Silvia Ionescu
10-24-2016

Assignment #5

Problem 5.1a: San Francisco Crime Prediction

The hours, days of the week and police district were extracted from the dataset. These three categorical values were converted into real-valued vectors. The DayOfWeek was represented as a 7-dimentional vector with "Sunday" = [1,0,0,0,0,0,0] and a histogram of the day of the week for all the incidents contained in the file data_SFcrime_train.mat is shown in Fig.

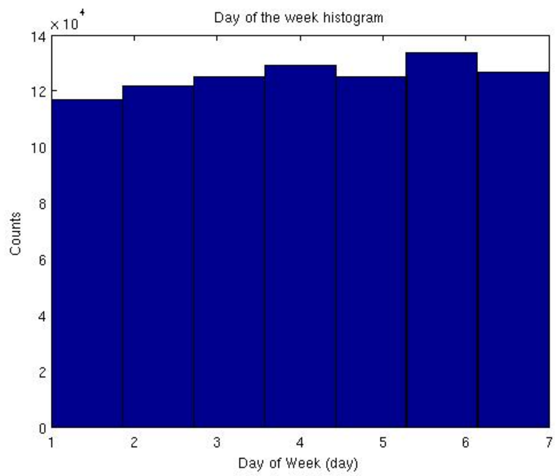


Fig.1: Day of the week histogram for all the incidents in the data_SFcrime_train dataset. Sunday is '1' and Saturday is '7'.

The Hours was represented as a 24-dimentional vector and a histogram of the for all the incidents contained in the file data_SFcrime_train.mat is shown below:

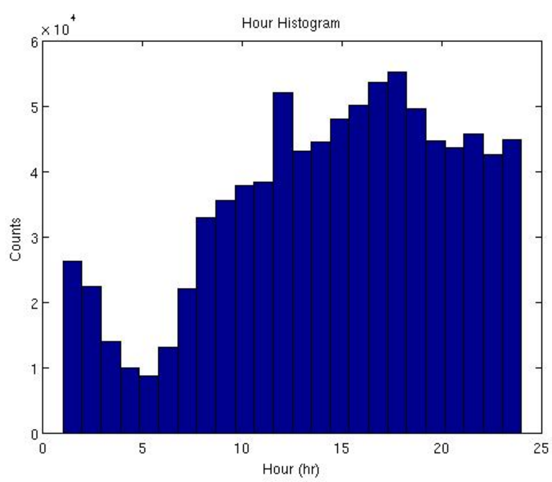


Fig.2: Hours histogram for all the incidents in the data_SFcrime_train dataset. Where 1 is 1am, and 24 is midnight.

The police district was represented as a 10-dimentional vector. The histogram is shown in Fig. 3. The hours, days, and police districts vectors were concatenated into a 41 feature vector.

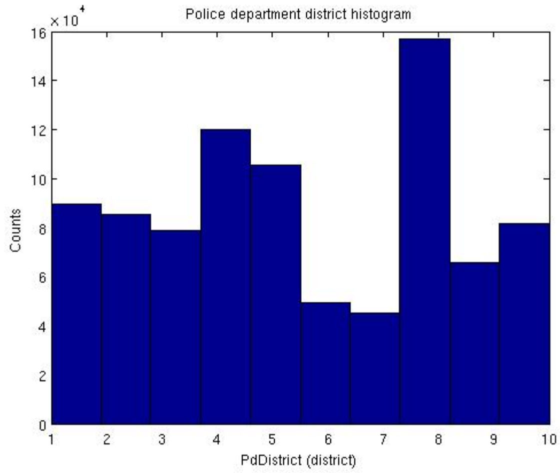


Fig.3: Police districts histogram for all the incidents in the data_SFcrime_train dataset. The districts were organized in alphabetical order, with

Most likely hour for each type of crime:

