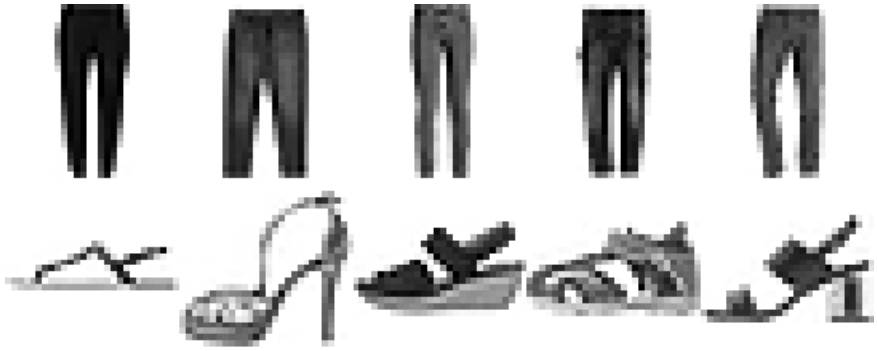
ENGR 421 / DASC 521: Introduction to Machine Learning
Homework 06: Support Vector Machine Classification

Deadline: April 29, 2022, 11:59 PM

In this homework, you will implement a support vector machine classifier using the histogram intersection kernel for image classification in Python. Here are the steps you need to follow:

1. You are given a binary classification data set, which contains 2000 clothing images of size 28 pixels × 28 pixels (i.e., 784 pixels). These images are from two distinct classes, namely, "trouser" and "sandal". The figure below shows five sample figures from each class. You are given two data files:
   a. `hw06_data_set_images.csv`: clothing images,
   b. `hw06_data_set_labels.csv`: corresponding class labels.



2. Divide the data set into two parts by assigning the first 1000 images to the training set and the remaining 1000 images to the test set.

3. You are going to represent each grayscale image using a color histogram over 64 bins. When there are 64 bins, the color bins are constructed as $[0, 4), [4, 8), \ldots, [252, 256)$. Calculate color histograms for training and test data points as the ratios of pixel values that fall into each bin. Your results should be like the following figures. (30 points)

```
print(H_train[0:5, 0:5])
print(H_test[0:5, 0:5])
[[0.86479592 0.00127551 0.         0.00255102 0.        ]
 [0.66836735 0.         0.00127551 0.00127551 0.        ]
 [0.66454082 0.00637755 0.00382653 0.00765306 0.00892857]
 [0.65816327 0.00765306 0.00892857 0.00127551 0.00382653]
 [0.5625     0.00255102 0.00255102 0.00127551 0.        ]]
[[0.68239796 0.00255102 0.00127551 0.00127551 0.00127551]
 [0.69770408 0.01658163 0.00510204 0.00382653 0.01020408]
 [0.73341837 0.02678571 0.01530612 0.00510204 0.00637755]
 [0.63903061 0.00892857 0.00255102 0.00127551 0.        ]
 [0.75382653 0.00765306 0.00127551 0.00127551 0.        ]]
```

4. You are going to use the histogram intersection kernel to calculate the similarities between input images. The histogram intersection kernel between two histograms can be written as

$$k(\boldsymbol{h}_i, \boldsymbol{h}_j) = \sum_{l=1}^{L} \min(h_{il}, h_{jl})$$

where L is the number of bins. Calculate the training and test kernel matrices using the histograms you calculated in the previous step. Please remember that the training kernel matrix should be calculated between training samples and training samples, whereas the test kernel matrix should be calculated between test samples and training samples. Your results should be like the following figures. (30 points)

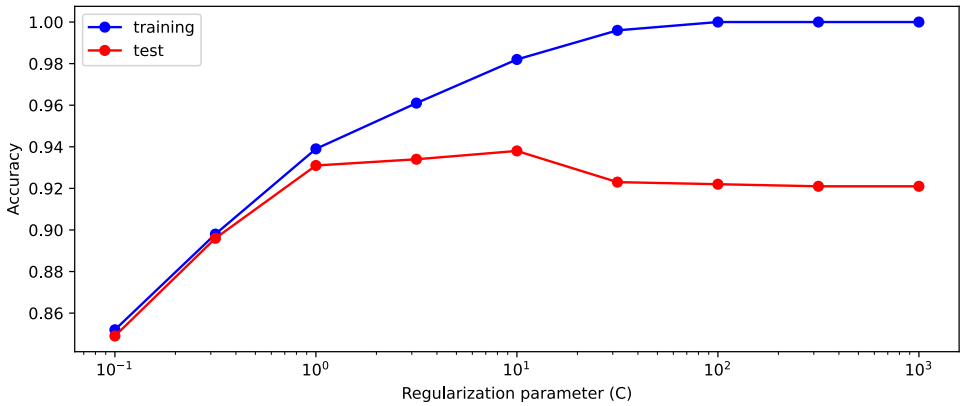
```
print(K_train[0:5, 0:5])
print(K_test[0:5, 0:5])
[[1.         0.72321429 0.77040816 0.75382653 0.62755102]
 [0.72321429 1.         0.73086735 0.78571429 0.68622449]
 [0.77040816 0.73086735 1.         0.84056122 0.70153061]
 [0.75382653 0.78571429 0.84056122 1.         0.76403061]
 [0.62755102 0.68622449 0.70153061 0.76403061 1.        ]]
[[0.77806122 0.80867347 0.82142857 0.88647959 0.79209184]
 [0.79464286 0.76403061 0.84566327 0.86607143 0.77933673]
 [0.8380102  0.74362245 0.85714286 0.83035714 0.68877551]
 [0.71556122 0.84438776 0.75       0.83418367 0.75765306]
 [0.84438776 0.76785714 0.82397959 0.84183673 0.73469388]]
```

5. Train a support vector machine classifier on the training kernel matrix you calculated in the previous step by setting the regularization parameter C to 10 (you must use the implementation discussed in the lab session). Calculate the confusion matrices for training and test data points using the training and test kernel matrices you calculated in the previous step. Your output should be like the following figures. (20 points)

```
y_train       -1    1
y_predicted
-1           484    9
 1             9  498

y_test        -1    1
y_predicted
-1           466   25
 1            37  472
```

6. Train support vector machine classifiers by setting the regularization parameter C to $10^{-1}, 10^{-0.5}, 10^{0}, 10^{0.5}, 10^{1}, 10^{1.5}, 10^{2}, 10^{2.5}, 10^{3}$. Draw accuracy for training and test data points as a function of C. Your figure should be like the following figure. (20 points)



What to submit: You need to submit your source code in a single file (.py file) named as ***STUDENTID.py***, where ***STUDENTID*** should be replaced with your 7-digit student number.

How to submit: Submit the file you created to Blackboard. Please follow the exact style mentioned and do not send a file named as ***STUDENTID.py***. Submissions that do not follow these guidelines will not be graded.

Late submission policy: Late submissions will not be graded.

Cheating policy: Very similar submissions will not be graded.