**HW 4 Yin Cao A 45106415**

1)

% ----------------------------------------------------logistic\_train.m---------------------------------------------------------------

% define function “logistic\_train”

% input: data, label, ntrain, epsilon, maxiter

% data: training data, N by (d+1) matrix, where d is the number of features, the last column of data are

% all ones

% output: weights,acc

% acc is the accuracy of the model, i.e. number of correct prediction/total number

function [weights,acc] = logistic\_train(data, label, ntrain, epsilon, maxiter)

%------training--------

[nrow,ncol]=size(data);

data=horzcat(data,ones(nrow,1));

[nrow,ncol]=size(data);

Phi=data(1:ntrain,:);

t=label(1:ntrain);

y=zeros(ntrain,1);

R=diag(ntrain);

iter=0;

w\_0=zeros(ncol,1);

if iter<=maxiter

for i=1:ntrain

y(i,1)=logsig(w\_0'\*Phi(i,:)');

end

for i=1:ntrain

R(i,i)=y(i,1)\*(1-y(i,1));

end

z=Phi\*w\_0-inv(R)\*(y-t);

w\_1=inv(Phi'\*R\*Phi)\*Phi'\*R\*z;

if mean(sqrt(sum((abs(w\_0-w\_1)).^2)))>=epsilon

w\_0=w\_1;

end

iter=iter+1;

weights=w\_0;

%--------testing--------

tdata=data(ntrain+1:nrow,:);

tlabel=label(ntrain+1:nrow);

[trow,tcol]=size(tdata);

err=zeros(trow,1);

tlabel\_p=zeros(trow,1);

for i=1:trow

tlabel\_p(i)=logsig(weights'\*tdata(i,:)');

if tlabel\_p(i)<0.5

tlabel\_p(i)=0;

else

tlabel\_p(i)=1;

end

end

for i=1:trow

if tlabel\_p(i)==tlabel(i)

err(i)=0;

else

err(i)=1;

end

end

acc=(trow-sum(err))/trow;

end

weights

acc

%---------------------------------------------main\_logistic.m---------------------------------------------------------------------

% accuracy contains the accuracies of different models by choosing different size of training data set.

ntrain=[200,500,800,1000,1500,2000];

n=length(ntrain);

accuracy=zeros(n,1);

for i=1:n

nn=ntrain(i);

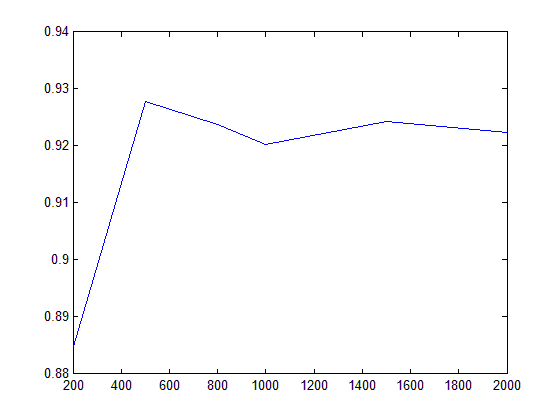
[weights,acc]=logistic\_train(spam\_data,spam\_label,nn,1e-5,1000);

accuracy(i)=acc;

end

accuracy

plot(ntrain,accuracy)



We train different models by taking different size of training data. The models and the corresponding prediction accuracy are plotted above. From the figure above, we see that the best model is trained by taking the first 500 samples, and the corresponding accuracy is about 92.78%. From this result we conclude that, larger size of training data could not guarantee a better prediction performance.

2)

%---------------------------------------------logistic\_l1\_train.m-------------------------------------------------------------------

function [w, c] = logistic\_l1\_train(data, labels, par)

% OUTPUT w is equivalent to the first d dimension of weights in logistic train

% c is the bias term, equivalent to the last dimension in weights in logistic train.

