Vision Challenge Submission Report

Paolo Climaco
(climaco@ins.uni-bonn.de)

July 11, 2023

1 The core principles and logic behind your algorithm

I developed two main selection approaches, both strongly relying on the farthest point sampling (FPS), also known as k-center algorithm. The FPS is a feature-based data selection algorithm that, given a dataset, aims to find a representative subset of points that are spread out evenly across the data space.

To know more about the FPS, take a look at the work I will present at the DMLR workshop, "Investigating minimizing the training set fill distance in machine learning regression", where the FPS is described, and it is investigated, theoretically and empirically, how it can benefit the learning process of machine learning regression algorithms.

The following two subsections describe the data selection approaches developed to address the vision challenge.

1.1 First approach: fpscv-ft2

Algorithm's input and output:
The first approach, called 'fpscv-ft2', takes in input the pool of available data points and a specific input-class name, e.g., Cupcake, Hawk or Sushi, and gives as output a .json file containing the IDs and associated labels (0 or 1) of a selected subset of the available pool.

The selected set is used to train a binary classifier aiming to label the specific input-class with 1 and all other classes in the dataset with 0.

Algorithm's main idea:
The fpscv-ft2 selection algorithm starts by pruning the dataset using the FPS to select a subset of 1000 points. Next, after using the machine-generated labels to ensure that the 1000 selected points are not associated with the input-class, they are labelled with 0. The logic is to use the points selected with the FPS, summarizing the data space not containing elements associated with the input-class of interest, as negative examples for the classification task. Subsequently, the algorithm again exploits machine-generated labels and human annotations

to identify the input-class points in the dataset and labels them with 1. These points are used as positive examples for the classification task. Finally, the fpscv-ft2 algorithm exploits the knowledge of the learning algorithm used for the binary classification task (logistic regression) to perform nested cross-validation to evaluate and select the best-performing training subsets. There are two nested loops. The outer loop selects different subsets associated with class 1 (the input-class of interest). The inner loop selects different subsets associated with class 0 (the points selected with the FPS, representing all other classes in the dataset).

Step-by-step:

1. Pruning the available pool of data points by selecting a subset of 1000 points using the FPS.
2. Assigning label 0 to the 1000 points selected with the FPS.
3. Identifying points associated with the input-class of interest and labelling them with 1.
4. Performing nested cross-validation. Notice that the results of the nested cross-validation are sensitive to the choices of various hyperparameters, such as the number of folds considered, the number of points to select labelled as 1 and as 0. Such hyperparameters have been heuristically fine-tuned for each input-class to optimize the final results.
5. The best-performing subset is selected and the selected points IDs, together with the associated labels (0 or 1), are saved in a .json file.

1.2 Second approach: fps-nn

The second approach, called 'fps-nn', is less performing than the first one (roughly 3% less in terms of the average F1 score over the three binary classification tasks) but, contrary to fpscv-ft2, has the potential to work also for a multi-class classification task.

Algorithm's input and output:
The fps-nn algorithm takes in input the pool of available data points, and the names of all the classes in which we are interested, i.e., Cupcake, Hawk and Sushi. The output consists of three .json files, one for each of the input classes, containing the IDs of the selected points (the same IDs for each .json file). Each selected point's ID gets label 1 only in one of the three output .json files and label 0 in the other two.

Algorithm's main idea:
The fps-nn algorithm utilizes the FPS to select a representative subset of 1000

points from the given dataset. It then exploits the knowledge of the machine-generated labels and human annotations to select all points in the dataset associated with the three input-classes and uses them to train a feedforward neural network (FNN) for a multi-class classification task. The trained FNN is then used to label with one of the input classes the pruned dataset, previously obtained with the FPS. Finally, the algorithm creates three .json files, one for each input class, each containing the same IDs, those associated with the 1000 points selected with the FPS. In the .json file associated with a specific class, the IDs of the points labelled with that same class by the trained FNN are associated with label 1, the others with 0.

The algorithm's logic is based on the fact that the FPS is assumed to provide a pruned dataset representative of the whole data space. Thus, in each generated .json file, the points labelled as 1 represent the portion of the data space associated with one of the classes of interest, even if their actual class differs from it. On the contrary, the points labelled with zero summarize the data space associated with all other classes.

Step-by-step:

1. Pruning the available pool of data points by selecting a subset of 1000 points using the FPS.
2. Identifying points associated with the three input classes by exploiting the knowledge of the machine-generated labels and human annotations and use them to train a multi-class feedforward neural network (FNN).
3. Labelling with one of the input classes the 1000 points selected with the FPS, using the trained FNN.
4. Creating three .json files, one for each input class, each containing the IDs of the points initially selected with the FPS from the available pool of data points. Each of the points' IDs is associated with label 1 only in the .json file associated with the class provided by the trained FNN to label the respective point.

2 The unique features or aspects of your approach

Both developed approaches strongly rely on the FPS, which seems to effectively summarize the data space. The advantage of using the FPS in comparison to other coresets approaches is that it is very efficient in terms of computational cost and memory requirements.

The unique feature of the fpscv-ft2 algorithm is that it can exploit all the information the participants were allowed to use in the challenge, namely the machine-generated labels, the human annotations and the knowledge of the learning model. Thanks to that, the algorithm can select effective sets containing just a few data points. In particular, the largest set selected with the fpscv-ft2 consists of 95 data points when the maximum amount we could choose was 1000.

The main aspect of the fps-nn approach is that it is model-agnostic, that is, it does not exploit the knowledge of the model that is used for the classification task, thus having the potential to be effective with different classes of learning algorithms and not only with logistic regression.

3 Any challenges you faced during the development or implementation of your algorithm

Given the size of the dataset, the main challenge was developing an effective algorithm that could be employed with low computational costs in terms of run time and memory usage.

4 How your algorithm addresses the problem presented in the challenge

In this challenge, the participants were asked to design a data selection strategy that chooses the best training examples from a candidate pool of training images for training an image classifier, which is then employed for a binary image classification task across a set of visual concepts (e.g., Cupcake, Hawk, Sushi).

The fpscv-ft2 approach provides class-specific training samples that aim at maximizing the classifier's ability to identify the class of interest from all other classes in the dataset.

The fps-nn has the same goal as the fpscv-ft2 approach with the difference that it provides the same set of samples for all considered classes, thus, having the potential to be effective also for a multi-class classification task.