信号恢复中稀疏解求解

黄军军

摘要

针对稀疏信号的恢复，本文利用两种经典的算法 OMP(Orthogonal Matching Pursuit，正交匹配追踪)和 CoSaMP(Compressive Sampling Match Pursuit，压缩采样匹配追踪)，对不同长度的稀疏信号分别做了两组实验，并讨论了测量次数与重构成功的概率关系，以及信号的稀疏度与重构成功的概率关系。由图像可看出随着测量次数的增加，重构成功的概率逐渐增加；随着信号稀疏度的增加（信号的非零项增加），信号重构成功的概率是逐渐递减的。

关键词：信号恢复 正交匹配追踪 压缩采样匹配追踪

1 问题介绍

设s是一个至多有m项非零元的d维实信号，这一类信号称之为m阶稀疏的。$\{x_1,\ldots x_N\}$是R^d中不依赖与信号的一组测量向量。我们用这些向量去采集信号的N个线性测量

$$\langle s,x_1\rangle,\langle s,x_2\rangle,\ldots,\langle s,x_N\rangle .$$

这里$\langle\bullet,\bullet\rangle$表示的是一般的内积运算。信号恢复通常要回答这样两个问题：

1) 恢复一个稀疏信号，多少次的测量是必须的?

2) 在已给出测量时，什么算法在重构信号的任务上表现的更好?

对于第一个问题，我们可以立即给出回答：重构一个m阶的稀疏信号应不少于m次测量。当然，d次测量总是足够的因为我们能够将信号的d个部分都列出。尽管它不是很明显，稀疏信号的重构确实是可以只用非常少的信息就能够做到。

这样做的方法源自于二战时期。美国军队对梅毒来筛查士兵有着天然的兴趣，但是梅毒测试是昂贵的，而军队也意识到对每个个体化验，以检测一个偶然的情况实在是一种浪费。他们的解决方案是储存一组士兵的血液，并检验所汇集的血液。如果这一批检查为阳性（表示血液中存在梅毒），就进行进一步的测试。这种方法，称之为组测试，随后在计算机科学和统计的文献被加以研究。

之后，计算调和分析团体提出了一个特定类型的组测试。其思想是，通过随机地组合一个稀疏信号的项，产生一组很小的统计汇总就能够使我们识别出信号中非零的项。接下来的定理，引自 Candès–Tao 的论文[1]和 Rudelson–Vershynin [2]。

定理 1： 设 $N \ge Km\ln\left(d/m\right)$，在 R^d 标准高斯分配中独立地取出 N 个向量 x_1，$x_2 .. x_N$,那么接下来的命题至少以 $1-e^{-kN}$ 的概率成立：从数据 $\left\{\left\langle s, x_n\right\rangle, n=1,2,\ldots N\right\}$ 重构任意一个 m 阶稀疏度的稀疏信号是可能的。

无噪声的情况下，重构信号问题归结为求如下的一个最优化问题：

$$
\begin{aligned}
(P_0): \qquad & \arg\min \|s\|_0 \\
& s.t \;\; \Psi s = v
\end{aligned}
$$

其中 $\|s\|_0$ 表示的是 s 中非零元的个数，$\left\{v: v_n=\left\langle s, x_n\right\rangle, n=1,2\ldots N\right\}$， $N \ll d$， Ψ 是 $N \times d$ 的测量矩阵，其行向量由 x_i， $i=1,2\ldots N$ 构成。由于 (P_0) 是一个 NP 难问题，且非凸优化问题，所以之后研究者考虑用 l_1 范数替代 l_0 范数，因为 l_1 范数不仅有 l_0 范数对稀疏向量的敏感性，又和 l_2 范数有类似的性质，有一个凸单位球，转化为一个凸优化问题。

$$
\begin{aligned}
(P_1): \qquad & \arg\min \|s\|_1 \\
& s.t \;\; \Psi s = v
\end{aligned}
$$

这就是 BP 算法（Basis Pusuit，基追踪）。虽然它能在多项式时间内求解，但是商业优化包对稀疏向量恢复却做的不好。原因是解向量是稀疏的而测量矩阵是稠密的。

针对 (P_0) 问题，下面我们介绍本文使用的两种算法 OMP(Orthogonal Matching Pursuit，正交匹配追踪)和 CoSaMP(Compressive Sampling Match Pursuit，压缩采样匹配追踪)。

2 算法简介

2.1 OMP 算法

输入：

- $N \times d$ 测量矩阵 Ψ
- N 维数据向量 v

- 理想信号s的稀疏度m

输出：

- 稀疏信号的估计$\hat{s} \in R^d$
- 包含从$\{1,...d\}$，m个元素的集合Λ_m
- 数据v的N维近似向量α_m
- N维残差向量$\gamma_m = v - \alpha_m$

步骤：

1) 初始残差$\gamma_0 = v$，指标集$\Lambda_0 = \varnothing$，迭代数$t = 1$.
2) 找出指标λ_t，使得满足如下的优化问题

   $$\lambda_t = \arg\max_{j=1,...d} \left|\left\langle \gamma_{t-1}, \varphi_j \right\rangle\right|$$

   如果最大值出现在多个指标时，确定一个即可。
3) 扩充指标集并用选择的原子扩充矩阵

   $$\Lambda_t = \Lambda_{t-1} \cup \lambda_t, \Psi_t = \begin{bmatrix} \Psi_t & \varphi_{\lambda_t} \end{bmatrix}$$ ，Ψ_0是一个空矩阵
4) 求解一个最小二乘问题来得到新的信号估计：

   $$x_t = \arg\min_x \left\| v - \Psi_t x \right\|_2$$
5) 计算新的数据逼近和新的残差

   $$\alpha_t = \Psi_t x_t$$
   $$\gamma_t = v - \alpha_t$$
6) 增加 t 的值，如果$t \prec m$则返回到第二步
7) 估计出的信号$\hat{s}$在Λ_m位置上是非零的，且$\hat{s}$的第λ_j个分量等于x_t的第j个分量

