Active Learning - When, How and Why it works

Jaideep Murkute
Department of Computer Science, Rochester Institute of Technology

***Abstract*—Today, many of the machine learning applications which perform with very high accuracy on practical use cases, rely on supervised learning strategies, that is they often require large amount of labeled data samples to learn from. Deep learning algorithms are able to learn even more complex phenomena but almost always they require even larger set of labeled data to learn from. However, the task of data labeling is often costly and is not always possible to accurately label large number of data samples. Active learning aims to alleviate this problem to lower the label complexity and still have an equal or in some cases more accurate model for decision making. In this paper, we will review what goes into the building blocks of active learning model, namely, how to choose seed data, how we can structure model to know that it should request data label for some unseen example to learn better and why active learning works. We also perform tests of active learning models to verify if they indeed can provide performance benefit in terms of label complexity and in some cases accuracy for given label complexity. We will also summarize scenarios dependent on either data distribution or structuring of active learning model when active learning may not work or can perform worse than traditional supervised learners and what we can do to avoid such scenarios.**

***Index Terms*—active learning, supervised machine learning, label complexity, optimal experimental design.**

I. Introduction

ACTIVE learning, also known as 'query learning' or 'optimal experimental design'[1] is a sub-field of machine learning which acts as an augmentation of supervised learning algorithms. Active learning frameworks are built upon an idea that rather than treating learning algorithm as a passive learner and hoping that it would converge to good enough local minimum if keep on feeding it more and more data, if we allow the learner to pick which of the data samples it wants to learn from, i.e. it can learn how to learn, the learning process would become much more efficient in terms of number of labeled data samples required, also known as the label complexity.

The reason we care about the label complexity is that, in most of the real world applications, the cost of labeling data is high, the process is time consuming and/or needs human expert to perform this task. Whereas unlabeled data is potentially in abundance. Consider an example of medical data of ECG recordings. Labeling data is essential to draw inference about patient's health. However, labeling the segments in ECG waveform is not the task that can be done by novices, even medical expert would have to do careful annotations for correct labeling and the process in turn gets very costly. If we desire to build accurate machine learning model for this task, we need large number of high quality segment labels. If such data is not abundant, progress of machine learning model would get hindered by the very problem it aims to solve. Active learning approach tries to alleviate these problems by actively looking for the data samples knowing true labels of which can be useful in the learning process and not requesting labels for other samples which do not seem 'interesting' to the learning algorithm.

Active learner would first learn with minimal or available labeled data, also known as the seed data. After learning from labeled data samples, model will make the predictions for unlabeled data samples and if some of the predictions have high uncertainty in them, then such sample can be treated as an interesting one from learner's perspective i.e. by learning the true label of such data sample model can learn some important features of function it's trying to learn. Learner would request the true labels for such data samples from human expert and upon receiving those, can continue with further iterations of learning[2],[1]. Such actively learning model not just understands how much error it made in each prediction and tries to improve its configuration, but it also tries to learn more about the more of the examples where it made mistakes. In essence, such models are learning the data distribution by using capability of how to learn such data distribution and for this reason, process is sometimes also termed as learning to learn.

Such methods have been shown to be beneficial not just in terms of the label complexity, but sometimes, even to get the better overall performance from the model[3]. Intuitive explanation of such behavior in literature often suggests that supervised learning algorithms can get overwhelmed by too much data and they cannot learn much of value from the majority of the data samples. This sometimes can cause them to converge at sub-optimal minimum with more data where better minimum would have been possible with less number of but useful examples. Also probably approximately correct (PAC) learning model [4] showed that it is possible to learn with such minimum number of examples to achieve some given value of the error rate. Later [5] showed interesting experimental results with show similar results, proving the validity of theoretical models of active learning approach. Active learning model will require only as much data as it needs to learn the function with sufficiently high accuracy and avoids the problem of trying to learn not much important data samples. This also results in low computational complexity of the training process.

In this paper, we explore the main approaches that are taken to create such active learning frameworks and apply these strategies on different data-sets to make sure that results we are getting are not caused by some feature of a particular data-set. We also experiment with different machine learning models, to make sure the validity of the active learning approach since, in theory, such frameworks should work irrespective of the type of the learner, as long as model has capabilities to learn in supervised setting and to yield probabilistic outcomes. Section 2 of the paper would review the related works in the field of improving model's performance with limited data and including the field of active learning. In section 3, we review details of the design framework of active learning models. Section 4 contains important details about the implementation approach. Section 5 gives a brief introduction to datasets we are use and the reuslts we get from our experiments with comparison with results in standard supervised learning setting.

