ECE 662 — mini project 2

Name: Wennan Chang
email: wnchang@iu.edu

Due Date: April. 6
Assignment: Number 2

Group information

The result of this mini project 1 based on the group discussion. Thanks to all my team member, Tina Chen and Bing He. They contributed a lot of idea and the nice discussion to help me understand the fundamental problem.

1 Task 1

Data generation method: Consider N-dimensional feature vectors coming from C classes. Assume that the distribution of the feature vectors for the two classes are (known) normal distributions (with same prior).

Study Parzen-window method:
- C=2, 80 samples for each class (40 for training, 40 for testing): Evaluate and plot error rate(@testing data) for varying Parzen window sizes (consider at least 4 different dimension : N)
- C=2, 1000 samples for each class (500 for training, 500 for testing): Evaluate and plot error rate (@testing data) for varying Parzen window sizes (consider at least 4 different dimensions: N)
- C=5, N=2, 80 samples for each class (40 for training, 40 for testing): plot error rate (@testing data) for varying Parzen window sizes to analyze the best window size.
- Analyze how number of training samples (e.g. from 10 to 10k) impact the error rate (@testing data), with different dimension: N. You can choose other parameters based on your own need.

Data simulation:

Benchmark datasets were generated based on the Gaussian distribution. If the class number is one, data points sampled from the univariate normal distribution. In addition, data points sampled from the multivariate normal distribution if the class number is greater than one. Our program provide the function to simulate all data points based on the random mean value and variance (or covariance) value. However, below experiments fixed all the mean value as (e.g. $\mu_1 = 1, \mu_2 = 3$) and covariance value (e.g. $\sigma = 1$ or $\Sigma = I$). In this way, we can compare the classification error rate fairly.

Specifically, the task 1 require us performs several experiments. For each scenario, we are supposed to explorer the classification error rate for at least 4 dimensions. Thus, we made the script (data.py) in order to generate and store all the datasets in the neat way.

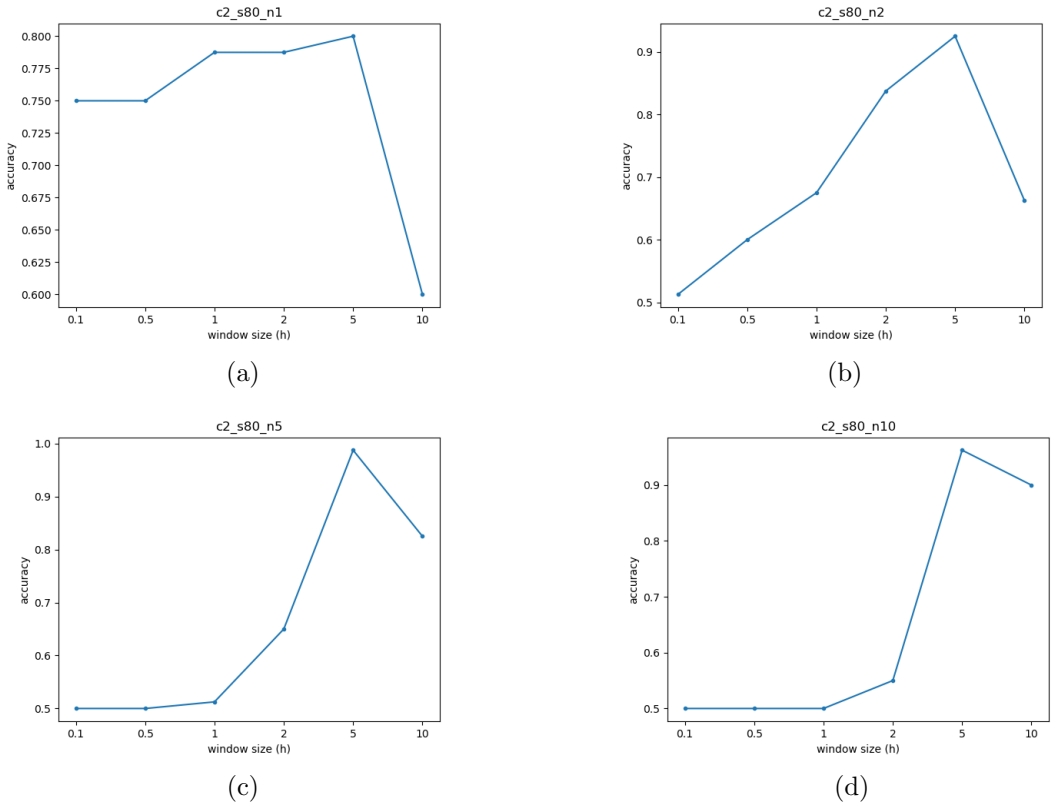
The dimensions in the task 1 were setting as N=1,2,5,10. For the last question about how of training samples impact the error rate, we use the 150,300,500,1200 training number to analysis the result.

