Fall 2017
BBM 409: Introduction to Machine Learning Lab.

Instructor: Aykut Erdem
TAs: Levent Karacan
Tugba Erdogan

Assignment 1

Due on October 20, 2017 (23:59:59)

Instructions. There are two parts in this assignment. The first part involves a series of theory questions and the second part involves coding. The goal of this problem set is to make you understand and familiarize with k-Nearest Neighbor (k-NN) classification algorithm.

Part I: Theory Questions

k-Nearest Neighbor Classification

1. Let k-NN(S) be the k Nearest Neighbor classification algorithm on sample set S, which takes the majority of the closest k points.
    - Show that if in both 1-NN(S_1) and 1-NN(S_2) the label of point x is positive, then in 1-NN($S_1 \cup S_2$) the label of x is positive.
    - Show an example such that in both 3-NN(S_1) and 3-NN(S_2) the label of x is positive, and in 3-NN($S_1 \cup S_2$) the label of x is negative.
2. Suppose you use 2 features (F_1, F_2) for classification. While F_1 can take values between [500, 1000], F_2 can take 1 or 2 which also gives the correct class. For example you have [1 900] and [2 700] features as training set and you want to classify [1 600] point(which is originally belongs to class 1). But according to 1-NN (with Euclidean distance), input data belongs to class 2. What can cause this misclassification? Comment about the reason(s) and offer solution(s).

Linear Regression

1. Suppose you have m=28 training examples with n=4 features (excluding the additional all-ones feature for the intercept term, which you should add). According to the closed form solution $\theta = (X^T X)^{-1} X^T y$, what are the dimensions of X, y, θ?
2. Suppose you have m=1000000 training examples with n=200000 features for each example. You want to use multivariate linear regression to fit paremeters θ to our data. Should you prefer gradient descent or the closed form solution?

PART II: Classification of Bird Species from Images and Attributes

In this part, you will implement a nearest neighbor algorithm to predict class of a given bird attributes or image features. Similarly, your algorithm will also be able to predict attributes of a given bird image features. Specifically, you are going to implement a KNN algorithm on given attribute space and feature space of images. You will also extend your implementation as weighted KNN algorithm.

You have a bird dataset[1] consists of images of 200 different bird species. Each image has bird attributes annotated by humans. For a given attribute or image features, your algorithm will try to find the correct class of bird images.

A dataset is provided for your training phase. Test images will be provided later and announced from Piazza group. Since test images will be provided later, you should use a subset of the training set as the test set. In other words, you should split your training dataset into two set: training set which will be used to learn model and validation set which will be used to measure the success of your model. You can use k-fold cross-validation method which is explained in the class.

Dataset

- You can download it from course page.
- Dataset contains images from 200 classes: species information of birds.
- Training set contains 5033 images.
- There are 288 bird attributes which describe the birds with high level expressions such as color, shape, etc . If an attribute exists on a bird, it takes 1 otherwise it takes 0 .
- There are rough bird segmentations and bounding boxes coordinates saved as ".mat" file under the annotations folder. You can use this information to improve your results.
- We provide feature extractor code to compute color histograms and SIFT descriptors of bird images.
- You can also use deep image features of VGG-19 net using Caffe framework to earn extra points. Related code will be given to you.
-

Features

- There is no limitation about features. You can use any feature that you think it's proper for your classification assignment. Some features are listed below:

  - Tiny images: You can resize images to a very small size (for example [16,16,3] as provided) and use raw RGB values as feature.
  - Color histograms: You can generate a RGB or CIE L*a*b* color histogram for each image. You can use openCV library to extract histogram of image via *cv2.calcHist()* method (`http://docs.opencv.org/3.1.0/d1/db7/tutorial_py_histogram_begins.html`) or any library you want.
  - HOG descriptors

- You can use more than one feature by concatenating them.

Steps to follow

1. Extract feature for each image in training set (HOG, color histogram, VGG-19, etc.).
2. For a given test sample, you will try to estimate its class.
3. Finally you will compute accuracy of your model to measure the success of your classification method for each setting you have used:

   Accuracy = 100 * ((number of correctly classified examples) / (number of examples))

   You will report mean accuracy by averaging your accuracy results for k folds.

Submit

- report.pdf (PDF file containing your report)
- code/ (directory containing all your codes as Python file .py)

The ZIP file will be submitted via the department's submission system.

NOTE: To enter the competition, you have to register kaggle in Class with your department email account. The webpage of the competition will be announced later. Top 5 assignment will earn extra points.

Grading

- Code (60): k-NN: 20, Weighted k-NN: 40
- Report(40): Theory part: 12 points, Analysis of the results for prediction: 28 points.

Notes for the report: You should explain your choices and their effects to the results. You can create a table to report your results like:

Feature	Accuracy
Hog	0.02
Color Histogram	0.08
Attribute	0.4

Late Policy

You may use up to four extension days (in total) over the course of the semester for the three problem sets you will take. Any additional unapproved late submission will be weighted by 0.5. You have to submit your solution in (rest of your late submission days + 4 days), otherwise it will not be evaluated.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.