Active Learning from Data

Submitted as part of the requirements for:
CE888 Data Science and Decision Making

Author: Ashkan Mohseni Hosseini

Lecturer: Dr Ana Matran-Fernandez

Date: 01 March 2020

Abstract: there are many situations in machine learning where obtaining labelled data is either expensive or it requires a human annotator to manually label each instance. Many datasets come with a mix of labelled and unlabelled instances, in many cases with much more unlabelled instances than labelled ones. Given the cost of labelling or the time that it takes, a techniques called active learning can be used to first train a model on labelled data, and then have the algorithm strategically choose the next instance that it likes to be labelled. This way, good performance can be achieved by training on a much smaller set of labelled data. In this project we will apply active learning to two datasets, a classification task and a regression task. The modAL framework developed by Tivadar Danka [1], which has a comprehensive set of modules for implementing active learning, is used to apply active learning to the aforementioned datasets. The code for this assignment can be found in git hub from the following link: https://github.com/Ashkan66/CE888Assignment.git

Contents

Introduction

Active learning is an area of machine learning in which the learning algorithm has the capability to analyse the unlabelled set of instances and decide which instance to query next in order to maximise learning. The queried instance is then labelled by an annotator, which in most literatures is referred to as the Oracle. Active learning is also referred to as "query learning" or optimal experimental design in the statistics community. Most machine learning algorithms, especially in the area of deep learning, need thousands or more labels to effectively learn [2]. In some cases, obtaining labels for these instances is easy, however there are other cases where obtaining labels is either difficult, expensive or time consuming. For instances obtaining labels for spam emails or ratings for media such as films or music is cheap, given that there are many people rating films or marking emails as spam every day. However, there are other cases where that is not the case, such providing labels for speech in a speech recognition supervised learning task or information extraction where detailed annotations need to be provided for documents. Therefore, active learning can be used in these situations where there is an abundance of unlabelled data.

An active learner can query instances under a few different scenarios. There are also several different query strategies which can be used to query new instances. These different scenarios and query strategies will be discussed in the upcoming sections of this report. In general, there is a small pool of labelled data $\mathcal{L}$ which is used in the initial learning. The algorithm will then query new instances from a large pool of unlabelled data $\mathcal{U}$. After having queried the new instance, The instance is added to $\mathcal{L}$ and it will then leverage its new knowledge to query more instances. We can query a set number of instances or continue querying until a desired performance has been achieved.

As an example to illustrate the power of active learning, Settles [2] gives a very intuitive example of active learning. In this example, a dataset is produced using two Gaussians cantered at (-2,0) and (2,0) with $\sigma = 1$. Each Gaussian represent a different class distribution as shown in figure 1(a). in figure 1(b), 30 instances are chosen at random to be used in a traditional supervised learning task without active learning and using a logistic regression algorithm. The blue line in figure 1(b) (adapted from Settles [2]) is the decision boundary obtained. The accuracy achieved in this scenario is 0.7. Figure 1(c) on the other hand, shows the decision boundary achieved using logistic regression with active learning. In this scenario, active learning is used to query instances closes to the decision boundary, i.e. where the decision is most uncertain, rather than from places where the label is redundant or more certain. This way, an accuracy of 0.9 can be achieved with the same number of labelled instances.

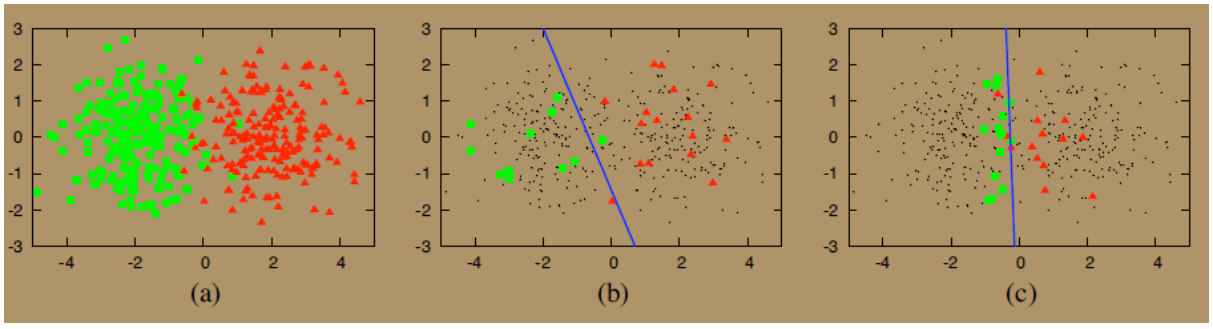

Figure 1. a dataset produced using two Gaussian distributions. in (b) a logistic regression is used without active learning to produce a decision boundary, where as in (c), logistic regression with active learning is used to create a decision boundary

We will then look at our results obtained from applying active learning to our two datasets and analyse the results and finally draw some conclusions from it.