Parzen-window implementation:

The Parzen-window method, also called Kernel density estimation, is a non-parametric way to estimate the probability density function of a random variable. The method does NOT require any knowledge or assumption about the underlying distribution. The kernel density estimator is

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} \phi(\frac{x - x_i}{h})$$

In high dimension, the h is the unit length of the hypercube. And, the ϕ is the kernel function to judge whether the current data point is in the hypercube of unit length h. Given window size h, we can count the number of data points in this hypercube in the center of the each sample. Thus, we can then get the probability for each training sample. Those probability within each window size contribution to the final density for each group. In our experiments, the parzen window size were 0.1, 0.5, 1, 2, 5, 10 respectively.



(a) (b) (c) (d)

Figure 1: C=2,80 sample with 4 different dimension. (a) is 1 dimension, (b) is 2 dimension, (c) is 5 dimension, (d) is 10 dimension.

Classification:

Based on the Bayesian Decision Rule, the posterior probability could be calculated based on the likelihood and the prior. Since the prior is same for each class, the posterior would depend on the likelihood, which is the Parzen window method estimated density function.

In the figure 1, we can find the optimal window size is close to 5. If the window size is two small, the density function would be impact by some noise/spike points which cause the classification accuracy decreased. On the other side, the density function would be too smooth to capture the shape of the true density function which also cause the classification accuracy decreased.

In addition, with the dimension increasing, the hypercube would become more complex and loose. If the window size is small, the number of points which inside the hypercube would be small thus the density function is not accurate anymore.