Crime	Most likely hr	Crime	Most likely hr
'ARSON'	24	'NON-CRIMINAL'	12
'ASSAULT'	24	'OTHER OFFENSES'	17
'BAD CHECKS'	12	'PORNOGRAPHY/OBSCENE MAT'	14
'BRIBERY'	17	'PROSTITUTION'	22
'BURGLARY'	17	'RECOVERED VEHICLE'	12
'DISORDERLY CONDUCT'	6	'ROBBERY'	21
'DRIVING UNDER THE INFLUENCE'	24	'RUNAWAY'	18
'DRUG/NARCOTIC'	14	'SECONDARY CODES'	12
'DRUNKENNESS'	24	'SEX OFFENSES FORCIBLE'	24
'EMBEZZLEMENT'	24	'SEX OFFENSES NON FORCIBLE'	24
'EXTORTION'	24	'STOLEN PROPERTY'	16
'FAMILY OFFENSES'	15	'SUICIDE'	18
'FORGERY/COUNTERFEITING'	24	'SUSPICIOUS OCC'	12
'FRAUD'	24	'TREA'	5
'GAMBLING'	13	'TRESPASS'	6
'KIDNAPPING'	24	'VANDALISM'	18
'LARCENY/THEFT'	18	'VEHICLE THEFT'	18
'LIQUOR LAWS'	17	'WARRANTS'	17
'LOITERING'	17	'WEAPON LAWS'	16
'MISSING PERSON'	8		

Most likely type of crime within each police district:

'BAYVIEW'	'OTHER OFFENSES'
'CENTRAL'	'LARCENY/THEFT'
'INGLESIDE'	'OTHER OFFENSES'
'MISSION'	'OTHER OFFENSES'
'NORTHERN'	'LARCENY/THEFT'
'PARK'	'LARCENY/THEFT'
'RICHMOND'	'LARCENY/THEFT'
'SOUTHERN'	'LARCENY/THEFT'
'TARAVAL'	'LARCENY/THEFT'
'TENDERLOIN'	'DRUG/NARCOTIC'

Table2: Most likely type of crime for each police district.

Part 5.3b – l_2 regularized multi-class logistic regression classifier

Implemented an gradient descent algorithm was described in the assignment handout in order to learn the parameters $\theta(w_1, ..., w_m)$ of a l_2 regularized multi-class logistic regression. I chose a fixed step size $\eta = 10^{-5}$ and all w_k's were initialized to zero (dxm zero matrix). We were given: $\lambda = 1000$ and t = 1, 2, …, 1000. The first 60% of the data samples in the data_SF_train dataset were used as a training set and the other 40% as a "test" set.

Part 5.3i

The value of the objective function over the 1000 iterations is shown below:

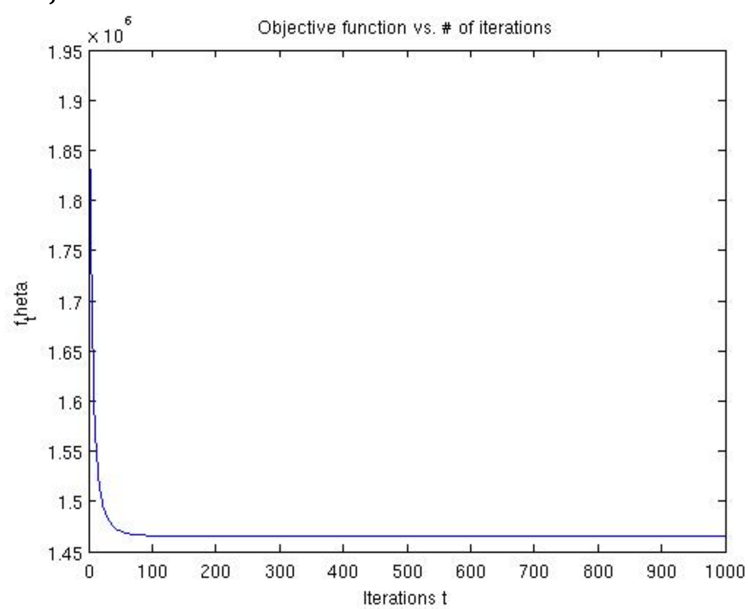


Fig.4: Objective function vs. # of iterations

The CCR and logloss ware calculated for t number of iterations and are shown below:

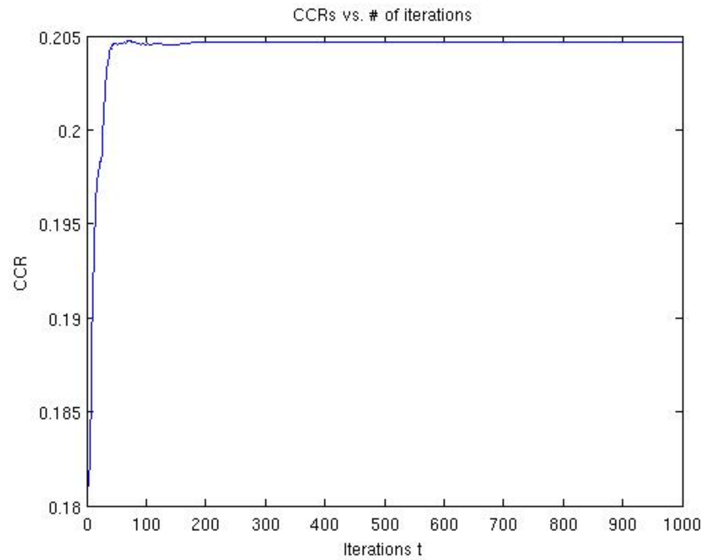


Fig. 5: CCR vs # iterations.