II. Related Work

Some of the early theoretical work to establish validity of learning with minimal data samples is with PAC leaning theory[4]. PAC theory suggests that if some data distribution can be perfectly classified with some classifier then to achieve the error rate of ϵ, it is enough to have O($1/\epsilon$) randomly chosen randomly chosen labeled data samples to learn from. This theory can be intuitively examined with toy example with data samples aligned in a straight line and then performing binary search over such space to find the data points that are desired and such toy example can be solved with O(log $1/\epsilon$) data samples. However, more robust results with defined querying framework can be found in the works on Freund et. al [6] with algorithm for Query-by Committee approach which proposes algorithm to be able to gain generalization error of ϵ after seeing O(d/ϵ) examples, where d is the Vapnik-Chervonenkis (VC) dimension[7]. Relatively recent works of Dasgupta et. al [5] propose a modified version of the perceptron learning in active learning setting which shows label complexity bounds to be O($1/\epsilon^2$).

It is worth noting that theoretical results as proposed above do not always show up in practice. Active learning approaches do benefit, sometimes to significant degree, if designed with care but we often we may not observe exponential reduction in requirement of labeled data as some of the theoretical works propose. Settles et. al [8] explains possible reasons including assumptions made in such experiments which do not show up in practice to issues like noisy oracles and varying labeling costs.

A notable and relatively recent results in the applied active learning are well known include work by Tong et. al[9] for use of active learning SVM and Bayesian networks for text classification with reduced label complexity which makes use of pool-based active learning with version-space formulation. This approach with SVM includes uncertainty based sampling by requesting labels for data samples closer to the model's assumed hyper-plane. The area of applied active learning is still being heavily researched, especially on challenging tasks like image classification with deep networks, recent work by Wang et. al [10] shows promising results and merges active learning with deep convolutional networks (CNN) and also proposes framework to reduce the cost of searching for data samples to query. Gal et. al [11] and Kendall et. al [12] show better performing deep learning models, on image classification tasks in terms of label complexity, with Bayesian uncertainty sampling methods of active learning.

III. Design of Active Learning strategy

A. Choosing seed data

Active learning model start off training with some minimal or available amount of available labeled data. This initial dataset is termed as the 'seed' dataset in active learning literature. Careful selection of seed dataset is important since problems may arise if the original data distribution that we are trying to learn is heavily biased to one or more classes. Dasgupta et. al [2] proposes use of clustering techniques to exploit the nature of data distribution and choosing data samples generated from these learned cluster in order to avoid the problem of heavily biased model. For our considerations, we can use uniform sampling since datasets we are trying to learn in the scope of this experiment MNIST-digits [13] and MNIST-fashion [14] do have almost uniform distribution of classes. Further work in seed dataset selection, if needed for Forest Type dataset[15] will be handled during implementation tasks.

B. Scenarios of Active Learning framework

Scenarios include different settings in which an active learner will be able to ask queries and commonly referred scenarios in the literature are Membership Query Synthesis, stream based sampling and pool based sampling[16].

(i) *Membership Query Synthesis*: In this scenario, learner when needs to learn more about a particular data sample, will generate some new examples by performing a few simple data augmentation tasks such as rotation, flipping etc. and will ask human annotator to label these samples [17], [16]. Such approaches can be efficient in simple and finite space problems but task of annotating these new samples can get difficult for human expert with increased complexity and can fail in tasks such as natural language processing.

(ii) *Stream based sampling*: In such design [18] , human experts or external program will randomly or intelligently choose samples from set of unlabeled samples and will allow learner model to determine if it needs label for that sample or not. In such scenario, it's assumed that getting unlabeled data samples is free. Such approach can reduce the labeling cost significantly. Care needs to be taken if data distribution is non-uniform since it would not provide model enough opportunity to choose examples from 'rare' clusters.

(iii) *Pool based sampling*: Proposed by Lewis et. al [19] , This is the most widely used scenario of active learning and differs from Stream based sampling in way that it allows learner to select from unlabeled data samples to query label of. This approach also assumes that we have a pool of unlabeled data instances for model to choose from. We are designing our experiments in this scenario.