要保证重构出原始的m阶稀疏信号，测量矩阵$\Psi_{N\times d}$的选取要遵循如下四条性质：

$(M0)$ 独立性：Ψ的各列相互独立

$(M1)$ 归一化：$\left\|\varphi_j\right\|_2^2 = 1,\ \ j = 1, 2..., d.$

$(M2)$ 联合相关：$\{u_t\}$ 是 m 个向量构成的序列，它的 l_2 范数不超过 1，φ 是 Ψ 任意一个列向量，则 φ 与 $\{u_t\}$ 是独立的，且

$$P\left\{\max_t \left|\langle \varphi, u_t \rangle\right| \le \varepsilon\right\} \ge 1 - 2m\, e^{-c\varepsilon^2 N}$$

$(M3)$ 最小奇异值：从 Ψ 中选择 $N \times m$ 的子矩阵 Z，第 m 大的奇异值 $\sigma_m(Z)$ 满足 $P\{\sigma_m(Z) \ge 0.5\} \ge 1 - e^{-cN}$.

通常我们选取归一化后的高斯随机测量矩阵或伯努利随机测量矩阵，本文只使用高斯随机测量矩阵实验。

OMP 对高斯测量矩阵重构信号有如下定理：

定理 2 固定 $\delta \in (0, 0.36)$，选取 $N \ge Km \ln(d/\delta)$.假定 s 是任意 d 维 m 阶稀疏的信号从 d 维标准高斯分布中取出 N 个独立的测量向量 $x_1, x_2, \ldots x_N$.给出数据 $\{\langle s, x_n \rangle, n = 1, 2, \ldots N\}$，OMP 能以至少 $1-2\delta$ 的概率重构信号。常数 $K \le 20$，对于较大的 m， K 可以减小到 $K \approx 4$

OMP 算法的思想是每一步的迭代选取测量矩阵 Φ 与 v 剩余部分最接近的那一列，用新得到的原子构成 Φ_t 来近似 Φ，经过 m 此迭代或残差足够小停止。

2.2 CoSaMP 算法

输入：

(1) 测量矩阵 Φ
(2) $N \times 1$ 维观测向量
(3) 信号的稀疏度 K

输出：

(1) 重构的信号 x_t
(2) 残差 $r_s = y - \Phi_t x_t$

以下流程中：r_t 表示残差，t 表示迭代次数，J_0 表示每次迭代找到的索引（列序号），Λ_t 表示 t 次迭代的索引（注意：设元素个数为 Lt，一般 $Lt \ne t$，因为每次迭代找到的索引 J_0 一般并非只含有一个列序号），φ_j 表示矩阵 Φ 的第 j 列，A_t 表示按索引 Λ_t 选出的矩阵 A 的列集合，θ_t 为 $Lt \times 1$ 的列向量，符号 $\cup$ 表示集合并运算，$\langle \bullet, \bullet \rangle$ 表示求向量内积。$abs(\bullet)$ 表示求绝对值

(1) 初始化$r_0 = y, \Lambda_0 = \varnothing, \Phi_0 = \varnothing, t = 1$

(2) 计算$u = abs\left[\Phi^T r_{t-1}\right]$（即计算$\left\langle r_{t-1}, \varphi_j \right\rangle, 1 \le j \le N$），选择$u$中$2K$个最大值，将这些值对应$\Phi$的列序号$j$构成集合$J_0$

(3) 令$\Lambda_t = \Lambda_{t-1} \cup J_0$，$\Phi_t = \Phi_{t-1} \cup \varphi_j$（for all $j \in J_0$）

(4) 求$y = \Phi_t \theta_t$的最小二乘解：$\hat{\theta}_t = \arg\min \left\| y - \Phi_t \theta_t \right\| = (\Phi_t^T \Phi_t)^{-1} \Phi_t^T y$

(5) 从$\hat{\theta}_t$中选出绝对值最大的K项记为$\hat{\theta}_{tK}$，对应的Φ_t中的K列记为$\hat{\theta}_{tK}$，对应的Φ的列序号记为Λ_{tK}，更新集合$\Lambda_t = \Lambda_{tK}$

(6) 更新残差$r_t = y - \Phi_{tK} \hat{\theta}_t$

(7) $t = t+1$,如果$t \le s$返回到（2）继续迭代，如果$t \succ s$货$r_t = 0$则停止迭代进入（8）

(8) 重构所得$\hat{\theta}$在Λ_{tK}有非零项，其值分别为最后一次迭代所得，$\hat{\theta}$就是恢复的信号x

定理（CoSaMP）假设Φ是一个$m \times N$采样矩阵，严格等距常数$\delta_{2s} \le c$。$u = \Phi x + e$是一个任意信号的样本向量，包含任意噪声。对一个给定的精确参量η，CoSaMP 算法能产生一个$2s$稀疏的近似a，且满足：

$$\left\| x - a \right\|_2 \le C \cdot \max\left\{ \eta, \frac{1}{\sqrt{s}} \left\| x - x_{s/2} \right\|_1 + \left\| e \right\|_2 \right\}$$

$x_{s/2}$是x的一个最佳的$s/2$稀疏近似。运行时间$O(\ell * log(\left\| x \right\|_2 / \eta))$，$\ell$是矩阵-向量与$\Phi$或$\Phi^*$乘法所耗费的时间边界。算法储存空间花费大约$O\left(N\right)$。[4]

压缩采样匹配追踪(Compressive Sampling MP)是 D. Needell 继 OMP 之后提出的又一个具有较大影响力的重构算法。CoSaMP 是对 OMP 的一种改进，每次迭代选择多个原子，除了原子的选择标准之外，它有一点不同于 OMP：OMP 每次迭代已经选择的原子会一直保留，而 CoSaMP 每次迭代选择的原子在下次迭代中可能会被抛弃。

