Data Mining Project 1 Report

Thang Le

What is the classification problem you are solving?

1. Classify three types of iris flower plants into their respective species.
2. Classify sets of 8-bit grey level pixels into 1 of 40 distinct face images.
3. Classify whether a banknote is genuine or forged based off extracted features.

What data do you use? What is the dimension? What are the attributes? Where is the data from? Cite any reference.

1. Iris data set.
    a. The dimension is 150 (instances) by 4 (attributes).
    b. Attributes information:
        1. sepal length in cm
        2. sepal width in cm
        3. petal length in cm
        4. petal width in cm
        5. class:
        -- Iris Setosa
        -- Iris Versicolour
        -- Iris Virginica
    c. source: https://archive.ics.uci.edu/ml/datasets/Iris
2. ATT faces images data set
    a. The dimension is 400 (instances) by 644 (attributes).
    b. Attributes information:

i. Attributes are pixels location whose value is [0-255] 8-bit grey level.

c. source: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

3. Banknote authentication data set

a. The dimension is 1372 (instances) by 5 (attributes).

b. Attributes information:

1. variance of Wavelet Transformed image
2. skewness of Wavelet Transformed image
3. curtosis of Wavelet Transformed image
4. entropy of image
5. class: 0 or 1 representing genuine or forged

c. source: https://archive.ics.uci.edu/ml/datasets/banknote+authentication#

What algorithms do you use? Describe the algorithm briefly.

1. The first algorithm is KNN(k-Nearest Neighbors). KNN classification works by computing the distance from the unknown data point to all training data points. It then takes N nearest neighbors by distance and use those to determine the label of the unknown by various methods such as majority voting.
2. The other algorithm is SVM(Support vector machine). SVM classification works by computing hyperplane(s) that classifies all training data into two or more classes. The label of the unknowns are determined by which side of the hyperplane they fall on.

How will you evaluate your result?

First, I will compute the true positive, true negative, false positive, and false negative from the results of the experiments. Using those values I can then calculate the accuracy and the F1 score from each of the experiments. Finally, I will average accuracies and the F1 scores respectively to evaluate my result.

What algorithms do you use? Which one do you write it by your own? Which toolbox or software do you use ?

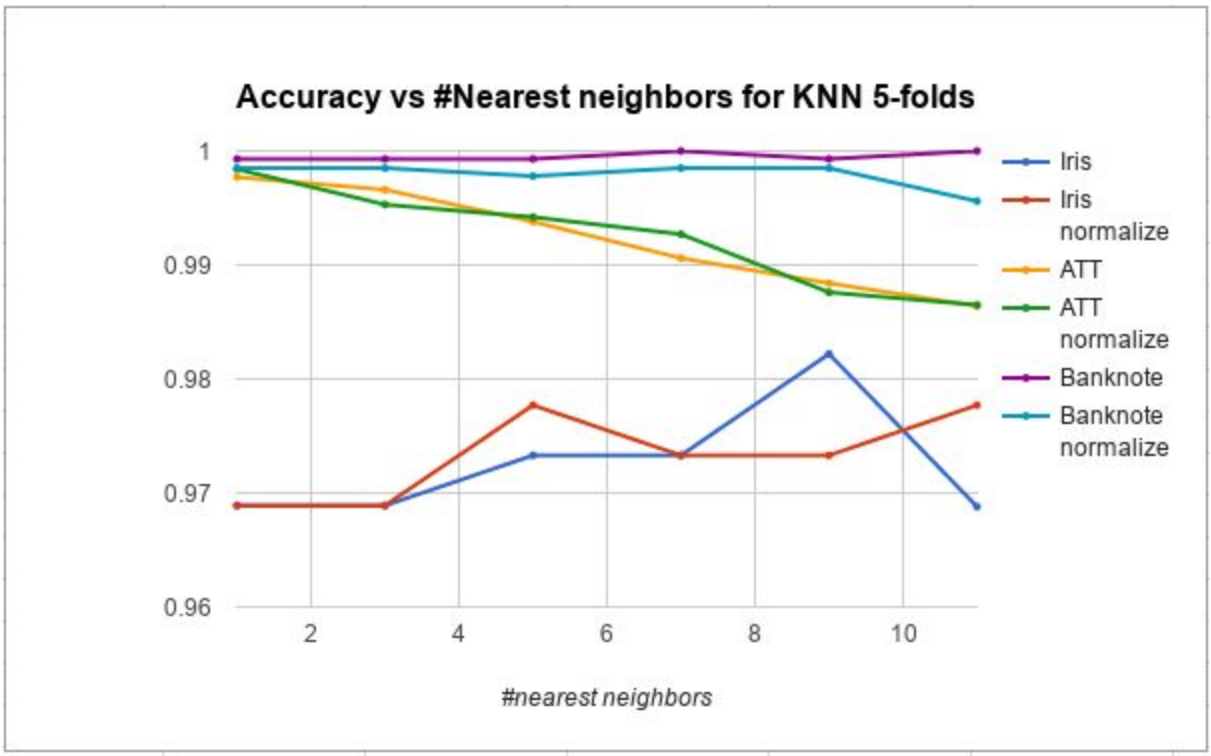
I implemented my own KNN algorithm. As for the SVM, I used the open source Java-ML library located here: http://java-ml.sourceforge.net/

