矩阵论及其应用实习题

陈轶洲 MF20330010

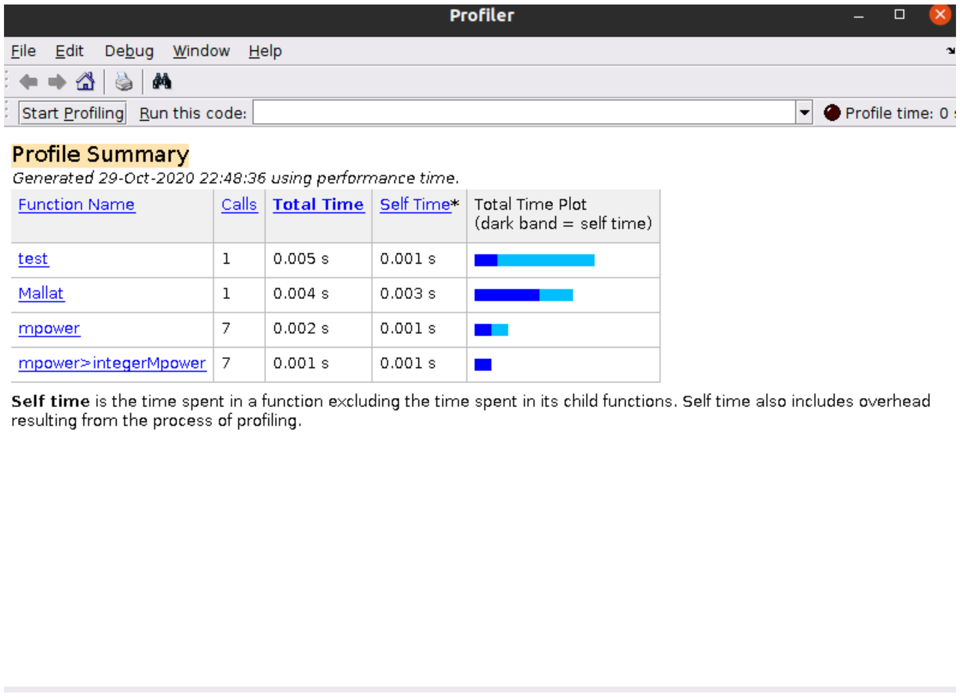
November 5, 2020

1 实验环境

计算机型号：MacBook Air (13-inch, Early 2015)
操作系统：Mac os
编程语言：Matlab

2 CPU 时间

输入第一组数据：



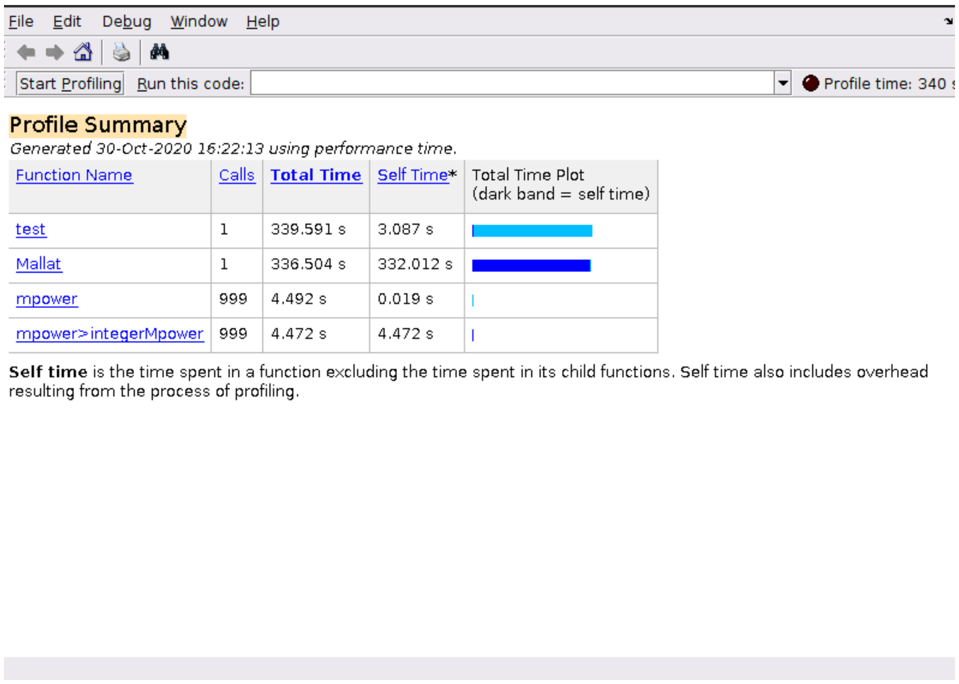
Profile Summary

Generated 29-Oct-2020 22:48:36 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
test	1	0.005 s	0.001 s	
Mallat	1	0.004 s	0.003 s	
mpower	7	0.002 s	0.001 s	
mpower>integerMpower	7	0.001 s	0.001 s	

Self time is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.

输入第二组数据：



Profile Summary

Generated 30-Oct-2020 16:22:13 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
test	1	339.591 s	3.087 s	
Mallat	1	336.504 s	332.012 s	
mpower	999	4.492 s	0.019 s	
mpower>integerMpower	999	4.472 s	4.472 s	

Self time is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.

3 算法说明

3.1 算法步骤

Algorithm 1 正交匹配追踪算法

```
Input: 观测数据向量 $y \in R^{M\times 1}$, 字典矩阵 $A \in R^{M\times N}$, 稀疏性指标 $K \in N$
Output: 稀疏的信号向量 $x \in R^{N\times 1}$
1: 初始化: $\Omega_0 = \phi, r_0 = y, k = 1$
2: while $k \le K$ do
3:     $j_k \in \arg\max_j |(r_{k-1}, a_j)|$
4:     $\Omega_k = \Omega_{k-1} \cup \{j_k\}$
