Homework 10
CSCI4100

Han Hai
Rin:661534083
haih2@rpi.edu

November 19th 2018

1. **Exercise 6.1**

   (a) Give two vectors with very high cosine similarity but very low Euclidean distance similarity. Similarly, give two vectors with very low cosine similarity but very high Euclidean distance similarity.

   Case 1 : high cosine similarity but very low Euclidean
   Let $x = [1, 1], x' = [10, 10]$
   $d(x, x') = 12.7$
   long distance→ low similarity
   $CosSim(x, x') = 1 \rightarrow$ high similarity
   Case 2: very low cosine similarity but very high Euclidean distance similarity
   Let $x = [0, 1], x' = [0, -1]$
   $d(x, x') = 4$ short distance → high similarity
   $CosSim(x, x') = -1 \rightarrow$ low similarity

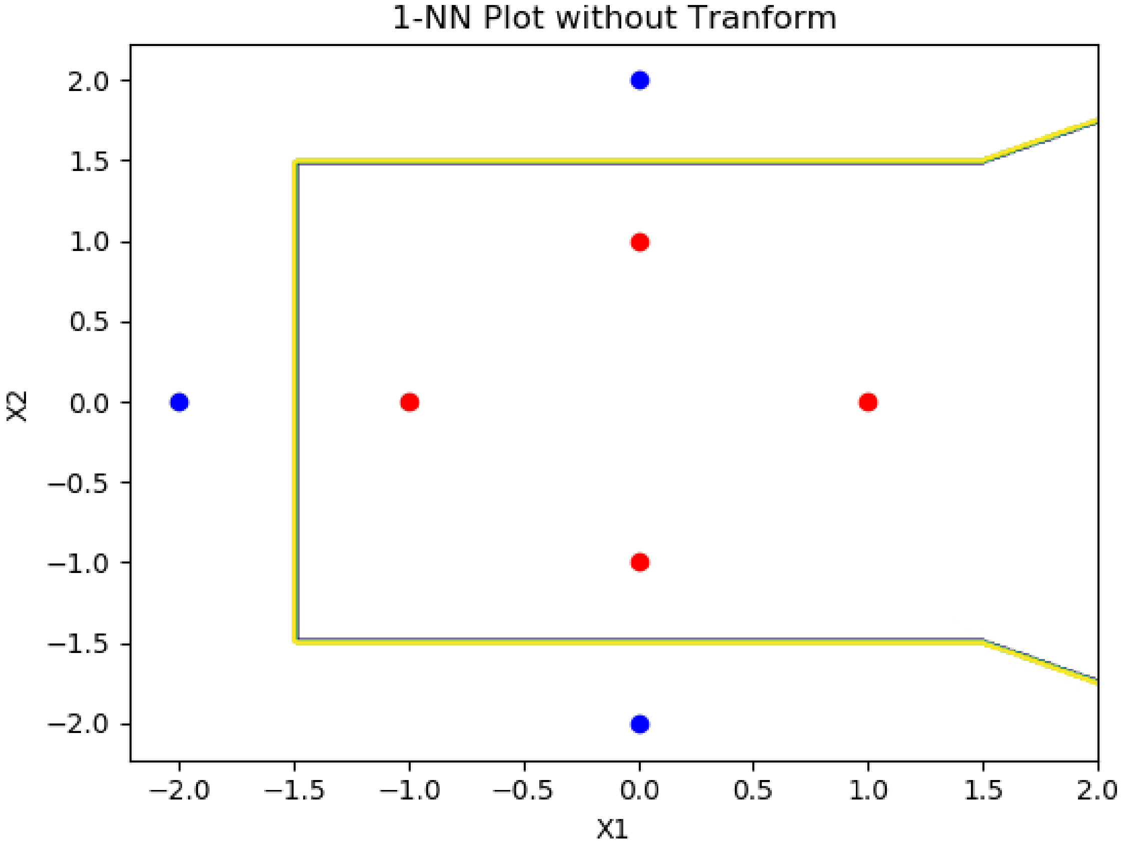
   (b) If the origin of the coordinate system changes, which measure of similarity changes? How will this affect your choice of features?
   Changing origin will change the euclidean distance between two vectors, thus euclidean similarity will be changed. And their relative angle will not be changed, so CosSim stays the same. In this way, cossim is superior when choosing features.

