Homework 5
M1522.001000 Computer Vision (2020 Fall)
Out: Nov 24 Tuesday 02:00PM.
Due: Dec 8 Tuesday 11:59PM.

1 Overview

The goal of this homework is to explore SVM, K-means clustering, BoW classification. There are **2 Theory questions and 1 Python programming questions** on this assignment, and 100 points in total.

Put your **code and writeup** into a directory called "`(studentid)-(yourname)-HW5`" and pack it into a `zip` named "`(studentid)-(yourname)-HW5.zip`".

For example, `202012345-gildonghong-HW5.zip`.

Your writeup should be **typed** with **English**. Please do **not** attach your code to writeup. Upload your `zip` file to **ETL** until due date. Refer to the **ETL** page for the policies regarding collaboration, due dates, extensions, and late days.

Your homework should be formatted as following:

```
(studentid)-(yourname)-HW5
├─ writeup.pdf
├─ BoW_classification.ipynb
└─ (Do not include other file/directory.)
```

Make sure you have Python 3.6.9 installed.(Colab default python version) You can check your version with command `python -V`. All packages you can use for this homework are in `requirements.txt`. To install them with the package manager for Python and to run jupyter notebook, see the following command :

```
# Unzip hw5.zip and go to hw5 directory
$ pip install -r requirements.txt
$ jupyter notebook
```

You can use jupyter notebook or colab for this programming question.

Note that we will use a code similarity checker to detect plagiarism. You are expected to work on the assignment individually. I firmly believe that every student can do his or her own work. For your sake, please do not copy and paste others code. Good Luck!

2 Theory Questions

2.1 SVM (20 points)

Suppose that training examples are points in 2-D space. The positive examples are $X_+ = \{(1,1),(-1,-1)\}$. The negative examples are $X_- = \{(1,-1),(-1,1)\}$.

(a) [4 pts] Are the positive examples linearly separable from the negative examples? (i.e., Can you draw a line to separate the positive examples from negative examples?)

(b) [8 pts] Consider the feature transformation $phi(x) = [1, x, y, xy]^T]$, where x and y are the first and second coordinates of an example. Write down the transformed coordinates of X_+ and X_-. (i.e., $\phi(X_+)$ and $\phi(X_-)$ for all four examples.)

(c) [8pts] Consider the prediction function $y(x) = w^T\phi(x)$. Give the coefficient w of a maximum-margin decision surface separating the positive from the negative examples. (hit: w is $[4 \times 1]$ vector, whose elements are only 0 or 1.)

2.2 K-means (20 pts)

With a dataset $\mathcal{X}$ to be partitioned into k clusters, recall that the initialization step of the k-means algorithm chooses an arbitrary set of k centers $\mathcal{C} = \{c_1, c_2, ..., c_k\}$. Now, consider the following initialization method which we denote as the "Greedy" initialization method, which picks the first center at random from the dataset, and then iteratively picks the datapoint that is furthest from all the previous centers. More exactly:

1. Choose c_1 uniformly at random from $\mathcal{X}$
2. Choose the next center c_i to be $\text{argmax}_{x\in\mathcal{X}}\{D(X)\}$.
3. Repeat step 2 until k centers are chosen

   where at any given time, with the current set of cluster centers $\mathcal{C}$,

$$D(x) = \min_{c\in\mathcal{C}} \|x - c\|$$

.

With an example show that with the greedy initialization, the k-means algorithm may converge to a clustering that has an arbitrarily larger cost than the optimal clustering (i.e., the one with the optimal cost). That is, given an arbitrary number $r > 1$, given an example where k-means with greedy initialization converges to a clustering whose cost is at least r times larger than the cost of the optimal clustering. Remember that the cost of a k-means clustering was defined as :

$$\phi = \sum_{x\in\mathcal{X}} \min_{c\in\mathcal{C}} \|x - c\|^2$$

3 Programming questions [60 pts]

First of all, make sure you have **Python** 3.6.9 installed. You can check your version with command `python -V`. To run the program, you will also need numpy, matplotlib, appropriate version of opencv, and so on. To install them with the package manager for Python, run `pip install -r requirements.txt`. Then, you should complete the code. Specifically, you have to fill in **four blocks** starting with `### START CODE HERE_1∼4 ###` and ending with `##### END CODE HERE ####`.

It is forbidden to use anything other than the given package.

We have learned that bag of words is commonly used in image categorization. The general idea is to represent an image as a set of features and categorize it with learned classifier. In this homework, you will complete `BoW_Classification.ipynb` which have four blocks to fill.

1. Complete `getImageDescriptor` function to get histogram based on 2×2, 4×4 spatial bins. [15 pts]
2. Get `BoW_test_ll` which is extracted spatial histogram of all test images. [15 pts]
3. Get `test_bow,best_labels` which are Bag of Words and Labels of all test images. [15 pts]
4. Complete the code for classifying sample image. [5 pts]
5. Explain why denseSIFT would be better than SIFT. [10 pts]