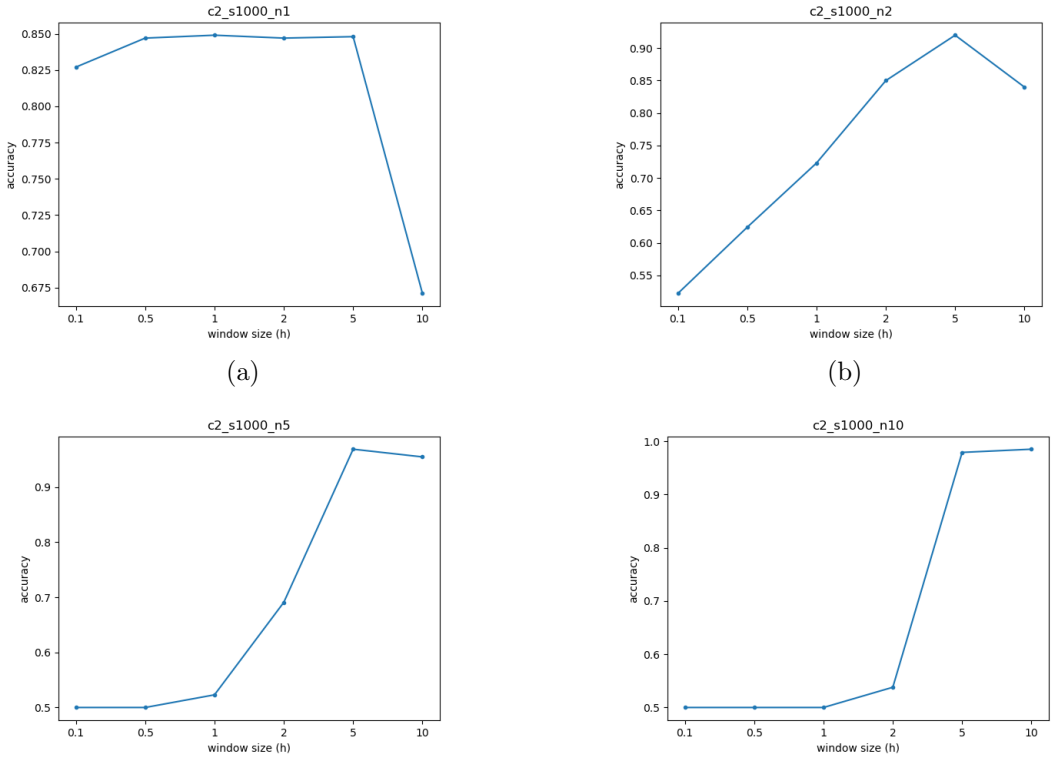


Figure 2: C=2,1000 sample with 4 different dimension. (a) is 1 dimension, (b) is 2 dimension, (c) is 5 dimension, (d) is 10 dimension.

In the figure 2, the sample number is 1000. Even the window size is small, each hypercube can still have some data points thus the density is more accurate than the scenario of 80 samples. Same as the figure 1, the accuracy decreased as the dimension increased.

The figure 3 shows the cases of 5 classes with two dimensions. Comparing with fig. 1a, the 5 classes spanned a complex space than 2 classes. The small window size causes few point fall in the hypercube and thus classification accuracy is low.

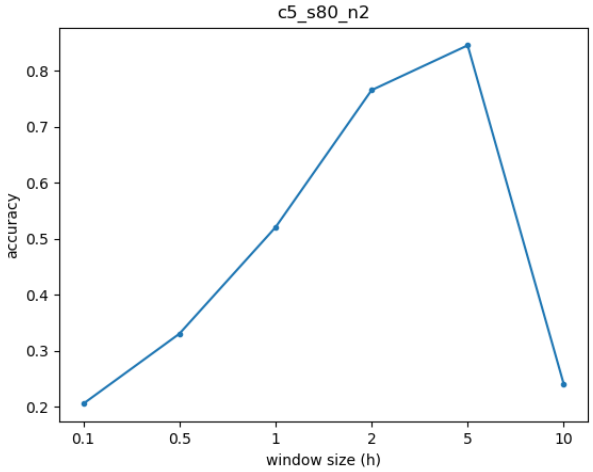
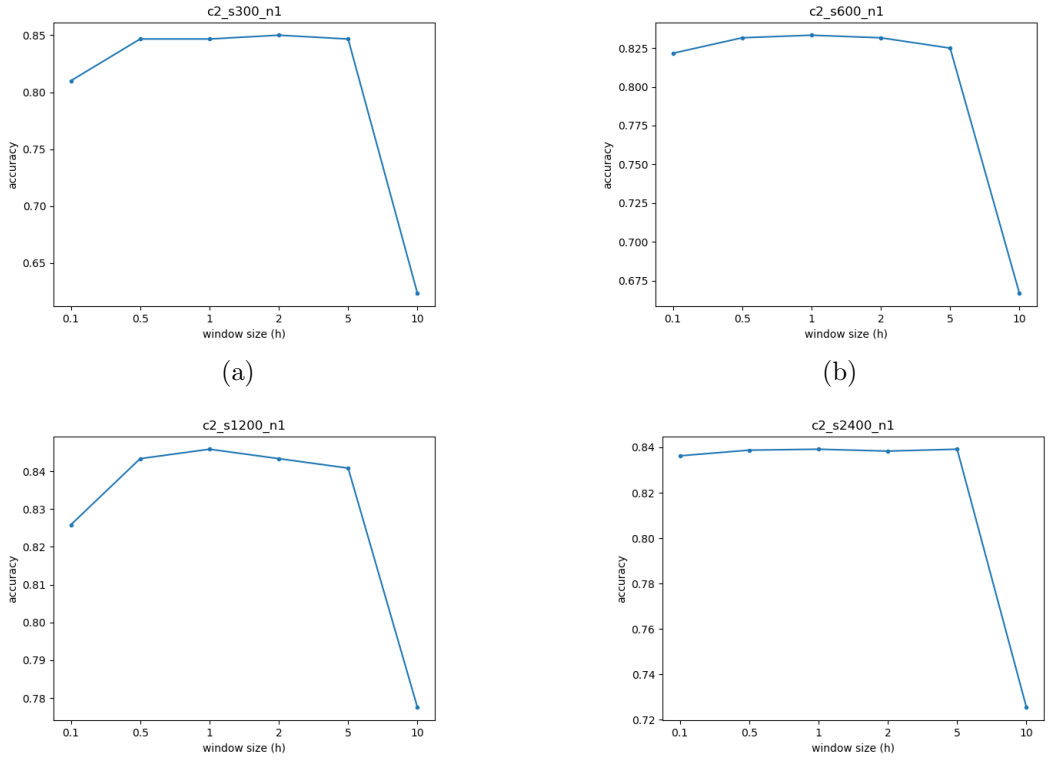