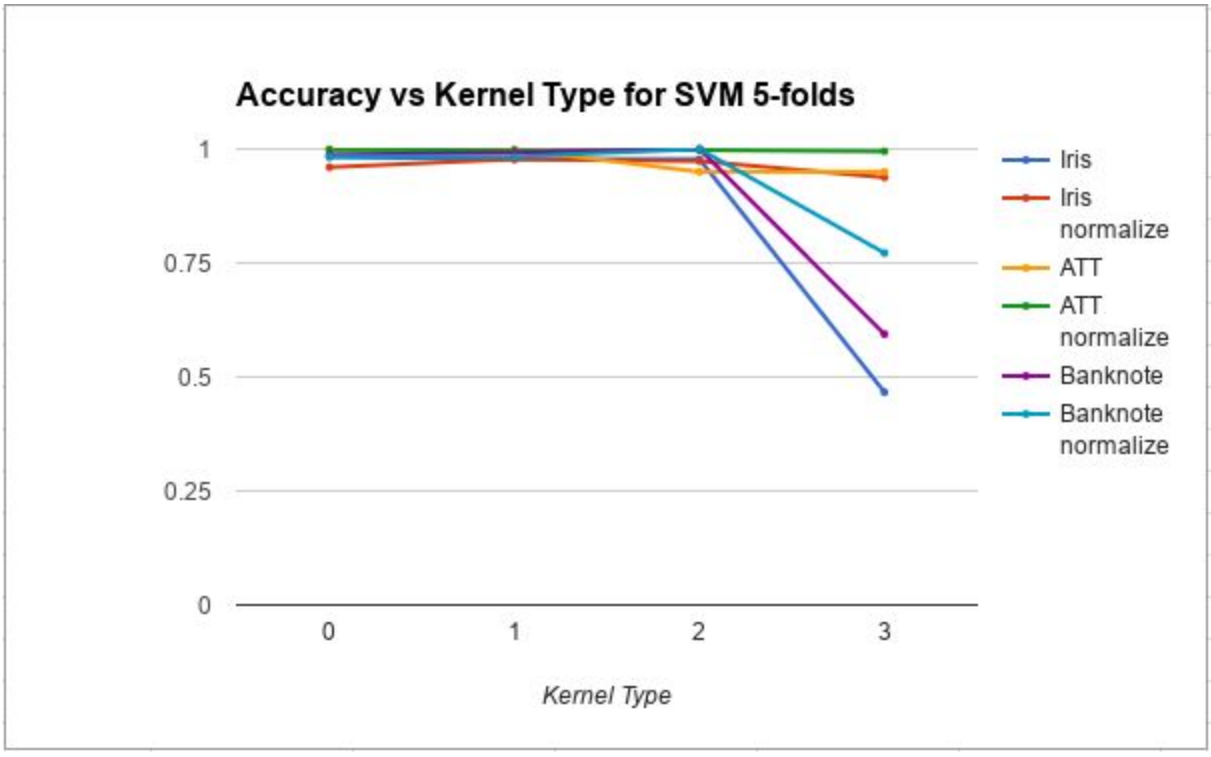
How many parameters does this algorithm have?