5:     $x_k = (A^T_{\Omega_k} A_{\Omega_k})^{-1} A^T_{\Omega_k} y$
6:     $r_k = y - A_{\Omega_k} x_k$
7:     $k+ = 1$
8: $\hat{x}(i) = \begin{cases} x_K(i) & i \in \Omega_K \\ 0 & others \end{cases}$
9: return $\hat{x}$
```

3.2 变量说明

使用 Matlab 设计的正交匹配追踪算法中，对变量做如下定义：
1.A 为测度矩阵，规模为 $M \times N$；

2.y 为观测所得向量，规模为 $M \times 1$;
3.K 为稀疏性指标，为正整数;
4.x 为待还原向量，规模为 $N \times 1$;
5.r 为残差向量，规模为 $M \times 1$;
6.At，每轮迭代时被选中的列
7.x_l 为最小二乘解

3.3 程序清单

首先在 Mallat.m 中实现正交匹配追踪算法:

```
function [ x ] = Mallat(y,A,K)
        [M,N] = size(A);%字典矩阵的规模
        x = zeros(N, 1);%需要重构的解向量x
        At = zeros(M, K);%存储每轮迭代时A中被选中的列
        x_pos = zeros(1, K);%存储每轮迭代时A中被选中列的序号
        r = y;%残差向量，初始化为y

        for ii = 1:K%迭代K轮
                product = A'*r;%计算矩阵各列与残差的点积
                [val, pos] = max(abs(product));%找到与残差最相关的列
                At(:,ii) = A(:,pos);%将被选中的列存储到At中
                x_pos(ii) = pos;%记录被选中列的序号
                A(:,pos) = zeros(M,1);%清零A的这一列，其实此行可以不要，因为它与
                x_l = (At(:,1:ii)'*At(:,1:ii))^(-1)*At(:,1:ii)'*y;%最小二乘解
                r = y - At(:,1:ii)*x_l;%更新残差
        end

        x(x_pos) = x_l;
end
```

