CIS 4526 Foundations of Machine Learning

K-Nearest Neighbor
&
Pocket/Perceptron Learning Algorithm

Khai Nguyen

CIS 4526 Foundations of Machine Learning

1. **Design**:

- k-nn: **test_knn()**
    - Loop through test_X,
        - Pick a point, compute the Euclidean distance to all train_X data points
        - Find the k nearest points
        - Perform majority voting
        - Produce a label

- Pocket algorithm:
  For the Pocket Algorithm, there are 3 functions:
    - **sign_function**(weight_vector, train_data)
      Input: weight array & a training data point
      Output: a label +1/-1
    - **test_pocket**(w, test_x)
      Input: a weight array & a test data array
      Output: a pred_y (+1/-1) array
    - **train_pocket**(train_x, train_y, num_iters)
      Input: train data array & train label array, with a number of iterations
      Output: a weight array

  The algorithm lies in **train_pocket()**:
      Initialize a weight array
      Pick a missed-classified point
      Produce a new weight
      Compute accuracy of the old & new weight
          If accuracy increases, update weight

2. **Result:**

    a) **k-NN:**

k Num_train	1	3	5	7	9
100	0.3722	0.3542	0.3272	0.3148	0.2818
1000	0.7652	0.7418	0.7112	0.6964	0.6658
2000	0.8376	0.8442	0.8204	0.7956	0.7812
5000	0.9082	0.9092	0.9066	0.8942	0.8912
10000	0.9416	0.9406	0.9374	0.933	0.936
15000	0.9544	0.9544	0.9538	0.9518	0.9492

By looking at the result table, we can see that:
+ as **k** increases from 1 to 9, accuracy decreases. The reason is as we get more neighbors, we are more likely to get more neighbor of the other classes, hence gives lower accuracy

CIS 4526 Foundations of Machine Learning

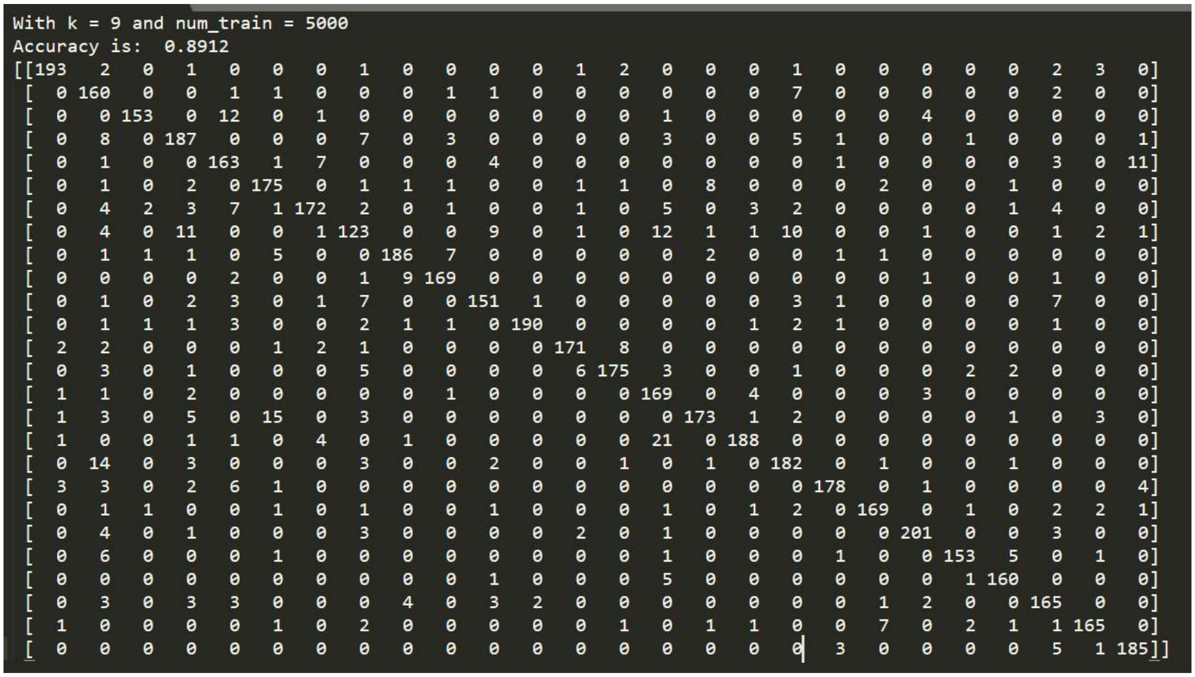
+ as **num_train** increases from 100 to 15000, we definitely have increased running time, and as observed, accuracy increases. The reason would be when num_train is lower, the neighbors are further away from a current data point, which result in higher chance of misclassifying.