Scenarios

As mentioned earlier, there are several different problem scenarios in which active learning can be carried out. The three main categories of scenarios are stream-based selective sampling, membership query synthesis, and pool-based active learning.

Membership Query Synthesis

Membership query synthesis was first proposed by Angluin [3]. In this scenario the learner request a label from the set of unlabelled instance. It could also request labels that are generated de novo. In finite problem domains, generating instances is often efficient and tractable [3]. However, in problems where instances come from some underlying natural distribution, this approach can produce instances that are problematic for the oracle to label. For example, Baum et al [4] used membership query synthesis in a character recognition problem in which this scenario created characters that could not be recognised by the human annotator. Generally, in domains where the labels come from non-human annotators, such as scientific experiment results etc. this scenario may be a promising approach for scientific discovery.

Stream-Based Selective Sampling

Another approach or scenario is stream-based elective sampling, first proposed by Cohn et al [5]. One important assumption in this approach is that acquiring unlabelled samples is free or low-cost. This way an instance can first be sampled and then the leaner can decide as to whether request its label or not. In this approach, instances are sampled on at a time, hence the name stream-based. There are several approaches when deciding to whether query an instance or not. In the first approach the sample is evaluated using some usefulness measure or some query strategy, as will be discussed in more detail in the next section. This information is then used to increase the probability of certain instance being drawn from $\mathcal{U}$ as illustrated by Dagan et al [6]. Cohn et al [5] use a different approach, in which they calculate the most uncertain regions in the underlying distribution of $\mathcal{U}$ which, put another way, is the region of the distribution which the learner is most uncertain about. Instances are then chosen only from that region. This method is widely used in the literature such as worked carried out by Dagan et al [6] in which they use this scenario in part-of-speech tagging, in sensor scheduling by Krishnamurthy [7], and in SVM selective sampling in information retrieval by Yu [8].

Pool-Based Active Learning

In this active learning scenario, the learner scans through the entire unlabelled dataset and then makes a decision regarding the most informative instance, as opposed to stream-based scenario. Pool-based active learning seems to be the dominant scenario used in the literature [2]. However, on platforms where computing power is limited, stream-based active learning would have an advantage over pool-based. Pool-based active learning has been applied to many real-world machine learning problems such as speech recognition (Tur et al [9]), cancer classification (Lui [10]) and object detection (Tong and Chan [11]). This is also the approach taken in this project.

Query strategies

To choose the instance to be queried for its label, there needs to be a measure of informativeness or usefulness that the algorithm uses to select the most suitable instance. There is a wide variety of query strategies used in the literature. In this section we will list some of them and we will explore the ones used in this project in more details. From this point, we will refer to the most informative instance as x_Q^*, where Q is some query selection algorithm.

Uncertainty Sampling

Uncertainty sampling is one of the most widely used query strategies in classification tasks. This query strategy was first proposed by Lewis and Gale [12]. in uncertainty sampling, the instance about whose label the algorithm is least certain will be queried. This can be formulated using the concept of entropy from information-theory as follows:

$$x^*_{ENT} = \underset{x}{\text{argmax}} - \sum_i P(y_i|x;\theta)\log P(y_i|x;\theta)$$

Where y_i is a particular label. This approach can be generalised to be applied to more complex data structures such as sequences (Settles et al [13]) and trees (Hwa et al [14]).

Query-by-committee

In Query-by-committee (QBC) (Seung et al [15]) a collection of models are trained on $\mathcal{L}$ with competing hypothesis. Then the most informative instance is the one for which there is the most disagreement among the trained models. The idea in QBC is to minimize the version space of the models or hypothesis. A version space is the set of hypothesis that are consistent with the labelled dataset, and QBC could be viewed as an attempt to make this version space as small as possible. There are two steps in implementing QBC. First a committee of models are constructed such that they each represent a different region of the version space. Secondly, some measure of disagreement needs to be implemented [2]. There are generally no agreement regarding the optimal size of the committee [2] which could vary based on the application or model class. However, worked carried out by Seung et al [15], McCallum et al [16] shows that committee sizes as small as two could work well in practice. In this project, we also use only three committee.

