Report for Assignment-II 2019-20
BITS F464 - Machine Learning
Active Learning Assignment

SUBMITTED BY

Akshit Khanna 2017A7PS0023P
Jhaveri Ayush Rajesh 2017A7PS0215P
Vitthal Bhandari 2017A7PS0136P

[To run the code: run all the cells in the Ipython Notebook. The last cell is a menu with all the different sampling options.]

What is Active Learning?

Active Learning is the process of guiding the sampling process by querying for certain types of instances based upon the data that we have seen so far. It enables machine learning models to learn effectively with less data.

What are its types?

1) Stream-based Active Learning:
    a) Consider one unlabeled example at a time
    b) Decide whether to query its label or ignore not
2) Pool-based Active Learning
    a) Given a large unlabeled pool of examples, rank examples in order of informativeness
    b) Query the labels of the most informative examples.

These queried labels are added to the training set to enable the classifier to predict effectively with less training data.

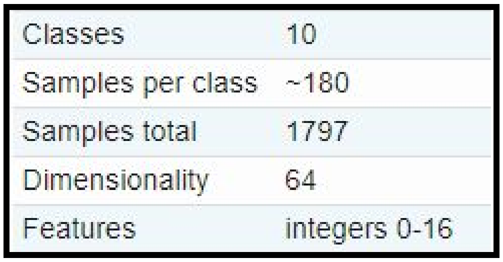
When is an example informative?

In stream-based active learning, we decide if an unlabelled example is informative enough to be queried for it's label. In pool-based learning, we rank and query their labels on the basis of 'informativeness'. But how is this 'informativeness' measured? What is our query strategy?

How did we proceed?

Several query strategy frameworks exist and have been explained below. Their implementation has been done in **Python** without the aid of any active learning library (**only pandas, numpy, matplotlib and sklearn classifier libraries**).

Dataset Used



Classes	10
Samples per class	~180
Samples total	1797
Dimensionality	64
Features	integers 0-16

The sklearn digits dataset was used(classification).

The query strategy frameworks used with both pool-based and stream-based scenarios are explained below:

1. Uncertainty Sampling

The event that the current classifier is most uncertain about is queried. We have implemented three uncertainty measures:

a) <u>Margin</u>

The difference between the top two most confident predictions. The one with minimum margin is the most uncertain label.

$$x^*_{SM} = \mathrm{argmin}_x \, P_\theta(y_1|x) - P_\theta(y_2|x)$$

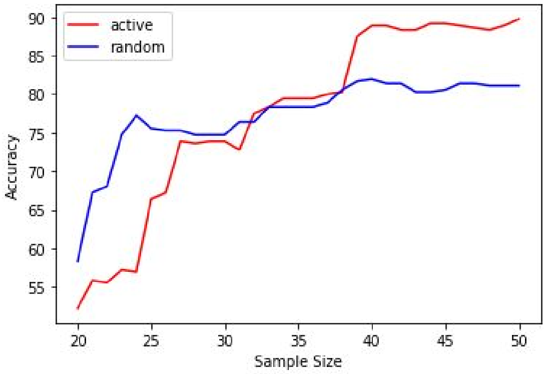


Fig 1: Pool Based Scenario

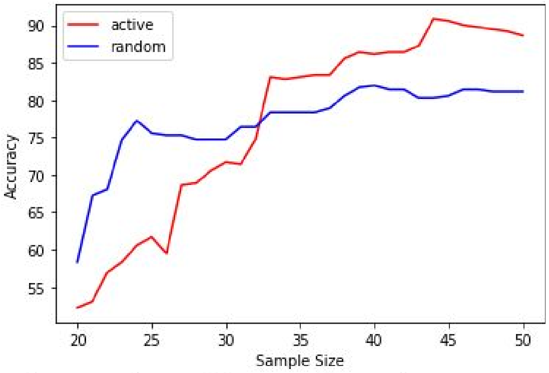


Fig 2: Stream Based Scenario

The performance of active learning in both pool and stream based scenarios is better than random selected labels as seen from the graphs. Pool based is also narrowly better than stream based as is expected. This measure performs the best out of the other two measures.

b) Entropy

The difference between all predictions. The one with highest entropy is the most uncertain entropy.