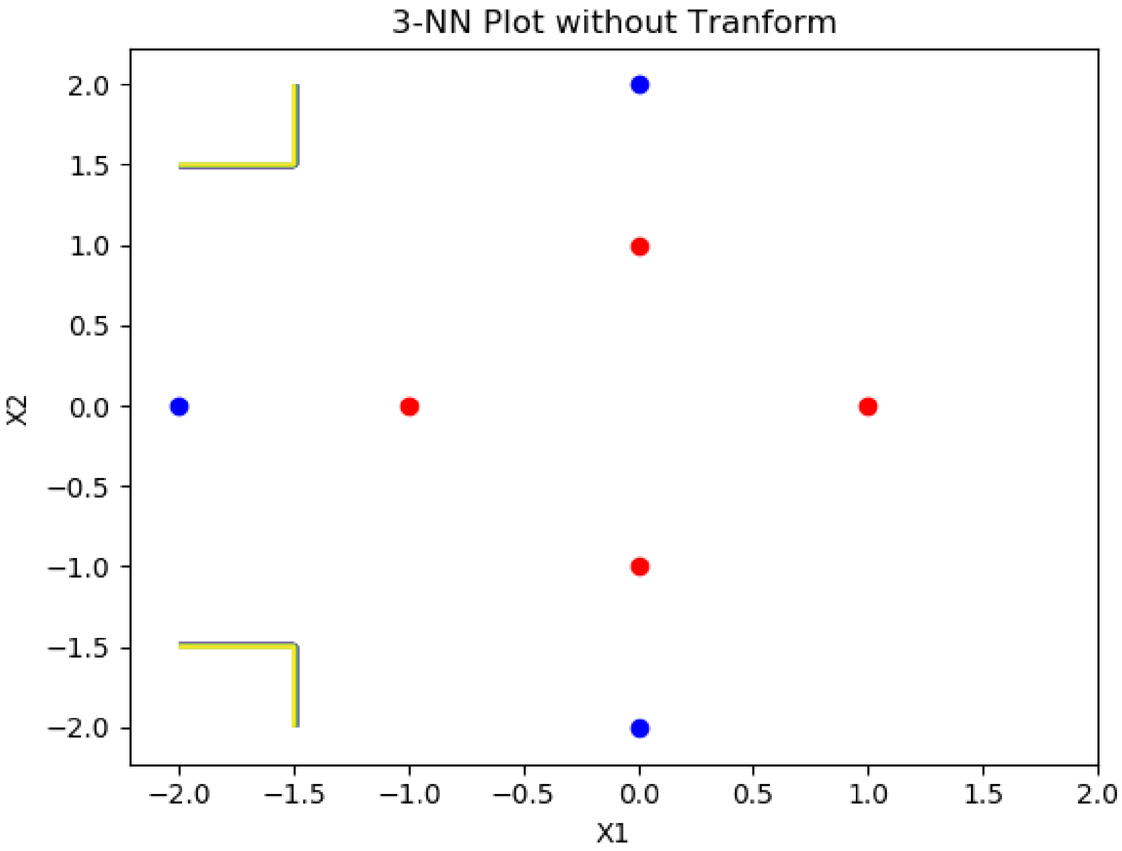
2. **Exercise 6.2** Let
   $f(x) = +1$ if $\pi(x) \geq \frac{1}{2}$
   $= -1$ otherwise
   Show that the probability of error on a test point x is $e(f(x)) = P[f(x) \neq y] = min(\pi(x), 1 - \pi(x))$ and $e(f(x)) \leq e(h(x))$ for any other hypothesis h (deterministic or not)