3 实验算例

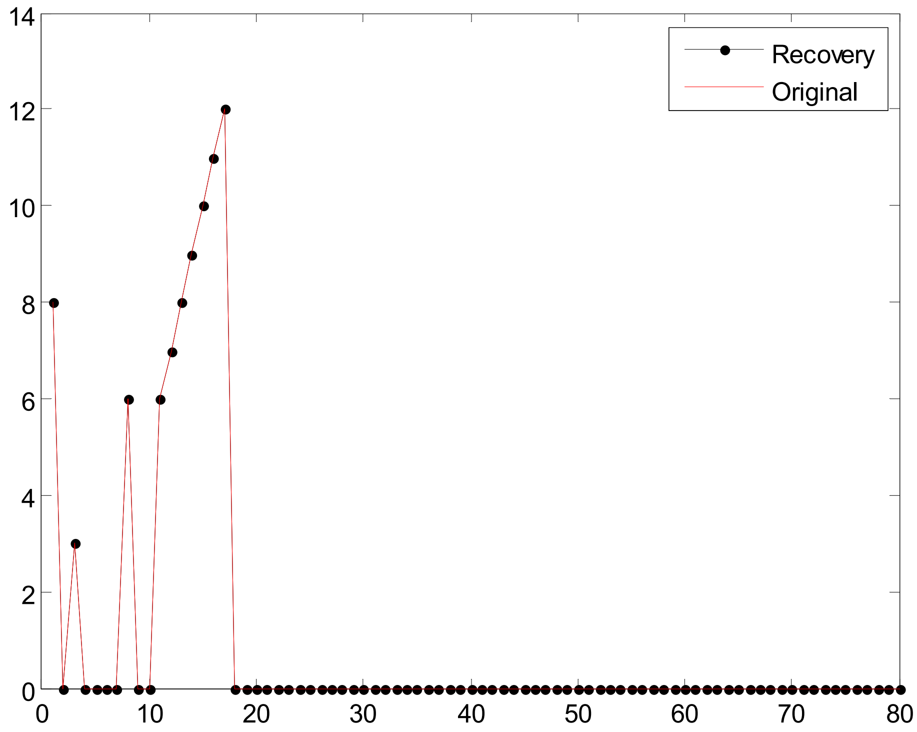
3.1 OMP 实验结果

3.1.1 实验一

人工生成一个稀疏度为 10，维数为 80 的待重构的稀疏信号，测量矩阵选择40×80高斯测量矩阵(稀疏度是测量次数的$1/4$)，由于高斯测量矩阵生成的随机性，每次重构信

号只是以高概率达到，因此多次运行程序尝试（多次重构效果都很差的概率很低），得到如下最好的重构结果。

重构后信号与原信号的图像如下：



程序共用的时间是：Elapsed time is 0.015860 seconds.

恢复残差：

ans =

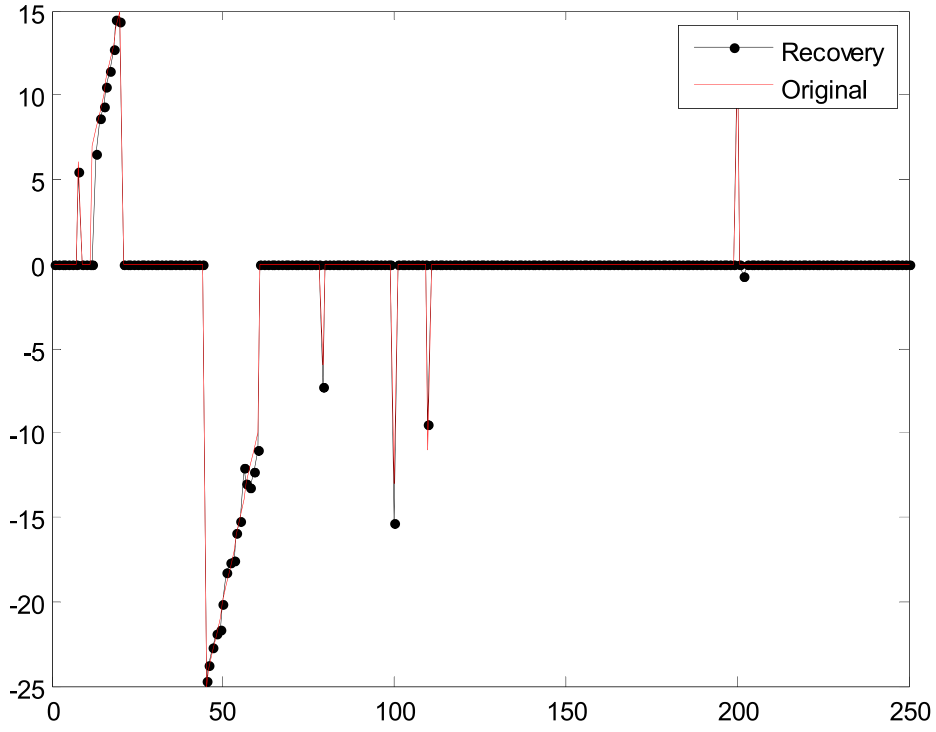
  4.7207e-015

由图可看出恢复的信号与原始信号结果很近。

3.1.2　实验二

同样地，人工生成一个稀疏度为 30，维数为 250 的待重构的稀疏信号，测量矩阵选择 90×250 高斯测量矩阵(稀疏度是测量次数的 $1/3$)，重构结果如下：