$$x^*_{LE} = \underset{x}{\operatorname{argmax}} - \sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)$$

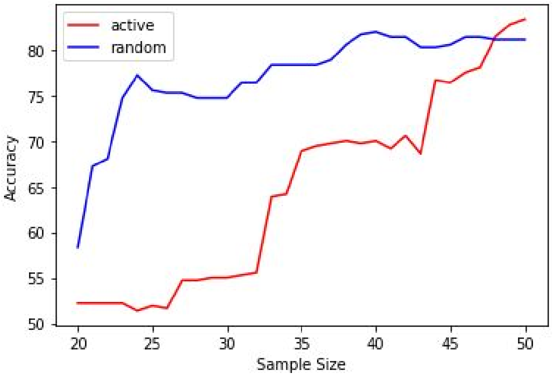


Fig 3: Pool Based Scenario

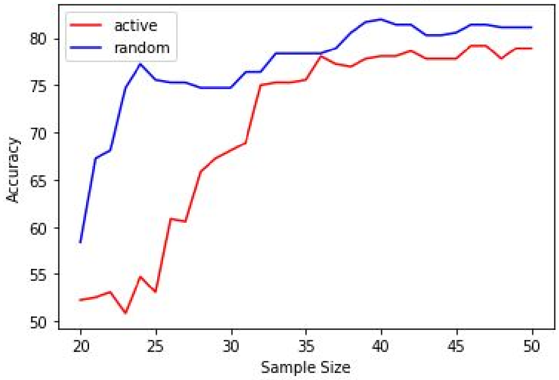


Fig 4: Stream Based Scenario

The performance of pool based is better in this measure also than stream based as is expected. The lower performance than random labels can be attributed to poor starting labels in case of active learning tests. It can be observed that the initial difference of accuracies reduce in the stream based scenario.

c) Least Confidence

The difference between the most confident label and 100% confidence. The one with this highest value is the most uncertain entropy.

$$x^*_{LC} = \operatorname{argmax}_x 1 - P_\theta(\hat{y}|x)$$

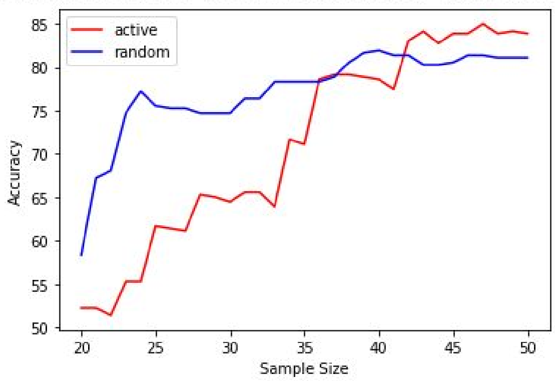


Fig 5: Pool Based Scenario

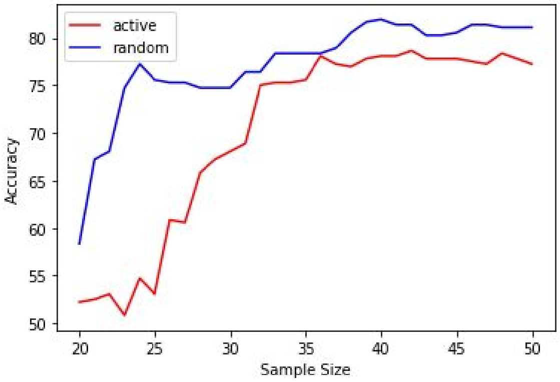


Fig 6: Stream Based Scenario

The performance of pool based is better in this measure also than stream based as is expected. The lower performance than random labels can be attributed to poor starting labels in case of active learning tests. This measure performs similarly as entropy.

2. Query by Committee

Query by Committee (QBC) approach involves maintaining a committee of models, all trained on the current labelled data. The example with the most disagreement among models is queried for its label. The disagreement measures we implemented are:

a) Vote Entropy

$$x^*_{VE} = \underset{x}{\text{argmax}} - \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}$$

y_i ranges over all possible labels, $V(y_i)$: number of votes received to label y_i

C is the number of models in the committee. The one with the maximum vote entropy is the one with the highest disagreement among models and is queried.