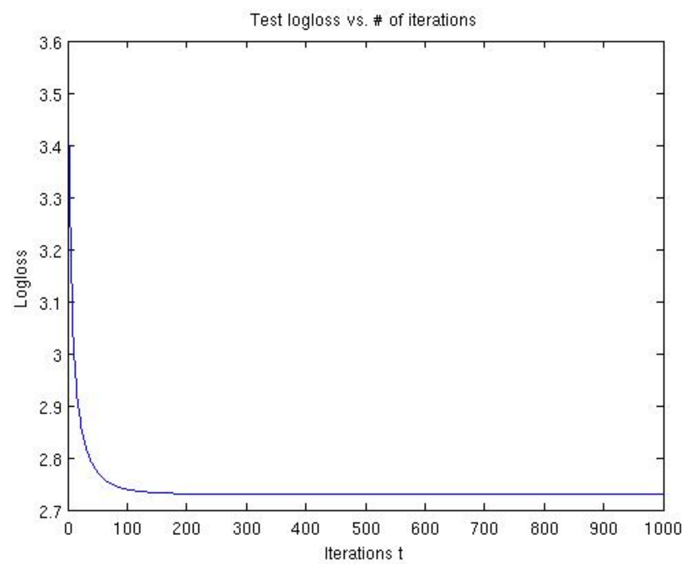


Fig. 6: Logloss vs. t number of iterations.

Silvia Ionescu
10-24-2016

Part 5.1c

For this part I ran part b for different values of lambda betweeb 10^{-5} and 10^{5}. I took 60% random samples from the training set for training and the other 40% for testing. I tried two different seeds for the random numbers. The CCR and logloss values are shown below for the two seeds. It seams that both show a lambda = 1. After that I took a range of lambda values around 1 (0.4, 0.8, 1, 4, 8). Shown below for both seeds. From this I concluded that lambda = 8;

Seed 2 – 700 iterations

Lambda	CCR	Logloss
10^-5	0.224395605021368	2.58584451865987
10^-4	0.224495257944473	2.58260635204306
10^-3	0.224495257944473	2.58167543560010
10^-2	0.224495257944473	2.58126580526538
10^-1	0.224495257944473	2.58104493707737
1	**0.224495257944473**	**2.58092091028806**
10	0.224483869038976	2.58102836857577
100	0.224267479834519	2.58931538684664
1000	0.223333589583707	2.67134671085306
10000	0.214498646143859	3.04302770377233
100k	0.202244183828324	3.54772760565007

Lambda	CCR	Logloss
0.4	0.224395605021368	2.58586757991501
0.8	0.224495257944473	2.58264721436103
4	0.224409841153241	2.58608030176961
8	0.224509494076346	2.58306148240538

Seed 3 – 700 iterations

Range	CCR	Logloss
10^-5	0.224372827210373	2.58526126078873
10^-4	0.224372827210373	2.58201726969759
10^-3	0.224372827210373	2.58109055437519
10^-2	0.224375674436747	2.58068292492764
10^-1	0.224375674436747	2.58046292333094
1	**0.224375674436747**	**2.58034136688479**
10	0.224341507720254	2.58047161207457
100	0.224273174287268	2.58889161147749
1000	0.223174144906739	2.67127021616956
10000	0.214407534899877	3.04354492040141