重构后信号与原信号的图像：



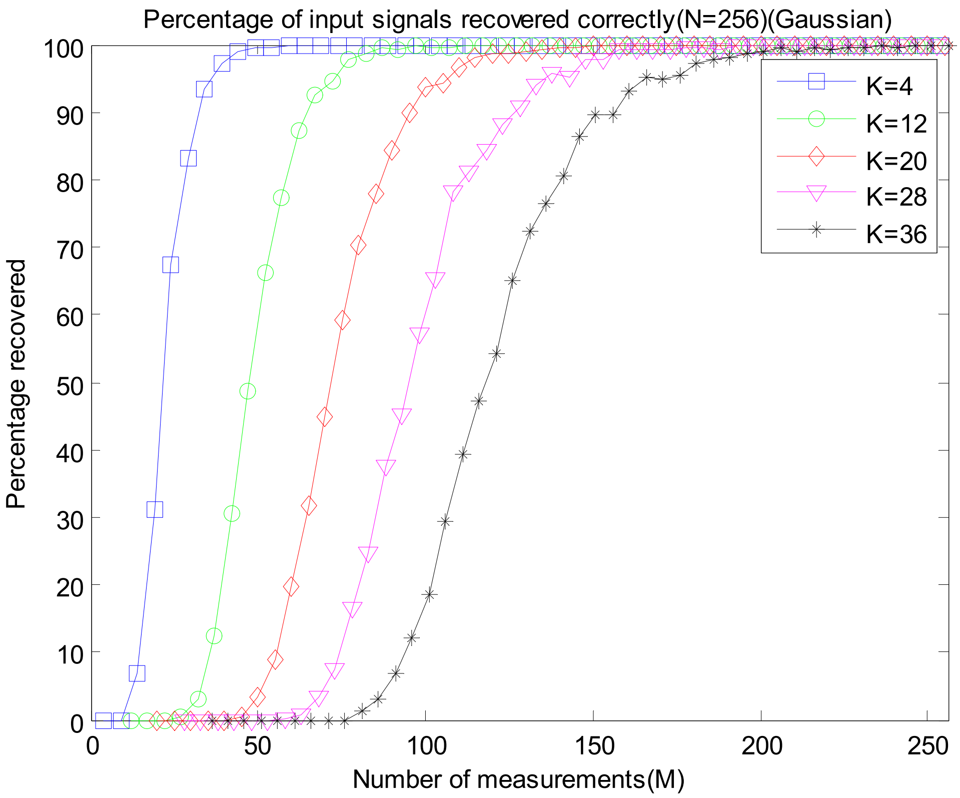
程序共用的时间是：Elapsed time is 0.043242 seconds.

恢复残差：

ans =

    8.6341

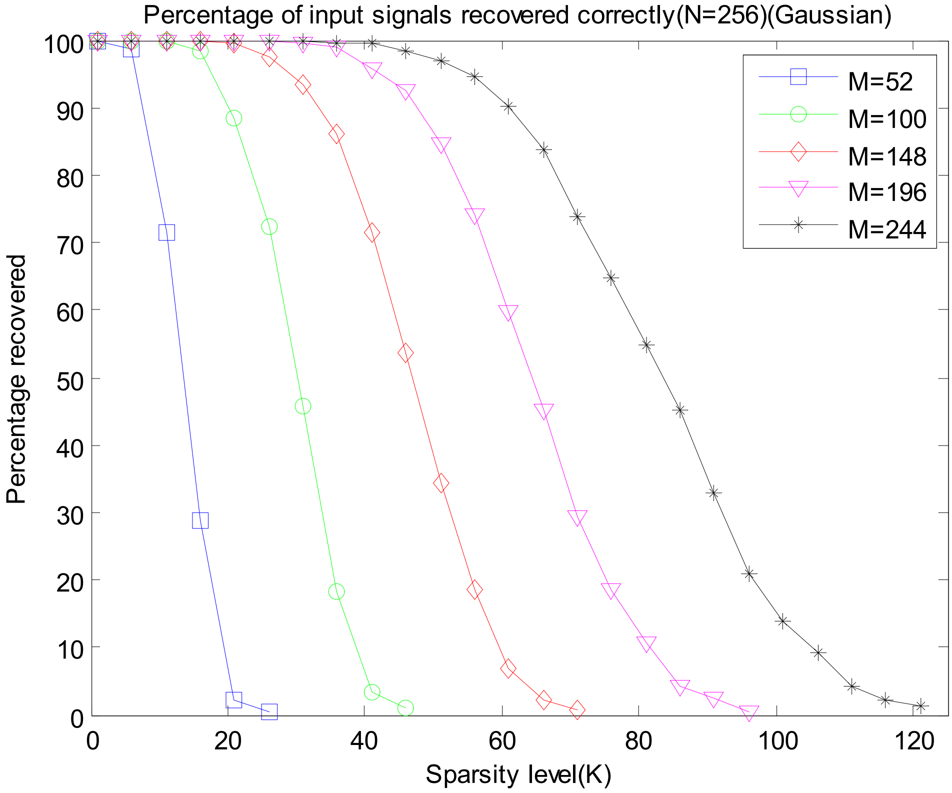
3.1.3 测量数 M 与重构成功概率关系曲线



程序共用时间：Elapsed time is 1910.100623 seconds.