➢ **Confusion matrix**:

By using the confusion matrix, we can see the number of points, for example “A”, classified as “A”, which is 193, and the number of points “A” wrongly classified as “B”, “C”,…

If we have a smaller number of features that can produce a 2x2 matrix, we will be able to tell the number of points that is True Positive, True Negative, False Positive, and False Negative. Hence, calculating Recall and Precision is possible.

For k = 9 and num_train = 5000



```
With k = 9 and num_train = 5000
Accuracy is:  0.8912
[[193   2   0   1   0   0   0   1   0   0   0   0   1   2   0   0   0   1   0   0   0   0   0   2   3   0]
 [  0 160   0   0   1   1   0   0   0   1   1   0   0   0   0   0   0   7   0   0   0   0   0   2   0   0]
 [  0   0 153   0  12   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   4   0   0   0   0   0]
 [  0   8   0 187   0   0   0   7   0   3   0   0   0   0   3   0   0   5   1   0   0   1   0   0   0   1]
 [  0   1   0   0 163   1   7   0   0   0   4   0   0   0   0   0   0   0   1   0   0   0   0   3   0  11]
 [  0   1   0   2   0 175   0   1   1   1   0   0   1   1   0   8   0   0   0   2   0   0   1   0   0   0]
 [  0   4   2   3   7   1 172   2   0   1   0   0   1   0   5   0   3   2   0   0   0   0   1   4   0   0]
 [  0   4   0  11   0   0   1 123   0   0   9   0   1   0  12   1   1  10   0   0   1   0   0   1   2   1]
 [  0   1   1   1   0   5   0   0 186   7   0   0   0   0   0   2   0   0   1   1   0   0   0   0   0   0]
 [  0   0   0   0   2   0   0   1   9 169   0   0   0   0   0   0   0   0   0   0   1   0   0   1   0   0]
 [  0   1   0   2   3   0   1   7   0   0 151   1   0   0   0   0   0   3   1   0   0   0   0   7   0   0]
 [  0   1   1   1   3   0   0   2   1   1   0 190   0   0   0   0   1   2   1   0   0   0   0   1   0   0]
 [  2   2   0   0   0   1   2   1   0   0   0   0 171   8   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   3   0   1   0   0   0   5   0   0   0   0   6 175   3   0   0   1   0   0   0   2   2   0   0   0]
 [  1   1   0   2   0   0   0   0   0   1   0   0   0   0 169   0   4   0   0   0   3   0   0   0   0   0]
 [  1   3   0   5   0  15   0   3   0   0   0   0   0   0   0 173   1   2   0   0   0   0   1   0   3   0]
 [  1   0   0   1   1   0   4   0   1   0   0   0   0   0  21   0 188   0   0   0   0   0   0   0   0   0]
 [  0  14   0   3   0   0   0   3   0   0   2   0   0   1   0   1   0 182   0   1   0   0   1   0   0   0]
 [  3   3   0   2   6   1   0   0   0   0   0   0   0   0   0   0   0   0 178   0   1   0   0   0   0   4]
 [  0   1   1   0   0   1   0   1   0   0   1   0   0   0   1   0   1   2   0 169   0   1   0   2   2   1]
 [  0   4   0   1   0   0   0   3   0   0   0   0   2   0   1   0   0   0   0   0 201   0   0   3   0   0]
 [  0   6   0   0   0   1   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0 153   5   0   1   0]
 [  0   0   0   0   0   0   0   0   0   0   1   0   0   0   5   0   0   0   0   0   0   1 160   0   0   0]
 [  0   3   0   3   3   0   0   0   4   0   3   2   0   0   0   0   0   0   0   1   2   0   0 165   0   0]
 [  1   0   0   0   0   1   0   2   0   0   0   0   0   1   0   1   1   0   0   7   0   2   1   1 165   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0   0   0   0   5   1 185]]
```

b) Pocket Learning Algorithm:

Num_train = 100	Num_train = 1000
Letter 0 has accuracy: 0.9588	Letter 0 has accuracy: 0.9786
Letter 1 has accuracy: 0.9654	Letter 1 has accuracy: 0.9352
Letter 2 has accuracy: 0.9658	Letter 2 has accuracy: 0.9658
Letter 3 has accuracy: 0.9568	Letter 3 has accuracy: 0.9568
Letter 4 has accuracy: 0.9618	Letter 4 has accuracy: 0.9618
Letter 5 has accuracy: 0.9612	Letter 5 has accuracy: 0.9612
Letter 6 has accuracy: 0.9584	Letter 6 has accuracy: 0.9584
Letter 7 has accuracy: 0.9644	Letter 7 has accuracy: 0.9644
Letter 8 has accuracy: 0.959	Letter 8 has accuracy: 0.959
Letter 9 has accuracy: 0.9634	Letter 9 has accuracy: 0.9634
Letter 10 has accuracy: 0.9646	Letter 10 has accuracy: 0.9646

