CSC 730 Assignment 09
Active Learning Based Rare Class Discovery

Mangesh Sakordekar

Overview

The aim of this assignment is to:

- Get the provided datasets from D2L
- Visualize the data w/ labels using 2 or 3-D tSNE.
- Write your own version of an active learning rare class discovery algorithm.
- Run your code on the dataset and keep track of the number of classes discovered vs. number of queries.
- Plot the number of classes discovered vs. number of queries.
- Rerun the same experiment using a random query strategy and plot it.

Visualizing Dataset

The datasets were imported from D2L, normalized and plotted using 2D tSNE. The plots of the datasets can be seen below.

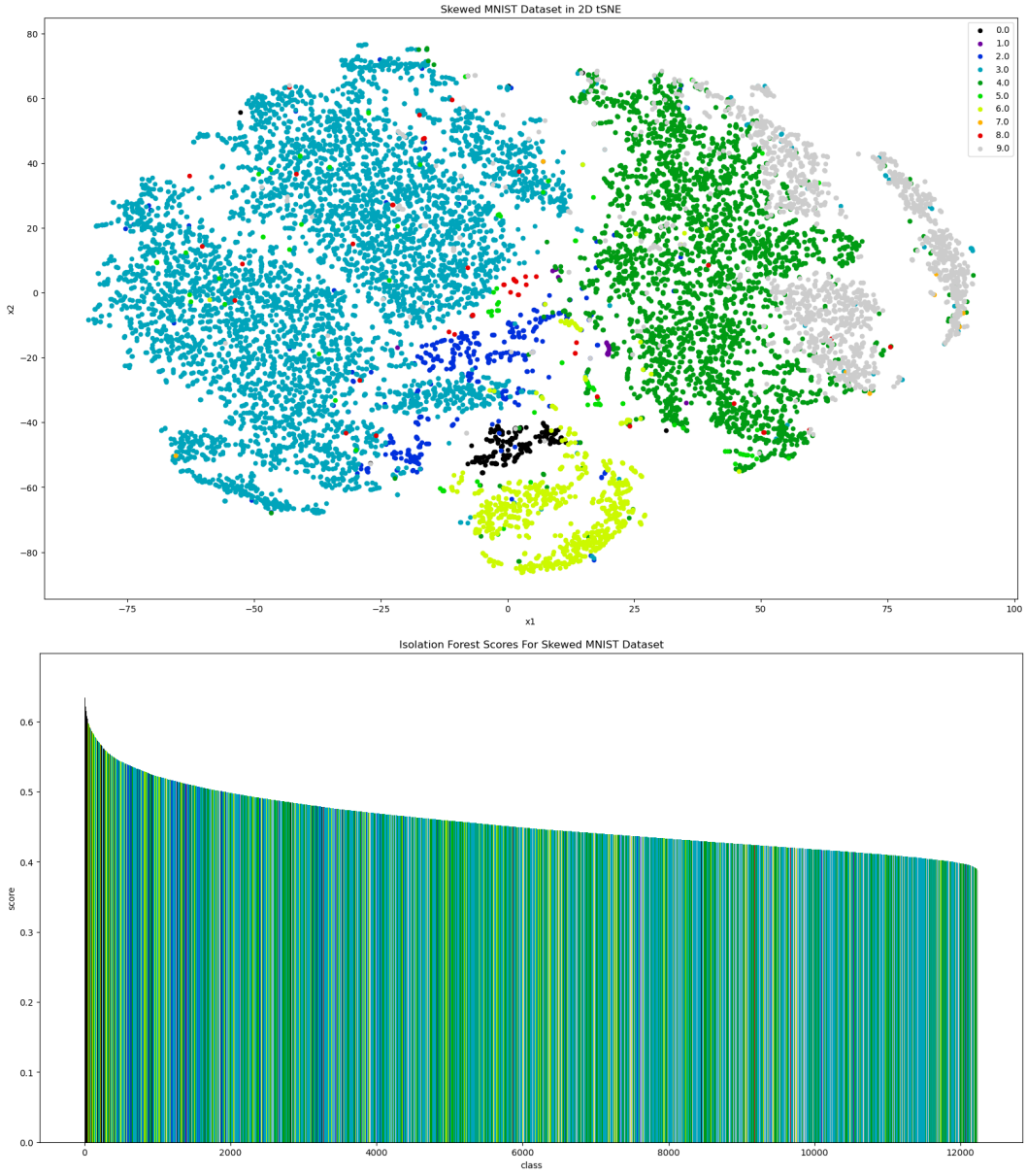

Figure 1: Skewed MNIST Top: 2D tSNE Bottom: Isolation forest scores

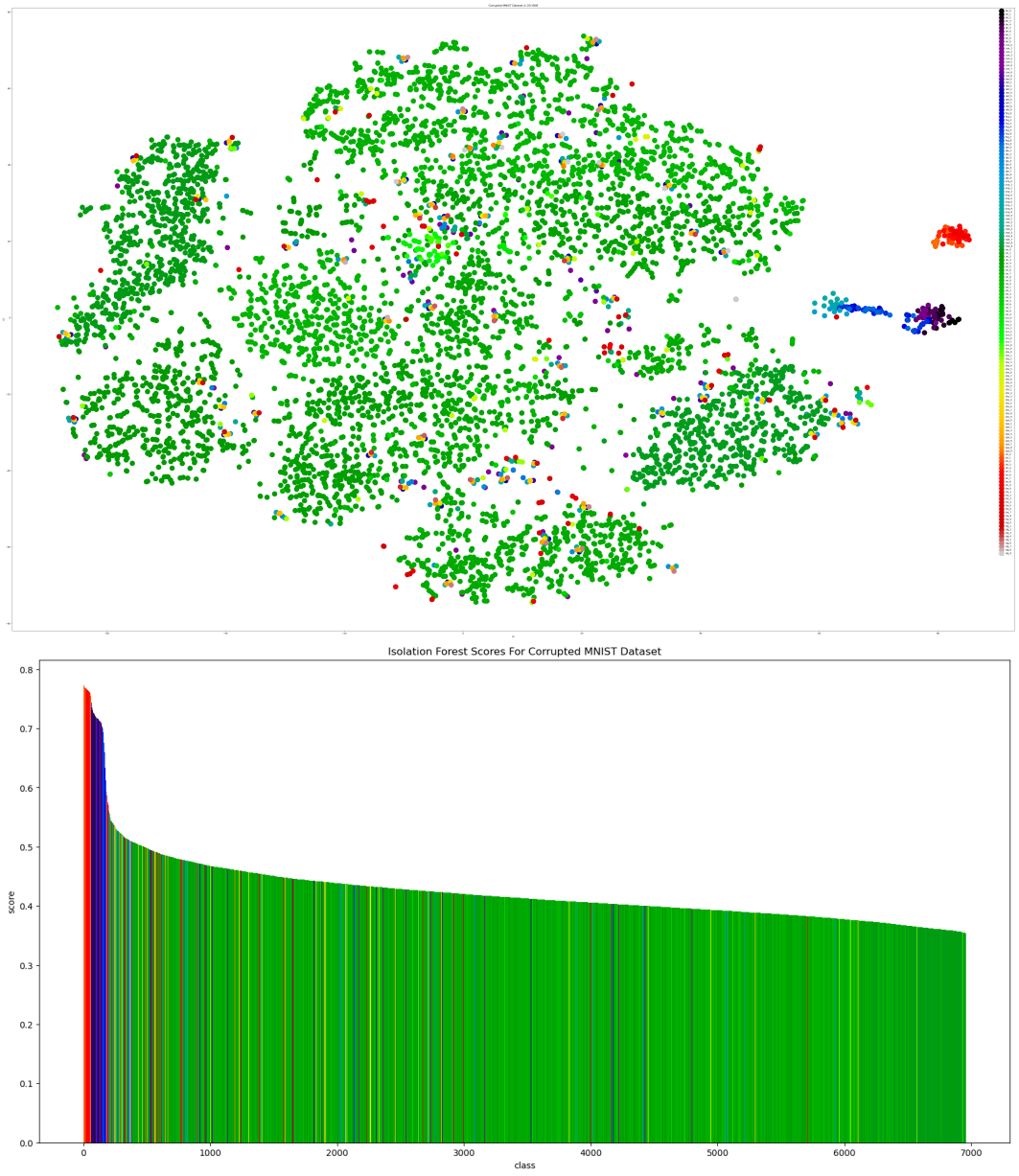

Figure 2: Corrupted MNIST Top: 2D tSNE Bottom: Isolation forest scores

In the bar graphs of anomolusness scores we can see that a variety of classes are packed in the region of high anomaly scores where as in the region of low anomaly scores we only see classes which are in the majority. We can also see that some of the corruption types are easier to detect and are grouped together on the graph. We can use this information to sample the regions with high density of varying classes first.

