FACULTY OF
COMPUTER SCIENCE

Advanced Topics in Machine Learning - Project Proposal (Extended proposal with Semi-Supervised Learning strategy) "Object detection using Semi-Supervised Classification"

Group - T02

May 24th, 2020

TEAM MEMBERS

- Shubham Kumar Agrawal
- Syed Muhammad Laique Abbas
- Usama Ashfaq
- Ranjiraj Rajendran Nair
- Pavan Tummala

Object detection using Semi-Supervised Classification

DATA LOADING

We propose to implement a `DataLoader` where we feed the images in an array along with their labels which further goes to the partition for train-test-validate.

FEATURE EXTRACTION

We target to extract the three key features `MPEG-7 Color Layout Descriptor`, `Visual BoW`, `SURF`. For the `Color Layout Descriptor`, the extraction process comprises 4 phases namely: `Image partitioning`, `Representative color selection`, `DCT transformation`, `Zigzag scanning`. For `Visual BoW` and `SURF` combined we extract the features for each of the image using the functionality provided by `OpenCV`. We then intend to construct a codebook vector using the k-means clustering algorithm of a certain vocabulary size using the extracted features from the first step. Thereafter, for each image feature we assign a code from the above-created codebook, and then produce a histogram for the codes, which we use as a feature to the model. We intend to follow and apply the above outline for the `SIFT` feature as well. Additionally, for the extra fourth feature, we plan to go for the `Frequency Density Histogram`.

FEATURE SELECTION

Once we successfully perform the feature extraction on all train-test-validate set separately, we aim to conduct feature selection (using PCA, Stepwise selection) and normalization and in the further development stages we plan on using the SSL techniques with focus on multi-class classification followed by model evaluation (using statistical methods)and re-iterate the steps for best results.

SEMI-SUPERVISED LEARNING PLAN

On the extracted features, we perform different feature reduction techniques like `PCA` and feature selection techniques like `ANOVA`, to select the most informative features for the modeling. After the feature selection, for different split values ranging from 0.1 to 0.9, we split the data into two different sets consisting of labeled and unlabeled instances and use these sets in a combination to be fed to the following *Transductive Semi-Supervised Learning* approaches:

1. **Label Propagation**

- For the **Label propagation** model, we are assuming that since the similar images would have similar descriptors (features) and so they would be mapped closely in the graph with high weights to the edges connecting to them.
- We are using the **Graph-based transductive semi-supervised learning approach - "label propagation"** offered by `sklearn`. Once training the model by tuning the hyperparameters such as choice of kernel, gamma, and the number of iterations, we predict the labels for the unlabeled dataset and calculate the accuracy score between the predicted and the actual values.

2. **Label Spreading**

- For the **Label Spreading** model, we assume that it works on the manifold assumption - the graphs, constructed based on the local similarity between data points, provide a lower-dimensional representation of the potentially high-dimensional input data (data points on the same low-dimensional manifold should have the same label).
- This graph-based method generally consists of three steps: graph construction, graph weighting, and inference. In the first step, the set of objects, X, is used to construct a graph where each node represents a data point and pairwise similar data points are connected by an edge. In the second step, these edges are weighted to represent the extent of the pairwise similarity between the respective data points. In the third step, the graph is used to assign labels to the unlabeled data points.
- We are using the **Graph-based transductive semi-supervised learning approaches - "Label Spreading"** provided by `sklearn`. Once training the model by tuning the hyperparameters such as choice of the kernel (RBF/kNN), α (clamping factor), γ and number of iterations, we predict the labels for the unlabeled dataset and calculate the accuracy score between the predicted and the actual values.

3. **Semi-Supervised Support Vector Machine**

- For the **S3VM** model, we assume that the cluster of similar images are likely to have the same features such that the decision boundary should not cross high-density regions, but instead, lie in low-density regions so that we get a clear separation.
- We plan to split on a set of ranges from 0.3 to 0.7, and further split the training data into labeled and unlabeled instances (creating a "no label"), and later recombine the training dataset labels. These merged sets will be given as input to the **Transductive semi-supervised support vector machine (TSVM)** from the `semisupervised` library. After which, we propose to carry out hyperparameter tuning like with kernel choice, initial weight for labeled instances, initial weight for unlabeled instances, balance parameters (α, β, γ), with several iterations. We then predict the labels for the unlabeled set and report the accuracy scores.
- We also aim to extend this process for the several split values mentioned above and generate learning curves to look out for the variation in the accuracy scores with an increase in the number of unlabeled instances in the dataset.

- Into this last point, we also plan to evaluate the `Safe SSL` assumptions concerning improving the learning performance from the `safeu` API from the respective `safeu.classification.TSVM.TSVM` sub-package with the same set of hyperparameters.

We carry out this approach for the above-mentioned algorithms for all the split values and plot a learning curve to investigate how the accuracy varies with an increase in the number of unlabeled instances in the dataset.

Moreover, We also plan to come with a confusion matrix and all evaluation metrics such as accuracy, precision, F-score, etc.