C. Sampling Strategies

Active learning literature proposes many different sampling strategies but the Uncertainty based sampling strategies are widely used and often help to gain desired behavior of the model. Uncertainty based sampling methods look at the probabilistic prediction values of data instances which are unlabeled and would pass the ones for which model is uncertain about to be annotated by human expert. There are three main types of Uncertainty based sampling as discussed below:

(i)*Least Confidence (LC)*: Such strategy would look at the probability assignments for different classes for different data samples and then pick the ones for which model was most uncertain about i.e. probability values were closely distributed [2]. Formulation for Least Confidence measure can be given as:

$$x_{LC} = argmax_x(1 - P(\hat{y}|x))$$

, where $\hat{y}$ are prediction values for sample x.

(ii)*Margin Sampling*: Least confidence strategy can sometimes miss the data samples where learner was not quite sure about the classification among two or few of the possible classes. Such instances can still pass through the least confidence test but learner can definitely benefit from true label in such cases.[2]

$$x_M = argmin_x(P(\hat{y_1}|x) - P(\hat{y_2}|x))$$

, where $\hat{y_1}$ and $\hat{y_2}$ are the most probable class values for each data sample predictions.

(iii)*Entropy Sampling*: This is the commonly used strategy since it does not rely on the specific label values or difference between label prediction probabilities, but computes overall entropy of the prediction, which is desired to be low. If entropy value summed up across classes for some data sample exceed certain threshold, such strategy would have learner request the label for that data samples [2].

$$x_H = argmax_x - \Sigma P(\hat{y_i}|x)log(P(\hat{y_i}|x))$$

, where $\hat{y_i}$ are prediction values over all possible classes.

Another widely used strategy is Query by committee[20] which relied on the maximal disagreement decision i.e. Data samples where most of the classifiers which can be in ensemble or in one versus all setting, disagree on the classification decision.

IV. Implementation

We use MNIST handwritten digits and MNIST Fashion datasets to test the theory behind the active learning. Details about these datasets have been specified in the next section. We use Logistic Regression as a supervised learning model and test the performance of the model in terms of accuracy versus label complexity. Similar results can be expected with different supervised learning techniques. For implementation, we use scikit-learn[21] library functions. The two datasets mentioned above have 10 classes in them. For this purpose, we use ten logistic regression models, each fitting to one of the classes in one versus all manner and making classification decision by majority voting.

Enabling model with active learning requires additional functions to select data points from a pool of unlabeled data based on the model's prediction results. For this, we are using alp repository[22]. This library provides functions for ranking data points based on sampling strategy chosen - least confidence, entropy sampling or query by committee by majority voting.

We first sample seed data in from shuffled dataset in uniform sampling manner, since datasets under consideration have balanced class distributions. Train both active learning and standard supervised learning logistic regressors(10 each) on this data and then feed in new data points incremently to these models. For new points, active learning models will be fed in with the top k data points ranked by the sampling strategy score, where k is the batch size. Passive learners will be fed in with random data points sampled uniformly from pool of unlabeled data. We monitor the accuracy of both of these models on test datasets after each new iteration of models fitting on newly available labeled data.

In real world scenarios, it is possible to create an interactive setting in which model, where models will be trained on all the labeled data available and then when model requets new data points to be labeled, expert can come in and label these points. For experimental setting, although we have all of the data points labeled, we assume that only 1 percent of total available data is labeled and treat other data as unlabeled data. When model needs labels for 'unlabeled' data points, it will lookup into labels data.

Previous plan for testing was on one more datasets and with multiple different models. However, given the time required to train models in multiple settings, we are restrciting to the logisitc regression. Althouth early tests with convolutional neural networks also showed similar performance of active learning models.

Both active and passive learning models have the same configurations which for logistic regression are number of iterations, optimization method and convergence criterion.

V. Datasets and Results

First data-set is well-known MNIST handwritten digit dataset[13] which has 10 classes with 70,000 gray-scale images of size 28*28 and the task is to predict the class or digit written in the image. Second dataset is MNIST-Fashion dataset[14] which maintains the same structure as MNIST digits dataset with 70,000 examples and 28*28 grayscale images of clothing items with 10 classes, but is more challenging than MNIST-digits dataset, which allow us to compare active learning models with robust supervised learning models and on more challenging classification task.