Query Strategy

Random / Base Strategy

This strategy randomly shuffles the indices of the labels array and pops the element at front for each query.

Strategy 1 - Isolation Forest

Isolation forest was used to sort the points by their anomaly scores and then the list was queried from the most anomalous point to the least.

Strategy 2 - Lowest Confidence Point

This strategy queries the point which has the lowest probability for the class that it has been classified. This helps the classifier to learn about the region which it is the least confident with. A neural net was used as the classifier in this strategy.

Strategy 3 - GMM Clustering

This strategy runs GMM clustering algorithm on the data. Points from each cluster are given sequentially. By doing so I wanted to provide the model with points from different regions of the data-space.

Strategy 4 - IFDFSLC (Isolation Forest + DFS + Lowest Confidence)

In this strategy, the points are sorted according to the anomaly scores provided by isolation forest and then sampled as if traversing a pre-processing Depth First Search algorithm on a binary search trees. This allows the algorithm to skip sections of the sorted list which are likely to be filled with similar labels, increasing the likelihood of encountering new classes. As it assumes labels for some points, this algorithm starts to plateau after a certain number of classes are found. To overcome this, the model switches to a strategy of querying the least confident point from a neural net model which is trained on the labeled points found previously.

```
        if self._switch:
            self._clf._fit(self._X_L,self._y_L)
            return

        if plen == len(self._lbl_cnt):
            self._prev_ctr += 1
        else:
            self._prev_ctr = 0

        if self._prev_ctr == self._threshold:
            self._switch = True
            self._clf._fit(self._X_L,self._y_L)

        if self._q_cnt == 0:
            self._q_cnt += 1
            self._left.add(lbl)
            return

        self._curr_node.lbl = lbl


        if lbl in self._left or len(self._q) == 0:
            if len(self._stack) != 0:
                temp = self._stack.pop(-1)
                self._left.add(temp.lbl)
                self._q = []
                self._add_to_q(temp.ind + temp.cut, temp.cut // 2)

        elif lbl not in self._right:
            self._right.add(lbl)
            self._stack.append(self._curr_node)
```

Figure 3: IFDFSLC Post Processing Implementation

Active Learner Based Rare Class Discoverer Implementation

Active Learner Based Rare Class Discoverer was implemented as a class which takes in the data, a Strategy instance, an Oracle instance and a value for maximum queries to perform. The Active Learner's implementation follows the pseudo code provided in the slides. Strategy and Oracle are classes implemented separately. The Oracle class takes in the labels of the data set and returns the label of the index queried. The Strategy classes are implemented to carry out each of the strategies which all have a "get_query" and "process" functions to get the index to be queried and post processing once the label is obtained.

Results

Each strategy was tested on both the datasets. On the skewed MNIST dataset, i ran the algorithms for 20 queries and 200 queries on the corrupted MNIST dataset. The number of classes discovered were stored at each query and plotted. The graphs can be seen below.

