PR2: Image Classification

Published Date:
Apr. 21, 2020, 10:00 p.m.

Deadline Date:
Apr. 30, 2020, 11:59pm

Description:

****
This is an individual assignment.
****
Overview and Assignment Goals:

The objectives of this assignment are the following:

- Use/implement a feature selection/reduction technique. Some sort of feature selection or dimensionality reduction must be included in your final problem solution.
- Experiment with various classification models.
- Think about dealing with imbalanced data.
- F1 Scoring Metric

Detailed Description:
Develop predictive models that can determine, given an image, which one of 11 classes it is.

In this assignment, you will analyze features extracted from traffic images depicting different objects to determine their type as one of 11 classes, noted by integers 1-11: car, suv, small_truck, medium_truck, large_truck, pedestrian, bus, van, people, bicycle, and motorcycle. The object classes are heavily imbalanced. For example, the training data contains 8855 cars but only 2 bicycles and 0 people. Classes in the test data are similarly distributed.

The input to your analysis will not be the images themselves, but rather features extracted from the images. An image can be can be described by many different types of features. In the training and test datasets, images are described as 887-dimensional vectors, composed by concatenating the following features:
- 512 Histogram of Oriented Gradients (HOG) features
- 256 Normalized Color Histogram (Hist) features
- 64 Local Binary Pattern (LBP) features
- 48 Color gradient (RGB) features
- 7 Depth of Field (DF) features

Since the dataset is imbalanced the scoring function will be the F1-score instead of Accuracy.

Caveats:

+ Remember not all features will be good for predicting the object class. Think of feature selection, engineering, reduction (anything that works).

+ Use the data mining knowledge you have gained until now, wisely, to optimize your results.

Data Description:

The training dataset consists of 18000 records and the test dataset consists of 3000 records. We provide you with the training class labels and the test labels are held out. The attributes are floating point values and are presented in a dense matrix format within train.dat and test.dat. The numpy loadtxt function can be used to read the data in Python. The data are included in the **data.zip** file. While the .zip file is fairly small, note that it expands to over 400 MB. Ensure you have enough space on your drive before expanding the file. Moreover, you will need to think carefully about how you will organize computations so you do not run out of RAM during training.

- **train.dat**: Training set (dense matrix, samples/images in lines, features in columns).
- **train.labels**: Training class labels (integers, one per line).
- **test.dat:** Test set (dense matrix, samples/images in lines, features in columns).
- **format.dat:** A sample submission with 3000 entries randomly chosen to be 1-11.

Rules:

- This is an individual assignment. Discussion of broad level strategies are allowed but any copying of prediction files and source codes will result in an honor code violation.
- Feel free to use the programming language of your choice for this assignment.
- You are allowed 5 submissions per day.
- After the submission deadline, only your last submission is considered for the leaderboard.

Deliverables:

- Valid submissions to the Leader Board website: https://www.kaggle.com/c/cmpe-255-hw2 (use the last 4 digits of your SJSU id as your team name).
- **Canvas Submission of source code and report:**
  - Create a folder called pr2_SJSU-ID
  - Include a 2-page, single-spaced report describing details regarding the steps you followed for feature selection and classifier model development. The report should be in PDF format and the file should be called **report.pdf**. Be sure to include the following in the report:
    1. Name and SJSU ID.
    2. Rank & F1-score for your submission (at the time of writing the report). If you chose not to see the leaderboard, state so.
    3. Your approach.

4. Your methodology of choosing the approach and associated parameters.

- Create a subfolder called src and put all the source code there.
- Archive your parent folder (.zip or tar.gz) and submit via Canvas for PR2.

Grading:

Grading for the Assignment will be split on your implementation (70%), report (20%) and ranking submissions (10%). Extra credit (1% of final grade) will be awarded to the top-3 performing algorithms. Note that extra credit throughout the semester will be tallied outside of Canvas and will be added to the final grade at the end of the semester.

Files: In Canvas, you can find `data.zip` containing

- *Train Data:* train.dat
- *Train Labels:* train.labels
- *Test Data:* test.dat
- *Format File:* format.dat