Figure 3: C=5, 80 samples in 2 dimensions.



(a) (b)

(c) (d)

Figure 4: C=2, validate the impact of training samples' number to classification accuracy. The size of train set is 150, 300, 600, 1200 from (a) to (d).

To validate the impact of the training samples' number, the size of training set is 150, 300, 600, 1200 respectively. The total sample is 300, 600, 1200, 2400 and only half of them were in the train set. In figure 4, we found that even very small window size can still have good performance when the size of train set is large. Figure 5 shows the similar experiment except the dimension is TWO. It give the similar conclusion as question 1 or 2.

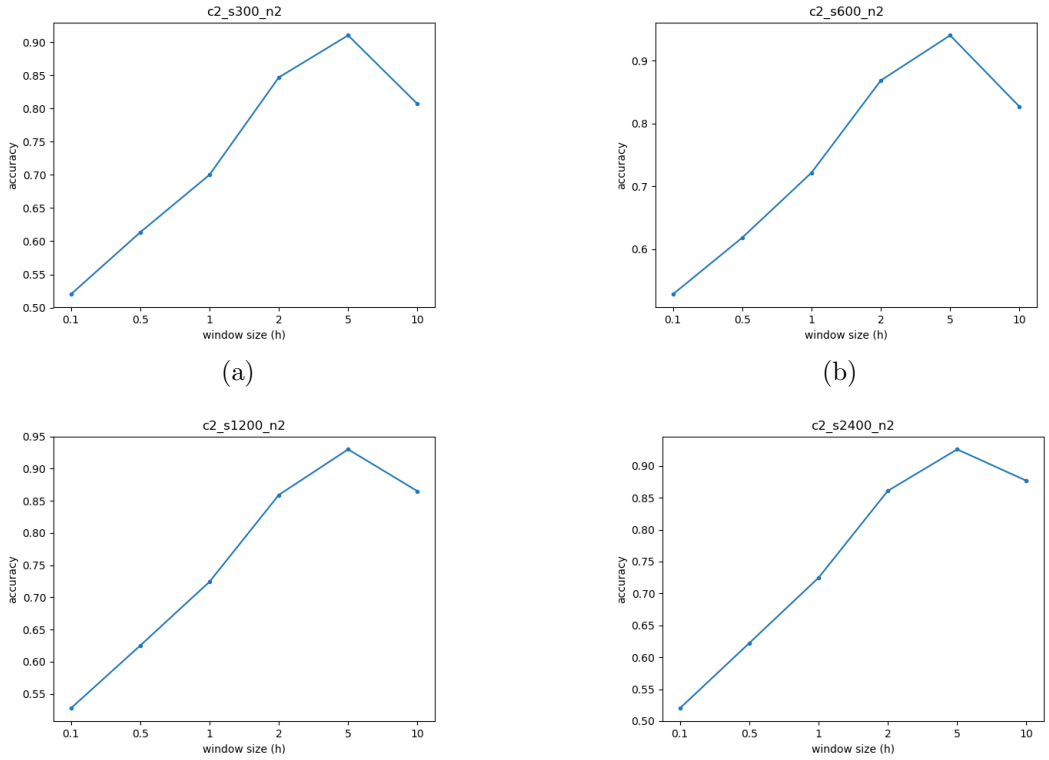


Figure 5: C=2, validate the impact of training samples' number to classification accuracy in TWO dimensions. The size of train set is 150, 300, 600, 1200 from (a) to (d).

2 Task 2

You can use the mehtod in Task 1 for data generation, for reuse its data.

