First, we need to clean the data(trainX and testX).We need to ditch the

columns with less than 600 nonzeros

rescale data to range 0 to 1

```
trainX = rescale(clean_data(trainX));
testX = rescale(clean_data(testX));
```

add a column of one to the matrix

```
col1 = ones([60000,1]);
trainX = [trainX col1];
```

use binary classifier to recognize digits

```
trainY_zero = bi_clasi(trainY,0);
trainY_one = bi_clasi(trainY,1);
trainY_two = bi_clasi(trainY,2);
trainY_three = bi_clasi(trainY,3);
trainY_four = bi_clasi(trainY,4);
trainY_five = bi_clasi(trainY,5);
trainY_six = bi_clasi(trainY,6);
trainY_seven = bi_clasi(trainY,7);
trainY_eight = bi_clasi(trainY,8);
trainY_nine = bi_clasi(trainY,9);
```

Solve for least squares min{|y-Ax|}. Here A is trainX. x is one vs all model parameter. y is {-1,1} after going through binary classifiers for each digit.

```
result_zero = lsqlin(double(trainX),double(trainY_zero.'));
result_one = lsqlin(double(trainX),double(trainY_one.'));
result_two = lsqlin(double(trainX),double(trainY_two.'));
result_three = lsqlin(double(trainX),double(trainY_three.'));
result_four = lsqlin(double(trainX),double(trainY_four.'));
result_five = lsqlin(double(trainX),double(trainY_five.'));
result_six = lsqlin(double(trainX),double(trainY_six.'));
result_seven = lsqlin(double(trainX),double(trainY_seven.'));
result_eight = lsqlin(double(trainX),double(trainY_eight.'));
result_nine = lsqlin(double(trainX),double(trainY_nine.'));
```

alpha is the last element of x since I add a column to the last column of A.

```
alpha_zero = result_zero(end);
alpha_one = result_one(end);
alpha_two = result_two(end);
alpha_three = result_three(end);
alpha_four = result_four(end);
alpha_five = result_five(end);
alpha_six = result_six(end);
alpha_seven = result_seven(end);
```

```
alpha_eight = result_eight(end);
alpha_nine = result_nine(end);
```

beta is the remaining x.

```
beta_zero = result_zero(1:end-1);
beta_one = result_one(1:end-1);
beta_two = result_two(1:end-1);
beta_three = result_three(1:end-1);
beta_four = result_four(1:end-1);
beta_five = result_five(1:end-1);
beta_six = result_six(1:end-1);
beta_seven = result_seven(1:end-1);
beta_eight = result_eight(1:end-1);
beta_nine = result_nine(1:end-1);
```

turn alpha into a ten times one vector

```
alpha = [alpha_zero; alpha_one; alpha_two; alpha_three; alpha_four;
    alpha_five; alpha_six; alpha_seven; alpha_eight; alpha_nine];
alpha
```

```
alpha = 10×1
   -0.6897
   -0.5228
   -0.8951
   -0.9576
   -0.5904
   -0.6225
   -0.8582
   -0.7080
   -1.2440
   -0.9119
```

turn beta into a ten times 493 matrix

```
beta = [(beta_zero.'); (beta_one.'); (beta_two.'); (beta_three.');
    (beta_four.'); (beta_five.'); (beta_six.'); (beta_seven.');
    (beta_eight.'); (beta_nine.')];
beta
```

```
beta = 10×493
   -0.0197   -0.0130   -0.0353    0.0048   -0.1032   -0.0320   -0.0221   -0.0960 ···
   -0.0209    0.0366   -0.0710    0.0082   -0.0324   -0.0118   -0.0340   -0.0377
   -0.2377   -0.1705   -0.1524   -0.0752   -0.1391   -0.1331   -0.0864   -0.1509
    0.0709    0.0536    0.0612    0.0291    0.0605    0.0842    0.0285    0.0882
   -0.4432   -0.0773   -0.1718   -0.0561   -0.1309   -0.0699   -0.0654   -0.0766
   -0.0123   -0.0433   -0.0199   -0.0508   -0.0102   -0.0193   -0.0175   -0.0627
    0.6271    0.2562    0.3797    0.1243    0.3575    0.1623    0.1809    0.2679
   -0.0027    0.0209    0.0281    0.0306   -0.0137    0.0347   -0.0186    0.0565
    0.0671   -0.0167    0.0045    0.0044    0.0388    0.0117    0.0396    0.0449
   -0.0284   -0.0465   -0.0231   -0.0192   -0.0272   -0.0267   -0.0050   -0.0338
```

Recognizing the class with highest weight. Put them into an array.

```
%trainX(:,end)=[];
result_ova = [];
for i = 1:60000
    a = trainX(i,:);
    b = a*beta.'+alpha.';

    result_ova = [result_ova one_v_all(b)];
end
```

Calculate error rate for one vs all

```
count = 0;
for j = 1:60000
    if trainY(j) == result_ova(j)
        count = count + 1;
    end
end
ova_error = 1-(count/60000);
ova_error
```

```
ova_error = 0.1445
```

Produce confusion matrix for one vs all

```
cof_matrix = zeros(10);
for i = 1:10
    for j = 1:10
        for k = 1:60000
            if trainY(k) == i-1 && result_ova(k) == j-1
                cof_matrix(i,j) = cof_matrix(i,j)+1;
            end
        end
    end
end

sum_col = [];
for i = 1:10
    sum_col(i) = sum(cof_matrix(i,:));
end

cof_matrix = [cof_matrix sum_col.'];

sum_row = [];
for i = 1:11
    sum_row(i) = sum(cof_matrix(:,i));
end

cof_matrix = [cof_matrix; sum_row];
cof_matrix
```

```
cof_matrix = 11×11
        5669           8          21          19          25          46 ···
           2        6543          36          17          20          30
```

99	278	4757	153	116	17
38	172	174	5150	31	122
13	104	41	5	5189	52
164	94	30	448	103	3974
104	78	77	2	64	106
55	191	36	48	165	9
69	492	64	225	102	220
67	66	26	115	365	12
⋮					