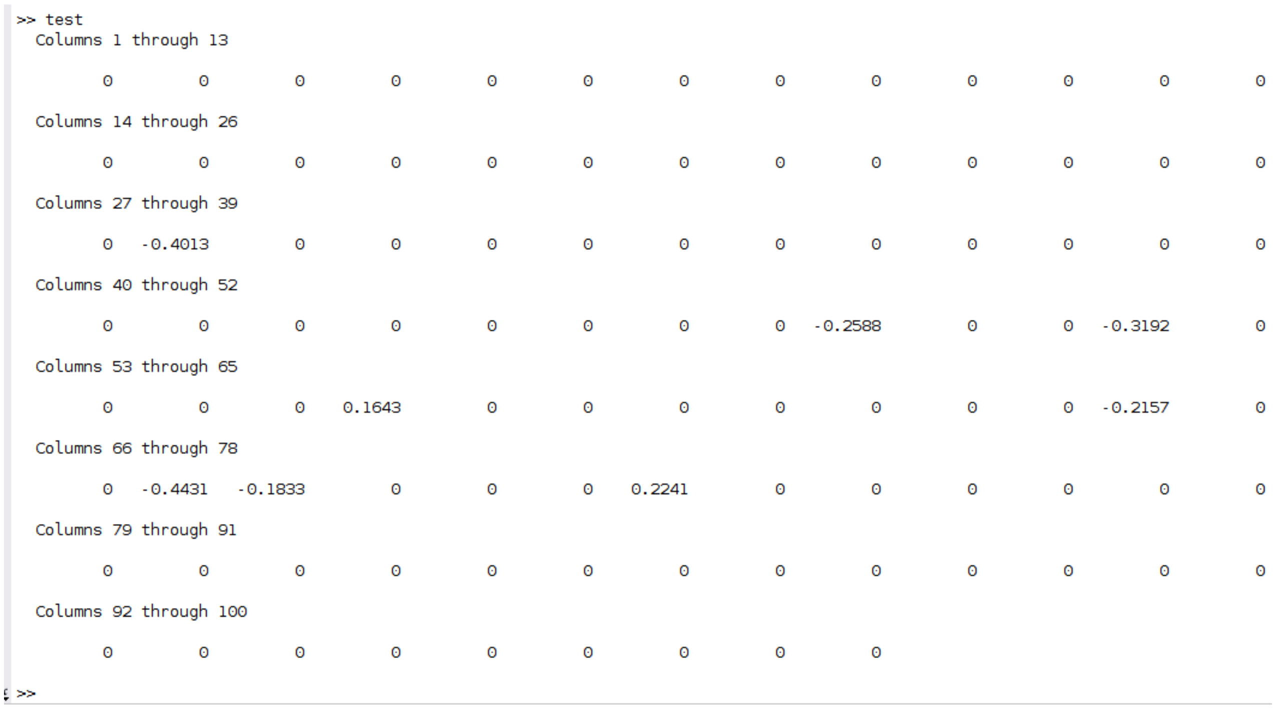
接着在 test.m 中对两组规模的数据进行测试

```
        K=8;
        M=50;
        N=100;
        A=randn(M,N);
        y=randn(M,1);
        x1 = Mallat(y,A,K);

        K=1000;
        M=15000;
        N=23000;
        A=randn(M,N);
        y=randn(M,1);
        x2 = Mallat(y,A,K);
```

4 运行结果及分析

第一组数据的解向量:



```
>> test
  Columns 1 through 13

         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 14 through 26

         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 27 through 39

         0   -0.4013         0         0         0         0         0         0         0         0         0         0         0

  Columns 40 through 52

         0         0         0         0         0         0         0         0   -0.2588         0         0   -0.3192         0

  Columns 53 through 65

         0         0         0    0.1643         0         0         0         0         0         0         0   -0.2157         0

  Columns 66 through 78

         0   -0.4431   -0.1833         0         0         0    0.2241         0         0         0         0         0         0

  Columns 79 through 91

         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 92 through 100

         0         0         0         0         0         0         0         0         0

>>
```

从结果可以看出，使用正交匹配追踪算法可以有效恢复信号，并通过稀疏性指标控制向量的稀疏性

5 实验总结

通过本次实验，让我对信号处理有了初步的了解，其中所涉及的矩阵论知识也强化了我对矩阵的特性以及矩阵运算有了进一步的认识。