication
            for(int row = 0; row < N; row++)
            {
                double temp = 0;
                for(int col = 0; col < N; col++)
                {
                    temp += Markov.m[row][col]*people[col];
                }
                new_people[row] = temp;
            }

            //store calculation result of area population
            //and update the area population for next round calculation
            for(int i = 0 ; i < N; i++)
            {
                result[t][i] = new_people[i];
                people[i] = new_people[i];
            }
        }
    }
}
void output_data()
{
    //write the calculation results to output_data.txt
    FILE *fp = NULL;
    fp = fopen("output_data.txt", "w+");

    //output the years from data end year to cal year
    fputs("YEAR : ", fp);
    for(int t = 1; t <=  cal_year_interval; t++)
```

```
    {
        sprintf(year_data[t],"%d ", data_end_year + t*5);
        fputs(year_data[t], fp);
    }
    fputs("\n", fp);

    //output corresponding area population calculation result from data end year to cal year
    for(int i = 0; i < N; i++)
    {

        fputs(area_name[i], fp);
        fputs(" : ", fp);
        for(int t = 1; t<= cal_year_interval; t++)
        {
            sprintf(population_data[t][i],"%I64u ", (unsigned long long int)result[t][i]);
            fputs(population_data[t][i], fp);
        }
        fputs("\n", fp);
    }
    fclose(fp);
}
int main(void)
{
    //get input data from txt file
    //and use least square equation coefficient to calculate markov matrix of the corresponding year
    input_data();
    //use markov matrix and initial
    Calculate_People(cal_year_interval);
    //out the txt file including the calculation result of area population of the corresponding year
    output_data();

    return 0;
}
```

C. Input and Output Files:

Input Data.txt (1990-2015 to 2020)

```
1990 2015 2020
-0.007 1.088
-0.0004 0.0107
0.00003 0.00417
0.0011 0.9931
ASIA 4437209000
EUROPE 741540000
```

Output Data.txt (1990-2015 to 2020)

```
YEAR : 2020
ASIA : 4647475396
EUROPE : 760619397
```

Input Data.txt (1990-2020 to 2025-2050)

```
1990 2020 2050
-0.007 1.0881
-0.0003 0.0104
0.00004 0.00413
0.0009 0.9937
ASIA 4647858000
EUROPE 746597000
```

Output Data.txt (1990-2020 to 2025-2050)

```
YEAR : 2025 2030 2035 2040 2045 2050
ASIA : 4835786002 4997151486 5128657626 5227503725 5291483006 5319063477
EUROPE : 767094053 789303686 813161642 838589998 865498614 893787094
```