   When $\pi(x) \geq \frac{1}{2}$ f(x)=1 e(f(x))=$1 - \pi(x)$, since $\pi(x) > 1 - \pi(x)$ we have $e(f(x)) = min(\pi(x), 1 - \pi(x))$
   When $\pi(x) < \frac{1}{2}$, f(x)=-1, e(f(x))=$\pi(x)$,since $\pi(x) < 1 - \pi(x)$we have the same expression.
   Its error is smaller than other hypothesis because it is the minimum error possible on a test point. It sufficiently chooses the classification that has larger possibility due to x as evidence, its mistake only contain stochastic noise.

3. **Problem 6.1**

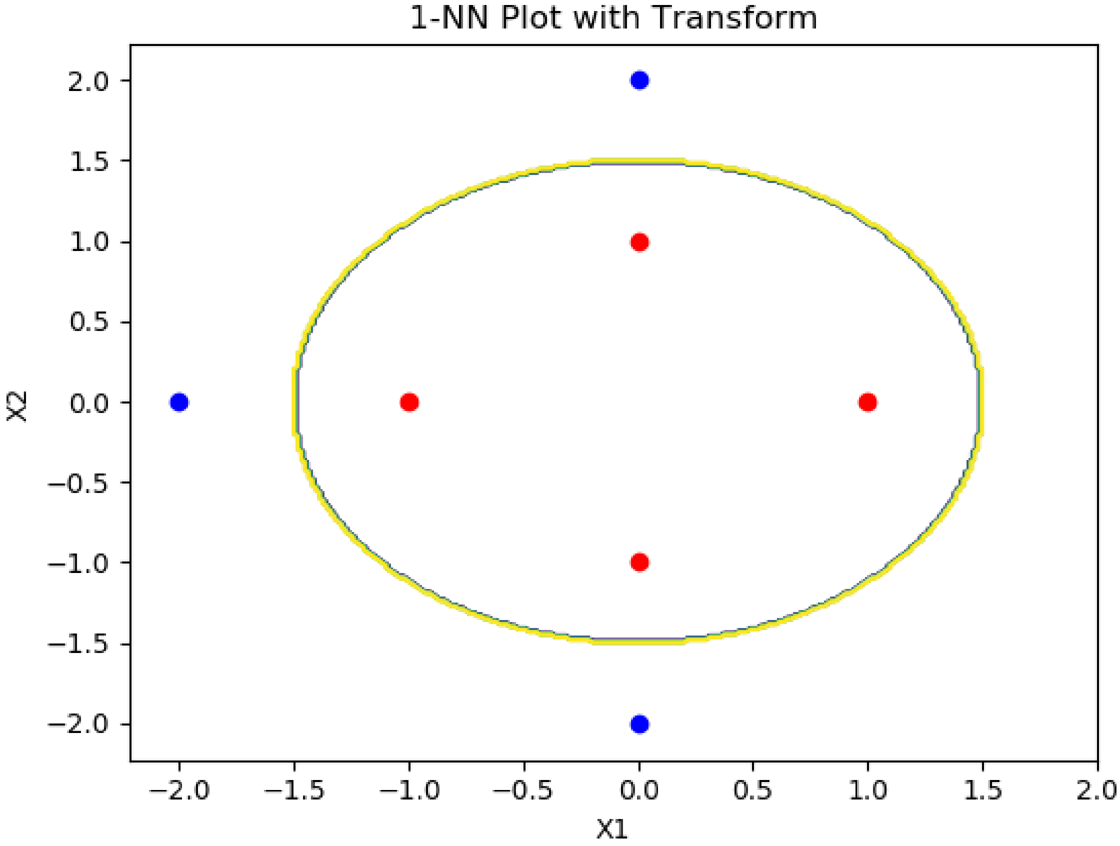
(a) Show the decision regions for the 1-NN and 3-NN rules.
note: blue is +1, red is -1. enclosed region is classified as -1