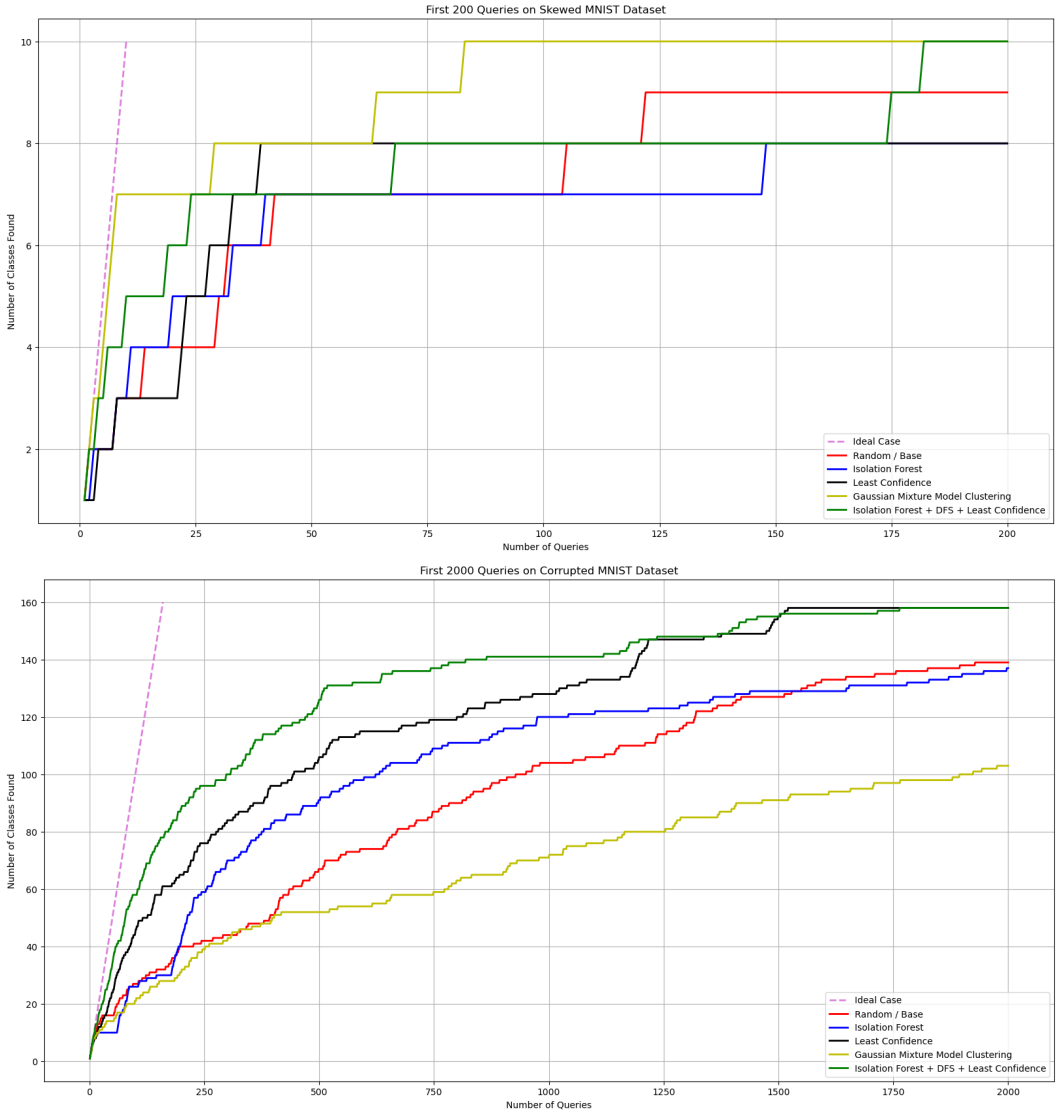

Figure 4: # of Classes vs # of Queries Top: skewed MNIST bottom: corrupted MNIST

As we can see in the plots above, the IFDFSLC algorithm outperforms the random strategy as well as the individual strategies which were used to create this algorithm. Testing the Skewed MNIST dataset, IFDFSLC algorithm found 10 out of the 180 classes in 20 queries, the next best is Random with 9 classes in 200 queries followed by isolation forest and least confidence with 8 classes. The representation based strategy using clustering with Gaussian Mixture Models outperformed all other algorithms in this dataset detecting all 10 classes within the first 80 queries. With the corrupted MNIST dataset, IFDFSLC algorithm found around 158 classes in 2000 queries. An interesting observation was, it managed to find a new class at every two queries for the first 100 queries. Second best performing strategy was the Least Confidence Point strategy which found the same number of classes as well, followed by Random and Isolation Forest, though, the random strategy struggled with this highly imbalanced dataset at the beginning. Surprisingly, the representation based strategy performed the worst with this dataset. I believe that it failed with this dataset as the class sizes are really small and since GMM form similarly sized clusters it ends up sampling only the classes in the majority.

Conclusion

An active learning based rare class discovery algorithm was successfully implemented and tested on both the e MNIST_C derived dataset and the MNIST-skewed dataset. For both the datasets, the developed algorithm outperformed the random strategy.