由图像可看出随着测量次数的增加，信号重构成功的概率是逐渐增加；在测量次数相同情况下，稀疏度越小的信号重构成功的概率越高。

3.1.4 信号稀疏度 K 与重构成功的概率关系曲线



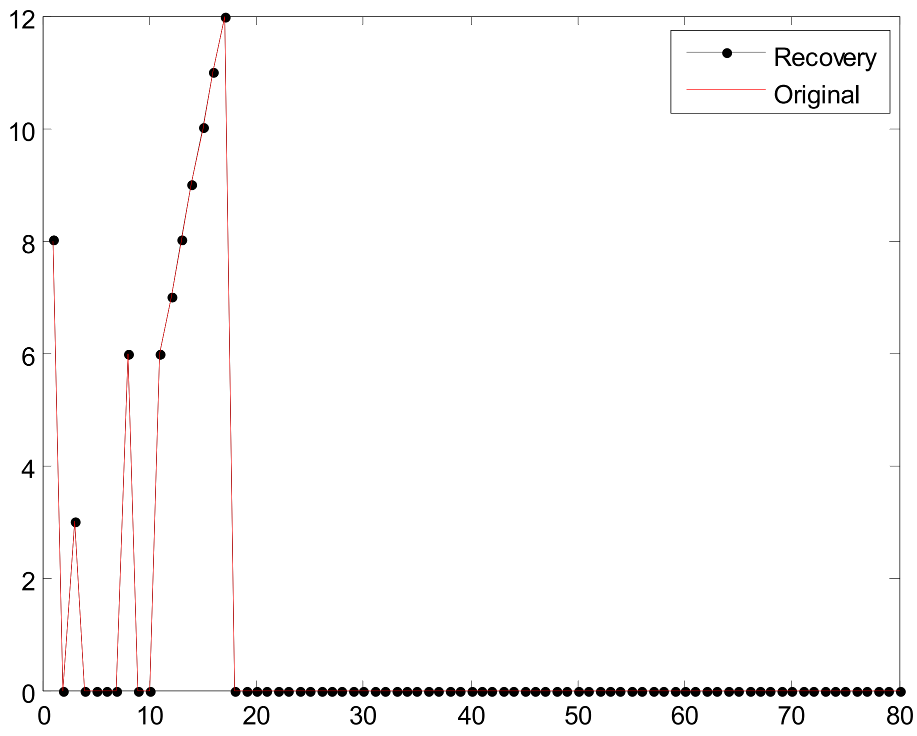
程序共用时间：Elapsed time is 2623.643396 seconds.

由图像看出在测量次数相同情况下，随着稀疏度增加，信号重构成功的概率越低。

3.2 CoSaMP 实验结果

3.2.1 实验一

与 3.1.1 原始稀疏信号相同，重构结果如下：



程序共用的时间是：Elapsed time is 0.059481 seconds.

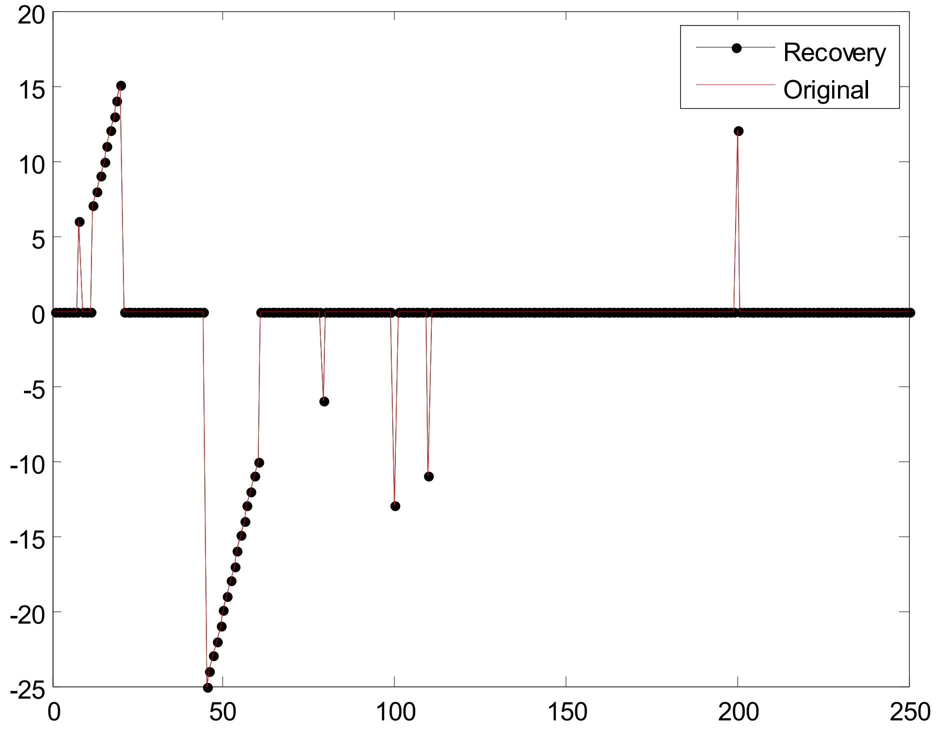
恢复残差：

ans =

  2.1130e-014

3.2.2 实验二

与 3.1.2 原始稀疏信号相同，重构结果如下：



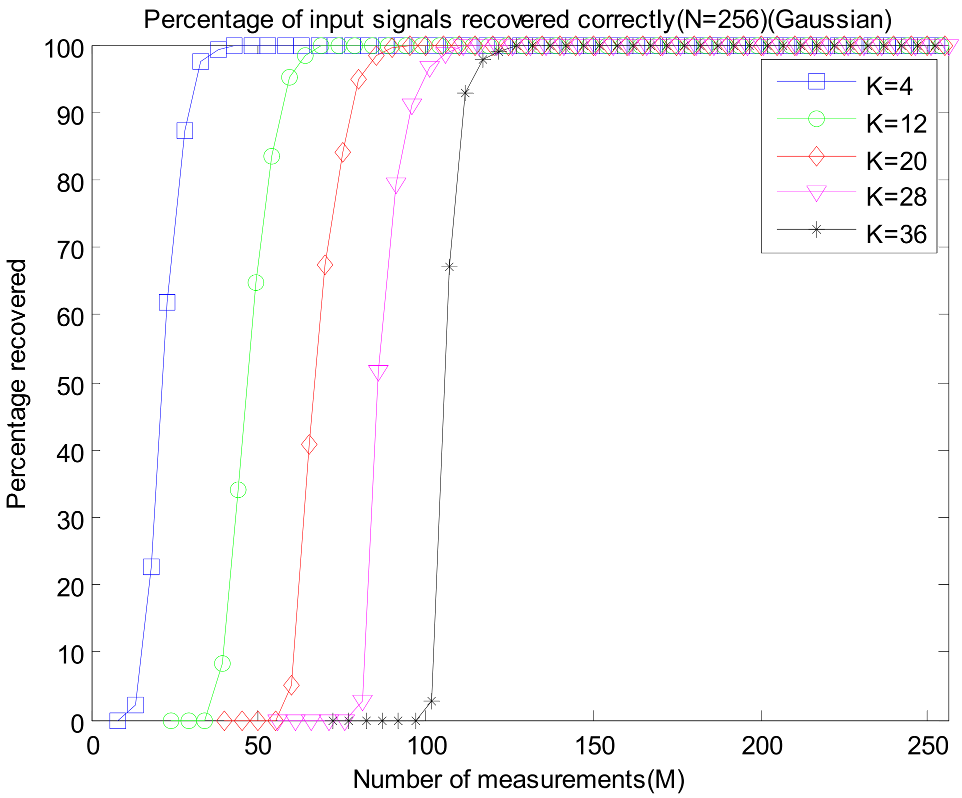
程序共用的时间是：Elapsed time is 0.079302 seconds.