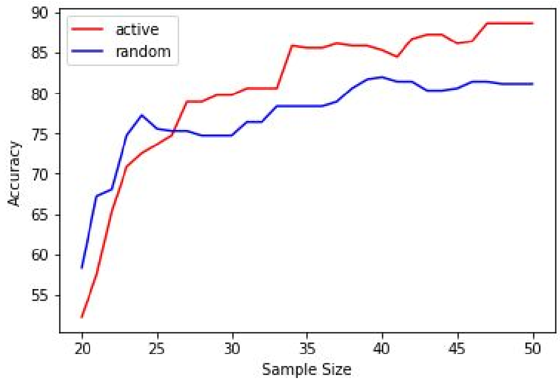


Fig 7: Pool Based Scenario

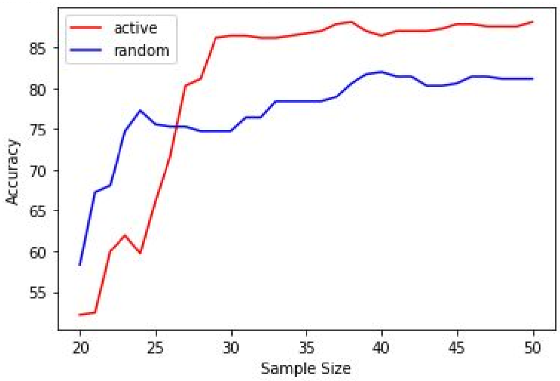


Fig 8: Stream Based Scenario

The performance of pool based is better in this measure also than stream based as is expected. The accuracies are comparable to the best case in uncertainty sampling with margin measure.

b) KL Divergence

p(xi) is the probability of class i by a model.
q(xi) is the average of the probabilities of class i by all the models in the committee.
N is the number of classes

$$D_{KL}(p||q) = \sum_{i=1}^{N} p(x_i) \cdot log \frac{p(x_i)}{q(x_i)}$$

The above statistic is calculated for each model in the committee, which is averaged over all the models to give the KL Divergence of a single example. The one with the highest KL Divergence is the example with the most disagreement among models.

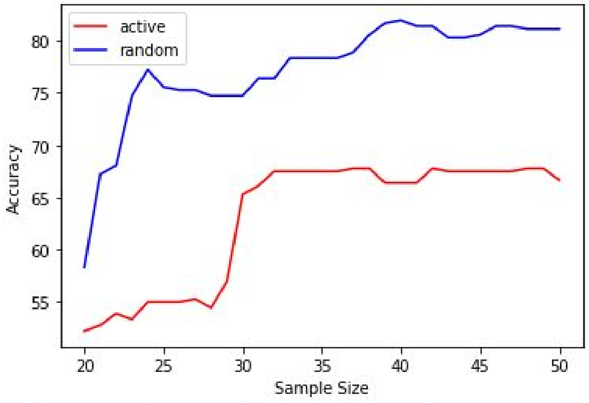


Fig 9: Pool Based Scenario

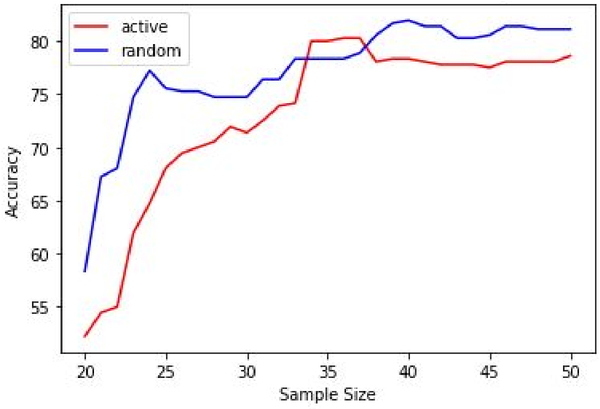


Fig 10: Stream Based Scenario

The performance of pool based scenario with KL divergence measure is consistently poor in contrast to vote entropy. This may be due to KL divergence not being fit for our dataset.

