2018

程式設計

沒有加強班

程式設計與實習(二)

BY 孫茂勛

Email:JOHN85051232@GMAIL.COM

Inverse Matrix with Guassian Jordan

- 繳交期限延長到下禮拜一晚上11:59
- 不過這禮拜還是有作業。

Guassian Jordan

- A 是個N*N的矩陣

矩陣列運算：

- 將一矩陣的某一列乘以一數值加入另一列
- 將一矩陣的某一列乘以一個不為零的數
- 將一矩陣的某兩列互換位置

Guassian Jordan

- 反矩陣的特性：A * A^{-1} = I(單位矩陣)
- 如果能找到一個矩陣B，使得A*B = I，則B = A^{-1}
- [A | I] -> [AB | IB] -> [I | B]

- Goal： [A | I] -> [I | A]

Persudo Code

- 1.輸入矩陣A
- 2.建立一個N*N單位矩陣 I

```
float Identity [N][N] = {0};
for i from 0 to row:
    for j from 0 to col:
        if i == j then Identity[i][j] = 1
```

Persudo Code

- 3.把A的對角線透過矩陣列運算化成1

```
for i from 0 to row:
    float temp = A[i][i];
    for j from 0 to col:
        A[i][j] /= temp;
        I[i][j] /= temp;
```

Persudo Code

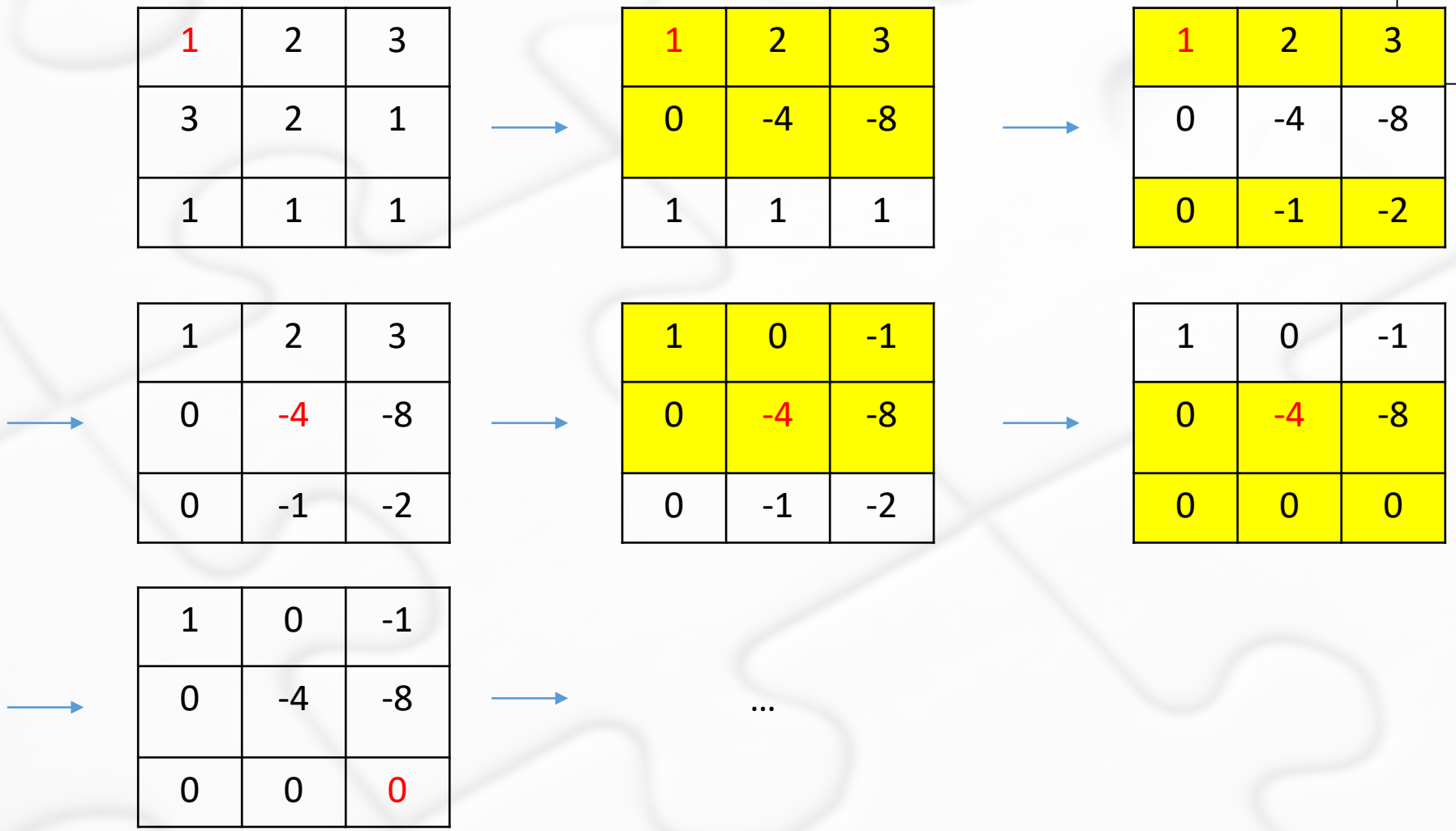
- 4.透過矩陣列運算，逐步把A變成I

```
for d in Diagonal:
    for i from 0 to row:
        if i == d then continue;
        float temp = A[i][d] / A[d][d];
        for j from 0 to col:
            A[i][j] -= temp * A[d][j];
            I[i][j] -= temp * A[d][j];
```