% Specify the options (use without modification).

opts.rFlag = 1; % range of par within [0, 1].

opts.tol = 1e-6; % optimization precision

opts.tFlag = 4; % termination options.

opts.maxIter = 5000; % maximum iterations

[w, c] = LogisticR(data, labels, par, opts);

end

%------------------------------------------main\_sparse\_logistic.m---------------------------------------------------------------

% auc records the AUC value of different models based on the l1 regularization parameters we pick

% nnzero records number of selected features of different models based on the l1 regularization

% parameters we pick

parameters = [1e-8, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1-1e-8];

np=length(parameters);

[trow,tcol]=size(X\_test);

accuracy=zeros(np,1);

nnzero=zeros(np,1); %nnzero is the number of non-zero elements(number of selected features) in weights

auc=zeros(np,1); %auc records the area under the roc curve corresponding to each parameters

for i=1:np

par=parameters(i);

[w,c] = logistic\_l1\_train(X\_train,y\_train,par);

nnzero(i)=nnz(w);

labelp=zeros(trow,1);

err=zeros(trow,1);

for j=1:trow

labelp(j)=w'\*X\_test(j,:)'+c;

% if labelp(j)>=0

% labelp(j)=1;

% else

% labelp(j)=-1;

% end

%

% if labelp(j)==y\_test(j)

% err(j)=0;

% else

% err(j)=1;

% end

%

end

%rescale labelp between -1 and 1

% clear max

% clear min

maxx = max(labelp);

minn = min(labelp);

% if par=1 , then we got maxx=minn, and the rescaling caught error,

% soln: change the last parameter into 1-1e-8

for k=1:trow

labelp(k)=(labelp(k)-((maxx+minn)/2))/((maxx-minn)/2);

end

%X returns the false positive rate; Y returns the true positive rate;

%AUC returns the auc value

[X,Y,~,AUC]=perfcurve(y\_test,labelp,1);

auc(i)=AUC;

% accuracy(i)=(trow-sum(err))/trow;

%

% plot(parameters,accuracy)

end

plot(parameters,nnzero)

xlabel('L1 regularization parameter'); ylabel('Number of selected features')

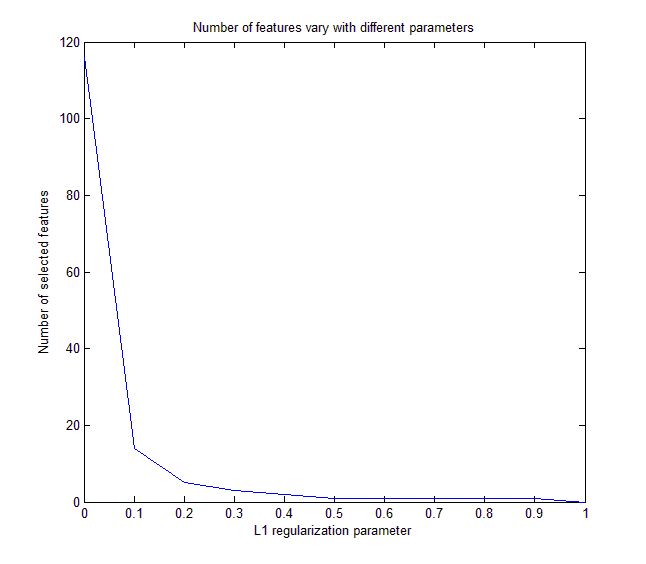
title('Number of features vary with different parameters');

figure;

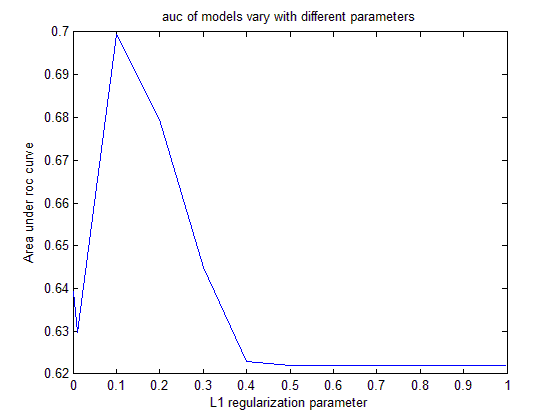
plot(parameters,auc)

xlabel('L1 regularization parameter'); ylabel('Area under roc curve')

title('auc of models vary with different parameters')



We plotted different models(by taking different l1 regularization parameter par) and their corresponding number of selected features in the above figure. From the above figure, we see that the number of features selected was increasing as the regularization increases, it was close to zero when par is large enough.



We also plotted the auc value against the corresponding parameters. From the result we can see that the maximal auc value is achieved as par=0.1. When par is large enough, the auc value stays constant 0.6220 (since the number of features selected are the same when par is large enough).