Name : Tamoghna Chattopadhyay

EE 588 : HW 6

(a)

data = csvread('Hill\_Valley\_without\_noise\_Training.data.txt',1,0);

y = data(:,end);

X = data(:,1:end-1);

(b)

X = X - repmat(mean(X.').',1,size(X,2));

X = normr(X);

(c)

The average error is 4.7170e-04. The code is:

clear all

close all

clc

% Part (a)

data = csvread('Hill\_Valley\_without\_noise\_Training.data.txt',1,0);

y = data(:,end);

X = data(:,1:end-1);

% Part (b)

X = X - repmat(mean(X.').',1,size(X,2));

X = normr(X);

% Part (c)

X\_aug = [ones(size(X,1),1),X];

trial = 100;

mean\_errs = zeros(1,trial);

for i = 1:trial

indall = randperm(size(X,1));

indtest = indall(1:106);

indtrain = indall;

indtrain(indtest) = [];

Xtrain = X\_aug(indtrain,:);

Xtest = X\_aug(indtest,:);

ytrain = y(indtrain,1);

ytest = y(indtest,1);

XtrainT = Xtrain.';

theta = zeros(size(X,2)+1,1);

thetaprev = zeros(size(X,2)+1,1);

lambda = 0.01;

T = 500;

mu = 0.01;

errs = zeros(T,1);

for t = 1:T

yvals = (1)./(1 + exp(-Xtrain\*theta));

gradf = -XtrainT\*ytrain + XtrainT\*yvals + lambda\*[0;theta(2:end)];

f = sum(-(Xtrain\*theta).\*ytrain + log(1 + exp(Xtrain\*theta)) + lambda/2\*norm(theta(2:end))^2);

thetaprev = theta;

theta = theta - mu\*gradf;

ypredreg = (1)./(1 + exp(-Xtest\*theta));

errs(t) = norm(gradf)^2/(1+abs(f));

end

ypredreg = (1)./(1 + exp(-Xtest\*theta));

ypred = ypredreg;

ypred(ypredreg>=0.5) = 1;

ypred(ypredreg<0.5) = 0;

mean\_errs(i) = sum(ypred ~= ytest)/length(ypred);

end

mean(mean\_errs)

(d)

The average number of iterations to get an accuracy of 10^-6 is 42.6900. The code is:

clear all

close all

clc

% Part (d)

data = csvread('Hill\_Valley\_without\_noise\_Training.data.txt',1,0);

y = data(:,end);

X = data(:,1:end-1);

X = X - repmat(mean(X.').',1,size(X,2));

X = normr(X);

X\_aug = [ones(size(X,1),1),X];

trial = 100;

iterstop = zeros(1,trial);

for i = 1:trial

indall = randperm(size(X,1));

indtest = indall(1:106);

indtrain = indall;

indtrain(indtest) = [];

Xtrain = X\_aug(indtrain,:);

Xtest = X\_aug(indtest,:);

ytrain = y(indtrain,1);

ytest = y(indtest,1);

XtrainT = Xtrain.';

theta = zeros(size(X,2)+1,1);

thetaprev = zeros(size(X,2)+1,1);

lambda = 0.01;

T = 10000;

mu = 0.01;

errs = zeros(T,1);

for t = 1:T

yvals = (1)./(1 + exp(-Xtrain\*theta));

gradf = -XtrainT\*ytrain + XtrainT\*yvals + lambda\*[0;theta(2:end)];

f = sum(-(Xtrain\*theta).\*ytrain + log(1 + exp(Xtrain\*theta)) + lambda/2\*norm(theta(2:end))^2);

thetaprev = theta;

theta = theta - mu\*gradf;

ypredreg = (1)./(1 + exp(-Xtest\*theta));

errs(t) = norm(gradf)^2/(1+abs(f));

if rem(t,1000)==0

disp(['t=',num2str(t), ', error = ',num2str(errs(t))]);

end

end

myind = find(errs < 1e-6);

iterstop(trial) = myind(1);

end

mean(iterstop)

(e)

(i) Using Heavy Ball Method, the best tuning parameter is 0.92.

val = 3.8900

ind = 4

ans = 0.9200

The code is:

clear all

close all

clc

% Part (e) - Heavy Ball Method

acceleratedopt = 1;

data = csvread('Hill\_Valley\_without\_noise\_Training.data.txt',1,0);

y = data(:,end);

