Question_2.m

```
clear all;
close all;
clc;

rng(0); % in Matlab
load nuclear.mat;
lambda = .001;
theta = [1 1 1];
obj = zeros(100, 1);
for iter = 1:100
    [p, n] = size(x);
    x_new = [ones(n, 1)'; x];
    a = theta * x_new;
    slack = 1 - (y .* (a));
    obj(iter) = sum(slack(slack > 0)) / n + lambda/2 *
sum(theta(2:end) .^2);
    subGrad = subGradient(x, y, theta, lambda);
    theta = theta - 100/iter * subGrad;
    % stopping criteria
    if obj(iter) < 1
        break
    end
end

plot(obj(1:iter))
```

subgradient_m

```
function [subGrad ] = subGradient( x, y, theta, lambda)
    %UNTITLED Summary of this function goes here
    %   Detailed explanation goes here
    [p, n] = size(x);
    x_new = [ones(n, 1)'; x];
    a = theta * x_new;
    slack = 1 - (y .* (a));
    b = (slack > 0);
    Ji_1 = [ 0 (lambda / n) * theta(2:end)];
    Ji_2 = [-y/n ; 1/n *(-[y;y]  .* x) + (lambda/n * (ones(n, 1) *
theta(2:end)))'];
    subGrad = (Ji_2 * b')' + sum(1-b) * Ji_1;
end
```

Question_3.m

```
clear all;
close all;
clc;

rng(0); % in Matlab
load nuclear.mat;
```

```
lambda = .001;
theta = [1 1 1];
[p, n] = size(x);
obj = zeros(n * 20000, 1);
flag = 0;
for iter = 1:100
    rng(0)
    x = x(:, randperm(n));
    x_new = [ones(n, 1)'; x];
    for i = 1:n
        a = theta * x_new;
        slack = 1 - (y .* (a));
        obj(i + (iter-1)*n) = sum(slack(slack > 0)) / n + lambda/2 *
sum(theta(2:end) .^2);
        subGrad = subGradient(x(:,i), y(:,i), theta, lambda);
        theta = theta - 100/iter * subGrad;
        % stopping criteria
        if obj(i + (iter-1)*n) < 1
            flag = 1;
            break
        end
    end
    if flag == 1
        break
    end
end

plot(obj(1:(i + (iter-1)*n)))
```