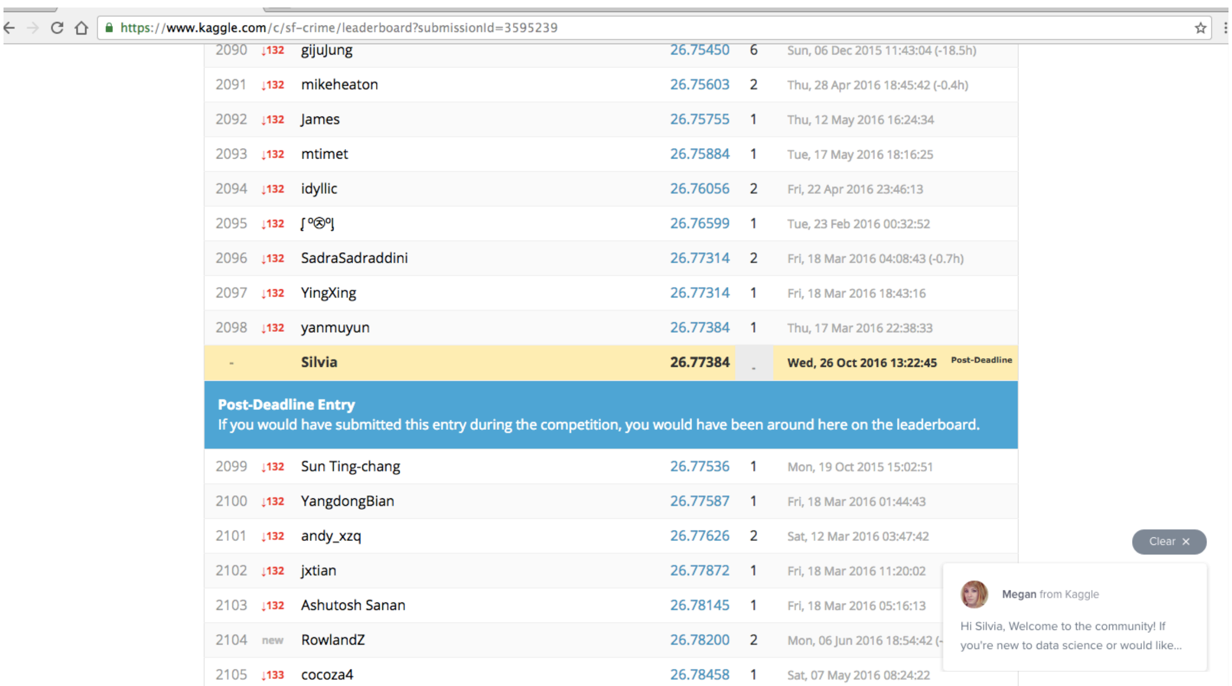
Silvia Ionescu
10-24-2016

100k	0.202358072883301	3.54778836553439

Seed 3 -700 iterations

Range	CCR	Logloss
0.4	0.224372827210373	2.58528489077001
0.8	0.224372827210373	2.58205968622982
4	0.224378521663122	2.58550267024960
8	**0.224378521663122**	**2.58248737518388**

Using lambda = 8 I calculated the labels for the test set using all samples from the training set. The screen shot for the Kaggle submission is below.