There are two main approaches for measuring the level of disagreement [2]. The first on is known as vote entropy proposed by Dagan et al [6] and the other approach is called Kullback-Leibler (KL) divergence

Expected Model Change

Expected model change queries the instance which would cause the greatest change to our model if we already knew its label. Put another way, this algorithm prefers instances that have the greatest impact on its parameters. This query strategy has been introduced by Settles et al [13]. Discriminative probabilistic models can use a variant of this approach known as expected gradient length. This approach has been shown to work well in practice [2], however if the feature space and the labelled dataset is large, this algorithm becomes computationally expensive.

Fisher Information Ratio and Variance Reduction

Another approach used to query instance is using statistical analysis in which instances are chosen based on how much reduction in learner's future error by reducing its variance. However the aforementioned technique can only be used for regression tasks. However, Zhang et al [17] have proposed an approach which can be used to query classification tasks using Fisher information. These two approaches can be grouped together under a more general framework which Settles [2] calls variance minimisation. However, these methods are not well suited for models with a high number of parameters as the time complexity of these models is $O(k^3\mathcal{U})$ [2] where K is the number of parameters and $\mathcal{U}$ is the size of the unlabelled set.

Estimated Error Reduction

In the previous section, we saw that we one can use model variance to select the next instance, However, this cannot be done for all models in closed form. We can instead calculate an estimation of the future error resulting from labelling a new instance and adding it to $\mathcal{L}$, and then the instance that minimises that expectation is chosen. Zhu et al [18] have shown that using this approach coupled with semi-supervised learning can dramatically improve performance compared to uncertainty sampling. However this query strategy might be the most expensive strategy out of all of the ones discussed so far [2].

Experimentation and Methodology

We have selected two datasets from the UCI repository [19]. The datasets are chosen strategically so that most of the models present in the modAL framework can be used. We have chosen one dataset containing a regression task and the other one a classification task. We have used python as the programming language for this assignment and Jupyter notebook is used as the development environment. in the following sections we will delve more into the details of each dataset and lay out the results obtained.

Active Learning on Classification tasks

In the first active learning task, a classification problem is considered. The dataset comes from a motion capture camera system [20] and the task is to classify the hand gesture based on the given coordinates. A Vicon motion capture camera system was used to record 12 users performing 5 hand postures with markers attached to a left-handed glove. A rigid pattern of markers on the back of the glove was used to establish a local coordinate system for the hand, and 11 other markers were attached to the thumb and fingers of the glove. 3 markers were attached to the thumb with one above the thumbnail and the other two on the knuckles. 2 markers were attached to each finger with one above the fingernail and the other on the joint between the proximal and middle phalanx. In total there are 38 features and 5 classes.

The data set is balanced, meaning there are nearly identical number belonging to each class and hence accuracy measure was chosen as the performance metric. Given the nature of the data and through some trial and error, we concluded that only the first 9 features were enough to achieve a high accuracy score. First the dataset is shuffled. Next, features are standardised using the StandardScaler class of the sklearn module. The data is then divided into train set, test set and a pool set. 0.5% of the whole dataset was used for training. 1% is used for testing and the rest is used for the pool set (which is used to query labels for instances chosen by the active learner.

First, a random forest classifier is trained on the training set to obtain the accuracy scores. Then three different query strategies are used to obtain the result of applying active learning using 200 samples drawn from the pool. The results obtained are as follows.

Query strategy	accuracy
Before applying active learning	62%
Uncertainty Sampling	69%
Margin Sampling	69%
Entropy Sampling	67%

Table 1. accuracy score obtained for different query strategies

As can be seen from table 1 there is an increase of 11 percent in accuracy.

We then used the committee based active learning with the above data with three different query strategies, namely, vote entropy sampling, and consensus entropy sampling. The algorithms used in the committee are random forest classifier and logistic regression. The results are shown in table 2. As can be seen from table 2, there is an improvement n accuracy score of about 11%.