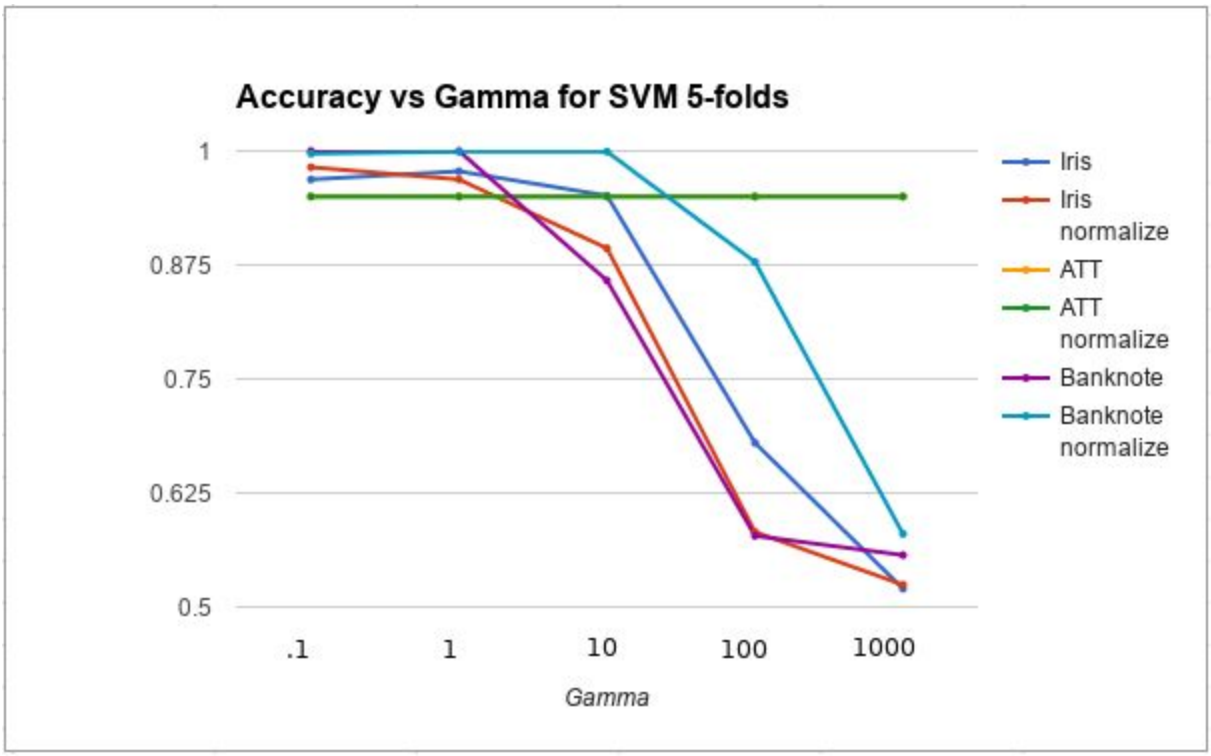
For my program, the KNN algorithm has two configurable parameters: the number of k-folds and the number of nearest neighbors use for majority voting. The SVM algorithm has three configurable parameters: the number of k-folds, kernel type, and the gamma value.

Preprocess the data: is it necessary to preprocess the data?

For all 3 of my datasets, normalization did not help increase the accuracy in most cases as seen in the tables below. I think this is due to the domain values in each respective datasets being relatively close to each other.


Accuracy vs #Nearest neighbors for KNN 5-folds
1
0.99
0.98
0.97
0.96
2
4
6
8
10
#nearest neighbors
Iris
Iris normalize
ATT
ATT normalize
Banknote
Banknote normalize


Accuracy vs Kernel Type for SVM 5-folds
1
0.75
0.5
0.25
0
0
1
2
3
Kernel Type
Iris
Iris normalize
ATT
ATT normalize
Banknote
Banknote normalize


Accuracy vs Gamma for SVM 5-folds
1
0.875
0.75
0.625
0.5
.1
1
10
100
1000
Gamma
Iris
Iris normalize
ATT
ATT normalize
Banknote
Banknote normalize