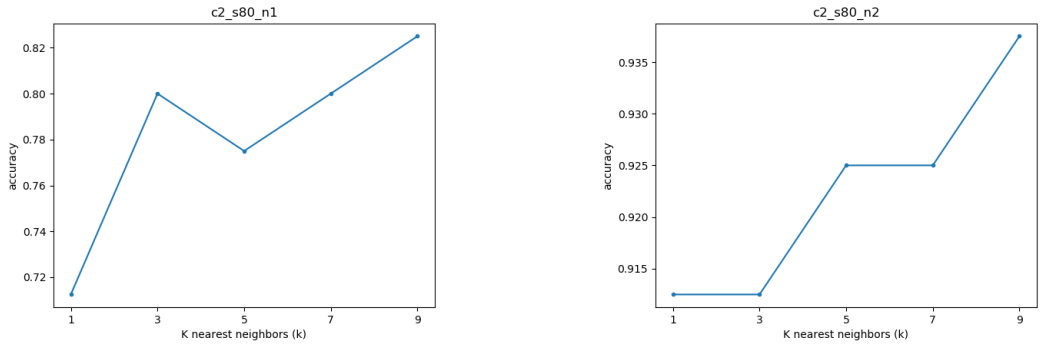
Study KNN method:
-C=2, 80 samples for each class (40 for training, 40 for testing): Evaluate and plot error rate (@testing data) for varying K (consider at least 4 different dimension: N).
-C=2, 1000 samples for each class (500 for training, 500 for testing): Evaluate and plot error rate (@testing data) for varying K (consider at least 4 different dimensions: N).
-C=5,N=2,80 samples for each class (40 for training, 40 for testing): plot error rate (@testing data) for varying K to analyze the best K.
-Analyze how number of training samples (e.g. from 10 to 10k) impact the error rate (@testing data), with different dimension: N. You can choose other parameters based on your own need.
-Study the difference of Euclidean distance and Manhattan distance. you can choose C=2, varying K. You can choose other parameters based on your own need.

KNN implementation:

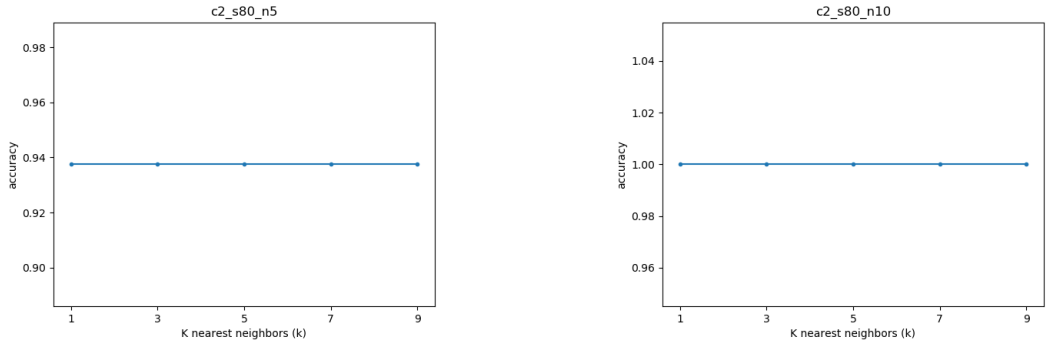
Comparing with the Parzen-window method, KNN estimate the density function by fixing the number of neighbor instead of fixing the volumn. The volumn is the hypercube of the Parzen window. The best choice of K depends upon the data; generally, larger values of k reduces effect of the noise on the classification; but make boundaries between classes less distinct. In our experiment, we set the K as 1, 3, 5, 7, 9 respectively for all the scenario.

Classification:

The idea of KNN classification is similar to voting. A predicted label of a new sample depend on the K nearest neighbors' class label. Thus, we need to calculate the distance between this new sample with all the data points (in Euclidean distance or Manhattan distance). Sorting all data points and let the K nearest neighbor 'vote' the predict label.



(a) (b)



(c) (d)

Figure 6: C=2, 80 samples with four different dimensions. (a) is one dimension, (b) is two dimension, (c) is five dimension, (d) is ten dimension.