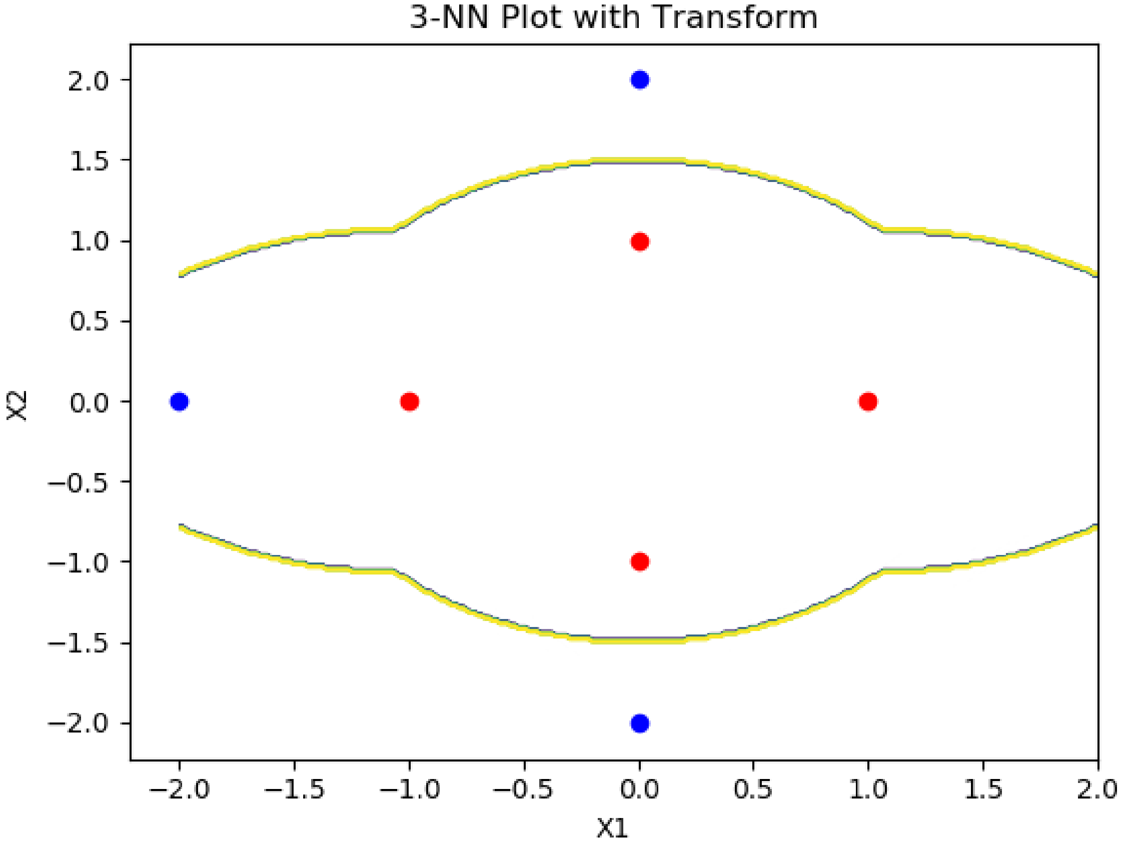




note: blue is +1, red is -1. large region is classified as -1, two corners are +1