恢复残差：

ans =

1.6909e-012

3.2.3 CoSaMP 测量数 M 与重构成功概率关系曲线绘制



程序共用时间：Elapsed time is 1774.610025 seconds.

由图像可看出随着测量次数的增加，信号重构成功的概率是逐渐增加；在测量次数相同情况下，稀疏度越小的信号重构成功的概率越高。

4　附录

% omp_test1

```
clc;clear all;close all;
A = randn(40,80); %生成一个 40*80 的高斯随机矩阵，40 为测量次数，80 为数据的维数
for i=1:80
    G(:,i) = A(:,i)/sqrt(norm(A(:,i),2));
    %将矩阵的每列进行规范化，使 G 的各列 L2 范数为 1，此 G 为需要的高斯测量矩阵
end
%-----------------------------------
%人工生成一个稀疏度为 10，维数为 80 的信号 x
x=zeros(1,80)';
x(1) = 8;
```

```
x(3) = 3;
x(8) = 6;
for i=11:17
   x(i) = i-5;
end
%-----------------------------------
y = G*x;                    %G 为范数归一后的测量矩阵
recovery_x=zeros(1,80);       %需要待重构的信号向量
f = [];                      %指标集对应的列待扩充到的空矩阵
r = y;                      %初始残差值
tic                          %计时开始
for t = 1:40/4               %迭代次数(稀疏度是测量的 1/4)
   for col = 1:80;
      product(col)=abs(G(:,col)'*r);
      % 恢复矩阵的列向量和残差的投影系数(内积值)
   end
   [val,pos]=max(product);   %最大投影系数对应的位置存储在 pos，相应最大值返
回到 val
    f=[f,G(:,pos)];          %将指标集对应的列扩充到矩阵 f
   G(:,pos)=zeros(40,1);     %选中的列置零（实质上应该去掉，为了简单把它置零）
   xt=(f'*f)^(-1)*f'*y;      %最小二乘法求解满足||y-fi*x||平方最小的 xt
   r=y-f*xt;                 %更新残差
   pos_array(t)=pos;         %纪录最大投影系数的位置
end
recovery_x(pos_array)=xt;      %重构的行向量
recovery_x = recovery_x';      %转为列向量，重构的信号 recovery_x
toc                          %计时结束
%  绘图
figure;
plot(recovery_x,'k.-');              %绘出 x 的恢复信号
hold on;
plot(x,'r');                  %绘出原信号 x
hold off;
legend('Recovery','Original')
fprintf('\n 恢复残差：');
norm(recovery_x-x)                %恢复残差
```

```
% omp_test2
```

```
clc;clear all;close all;
A = randn(90,250); %生成一个 90*250 的高斯随机矩阵，90 为测量次数，250 为数据的维数
for i=1:250
    G(:,i) = A(:,i)/sqrt(norm(A(:,i),2));
    %将矩阵的每列进行规范化，使 G 的各列 L2 范数为 1，此 G 为需要的高斯测量矩阵
end
%-----------------------------------
%人工生成一个稀疏度为 30，维数为 250 的信号 x
x=zeros(1,250)';
x(8) = 6;
x(200) = 12;
for i=12:20
    x(i) = i-5;
end
x(100) = -13;x(110) = -11;
x(79) = -6;
for i=45:60
    x(i) = i -70;
end
%-----------------------------------
y = G*x;                    %G 为范数归一后的测量矩阵
recovery_x=zeros(1,250);      %需要待重构的信号向量
f = [];                     %指标集对应的列待扩充到的空矩阵
r = y;                      %初始残差值
tic                         %计时开始
for t = 1:90/3               %迭代次数(稀疏度是测量的 1/3)
    for col = 1:250;
        product(col)=abs(G(:,col)'*r);
        % 恢复矩阵的列向量和残差的投影系数(内积值)
    end
    [val,pos]=max(product);   %最大投影系数对应的位置存储在 pos，相应最大值返回到 val
     f=[f,G(:,pos)];          %将指标集对应的列扩充到矩阵 f
    G(:,pos)=zeros(90,1);     %选中的列置零（实质上应该去掉，为了简单把它置零）
    xt=(f'*f)^(-1)*f'*y;       %最小二乘法求解满足||y-fi*x||平方最小的 xt
    r=y-f*xt;                 %更新残差
    pos_array(t)=pos;          %纪录最大投影系数的位置
end
recovery_x(pos_array)=xt;      %重构的行向量
recovery_x = recovery_x';      %转为列向量，重构的信号 recovery_x
toc                         %计时结束
```

```
% 绘图
figure;
plot(recovery_x,'k.-');       %绘出 x 的恢复信号
hold on;
plot(x,'r');                  %绘出原信号 x
hold off;
legend('Recovery','Original')
fprintf('\n 恢复残差：');
norm(recovery_x-x)            %恢复残差
```