1	2	3
3	2	1
1	1	1

Persudo Code

```
for d in Diagonal:
    for i from 0 to row:
        if i == d then continue;
        float temp = A[i][d] / A[d][d];
        for j from 0 to col:
            A[i][j] -= temp * A[d][j];
            I[i][j] -= temp * A[d][j];
```

Persudo Code

- 5.確認是否無解(A化簡後某一列皆為0)，若無解則輸出”無解”　；反之，輸出I。

Sparse Matrix

- 有一個很大的Matrix，並且裡面的資訊涵蓋著大量的0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	11	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	2	8	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	6	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2
14	0	0	0	0	0	3	0	0	0
0	0	0	0	0	0	0	0	0	0

Sparse Matrix

- 這種很大但實際上卻記錄很少資訊的矩陣稱之Sparse Matrix(稀疏矩陣)
- 這麼大的一個Matrix，實際上卻只記錄了8筆資料。
- 全部儲存所花的記憶體：13(col) * 10(row) * 4(int) = 520 Bytes

- Q：有沒有辦法只存放這8筆資料呢？

Sparse Matrix

- 建立一個Structure，紀錄Array中row、col、value

```
struct SparseMatrix
{
        int col;
        int row;
        double value;
};
```

- 如此儲存一筆資料只需要4+4+8 = 16Bytes
- 8筆資料 = 128Bytes

I / O

- Input

```
#define SIZE 100
SparseMatrix sm[SIZE];
…
int sm_count = 0;
for i from 0 to row:
    for j from 0 to col:
        scanf(“%f”,&input);
        if input != 0 then:
            sm[sm_count].row = i;
            sm[sm_count].col = j;
            sm[sm_count].value = input;
            sm_count++;
```

I / O

- Output

```
for i from 0 to row:
    for j from 0 to col:
        for k from 0 to sm_count:
            if i == sm[k].row and j == sm[k].col then:
                print sm[k].value
            else:
                print 0
```

矩陣在程式裡面的用途？

- 影像處理，一張1024*1024的圖片可以看成矩陣
- 圖論，用矩陣表示點和邊的關係，之後會說到...應該吧
- Algorithm，裡面有一堆數學會用到矩陣QQ

HW

- 使用Sparse Matrix針對N階方陣實作出下列運算
- 矩陣加法
- 矩陣減法
- 矩陣乘法
- 轉置矩陣
- 反矩陣(Optional)

關於排序這件事

- 上學期學到的sort：針對數字由小到大、由大到小

Q：如果想針對特定的資料做排序？

- 針對一堆字串按照英文字母的順序排序
- 針對結構中特定的資料進行排序?

關於排序這件事

- 一樣可以透過bubble sort去完成
- 不過你有更好的選擇...std::sort()

```
#include <algorithm>
std::sort(array_begin,array_end[,cmpare function]);
```

Case 1

- 給一個整數陣列，由小排到大

```
#include <stdio.h>
#include <stdlib.h>
#include <algorithm> //引入library

int main()
{
    int a[5] = {3,6,1,0,9};
    //透過std提供的function排序 std::sort()
    //第一個參數是起始的位址
    //第二個參數是終止的位址
    //第三個是compare function(預設是由小到大)
    std::sort(a,a+5);
    for(int i = 0 ; i < 5 ; i++)
    {
        printf("%d ",a[i]);
    }
    system("pause");
    return 0;
}
```

Compare Function

- compare function的型態是bool，也就是只會回傳0或1
- sort()會一次抓出資料中的前兩個資料，透過compare function進行比對，所以compare function回傳1的規則代表是正確的排序規則

```
bool cmp(int a,int b)
{
    //Compare function的型態是bool，也就是只會回傳0或1
    //C會一次抓出資料中的前兩個資料，透過compare function進行比對
    return a < b; //如果第一個資料(a)小於第二個資料(b)則return 1
}
```

```
std::sort(a,a+5,cmp);
```

Case 2

- 給一個整數陣列，由大排到小，如何做？

Case 3

- 給一個字元陣列：{ ‘z’ , ’ a’ , ‘l’ }，按照英文字母順序排序？
- 如果是要求反向排序呢？

Case 4

- 給一個struct記錄學生的相關資訊：

```
struct student
{
    char name[10];
    float height;
    int grade;
}
```

- 請輸入至少3位同學，按照成績由大到小進行排序？
- 如果要求按照身高排序呢？

Sorting String ?

- 能不能按照字串排序？答案是可以的
- 不過需要使用string來宣告字串(OOP會教...應該吧)，如果用字元陣列會無法使用std::sort()

- 另一個做法是使用map的資料結構(OOP好像不會教QQ...得自己google了)

THANK YOU