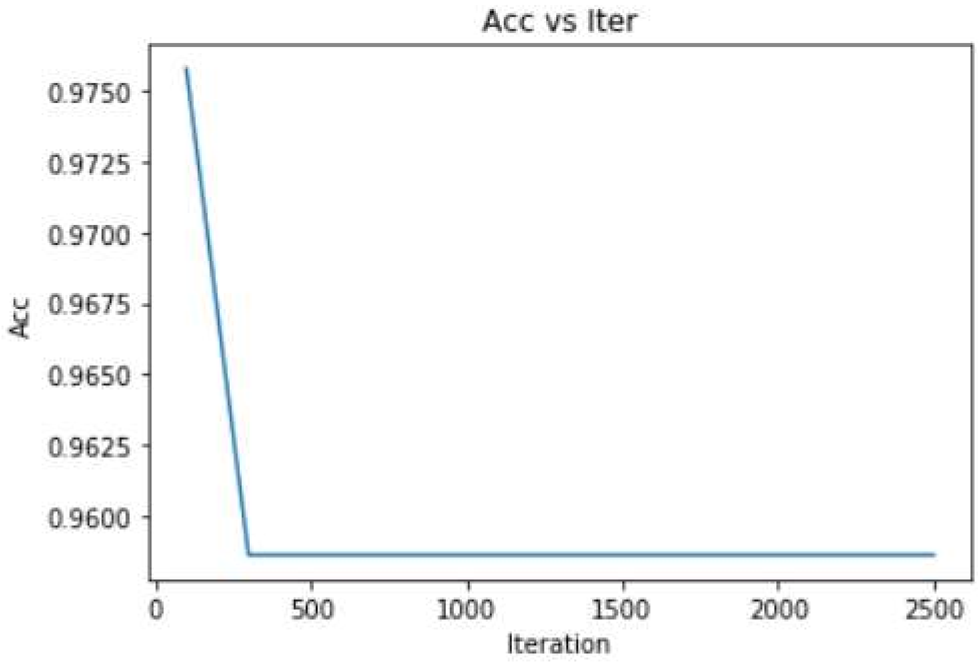
Letter 11 has accuracy: 0.959 Letter 12 has accuracy: 0.9626 Letter 13 has accuracy: 0.9604 Letter 14 has accuracy: 0.9638 Letter 15 has accuracy: 0.9586 Letter 16 has accuracy: 0.9566 Letter 17 has accuracy: 0.9584 Letter 18 has accuracy: 0.9604 Letter 19 has accuracy: 0.9632 Letter 20 has accuracy: 0.957 Letter 21 has accuracy: 0.9664 Letter 22 has accuracy: 0.9666 Letter 23 has accuracy: 0.9628 Letter 24 has accuracy: 0.9634 Letter 25 has accuracy: 0.9612	Letter 11 has accuracy: 0.959 Letter 12 has accuracy: 0.9626 Letter 13 has accuracy: 0.9604 Letter 14 has accuracy: 0.9638 Letter 15 has accuracy: 0.9586 Letter 16 has accuracy: 0.9566 Letter 17 has accuracy: 0.9584 Letter 18 has accuracy: 0.9604 Letter 19 has accuracy: 0.9632 Letter 20 has accuracy: 0.957 Letter 21 has accuracy: 0.9664 Letter 22 has accuracy: 0.9666 Letter 23 has accuracy: 0.9628 Letter 24 has accuracy: 0.9634 Letter 25 has accuracy: 0.9612
Num_train = 2000 Letter 0 has accuracy: 0.9884 Letter 1 has accuracy: 0.931 Letter 2 has accuracy: 0.9658 Letter 3 has accuracy: 0.9568 Letter 4 has accuracy: 0.9618 Letter 5 has accuracy: 0.9612 Letter 6 has accuracy: 0.9584 Letter 7 has accuracy: 0.9644 Letter 8 has accuracy: 0.959 Letter 9 has accuracy: 0.9634 Letter 10 has accuracy: 0.9646 Letter 11 has accuracy: 0.959 Letter 12 has accuracy: 0.9626 Letter 13 has accuracy: 0.9604 Letter 14 has accuracy: 0.9638 Letter 15 has accuracy: 0.9586 Letter 16 has accuracy: 0.9566 Letter 17 has accuracy: 0.9584 Letter 18 has accuracy: 0.9604 Letter 19 has accuracy: 0.9632 Letter 20 has accuracy: 0.957 Letter 21 has accuracy: 0.9664 Letter 22 has accuracy: 0.9666 Letter 23 has accuracy: 0.9628 Letter 24 has accuracy: 0.9634 Letter 25 has accuracy: 0.9612	Num_train = 5000 Letter 0 has accuracy: 0.9728 Letter 1 has accuracy: 0.9514 Letter 2 has accuracy: 0.9658 Letter 3 has accuracy: 0.9568 Letter 4 has accuracy: 0.9618 Letter 5 has accuracy: 0.9612 Letter 6 has accuracy: 0.9584 Letter 7 has accuracy: 0.9644 Letter 8 has accuracy: 0.959 Letter 9 has accuracy: 0.9634 Letter 10 has accuracy: 0.9646 Letter 11 has accuracy: 0.959 Letter 12 has accuracy: 0.9626 Letter 13 has accuracy: 0.9604 Letter 14 has accuracy: 0.9638 Letter 15 has accuracy: 0.9586 Letter 16 has accuracy: 0.9566 Letter 17 has accuracy: 0.9584 Letter 18 has accuracy: 0.9604 Letter 19 has accuracy: 0.9632 Letter 20 has accuracy: 0.957 Letter 21 has accuracy: 0.9664 Letter 22 has accuracy: 0.9666 Letter 23 has accuracy: 0.9628 Letter 24 has accuracy: 0.9634 Letter 25 has accuracy: 0.9612
Num_train = 10000 Letter 0 has accuracy: 0.9902 Letter 1 has accuracy: 0.9308 Letter 2 has accuracy: 0.9658 Letter 3 has accuracy: 0.9568 Letter 4 has accuracy: 0.9618 Letter 5 has accuracy: 0.9612 Letter 6 has accuracy: 0.9584 Letter 7 has accuracy: 0.9644 Letter 8 has accuracy: 0.959	Num_train = 15000 Letter 0 has accuracy: 0.9882 Letter 1 has accuracy: 0.9252 Letter 2 has accuracy: 0.9658 Letter 3 has accuracy: 0.9568 Letter 4 has accuracy: 0.9618 Letter 5 has accuracy: 0.9612 Letter 6 has accuracy: 0.9584 Letter 7 has accuracy: 0.9644 Letter 8 has accuracy: 0.959