X = data(:,1:end-1);

X = X - repmat(mean(X.').',1,size(X,2));

X = normr(X);

X\_aug = [ones(size(X,1),1),X];

trial = 100;

betavals = 0.89:0.01:0.99;

iterstop = zeros(1,trial);

iterstopbeta = zeros(size(betavals));

indbeta = 0;

for beta = betavals

indbeta = indbeta + 1;

for i = 1:trial

indall = randperm(size(X,1));

indtest = indall(1:106);

indtrain = indall;

indtrain(indtest) = [];

Xtrain = X\_aug(indtrain,:);

Xtest = X\_aug(indtest,:);

ytrain = y(indtrain,1);

ytest = y(indtest,1);

XtrainT = Xtrain.';

theta = zeros(size(X,2)+1,1);

thetaprev = zeros(size(X,2)+1,1);

lambda = 0.01;

T = 10000;

mu = 0.01;

errs = zeros(T,1);

for t = 1:T

if acceleratedopt ~= 1

yvals = (1)./(1 + exp(-Xtrain\*theta));

gradf = -XtrainT\*ytrain + XtrainT\*yvals + lambda\*[0;theta(2:end)];

f = sum(-(Xtrain\*theta).\*ytrain + log(1 + exp(Xtrain\*theta)) + lambda/2\*norm(theta(2:end))^2);

thetaprev = theta;

theta = theta - mu\*gradf;

else

z = theta + beta\*(theta - thetaprev);

yvalsz = (1)./(1 + exp(-Xtrain\*z));

gradfz = -XtrainT\*ytrain + XtrainT\*yvalsz + lambda\*[0;z(2:end)];

yvals = (1)./(1 + exp(-Xtrain\*theta));

gradf = -XtrainT\*ytrain + XtrainT\*yvals + lambda\*[0;theta(2:end)];

thetaprev = theta;

theta = z - mu\*gradf;

f = sum(-(Xtrain\*theta).\*ytrain + log(1 + exp(Xtrain\*theta)) + lambda/2\*norm(theta(2:end))^2);

end

ypredreg = (1)./(1 + exp(-Xtest\*theta));

errs(t) = norm(gradf)^2/(1+abs(f));

end

myind = find(errs < 1e-6);

iterstop(trial) = myind(1);

end

iterstopbeta(indbeta) = mean(iterstop);

end

[val,ind] = min(iterstopbeta)

betavals(ind)

(ii) Using Nesterov Method,

val = 3.3200

ind = 4

ans = 0.9200

The code is:

clear all

close all

clc

% Part (e) - Nesterov Method

acceleratedopt = 1;

data = csvread('Hill\_Valley\_without\_noise\_Training.data.txt',1,0);

y = data(:,end);

X = data(:,1:end-1);

X = X - repmat(mean(X.').',1,size(X,2));

X = normr(X);

X\_aug = [ones(size(X,1),1),X];

trial = 100;

betavals = 0.89:0.01:0.99;

iterstop = zeros(1,trial);

iterstopbeta = zeros(size(betavals));

indbeta = 0;

for beta = betavals

indbeta = indbeta + 1;

for i = 1:trial

indall = randperm(size(X,1));

indtest = indall(1:106);

indtrain = indall;

indtrain(indtest) = [];

Xtrain = X\_aug(indtrain,:);

Xtest = X\_aug(indtest,:);

ytrain = y(indtrain,1);

ytest = y(indtest,1);

XtrainT = Xtrain.';

theta = zeros(size(X,2)+1,1);

thetaprev = zeros(size(X,2)+1,1);

lambda = 0.01;