3. Divergent Sampling

The basis of the divergent sampling approach is that the sample selected should be a good representative of the original dataset, covering diverse points across the set. Along with this, the uncertain points should also be considered while querying for this sample. Keeping this in mind, we have devised the following algorithm to perform Divergent Sampling.

Step 1: For an initial sample size of N, perform K-Means Clustering with N Clusters on the training set. Randomly query an example from each cluster to form the initial sample. This would contain the representative points.

Step 2: The euclidean distance from each point in this sample to each point in the remaining remaining set is calculated. We get a vector of euclidean distances for each example in the training set containing distances from each sample point.

Step 3: The minimum distance in each training example’s vector is extracted. The one with the maximum value among these is the example to query. It is the example that is considered the most different from the examples already in the sample.

Step 4: The steps 2 to 3 are repeated until the maximum desired sample size is reached.

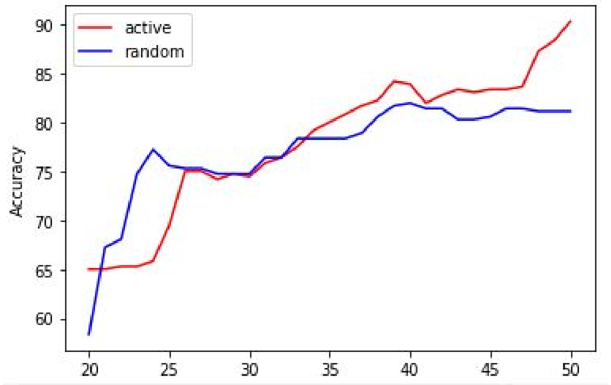


Fig 11: Pool Based Scenario

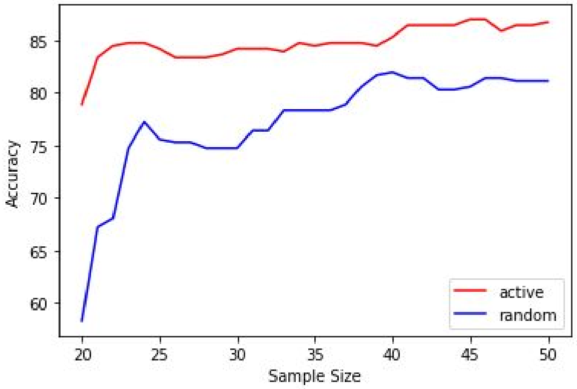


Fig 12: Stream Based Scenario

The performance of pool based is better in this measure also than stream based as is expected. The initial accuracies of both pool and stream-based are higher than the random ones due to the initializations using clustering of diverse sampling.

Cluster-based Strategy

A method is devised on the basis of clustering to query minimal labels and label the remaining unlabelled data. The application of this would be crucial wherever there is a limited budget for a machine learning classifier.

Step 1: Perform K-means Clustering on the training set with a suitable value of K (we used 20).

Step 2: From each cluster i, we query N1(i) labels of its examples.

N1(i) = (no. of examples in cluster i) * (limited budget) / (total no. of examples in training set)

Step 3: We calculate the majority label in each cluster.

Step 4: We label all the points in each cluster with its majority label to produce a training set.

Improvement to Cluster-based Strategy

A slight change is made to the algorithm stated above. In step 3, instead of labelling all the points in each cluster with the majority label, we only label the unlabelled points with this majority label. The labelled points initially queried hence retain their label, even if they are not the majority label in their cluster.

LIMITED BUDGET (30)	**Cluster-based Strategy**	**Improvement to Cluster-based Strategy**
Accuracy with only limited labelled points	0.86112	0.861112
Accuracy after labelling using clustering	0.87778	0.88889

Table 1 : Limited Budget (30 samples)

HIGHER BUDGET (100)	**Cluster-based Strategy**	**Improvement to Cluster-based Strategy**
Accuracy with only limited labelled points	0.94444	0.94445
Accuracy after labelling using clustering	0.90277	0.90278

Table 2 : Higher Budget (100 samples)

Judging from the above tables, we can conclude that the cluster-based strategies are only effective when the budget is low, not high.

Conclusion

Concluding from the results of our implementations, we can say that Active Learning enables machine learning models to learn effectively with less labelled data and is an integral part of today's world of enormous unlabelled data.