The classification accuracy continually increase as the K become larger. The fundamental idea is the higher accuracy we get when we use more neighbors to identify the label of the new sample. However, if we continue to increase the value of K, the classify will add some data points in another class into the current dependent group. Thus will cause the classification accuracy drop.

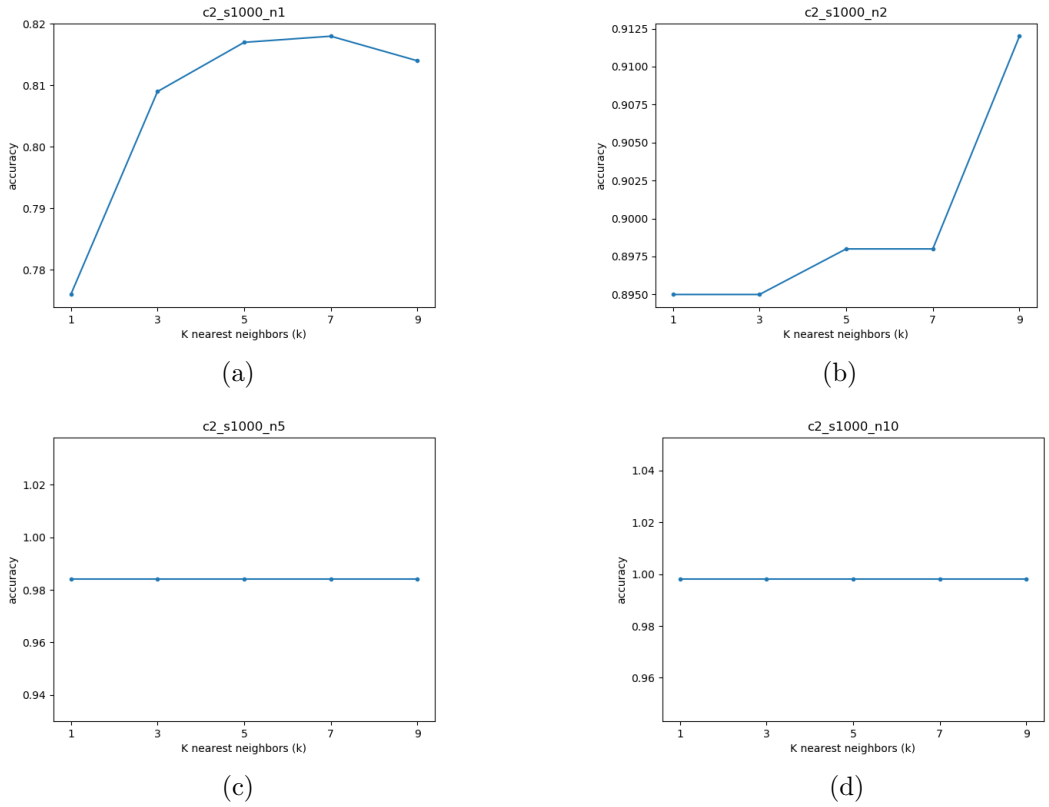


Figure 7: C=2, 1000 samples with four different dimensions. (a) is one dimension, (b) is two dimension, (c) is five dimension, (d) is ten dimension.

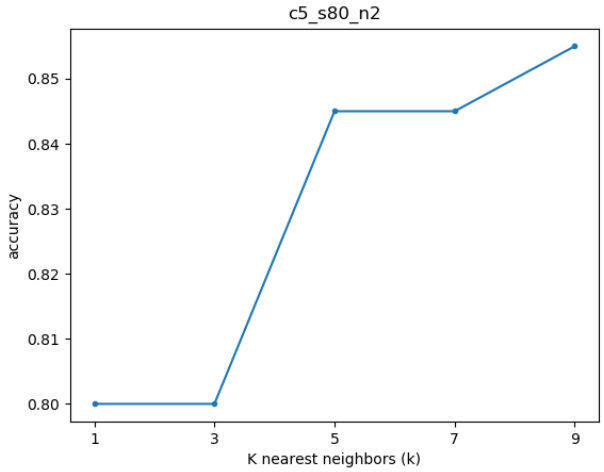


Figure 8: C=5, 80 samples in 2 dimensions.