T = 10000;

mu = 0.01;

errs = zeros(T,1);

for t = 1:T

if acceleratedopt ~= 1

yvals = (1)./(1 + exp(-Xtrain\*theta));

gradf = -XtrainT\*ytrain + XtrainT\*yvals + lambda\*[0;theta(2:end)];

f = sum(-(Xtrain\*theta).\*ytrain + log(1 + exp(Xtrain\*theta)) + lambda/2\*norm(theta(2:end))^2);

thetaprev = theta;

theta = theta - mu\*gradf;

else

z = theta + beta\*(theta - thetaprev);

yvalsz = (1)./(1 + exp(-Xtrain\*z));

gradfz = -XtrainT\*ytrain + XtrainT\*yvalsz + lambda\*[0;z(2:end)];

yvals = (1)./(1 + exp(-Xtrain\*theta));

gradf = -XtrainT\*ytrain + XtrainT\*yvals + lambda\*[0;theta(2:end)];

thetaprev = theta;

theta = z - mu\*gradfz;

f = sum(-(Xtrain\*theta).\*ytrain + log(1 + exp(Xtrain\*theta)) + lambda/2\*norm(theta(2:end))^2);

end

ypredreg = (1)./(1 + exp(-Xtest\*theta));

errs(t) = norm(gradf)^2/(1+abs(f));

end

myind = find(errs < 1e-6);

iterstop(trial) = myind(1);

end

iterstopbeta(indbeta) = mean(iterstop);

end

[val,ind] = min(iterstopbeta)

betavals(ind)

(iii) To compare, the code is:

clear all

close all

clc

data = csvread('Hill\_Valley\_without\_noise\_Training.data.txt',1,0);

y = data(:,end);

X = data(:,1:end-1);

X = X - repmat(mean(X.').',1,size(X,2));

X = normr(X);

X\_aug = [ones(size(X,1),1),X];

trial = 100;

beta = 0.92;

T = 500;

errstot = zeros(T,3);

indopt = 0;

for acceleratedopt = [0,1,2]

indopt = indopt + 1;

indall = randperm(size(X,1));

indtest = indall(1:106);

indtrain = indall;

indtrain(indtest) = [];

Xtrain = X\_aug(indtrain,:);

Xtest = X\_aug(indtest,:);

ytrain = y(indtrain,1);

ytest = y(indtest,1);

XtrainT = Xtrain.';

theta = zeros(size(X,2)+1,1);

thetaprev = zeros(size(X,2)+1,1);

lambda = 0.01;

mu = 0.01;

errs = zeros(T,1);

for t = 1:T

if acceleratedopt == 0

yvals = (1)./(1 + exp(-Xtrain\*theta));

gradf = -XtrainT\*ytrain + XtrainT\*yvals + lambda\*[0;theta(2:end)];

f = sum(-(Xtrain\*theta).\*ytrain + log(1 + exp(Xtrain\*theta)) + lambda/2\*norm(theta(2:end))^2);

thetaprev = theta;

theta = theta - mu\*gradf;

elseif acceleratedopt == 1

z = theta + beta\*(theta - thetaprev);

yvalsz = (1)./(1 + exp(-Xtrain\*z));

gradfz = -XtrainT\*ytrain + XtrainT\*yvalsz + lambda\*[0;z(2:end)];

yvals = (1)./(1 + exp(-Xtrain\*theta));

gradf = -XtrainT\*ytrain + XtrainT\*yvals + lambda\*[0;theta(2:end)];

thetaprev = theta;

theta = z - mu\*gradfz;

f = sum(-(Xtrain\*theta).\*ytrain + log(1 + exp(Xtrain\*theta)) + lambda/2\*norm(theta(2:end))^2);

else

z = theta + beta\*(theta - thetaprev);

yvalsz = (1)./(1 + exp(-Xtrain\*z));

gradfz = -XtrainT\*ytrain + XtrainT\*yvalsz + lambda\*[0;z(2:end)];

yvals = (1)./(1 + exp(-Xtrain\*theta));

gradf = -XtrainT\*ytrain + XtrainT\*yvals + lambda\*[0;theta(2:end)];

thetaprev = theta;

theta = z - mu\*gradf;

f = sum(-(Xtrain\*theta).\*ytrain + log(1 + exp(Xtrain\*theta)) + lambda/2\*norm(theta(2:end))^2);

end

ypredreg = (1)./(1 + exp(-Xtest\*theta));

errs(t) = norm(gradf)^2/(1+abs(f));

end

errstot(:,indopt) = errs;

end

dlmwrite('D:\EE 588\HW 6\HW6\_Partd' ,[ (1:T)', errstot ] ,'delimiter',',');

The convergence rate of Heavy Ball method is slightly faster. As there is very little difference and we know that Heavy Ball Method may not converge, we should probably use Nesterov’s method which surely converges.