% omp_test3

```
clear all;close all;clc;
%% 参数配置初始化
CNT = 1000;%对于每组(K,M,N)，重复迭代次数
N = 256;%信号 x 的长度
Psi = eye(N);%x 本身是稀疏的，定义稀疏矩阵为单位阵 x=Psi*theta
K_set = [4,12,20,28,36];%信号 x 的稀疏度集合
Percentage = zeros(length(K_set),N);%存储恢复成功概率
%% 主循环，遍历每组(K,M,N)
tic
for kk = 1:length(K_set)
   K = K_set(kk);%本次稀疏度
   M_set = K:5:N;%每隔 5 测试一个
   PercentageK = zeros(1,length(M_set));%存储此稀疏度 K 下不同 M 的恢复成功概
率
   for mm = 1:length(M_set)
     M = M_set(mm);%本次观测值个数
     P = 0;
     for cnt = 1:CNT %每个观测值个数均运行 CNT 次
        Index_K = randperm(N);
        x = zeros(N,1);
        x(Index_K(1:K)) = 5*randn(K,1);%x 为 K 稀疏的，且位置是随机的
        Phi = randn(M,N);%测量矩阵为高斯矩阵
        A = Phi * Psi;%传感矩阵
        y = Phi * x;%得到观测向量 y
        theta = CS_OMP(y,A,K);%恢复重构信号 theta
        x_r = Psi * theta;% x=Psi * theta
        if norm(x_r-x)<1e-6%如果残差小于 1e-6 则认为恢复成功
           P = P + 1;
        end
     end
```

```
      PercentageK(mm) = P/CNT*100;%计算恢复概率
    end
    Percentage(kk,1:length(M_set)) = PercentageK;
end
toc
save MtoPercentage1000 %把变量全部存储下来
%% 绘图
S = ['-bs';'-go';'-rd';'-mv';'-k*'];
figure;
for kk = 1:length(K_set)
    K = K_set(kk);
    M_set = K:5:N;
    L_Mset = length(M_set);
    plot(M_set,Percentage(kk,1:L_Mset),S(kk,:));%绘出 x 的恢复信号
    hold on;
end
hold off;
xlim([0 256]);
legend('K=4','K=12','K=20','K=28','K=36');
xlabel('Number of measurements(M)');
ylabel('Percentage recovered');
title('Percentage of input signals recovered correctly(N=256)(Gaussian)');
```

%信号稀疏度 K 与重构成功概率关系曲线代码

```
clear all;close all;clc;
%% 参数配置初始化
CNT = 1000;%对于每组(K,M,N)，重复迭代次数
N = 256;%信号 x 的长度
Psi = eye(N);%x 本身是稀疏的，定义稀疏矩阵为单位阵 x=Psi*theta
M_set = [52,100,148,196,244];%测量值集合
Percentage = zeros(length(M_set),N);%存储恢复成功概率
%% 主循环，遍历每组(K,M,N)
tic
for mm = 1:length(M_set)
    M = M_set(mm);%本次测量值个数
    K_set = 1:5:ceil(M/2);%信号 x 的稀疏度 K 没必要全部遍历，每隔 5 测试一个就可
以了
    PercentageM = zeros(1,length(K_set));%存储此测量值 M 下不同 K 的恢复成功概
率
    for kk = 1:length(K_set)
      K = K_set(kk);%本次信号 x 的稀疏度 K
      P = 0;
      for cnt = 1:CNT %每个观测值个数均运行 CNT 次
```

```
            Index_K = randperm(N);
            x = zeros(N,1);
            x(Index_K(1:K)) = 5*randn(K,1);%x 为 K 稀疏的，且位置是随机的
            Phi = randn(M,N);%测量矩阵为高斯矩阵
            A = Phi * Psi;%传感矩阵
            y = Phi * x;%得到观测向量 y
            theta = CS_OMP(y,A,K);%恢复重构信号 theta
            x_r = Psi * theta;% x=Psi * theta
            if norm(x_r-x)<1e-6%如果残差小于 1e-6 则认为恢复成功
                P = P + 1;
            end
        end
        PercentageM(kk) = P/CNT*100;%计算恢复概率
    end
    Percentage(mm,1:length(K_set)) = PercentageM;
end
toc
save KtoPercentage1000test %运行一次不容易，把变量全部存储下来
%% 绘图
S = ['-bs';'-go';'-rd';'-mv';'-k*'];
figure;
for mm = 1:length(M_set)
    M = M_set(mm);
    K_set = 1:5:ceil(M/2);
    L_Kset = length(K_set);
    plot(K_set,Percentage(mm,1:L_Kset),S(mm,:));%绘出 x 的恢复信号
    hold on;
end
hold off;
xlim([0 125]);
legend('M=52','M=100','M=148','M=196','M=244');
xlabel('Sparsity level(K)');
ylabel('Percentage recovered');
title('Percentage of input signals recovered correctly(N=256)(Gaussian)');
```

%CoSaMP

```
function[s]=CS_CoSaMP(v,phi,m)
%s 为重构出的信号
%phi 为测量矩阵
%m 为稀疏信号的稀疏度
%v = phi*s,为所测量的值
[r,col] = size(v);
if  r < col
    v = v';             %v 应当是一个列向量
```

```
end
[M,N] = size(phi);        %测量矩阵 phi 为 M*N
s = zeros(N,1);           %用来存储待恢复的 s 信号
pos_s  = [];              %存储 phi 被选择的列序号
r_n = v;                  %初始化残差
for t = 1:m
    product = phi'*r_n;  %phi 各列与残差做内积
    [val,pos] = sort(abs(product),'descend');
    Js = pos(1:2*m);      %选出内积最大的 2m 列
    Is = union(pos_s,Js);%pos_s 与 Js 并集
    if length(Is)<=M
        phi_t=phi(:,Is);
    else
        if t ==1
            s_ls = 0;
        end
        break;
    end
    s_ls = (phi_t'*phi_t)^(-1)*phi_t'*v; %最小二乘法求解
    [val,pos] = sort(abs(s_ls),'descend');
    pos_s=Is(pos(1:m));
    s_ls = s_ls(pos(1:m));
    %phi_t(:,pos(1:m))*s_ls 是 v 在 phi_t(:,pos(1:m))列空间上的正交投影
    r_n = v - phi_t(:,pos(1:m))*s_ls;  %更新残差
    if norm(r_n)<1e-6
        break;
    end
end
s(pos_s) = s_ls;                  %恢复出的 s
end
```