Since our simulation data is clear and separable, the optimal K should be all the neighbors within the current class group. In figure 8, we will find the optimal K if we increase K until the accuracy of classifier drop.

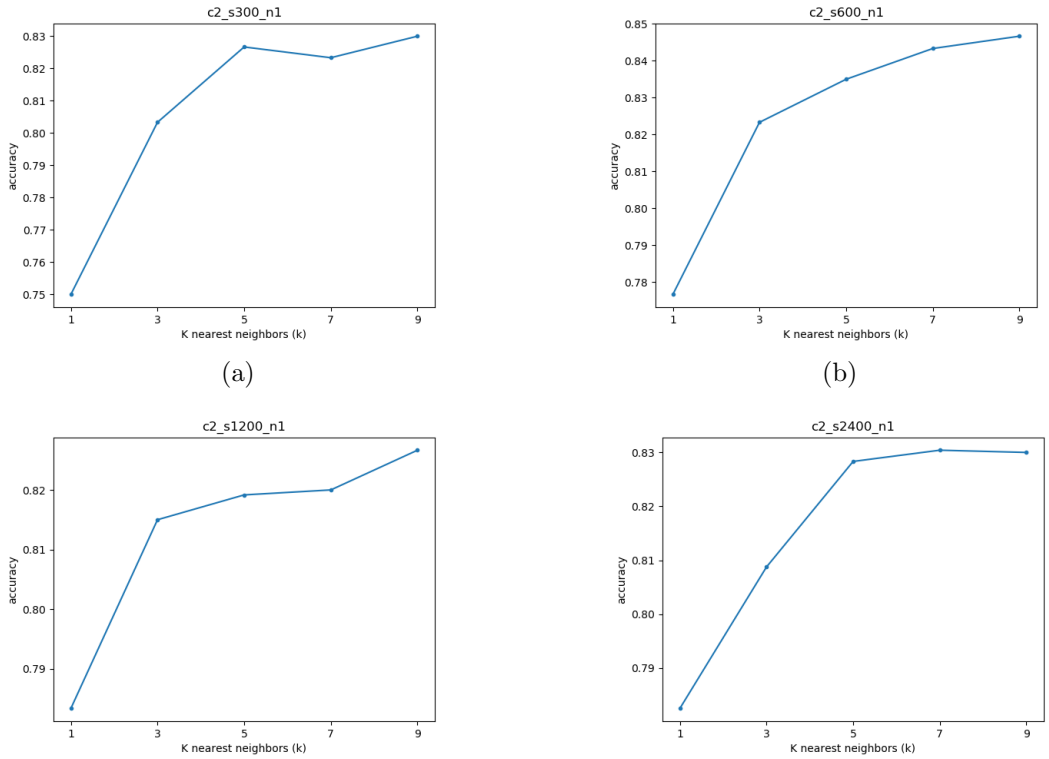


Figure 9: C=2, validate the impact of training samples' number to classification accuracy. The size of train set is 150, 300, 600, 1200 from (a) to (d).

In figure 9, we can find that the bigger training sample number, the better performance classifier have. The underlying reason is the estimated density would approximate the true density of distribution if we use more training sample.