Results for all such tests have been listed in tables below. 'Random' columns enlist the results with standard supervised models where data samples are fed to the model randomly. 'Active' column is performance when model actively chooses data sample to learn from. For comparison, we choose accuracy values at cut-off, where for given uper limit on number of data samples to perform test on, one of the models tends to converge. We test performance of three diference query strategies, least confidence, entropy sampling and majority voting query by committee.

TABLE I
MNIST HANDWRITTEN DIGIT DATASET - ACCURACY

Sampling Strategy	Passive Learner	Active Learner
Least Confidence	83 %	86%
Entropy Sampling	82 %	85%
Query by Committee	83%	84%

TABLE II
MNIST HANDWRITTEN DIGIT DATASET - LABEL COMPLEXITY

Sampling Strategy	Passive Learner	Active Learner
Least Confidence	1000	420
Entropy Sampling	1000	450
Query by Committee	1000	250

Label complexity compares the number of data samples required by active learner to achieve the same level of accuracy as passive learner.

Below plots show the results on test set, as listed above, during training.

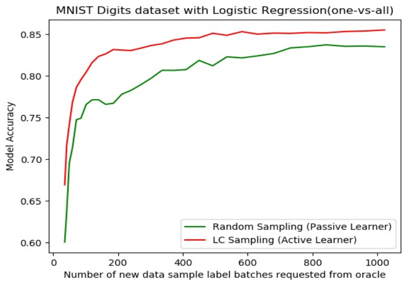

Fig. 1. Comparison on MNIST digits dataset with Least Confidence sampling for active learner

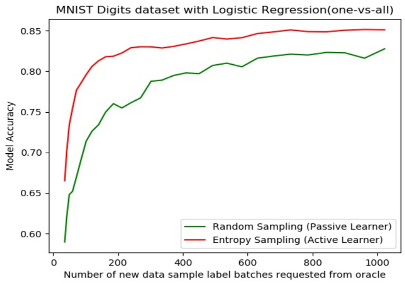

Fig. 2. Comparison on MNIST digits dataset with Entropy sampling for active learner

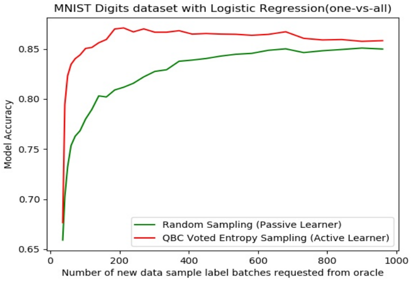

Fig. 3. Comparison on MNIST digits dataset with QBC sampling for active learner

TABLE III
MNIST FASHION DATASET - ACCURACY

Sampling Strategy	Passive Learner	Active Learner
Least Confidence	77%	74%
Entropy Sampling	78%	71%
Query by Committee	78%	76%

TABLE IV
MNIST FASHION DATASET - LABEL COMPLEXITY

Sampling Strategy	Passive Learner	Active Learner
Least Confidence	500	220
Entropy Sampling	1000	220
Query by Committee	1000	250

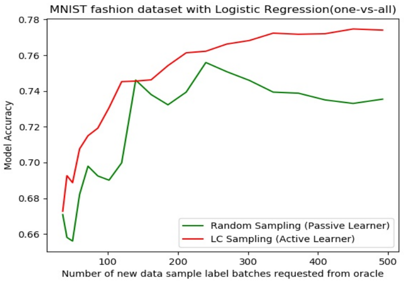

Fig. 4. Comparison on MNIST Fashion dataset with Least Confidence sampling for active learner

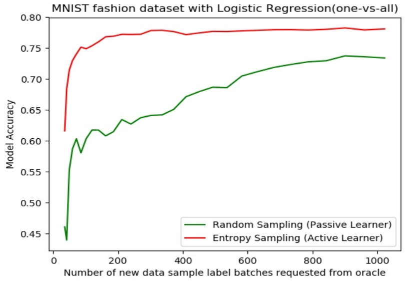

Fig. 5. Comparison on MNIST Fashion dataset with Entropy sampling for active learner

