SYDE 671 Advanced Image Processing

David Abou Chacra University of Waterloo `Fall 2021`

Assignment 3 - Questions Part

1 Instructions

- 5 questions.
- The answers to this document are to be submitted as a part of one Jupyter notebook file only.
- Feel free to include images and equations and comment on both your codes and results meticulously.

2 Questions

Q1: In machine learning, what are bias and variance ?
When we evaluate a classifier, what are overfitting and underfitting and how do these relate to bias and variance?

Q2: Given a linear classifier like an SVM, how might we handle data that is not linearly separable? How does the kernel trick help in these cases?

Q3: Given a linear classifier such as an SVM which separates two classes (binary decision), how might we use multiple linear classifiers to create a new classifier which separates k classes?

Below, we provide pseudocode for a linear classifier. It trains a model on a training set, and then classifies a new test example into one of two classes. Please convert this into a multi-class classifier. You can take either the one vs. all (or one vs. others) approach or the one vs. one approach in the slides; please declare which approach you take.
Be aware that:

1. The input labels in the multi-class case are different, and you will need to match the expected label input for the *train_linear_classifier* function.
2. You need to make a new decision on how to aggregate or decide on the most confident prediction.

Note: An actual SVM application would separate the classifier training and testing into two different functions for now we are not doing so.

```
# Inputs
#   train_feats: N x d matrix of N features each d descriptor long
#   train_labels: N x 1 array containing values of either -1 (class 0)
                                    or 1 (class 1)
#   test_feat: 1 x d image for which we wish to predict a label
# Outputs
#   -1 (class 0) or 1 (class 1)
#
# Please turn this into a multi-class classifier for k classes.
# Inputs: As before, except
# train_labels: N x 1 array of class label integers from 0 to k-1
# Outputs:
#    A class label integer from 0 to k-1
#

def classify(train_feats, train_labels, test_feat)
    #Train classification hyperplane
    weights, bias = train_linear_classifier(train_feats, train_label)
    # Compute distance from hyperplane
    test_score = weights * test_feats + bias
    return 1 if test_score > 0 else -1
```

Q4: Suppose we are creating a visual word dictionary using SIFT and k-means clustering for a scene recognition algorithm. Examining the SIFT features generated from our training database, we see that many are almost equidistant from two or more visual words. Why might this affect classification accuracy? Given the situation, describe two methods to improve classification accuracy, and explain why they would help.

Q5: The way that the bag of words representation handles the spatial layout of visual information can be both an advantage and a disadvantage. Describe an example scenario for each of these cases, plus describe a modification or additional algorithm which can overcome the disadvantage.
How might we evaluate whether bag of words is a good model?