

Contents

- [prepare workspace](#)
- [load the variables of the optimization problem](#)
- [set up the function and its gradient](#)
- [parameters of the gradient method](#)
- [optimize](#)

```
% GM with a fixed step
%
% Least squares: gradient method with fixed step
%
% U. S. Kamilov, CIG, WUSTL, 2021.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

prepare workspace

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear; close all; home;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

load the variables of the optimization problem

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
load('dataset.mat');
```

```
[m, n] = size(A); % m rows, n cols
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

set up the function and its gradient

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
stepSize = 1; % step-size of the gradient method
lambda = 0.02;
```

```
evaluateFunc = @(x) (1/2)*norm(A*x-b)^2;
evaluateGrad = @(x) A'*(A*x-b);
```

```
evaluate_g = @(x) lambda*norm(x,1);
evaluateGrad_g = @(y) abs(A').*sign(y); % This is the rule for subgrad of g
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

parameters of the gradient method

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

xInit = zeros(n, 1); % zero initialization
maxIter = 200; % maximum number of iterations
beta = 0.5; % Step reduction parameter
phi = 0.5;
thetaPast = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

optimize

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% initialize
x = xInit;
xPast = x;

% keep track of cost function values
objVals = zeros(maxIter, 1);

% iterate
for iter = 1:maxIter

    % gradient at w
    grad = evaluateGrad(x);

    % AGM
    theta = (1+sqrt(1+4*thetaPast^2))/2;
    beta_t = (thetaPast - 1)/theta;
    s = x + beta_t*(x - xPast);
    % Update x
    xPast = x;

    % update AGM
    x = s - stepSize*evaluateGrad(s);

    % stepSize = 100;

    % BLS
    while (evaluateFunc(x - stepSize*evaluateGrad(x)) > evaluateFunc(x) ...
          - phi*stepSize*norm(evaluateGrad(x))^2)
        stepSize = beta*stepSize;
    end

    % update GDM
    xNext = x - stepSize*evaluateGrad(x);

    % SGD
    xNext = x - stepSize*evaluateGrad_g(x);

    % evaluate the objective
    funcNext = evaluateFunc(xNext);

    % store the objective and the classification error
    objVals(iter) = funcNext;

    fprintf('%d/%d [step: %.1e] [objective: %.1e] [norm(grad): %.1e]\n',...
            iter, maxIter, stepSize, objVals(iter), norm(grad));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% begin visualize data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% plot the evolution
figure(1);
set(gcf, 'Color', 'w');
semilogy(1:iter, objVals(1:iter), 'b-',...
    iter, objVals(iter), 'b*', 'LineWidth', 2);
grid on;
axis tight;
xlabel('iteration');
ylabel('objective');
title(sprintf('cost: %.4e', objVals(iter)));
xlim([1 maxIter]);
set(gca, 'FontSize', 16);
drawnow;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% end visualize data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% update w
%thetaPast = theta;
x = xNext;
end

```

```

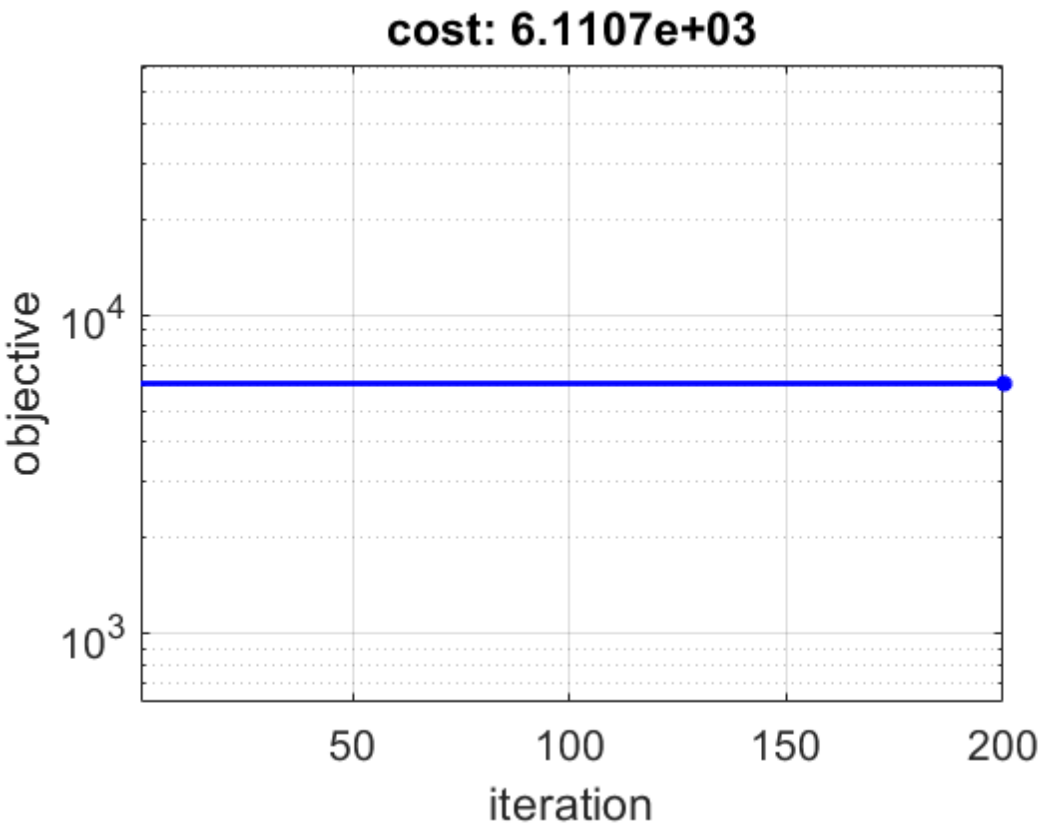
[1/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.2e+02]
[2/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[3/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[4/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[5/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[6/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[7/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[8/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[9/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[10/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[11/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[12/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[13/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[14/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[15/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[16/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[17/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[18/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[19/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[20/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[21/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[22/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[23/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[24/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[25/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[26/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[27/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[28/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[29/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[30/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]
[31/200] [step: 1.0e+00] [objective: 6.1e+03] [norm(grad): 1.0e+03]

```

[illegible]

[illegible]

[illegible]



Published with MATLAB® R2020b