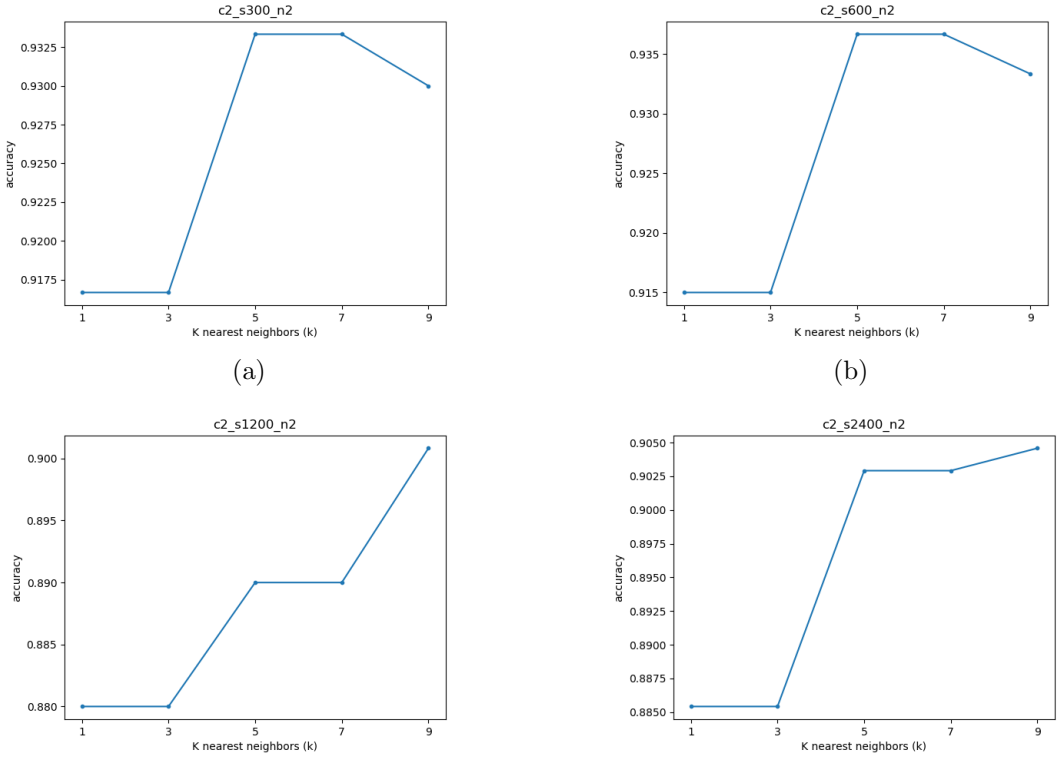
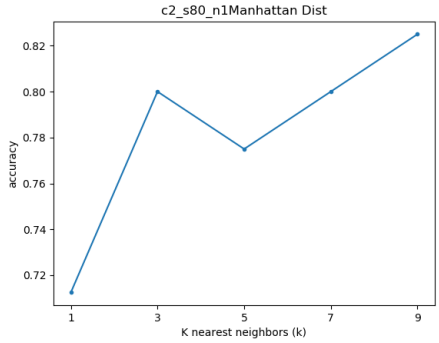
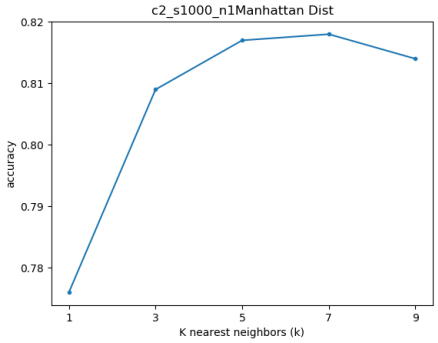


Figure 10: C=2, validate the impact of training samples' number to classification accuracy in TWO dimensions. The size of train set is 150, 300, 600, 1200 from (a) to (d).

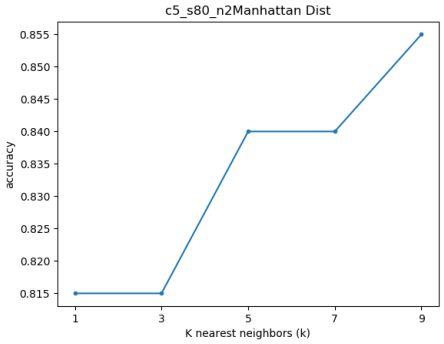
We also explorer the impact of different distance metric including Euclidean distance and Manhattan distance. The performance of two different metric depend on the dataset size, data dimension, noise level... The comprehensive experiment and the explanation refer to [5]. Our experiment executed the two distance on the three scenario including C=2,80 samples, N=1; C=2, 1000 samples, N=2; C=5, 80 samples, N=2. The results shown in figure 11.



(a)



(b)



(c)

Figure 11: Result of Manhattan distance. (a) is C=2,80 samples, N=1 ; (b) is C=2, 1000 samples, N=1; (c) is C=5, 80 samples, N=2

Reference

[1] https://en.wikipedia.org/wiki/Kernel_density_estimation
[2] https://sebastianraschka.com/Articles/2014_kernel_density_est.html
[3] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[4] https://en.wiktionary.org/wiki/Manhattan_distance
[5] https://arxiv.org/pdf/1708.04321.pdf