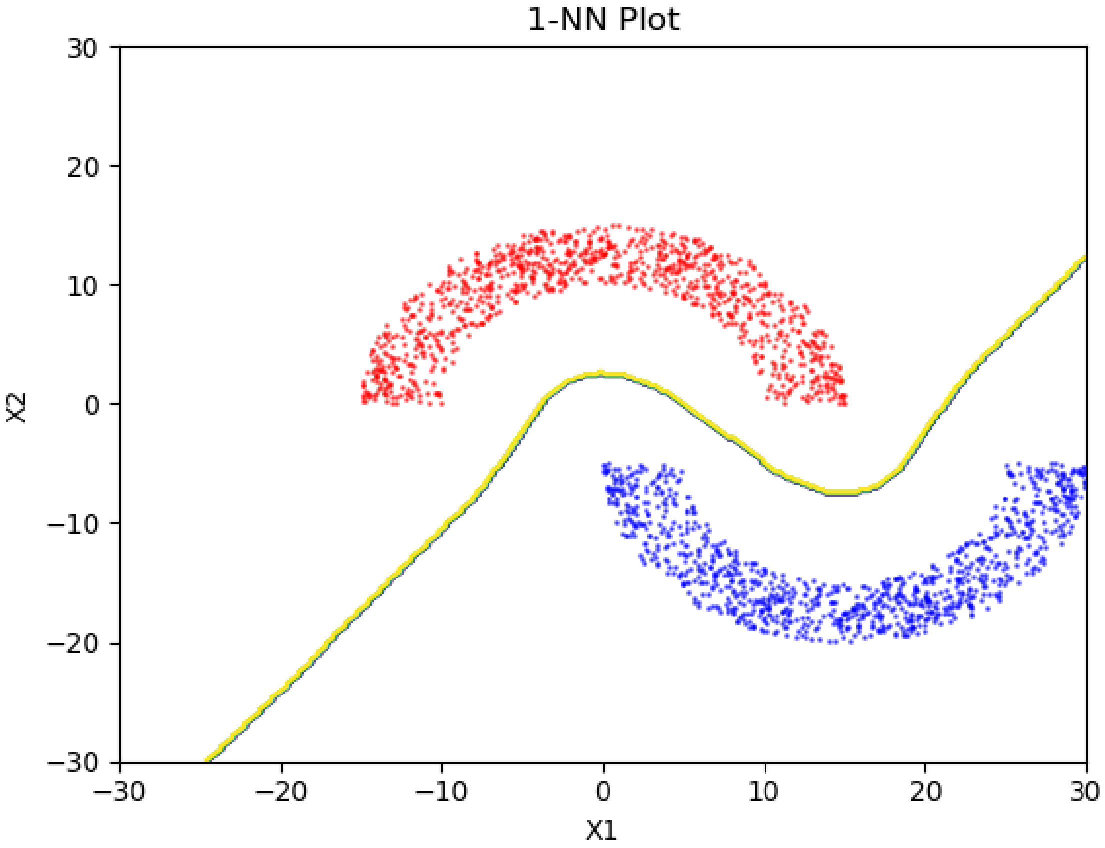
(b) nonlinear transform
note: blue is +1, red is -1. enclosed region inside the cricle is classified as -1,+1 otherwise



