CS405 Machine Learning
Lab07 SVM
(100 points)

1. Introduction

We have learnt the principle of SVM which is a famous and widely used classifier. Even now, many works are done by SVM. Recently, the boost of neural networks overtook SVM and almost all the complex classification tasks are done by the new methods. However this does not mean that SVM cannot do such complicated things. In this lab, our goal is to write a software pipeline to identify vehicles in a video and apply a tight bounding box around vehicle detected. Following steps were implemented to achieve the goal:

- Perform a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set of images and train a linear SVM classifier
- Additionally, apply a color transform and append binned color features, as well as histograms of color, to HOG feature vector
- Implement a sliding-window technique and use trained classifier to search for vehicles in images
- Run the pipeline on a video stream and create a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles
- Estimate a bounding box for vehicles detected

It doesn't matter that you do not understand most terms above. We will provide a frame with some vacancies for implement SVM, all you need is to fill in the blank area and make the pipeline work well. We really prefer you to read the code and figure out how the whole thing be done if you are interested in.

2. SVM classifier in scikit learn

Before you start, a few things you should know. Firstly, when you perform a SVM classifier, make sure you standardize the raw data, as the SVM algorithm is based on the distance, the unit will influence the result badly.

Then you should know how to implement SVM in scikit learn, and that is quite easy. Just like other machine learning model we learnt before, first import the right model:

```
# Import SVM model. SVC is Support Vector Classification
from sklearn.svm import LinearSVC
```

(SVC for non-linear classification, you should define the kernel; SVR for Support Vector Regression). Than instantiate an object:

```
svc = LinearSVC(C=1e9)
svc.fit(X_std, y)
```

```
LinearSVC(C=1000000000.0, class_weight=None, dual=True, fit_intercept=True,
          intercept_scaling=1, loss='squared_hinge', max_iter=1000,
          multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
          verbose=0)
```

These arguments define the classifier. For example, C stands for regularization factor for soft SVM. Please search for documents and figure out the meaning of each parameters. The usage of this model is just like all scikit learn models. Use *fit* to train, *predict* to test.

Lastly, this project uses other Python packages such as OpenCV, scikit-image. They should be correctly installed. For Anaconda user, here are the tips:

- Install scikit-image:

```
pip install  scikit-image
```

- install opencv:

```
pip install opencv-python
```

These packages are used to load data and extract features by HOG, you just need to use these features to train you SVM. Than you can use your classifier to detect vehicles in a test video which uses a sliding window method.

3. Requirement

Please download the code and data, complement the missing code following the references. You should firstly train your SVM, if the result is good enough, implement it on the video *test_video.mp4*. If the detecting result is good, save it as a video.

Things you should submit:

1) Files: *train.py, experiment.py*
2) Video: your detecting result
3) Report: including the output results and a brief comment.