Letter 9 has accuracy: 0.9634	Letter 9 has accuracy: 0.9634
Letter 10 has accuracy: 0.9646	Letter 10 has accuracy: 0.9646
Letter 11 has accuracy: 0.959	Letter 11 has accuracy: 0.959
Letter 12 has accuracy: 0.9626	Letter 12 has accuracy: 0.9626
Letter 13 has accuracy: 0.9604	Letter 13 has accuracy: 0.9604
Letter 14 has accuracy: 0.9638	Letter 14 has accuracy: 0.9638
Letter 15 has accuracy: 0.9586	Letter 15 has accuracy: 0.9586
Letter 16 has accuracy: 0.9566	Letter 16 has accuracy: 0.9566
Letter 17 has accuracy: 0.9584	Letter 17 has accuracy: 0.9584
Letter 18 has accuracy: 0.9604	Letter 18 has accuracy: 0.9604
Letter 19 has accuracy: 0.9632	Letter 19 has accuracy: 0.9632
Letter 20 has accuracy: 0.957	Letter 20 has accuracy: 0.957
Letter 21 has accuracy: 0.9664	Letter 21 has accuracy: 0.9664
Letter 22 has accuracy: 0.9666	Letter 22 has accuracy: 0.9666
Letter 23 has accuracy: 0.9628	Letter 23 has accuracy: 0.9628
Letter 24 has accuracy: 0.9634	Letter 24 has accuracy: 0.9634
Letter 25 has accuracy: 0.9612	Letter 25 has accuracy: 0.9612

Discussion:

Looking at the results for the pocket algorithm, there are “letters” that have the same accuracy when num_train increases.

In addition, by plotting the curve between Accuracy vs. Number of Iterations, we can see that after 500 iterations, the value of accuracy stabilizes. Therefore, we can choose any number of iterations larger than 500 but for the purpose of not creating extra computation time, we can pick **500** as the number of iterations.


Acc vs Iter
Acc
Iteration