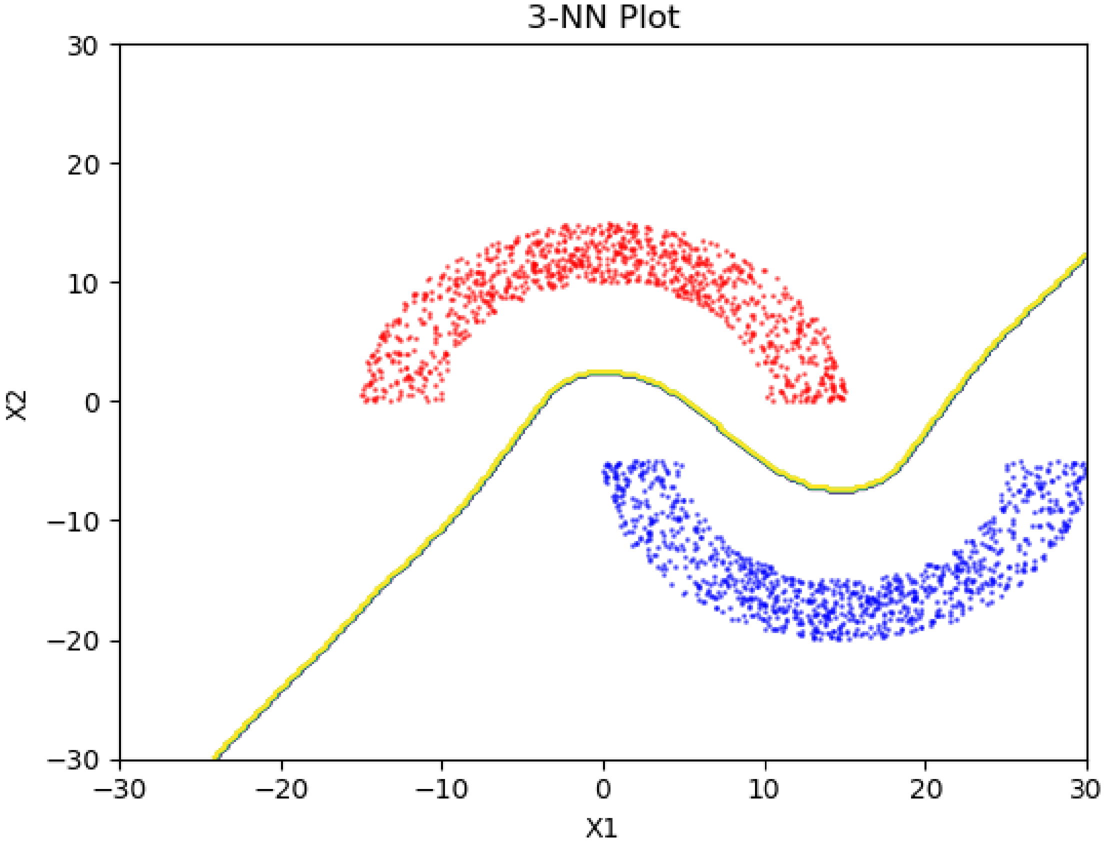


note: blue is +1, red is -1. enclosed region inside the pedal is classified as -1, +1 otherwise

4. **Problem 6.4** For the double semi-circle problem in Problem 3.1, plot the decision regions for the 1-NN and 3-NN rules.
   note: -1 (red) upward, +1(blue) downward



note: -1 (red) upward, +1(blue) downward


3-NN Plot
30
20
10
0
−10
−20
−30
X2
−30
−20
−10
0
10
20
30
X1

5. **Problem 6.16**

   (a) Generate a data set of 10,000 data points uniformly in the unit square to test the performance of the branch and bound method:
   (i) Construct a 10-partition for the data using the simple greedy heuristic described in the text.
   (ii) Generate 10,000 random query points and compare the running time of obtaining the nearest neighbor using the partition with branch and bound versus the brute force approach which does not use the partition.

   Solution: The brute force algirthom takes 541 seconds, while branch and bound search takes 86.4 seconds

   (b) Repeat (a) but instead generate the data from a mixture of 10 gaussians with centers randomly distributed in [0, 1]
   Solution: Here, I used Gaussian clusters instead, and the brute force took 575 seconds, while branch and bound took 76 seconds
   and identical covariances for each bump equal to σI where σ= 0.1.

   (c) Explain your observations.
   Solution: In part a, we can see that branch and bound is much faster than brute force. It is because branch and bound is able to skip data that are not important. In part b, brute force almost stays the same in terms of performance, while b and b is even faster, it's because gaussian cluster is better than randomly generate data.

   (d) Does your decision to use the branch and bound technique depend on how many test points you will need to evaluate?
   Solution: No, branch and bound seems to be better than brute force reagrdless of the size of the data. It is because its ability to skip data points is always useful