Silvia Ionescu
10-24-2016

```
% Assignment5 - Part 5.1a
% Load the train and test databases
% Extract the hour, day, and police district into three vectors
% Concatenate the three vectors into a binary vector of 41 features and
% save it.

clear; close all; clc;

Hr = 24;

%train and test datasets
train = load('data_SFcrime_train.mat');
test = load('data_SFcrime_test.mat');

train_Dates = train.Dates;
train_DayOfWeek = train.DayOfWeek;
train_PdDistrict = train.PdDistrict;
train_Category = train.Category;

% obtain unique values
train_Category_unique = unique(train_Category);
train_DaysOfWeek_unique = {'Sunday','Monday', 'Tuesday', 'Wednesday',
'Thursday', 'Friday', 'Saturday'};
train_PdDistrict_unique = unique(train_PdDistrict);

% initialize empty variables
Hour = zeros(length(train_Dates), Hr);
Day = zeros(length(train_Dates), 7);
PdDistrict = zeros(length(train_Dates), 10);
Label_crime_type = zeros(length(train_Dates),39);


for i = 1: length(train_Dates)
    % extract the hour and make a nx24 binary vector
    disp(i);
    sample_train = char(train_Dates{i});
    sample_hour = sample_train(end-4:end-3);
    hour_num(i) = str2num(sample_hour);
    if (hour_num(i) == 0)
        hour_num(i) = 24;
        Hour(i, 24) = 1;
    else
        Hour(i, hour_num(i)) = 1;
    end

    % extract the day and make a nx7 binary vector
    day_num(i) = find(strcmp(train_DayOfWeek{i},train_DaysOfWeek_unique));
    Day(i, day_num(i)) = 1;

    % extract the police distric and make a nx7 binary vector
    pd_distict_num(i) = find(strcmp(train_PdDistrict{i},
train_PdDistrict_unique));
    PdDistrict(i, pd_distict_num(i)) = 1;
```

```
    % get label
    label(i) = find(strcmp(train_Category{i},train_Category_unique));
    Label_crime_type(i,label(i)) = 1;
end

figure(1);
hist(hour_num,24);
title('Hour Histogram');
xlabel('Hour (hr)');
ylabel('Counts');

figure(2);
hist(day_num,7);
title('Day of the week histogram');
xlabel('Day of Week (day)');
ylabel('Counts');

figure(3);
hist(pd_distict_num,10);
title('Police department district histogram');
xlabel('PdDistrict (district)');
ylabel('Counts');

train_data = [Hour, Day, PdDistrict] ;

save('train_data','train_data','label');

% calculate the most likely hour for each type of crime
hour_max_per_label = zeros(1,39);
for k = 1:39
    disp('k:');
    disp(k);
    hour_per_label =  sum(Hour(find(label == k),:),1);
    find_hour = find(hour_per_label == max(hour_per_label));
    if length(find_hour) < 2
        hour_max_per_label(k) = find_hour;
    else
        hour_max_per_label(k) = find_hour(1,1);
    end

end

crime_per_PdDistrict = zeros(1,10);
for r = 1:10
   a = sum(Label_crime_type(find(pd_distict_num == r),:),1);
   crime_per_PdDistrict(r) = find(a == max(a));
end

%%%%%%%%% Test preprocessing %%%%%%%%%

test_Dates = test.Dates_test;
test_DayOfWeek = test.DayOfWeek_test;
test_PdDistrict = test.PdDistrict_test;
```

```
% obtain unique values
test_DaysOfWeek_unique = {'Sunday','Monday', 'Tuesday', 'Wednesday',
'Thursday', 'Friday', 'Saturday'};
test_PdDistrict_unique = unique(test_PdDistrict);

% initialize with zero
Hour_test = zeros(length(test_Dates), Hr);
Day_test = zeros(length(test_Dates), 7);
PdDistrict_test = zeros(length(test_Dates), 10);


for t = 1: length(test_Dates)
    % hours
    disp(t);
    sample_test = char(test_Dates{t});
    sample_hour_test = sample_test(end-7:end-6);

    hour_num_test(t) = str2num(sample_hour_test);
    if (hour_num_test(t) == 0)
        hour_num_test(t) = 24;
        Hour_test(t, 24) = 1;
    else
        Hour_test(t, hour_num_test(t)) = 1;
    end

    % day
    day_num_test(t) = find(strcmp(test_DayOfWeek{t},test_DaysOfWeek_unique));
    Day_test(t, day_num_test(t)) = 1;

    % police department
    pd_distict_num_test(t) = find(strcmp(test_PdDistrict{t},
test_PdDistrict_unique));
    PdDistrict_test(t, pd_distict_num_test(t)) = 1;

end

test_data = [Hour_test, Day_test, PdDistrict_test] ;

save('test_data','test_data');
```