Query strategy	accuracy
Before applying active learning	62%
Vote entropy Sampling	69%
Max disagreement Sampling	70%
Consensus entropy Sampling	68%

Table 2. accuracy scores for committee classification

Active Learning on Regression Tasks

Next we tried active learning for a regression task. The major difference between the two is that the queries used in the last section do not work for regression tasks. Therefore, we need other query strategies. we also need to use a learning algorithm that returns a standard deviation for each instance. To that end, we have used a Gaussian process regressor from sklearn library. we first use a none-committee approach with three different query strategies, namely maximum expected improvement, maximum probability of improvement, and highest confidence upper bound. We then use a committee base approach in which we use two algorithms. Both algorithms are a Bayesian process regressor, however, they each have different kernels.

Our dataset is obtained from UCI and it can be downloaded from [21]. There are 5 features and 1 output. All output and inputs are real numbers. Just as was the case with the last dataset, we first shuffle the dataset and then standardise the features. We use 0.5% of the data for training, 1% for testing and the rest as a pool from which the active learning draws the instances. In all active learning cases, 100 instances are queried. The performance metric used in this task is the mean squared error. The following table shows the results obtained.

Query strategy	accuracy
Before applying active learning	118
Max EI	73
MAX PI	73
MAX UCB	70
Committee based learning	84

Table 3. Performance comparison of different approaches in the regression task with active learning

As can be seen from table 3 there is an average decrease of MSE of about 39% for non-committee approaches and a decrease of 29% for the committee-based approach.

Conclusion

As we have observed form the work carried out in this assignment. Active learning can improve the performance with fewer datapoints as opposed to not using active learning. However, in both learning tasks, committee-based approaches yielded inferior results compared to non-committee-based

approaches. There could be a few reasons for this. First, we might need to use more models in the committee to achieve better results. But also, it might help if each model in the committee concentrated in a separate region of the data. furthermore, in the case of the regression task we have used the same learning method, namely Gaussian process regressor. It might be better to use different models. The reason we have used only Gaussian process regressor in this assignment is because it is one the few algorithms that returns the standard deviation for each instance.

References

[1] T. Danka. "modAL: A modular active learning framework for Python3." https://modal-python.readthedocs.io/en/latest/index.html (accessed 10/02/2020, 2020).

[2] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, 2009.

[3] D. Angluin, "Queries revisited," *Theoretical Computer Science,* vol. 313, no. 2, pp. 12-31, 2004.

[4] E. B. Baum and K. Lang, "Query learning can work poorly when a human oracle is used," in *International joint conference on neural networks*, 1992, vol. 8, p. 8.

[5] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning,* vol. 15, no. 2, pp. 201-221, 1994.

[6] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Machine Learning Proceedings 1995*: Elsevier, 1995, pp. 150-157.

[7] V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden Markov model sensors," *IEEE Transactions on Signal Processing,* vol. 50, no. 6, pp. 1382-1397, 2002.

[8] H. Yu, "SVM selective sampling for ranking with application to data retrieval," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 354-363.

[9] G. Tur, D. Hakkani-Tür, and R. E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Communication,* vol. 45, no. 2, pp. 171-186, 2005.

[10] Y. Liu, "Active learning with support vector machine applied to gene expression data for cancer classification," *Journal of chemical information and computer sciences,* vol. 44, no. 6, pp. 1936-1941, 2004.

[11] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the ninth ACM international conference on Multimedia*, 2001, pp. 107-118.

[12] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR'94*, 1994: Springer, pp. 3-12.

[13] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 1070-1079.

[14] R. Hwa, "Sample selection for statistical parsing," *Computational linguistics,* vol. 30, no. 3, pp. 253-276, 2004.

[15] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 287-294.

[16] A. K. McCallumzy and K. Nigamy, "Employing EM and pool-based active learning for text classification," in *Proc. International Conference on Machine Learning (ICML)*, 1998: Citeseer, pp. 359-367.

[17] T. Zhang and F. Oles, "The value of unlabeled data for classification problems," in *Proceedings of the Seventeenth International Conference on Machine Learning,(Langley, P., ed.)*, 2000, vol. 20, no. 0: Citeseer, p. 0.

[18] X. J. Zhu, "Semi-supervised learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, 2005.

[19] "Machine Learning Repository." https://archive.ics.uci.edu/ml/index.php (accessed 10/02/2020, 2020).

[20] "MoCap Hand Postures Data Set." UCI. https://archive.ics.uci.edu/ml/datasets/MoCap+Hand+Postures (accessed 10/02/2020, 2020).

[21] NASA. "Airfoil Self-Noise Data Set." UCI. https://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise (accessed 10/02/2020, 2020).