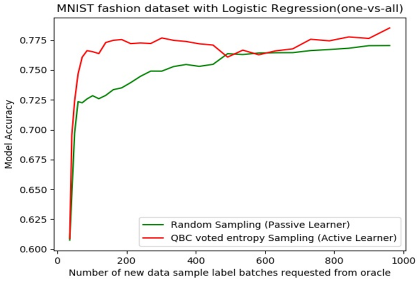

Fig. 6. Comparison on MNIST Fashion dataset with QBC sampling for active learner

VI. Conclusion

We observe that active learning can indeed help to reduce the label complexity of the supervised learning models to achieve the same level of accuracy. Active learning models by selecting 'interesting' data samples to learn from can achieve better performance than passive learners. Future enhancements in the above implementations are possible to mitigate the effect of drop in performance of active learning models as training progresses, which can happen when data samples that model is learning from gets biased because of lack of samples from a particular class requested by model, which model believes it has learned but in reality can make mistakes in predictions and which can be handled by active re-sampling methods.

References

[1] F. Olsson, "A literature survey of active machine learning in the context of natural language processing," SICS-T–2009-06–SE.pdf, April 2009. [Online]. Available: http://soda.swedish-ict.se/3600/1/SICS-T–2009-06–SE.pdf

[2] S. Dasgupta and J. Langford, "MATE: MPLS a tutorial on active learning," International Conference on Machine Learning, 2009.

[3] Z. G. D. Cohn and M. Jordan, "Active learning with statistical models," Journal of Articial Intelligence Research, 1996.

[4] L. G. Valiant, "Active learning with statistical models," Journal of Articial Intelligence Research, 1984.

[5] A. T. K. S. Dasgupta and C. Monteleoni, "Analysis of perceptron-based active learning," Journal of Machine Learning Research, 2009.

[6] E. S. Y. FREUND, H. SEBASTIAN SEUNG and N. TISHBY, "Selective sampling using the query by committee algorithm," Machine Learning, Chapter 28, Pages 133168, Kluwer Academic Publishers, 1997.

[7] V. N. VAPNIK and A. Y. CHERVONENKIS, "On the uniform convergence of relative frequencies of events to their probabilities," THEORY OF PROBABILITY AND ITS APPLICATIONS, Volume XVI, Number 2, 1971.

[8] B. Settles, "What uncertainties do we need in bayesian deep learning for computer vision?" JMLR: Workshop and Conference Proceedings - Workshop on Active Learning and Experimental Design, 2011. [Online]. Available: https://arxiv.org/pdf/1703.04977.pdf

[9] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," Journal of Machine Learning Research, 2001.

[10] D. Z. K. Wang and L. Lin, "Cost-effective active learning for deep image classification," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 2017. [Online]. Available: https://arxiv.org/pdf/1701.03551.pdf

[11] R. I. Y. Gal and Z. Ghahramani, "Deep bayesian active learning with image data," 2017. [Online]. Available: https://arxiv.org/pdf/1703.02910.pdf

[12] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" Thirty-first Conference on Neural Information Processing Systems, 2017. [Online]. Available: https://arxiv.org/pdf/1703.04977.pdf

[13] C. C. Chris Burges, Yan LeCun, "The mnist database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[14] Zalando, "A mnist-like fashion product database-version4," 2017. [Online]. Available: https://research.zalando.com/welcome/mission/research-projects/fashion-mnist/

[15] R. Sensing and C. o. N. R. GIS Program, Department of Forest Sciences, "A mnist-like fashion product database-version4," 1998. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/covertype

[16] B. Settles, "A literature survey of active machine learning in the context of natural language processing," SICS-T–2009-06–SE.pdf, April 2009. [Online]. Available: http://soda.swedish-ict.se/3600/1/SICS-T–2009-06–SE.pdf

[17] D. ANGLUIN, "Queries and concept learning," Machine Learning, 2, Pages 319-342, July 1988. [Online]. Available: https://link.springer.com/content/pdf/10.1023/A:1022821128753.pdf

[18] D. P. D. Cohn, L. Atlas, "Training connectionist networks with queries and selective sampling," Neural Information Processing Systems, 1990.

[19] D. Lewis and W. Gale, "A sequential algorithm for training text classifiers," ACM SIGIR Conference on Research and Development in Information Retrieval, 1994.

[20] M. O. H. S. Seung and H. Sompolinsky, "Query by committee," Proceeding COLT '92 Proceedings of the fifth annual workshop on Computational learning theory, 1992.

[21] "scikit-learn machine learning in python." [Online]. Available: https://scikit-learn.org/stable/

[22] "Active learning in python." [Online]. Available: https://github.com/davefernig