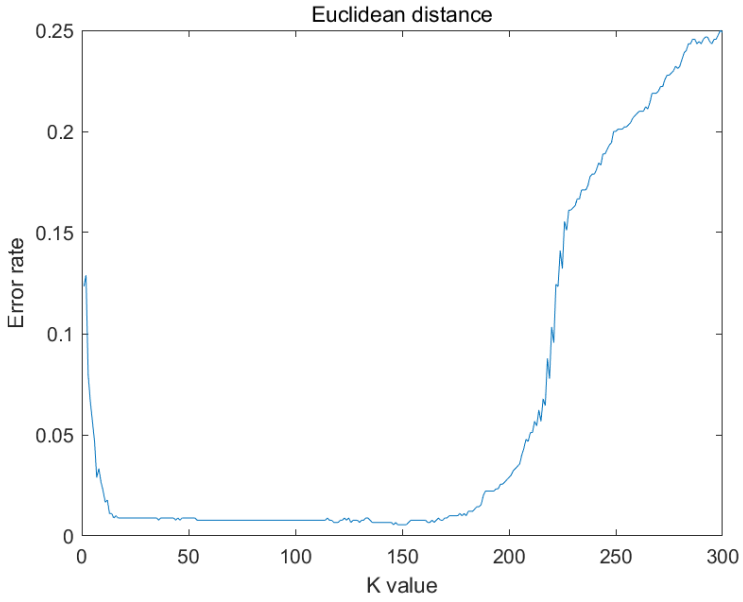
PA1 Report

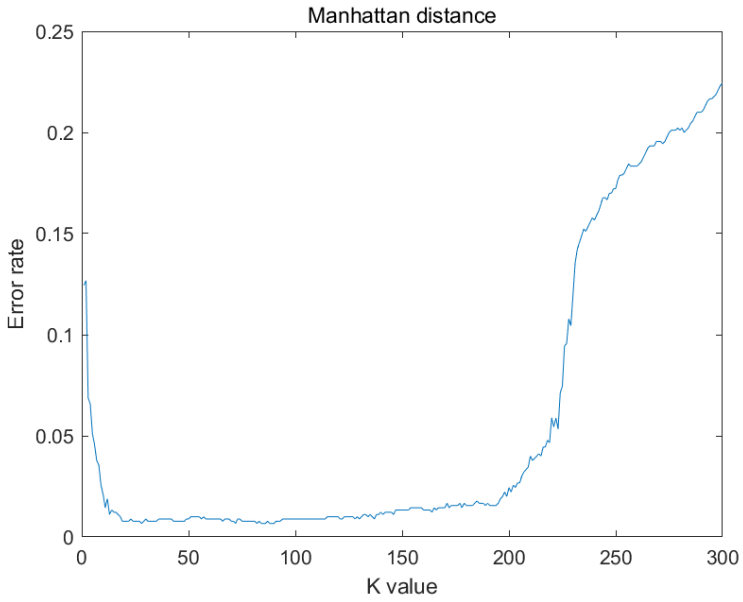
themis12@kaist.ac.kr / 010 2830 3651
20184448 Jo MinKi

1 Introduction

This report contains the summary of classic classifier matlab implementation and experiments. There are KNN classifier with Manhattan distance and Euclidean distance in various K value.

2 Result





KNN with Manhattan distance has the minimum error rate 0.0067 at k = 90 and KNN with Euclidean distance has **the minimum error rate 0.0056 at k = 150**. Since the number of training data and test data was too small, the error rate has same value at many different k value. For example, the all Error rate was 0.0056 at K = 146:152. In addition, according to this experiment, NN is worse than KNN by comparing the error rate at k = 1.

3 Implementation

KNN implementation

```
function ErrRate = classify(W1_train, W2_train, W3_train, W1_test,
W2_test, W3_test)

tr_data = [W1_train; W2_train; W3_train];
te_data = [W1_test; W2_test; W3_test];

% K-value
k = 150;

% prediction list, returns the predicted class
predict_list = [];
data_size = size(tr_data);
num_data = data_size(1);

for i = 1:num_data
    instance = te_data(i,:);
    % Euclidean distance
    diff = abs(tr_data - instance);
    euc_dist = sqrt(sum(diff.^2,2));
    % Manhattan distance
    man_dist = sum(diff,2);
    dist = euc_dist;

    % ranking of the close datapoints
    [~, ranking] = sort(dist, 'ascend');
    num_class = [0, 0, 0];
    for j = 1:k
        if ranking(j)<=300
            num_class(1) = num_class(1) + 1;
        elseif ranking(j) <= 600
            num_class(2) = num_class(2) + 1;
        else
            num_class(3) = num_class(3) + 1;
        end
    end

    % prediction based on KNN
    [~, predict] = max(num_class);
    predict_list = [predict_list, predict];
end

Error_W1 = 300 - sum(predict_list(1:300)==1);
Error_W2 = 300 - sum(predict_list(301:600)==2);
Error_W3 = 300 - sum(predict_list(601:900)==3);
Error_total = Error_W1 + Error_W2 + Error_W3;

ErrRate = Error_total/num_data;
end
```

KNN implementation

Calculate the number of closer data than Kth close data, and select the most likelihood class based on that statistic. You can choose the metric Euclidean distance or Manhattan distance by modifying dist = line.

Testing

```
result = [];
k_list = 1:300;
for i = k_list
    Err = classify_k(W1_train, W2_train, W3_train, W1_test, W2_test,
W3_test, i);
    result = [result; [i, Err]];
end

fig = plot(result(:,1), result(:,2));
title('Euclidean distance');
xlabel('K value');
ylabel('Error rate');

saveas(fig, 'Euclidean_distance.png');
[val, ind] = min(result(:,2));
```

There is classify_k function in the zip file which can be modified k value. This code calculate the error rate at 300 different k value from 1 to 300.