Silvia Ionescu
10-24-2016

```
% Part 5.1b
% Find the Objective function over 1000 iterations
% Find the CCR over 1000 iterations
% Find the Logloss over 1000 iterations

clear; close all; clc;

train = load('train_data.mat');

train_data = train.train_data;
train_label = train.label;
train_label = train_label';
s = RandStream('mt19937ar','Seed',0);
%sample_train = randperm(s, size(train_data,1),
ceil(0.60*size(train_data,1)));
%sample_train = randperm(s, size(train_data,1));
x = train_data(1:526830,:);
y = train_label(1:526830,:);

x_test = train_data(526831:end,:);
y_test = train_label(526831:end,:);

n = 10^-5;
lambda = 100;

% Initialize the parameters w to zero
w = zeros(39, size(train_data,2));

% loop over t = 1000 iterations
penalty = zeros(39, size(x,1));

tic
for i = 1:1000
    disp(i);
    exponential = exp(w * x');
    exp_sum = sum(exponential,1);

    penalty = zeros(39, size(x,1));
    for p = 1:39
        penalty(p,(y == p)) = 1;
    end

    exp_sum_rep = repmat(exp_sum,39,1);

    grad_a = (exponential./exp_sum_rep) - penalty;
    grad_NILL = grad_a * x;

    NILL_a = sum(log(exp_sum),2);

    % calculate NILL
    for k = 1:39

        % for f(theta)
        test = penalty(k,:)*x;
```

```
        NILL_b(k) = w(k,:)*test';

        % euclidian distamce
        eclidian(k) = w(k,:)*w(k,:)';
        %eclidian(k) = norm(w(k,:));
    end

  % calculate NILL
  NILL(i) = NILL_a - sum(NILL_b);
  f_theta(i) = NILL(i) + (lambda/2)* sum(eclidian);

  gradient_f_theta = grad_NILL + lambda*w;

  % update weights
  w = w - n*gradient_f_theta;

  % calculate CCR
  label_calc = w*x_test';
  [label_max, label_predicted] =  max(label_calc,[],1);
    prob = 0;
    for c = 1:size(x_test,1)
        %label_predicted(c) = find(label_calc(:,c) == label_max(c));
        prob_prime = exp(w(y_test(c,1),:)* x_test(c,:)');

        if prob_prime < 10^-10
            prob_prime = 10^-10 ;
        end

        prob = prob + log(prob_prime);
    end

    confusion_matrix = confusionmat(y_test, label_predicted);
    CCR(i) = sum(diag(confusion_matrix))/sum(sum(confusion_matrix));

    exponential_test = exp(w * x_test');
    exp_sum_test = sum(exponential_test,1);
    exp_sum_test(find(exp_sum_test < 10^-10)) = 10^-10;

   % prob_log = log(prob);
    exp_sum_test_log = sum(log(exp_sum_test));
    logloss(i) = -(1/size(x_test,1))*(prob - exp_sum_test_log);

end

figure(1);
plot(1:1000, f_theta);
title('Objective function vs. # of iterations');
xlabel('Iterations t');
ylabel('f_theta');

figure(2);
plot(1:1000, CCR);
title('CCRs vs. # of iterations');
xlabel('Iterations t');
ylabel('CCR');
```

```
figure(3);
plot(1:1000, logloss);
title('Test logloss vs. # of iterations');
xlabel('Iterations t');
ylabel('Logloss');


% Part 5.1c
% Predict the labels for the real test dataset

clear; close all; clc;

train = load('train_data.mat');
test = load('test_data.mat');

train_data = train.train_data;
train_label = train.label;
train_label = train_label';

test_data = test.test_data;

% s = RandStream('mt19937ar','Seed',0);
% %sample_train = randperm(s, size(train_data,1),
ceil(0.60*size(train_data,1)));
% %sample_train = randperm(s, size(train_data,1));
% x = train_data(1:526830,:);
% y = train_label(1:526830,:);
%
% x_test = train_data(526831:end,:);
% y_test = train_label(526831:end,:);

n = 10^-5;
lambda = 8;

% Initialize the parameters w to zero
w = zeros(39, size(train_data,2));

% loop over t = 1000 iterations
%penalty = zeros(39, size(train_data,1));
tic
for i = 1:1000
    disp(i);
    exponential = exp(w * train_data');
    exp_sum = sum(exponential,1);

    penalty = zeros(39, size(train_data,1));

    % calculate NILL
    for k = 1:39

        penalty(k,(train_label == k)) = 1;

        % for f(theta)
        partial = penalty(k,:)*train_data;
```

```
        NILL_b(k) = w(k,:)*partial';

        % euclidian distamce
        eclidian(k) = w(k,:)*w(k,:)';
        %eclidian(k) = norm(w(k,:));
    end

  exp_sum_rep = repmat(exp_sum,39,1);

  grad_a = (exponential./exp_sum_rep) - penalty;
  grad_NILL = grad_a * train_data;

  NILL_a = sum(log(exp_sum),2);


  % calculate NILL
  NILL(i) = NILL_a - sum(NILL_b);
  f_theta(i) = NILL(i) + (lambda/2)* sum(eclidian);

  gradient_f_theta = grad_NILL + lambda*w;

  % update weights
  w = w - n*gradient_f_theta;

  % calculate CCR
  label_calc = w*test_data';
  [label_max, label_predicted] =  max(label_calc,[],1);

end

figure(1);
plot(1:1000, f_theta);
title('Objective function vs. # of iterations');
xlabel('Iterations t');
ylabel('f_theta');

train_for_labels = load('data_SFcrime_train.mat');

train_category_crime = train_for_labels.Category;

first_colum = 1:size(label_predicted,2);
first_colum = first_colum';

final_label = zeros(size(label_predicted,2),39);
for l = 1:size(label_predicted,2)
    disp(l);
    final_label(l,label(1,l)) = 1;
end


final = [first_colum, final_label];
csvwrite('test_label.csv',final);
```