%CosaMP_test1

```
clc;clear all;close all;
A = randn(40,80); %生成一个 40*80 的高斯随机矩阵，40 为测量次数，80 为数据的维数
for i=1:80
    G(:,i) = A(:,i)/sqrt(norm(A(:,i),2));
    %将矩阵的每列进行规范化，使 G 的各列 L2 范数为 1，此 G 为需要的高斯测量矩阵
end
%-----------------------------------
%人工生成一个稀疏度为 10，维数为 80 的信号 x
x=zeros(1,80)';
x(1) = 8;
```

```
x(3) = 3;
x(8) = 6;
for i=11:17
    x(i) = i-5;
end
%-----------------------------------
y = G*x;                    %G 为范数归一后的测量矩阵
m = 10;                     %稀疏度为 10
tic                      %计时开始
recovery_x = CS_CoSaMP( y,G,m );
toc                       %计时结束
%绘图
figure;
plot(recovery_x,'k.-');       %绘出 x 的恢复信号
hold on;
plot(x,'r');                 %绘出原信号 x
hold off;
legend('Recovery','Original')
fprintf('\n 恢复残差：');
norm(recovery_x-x)           %恢复残差
```

%CosaMP_test2

```
clc;clear all;close all;
A = randn(90,250); %生成一个 90*250 的高斯随机矩阵，90 为测量次数，250 为数据的维数
for i=1:250
    G(:,i) = A(:,i)/sqrt(norm(A(:,i),2));
    %将矩阵的每列进行规范化，使 G 的各列 L2 范数为 1，此 G 为需要的高斯测量矩阵
end
%-----------------------------------
%人工生成一个稀疏度为 30，维数为 250 的信号 x
x=zeros(1,250)';
x(8) = 6;
x(200) = 12;
for i=12:20
    x(i) = i-5;
end
x(100) = -13;x(110) = -11;
x(79) = -6;
for i=45:60
    x(i) = i -70;
end
%-----------------------------------
y = G*x;                    %G 为范数归一后的测量矩阵
```

```
m = 30;                    %稀疏度为 10
tic                        %计时开始
recovery_x = CS_CoSaMP( y,G,m );
toc                        %计时结束
%绘图
figure;
plot(recovery_x,'k.-');      %绘出 x 的恢复信号
hold on;
plot(x,'r');                 %绘出原信号 x
hold off;
legend('Recovery','Original')
fprintf('\n 恢复残差：');
norm(recovery_x-x)           %恢复残差
```

%CoSaMP 测量数 M 与重构成功概率关系曲线绘制

```
clear all;close all;clc;
%% 参数配置初始化
CNT = 1000;%对于每组(K,M,N)，重复迭代次数
N = 256;%信号 x 的长度
Psi = eye(N);%x 本身是稀疏的，定义稀疏矩阵为单位阵 x=Psi*theta
K_set = [4,12,20,28,36];%信号 x 的稀疏度集合
Percentage = zeros(length(K_set),N);%存储恢复成功概率
%% 主循环，遍历每组(K,M,N)
tic
for kk = 1:length(K_set)
  K = K_set(kk);%本次稀疏度
   M_set = 2*K:5:N;%M 没必要全部遍历，每隔 5 测试一个就可以了
   PercentageK = zeros(1,length(M_set));%存储此稀疏度 K 下不同 M 的恢复成功概
率
   for mm = 1:length(M_set)
      M = M_set(mm);%本次观测值个数
      fprintf('K=%d,M=%d\n',K,M);
      P = 0;
      for cnt = 1:CNT %每个观测值个数均运行 CNT 次
           Index_K = randperm(N);
           x = zeros(N,1);
           x(Index_K(1:K)) = 5*randn(K,1);%x 为 K 稀疏的，且位置是随机的
           Phi = randn(M,N)/sqrt(M);%测量矩阵为高斯矩阵
           A = Phi * Psi;%传感矩阵
           y = Phi * x;%得到观测向量 y
           theta = CS_CoSaMP(y,A,K);%恢复重构信号 theta
```

```
            x_r = Psi * theta;% x=Psi * theta
            if norm(x_r-x)<1e-6%如果残差小于 1e-6 则认为恢复成功
                P = P + 1;
            end
        end
      PercentageK(mm) = P/CNT*100;%计算恢复概率
    end
    Percentage(kk,1:length(M_set)) = PercentageK;
end
toc
save CoSaMPMtoPercentage1000 %把变量全部存储下来
%% 绘图
S = ['-bs';'-go';'-rd';'-mv';'-k*'];
figure;
for kk = 1:length(K_set)
    K = K_set(kk);
    M_set = 2*K:5:N;
    L_Mset = length(M_set);
    plot(M_set,Percentage(kk,1:L_Mset),S(kk,:));%绘出 x 的恢复信号
    hold on;
end
hold off;
xlim([0 256]);
legend('K=4','K=12','K=20','K=28','K=36');
xlabel('Number of measurements(M)');
ylabel('Percentage recovered');
title('Percentage of input signals recovered correctly(N=256)(Gaussian)');
```

5 参考文献

[1] E.J.Candès and T.Tao, "Decoding by linear programming," IEEE Trans.Inf.Theory ,vol.51,no.12,pp.4203–4215,Dec.2005.

[2] M.Rudelson and R.Veshynin,"Geometric approach to error correcting

codes and reconstruction of signals," Int. Math. Res. Not., vol. 64, pp

4019–4041, 2005.

[3] Tropp J /, Gilbert A C. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit[J]. IEEE Transactions on Information Theory, 2008, 53(12):4655-4666.

[4]D. Needell, J.A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples, ACM Technical Report 2008-01, California Institute ofTechnology, Pasadena, 2008.