

第二题

matlab代码:

```
1 clear;clc;
2
3 m = 100;n = 500;s = 50;
4 A = randn(m,n);
5 xs = zeros(n,1);picks=randperm(n);xs(picks(1:s)) = randn(s,1);
6 b = A*xs;tau=0.001;
7
8 cvx_begin
9     variable x(n)
10    minimize 1/2*square_pos(norm(A*x-b))+tau*norm(x,1)
11 cvx_end
12
13 f = @(x) 1/2*square_pos(norm(A*x-b))+tau*norm(x,1);
14
15 % constant step size
16 x = zeros(n,1); alpha = 0.0001;
17 for k = 1 : 10000
18     funcv = f(x);
19     funcv_diff_const_step_size_1(k) = funcv-cvx_optval;
20     % 计算在x点处的一个次梯度
21     subgd = A'*(A*x-b)+tau*sign(x);
22     % 更新x
23     x = x-alpha*subgd;
24 end
25
26 x = zeros(n,1); alpha = 0.0005;
27 for k = 1 : 10000
28     funcv = f(x);
29     funcv_diff_const_step_size_2(k) = funcv-cvx_optval;
30     % 计算在x点处的一个次梯度
31     subgd = A'*(A*x-b)+tau*sign(x);
32     % 更新x
33     x = x-alpha*subgd;
34 end
35
36 x = zeros(n,1); alpha = 0.001;
37 for k = 1 : 10000
38     funcv = f(x);
39     funcv_diff_const_step_size_3(k) = funcv-cvx_optval;
40     % 计算在x点处的一个次梯度
41     subgd = A'*(A*x-b)+tau*sign(x);
42     % 更新x
43     x = x-alpha*subgd;
44 end
45
46 % constant step length
47 x = zeros(n,1); gamma = 0.0005;
48 for k = 1 : 10000
49     funcv = f(x);
50     funcv_diff_const_step_len_1(k) = funcv-cvx_optval;
51     % 计算在x点处的一个次梯度
52     subgd = A'*(A*x-b)+tau*sign(x);
```

```

53     % 更新x
54     x = x-gamma/norm(subgd)*subgd;
55 end
56
57 x = zeros(n,1); gamma = 0.001;
58 for k = 1 : 10000
59     funcv = f(x);
60     funcv_diff_const_step_len_2(k) = funcv-cvx_optval;
61     % 计算在x点处的一个次梯度
62     subgd = A'*(A*x-b)+tau*sign(x);
63     % 更新x
64     x = x-gamma/norm(subgd)*subgd;
65 end
66
67 x = zeros(n,1); gamma = 0.005;
68 for k = 1 : 10000
69     funcv = f(x);
70     funcv_diff_const_step_len_3(k) = funcv-cvx_optval;
71     % 计算在x点处的一个次梯度
72     subgd = A'*(A*x-b)+tau*sign(x);
73     % 更新x
74     x = x-gamma/norm(subgd)*subgd;
75 end
76
77 % diminishing step size
78 x = zeros(n,1);
79 for k = 1 : 10000
80     funcv = f(x);
81     funcv_diff_dimi_step_size_1(k) = funcv-cvx_optval;
82     % 计算在x点处的一个次梯度
83     subgd = A'*(A*x-b)+tau*sign(x);
84     % 更新x
85     x = x-0.0005/sqrt(k)*subgd;
86 end
87
88 x = zeros(n,1);
89 for k = 1 : 10000
90     funcv = f(x);
91     funcv_diff_dimi_step_size_2(k) = funcv-cvx_optval;
92     % 计算在x点处的一个次梯度
93     subgd = A'*(A*x-b)+tau*sign(x);
94     % 更新x
95     x = x-0.001/sqrt(k)*subgd;
96 end
97
98 x = zeros(n,1);
99 for k = 1 : 10000
100     funcv = f(x);
101     funcv_diff_dimi_step_size_3(k) = funcv-cvx_optval;
102     % 计算在x点处的一个次梯度
103     subgd = A'*(A*x-b)+tau*sign(x);
104     % 更新x
105     x = x-0.001/k*subgd;
106 end
107
108 % Polyak's step size

```

```

109 x = zeros(n,1);
110 for k = 1 : 10000
111     funcv = f(x);
112     funcv_diff_poly_step_size(k) = funcv-cvx_optval;
113     % 计算在x点处的一个次梯度
114     subgd = A'*(A*x-b)+tau*sign(x);
115     % 更新x
116     x = x-(funcv-cvx_optval)/sum_square(subgd)*subgd;
117 end
118
119 figure;
120 semilogy(funcv_diff_const_step_size_1, 'DisplayName', 'alpha(k)=0.0001',
121 'Linewidth', 1.5);
122 hold on
123 semilogy(funcv_diff_const_step_size_2, 'DisplayName', 'alpha(k)=0.0005',
124 'Linewidth', 1.5);
125 semilogy(funcv_diff_const_step_size_3, 'DisplayName', 'alpha(k)=0.001',
126 'Linewidth', 1.5);
127 xlabel('Iteration');
128 ylabel('f(xk)-f*');
129 legend;
130 hold off;
131
132 figure;
133 semilogy(funcv_diff_const_step_len_1, 'DisplayName',
134 'alpha(k)=0.0005/norm(subgrad)', 'Linewidth', 1.5);
135 hold on;
136 semilogy(funcv_diff_const_step_len_2, 'DisplayName',
137 'alpha(k)=0.001/norm(subgrad)', 'Linewidth', 1.5);
138 semilogy(funcv_diff_const_step_len_3, 'DisplayName',
139 'alpha(k)=0.005/norm(subgrad)', 'Linewidth', 1.5);
140 xlabel('Iteration');
141 ylabel('f(xk)-f*');
142 legend;
143 hold off;
144
145 figure;
146 semilogy(funcv_diff_dimi_step_size_1, 'DisplayName',
147 'alpha(k)=0.0005/sqrt(k)', 'Linewidth', 1.5);
148 hold on;
149 semilogy(funcv_diff_dimi_step_size_2, 'DisplayName',
150 'alpha(k)=0.001/sqrt(k)', 'Linewidth', 1.5);
151 semilogy(funcv_diff_dimi_step_size_3, 'DisplayName', 'alpha(k)=0.001/k',
152 'Linewidth', 1.5);
153 xlabel('Iteration');
154 ylabel('f(xk)-f*');
155 legend;

```

