Batch Acquisition for Deep Bayesian Active Learning with Imperfect Oracles

Pushkar Kolhe PUSHKAR@GATECH.EDU
Department of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA

Abstract

Active Learning is used to maintain accuracy while reducing the training set size in many machine learning applications. However active learning approaches are not yet common in practice because they make a strong assumption on the quality of labeled data from an oracle. For machine learning applications whose goal is to estimate a model with very high confidence, we propose a framework for querying data in active learning that works with noisy oracles. In this framework we extend BatchBALD (Kirsch et al., 2019) to create a batch query with a control example and multiple informative examples for the task of deep Bayesian active learning. This allows us to infer the proficiency of the labeler and associate a confidence estimate while using their labels.

d

Keywords: Active Learning, Applied Machine Learning, Human-Computer Interaction

1. Introduction

Active Learning is a family of machine learning methods which may query the data instances to be labeled for training by an oracle (e.g., a human annotator). These methods can achieve comparable performance to passive learning methods with fewer labeled examples. Active Learning is a promising approach to improve many machine learning applications, but is is rarely used in practice due to practical challenges. One of the strong assumptions of earlier active learning methods is that the oracle is perfect, however that is often not true. Even if labels come from human experts, they may not always be reliable: (i) some instances are implicitly difficult for both people and machines, and (ii) people can become distracted or fatigued over time, which introduces variability in the quality of their annotations. With the recent introduction of services like Mechanical Turk this problem has become even more relevant.

There are still many open research opportunities along these lines. In particular, how might the effect of payment influence annotation quality (i.e., if you pay a non-expert twice as much, are they sufficiently motivated to be more accurate)? What if some instances are inherently ambiguous regardless of which annotator is used, so repeated labeling is not likely to improve matters? In most crowd-sourcing environments, the users are not necessarily available "on demand," thus accurate estimates of annotator quality may be difficult to achieve in the first place, and might possibly never be applicable again since the model has no real choice over which to use. In this paper we do not optimize active learning with

respect to cost. We show how our labeling scheme can be used with imperfect oracles for active learning problems.

2. Related Work

One way to think about the problem of non-experts is agnostic active learning Balcan et al. (2009), a framework which relaxes the assumption that the oracle's labels are trustworthy, yet still has positive theoretical results. Other recent work assumes that a learner may repeat queries to be labeled by multiple annotators Sheng et al. (2008). They analyze the different strategies that could be used for re-labeling. This introduces another interesting research issues. When should the learner decide to query for the (potentially noisy) label of a new unlabeled instance, versus querying for repeated labels to de-noise an existing but suspicious training instance? How can the learner even decide that the quality of a label is suspect?

In the context of neural networks, Gupta et al. (2019) show how a denoising layer can be added to make active learning robust to label noises. Another approach more recently has been in designing an end-to-end framework such as Platanios et al. (2020) where a new loss function is introduced that includes instances difficulties and predictor competencies (we use the word proficiency instead of competency in our paper). The authors claim that due to their framework annotators are assigned to instances they are more likely to label correctly while performing crowdsourcing. Both these approaches introduces bias in the learning process, either by introducing a change in the model or adding new parameters in the loss function. To their credit they provide various examples and their results shows promise. In our approach we separate the process of learning from the labeling process, which means that in our approach the learning process is not biased by the actual labeling. In a practical setting someone could choose to label multiple batches together before performing one active learning loop to update the model. In our view separating the active learning and labeling process makes our approach more practical.

Ipeirotis et al. (2014) goes through the basics of repeated labeling and show that it is indeed useful. In their paper they discuss basic repeated labeling strategies such as majority voting, fixed round robin strategy with and without costs and selective repeated labeling. They also show that incorrectly labeled examples tend to have higher model uncertainty scores, compared to correctly labeled examples. They also introduce soft labeling and weighted labeling in the context of active learning. They conclude in their paper that selective repeated-labeling is preferable after taking into account both labeling uncertainty and model uncertainty. This conclusion has heavily influenced our approach.

3. Background

In this paper we focus on Bayesian Neural Networks and the MNIST example used by Kirsch et al. (2019). Compared to regular neural networks, bayesian neural networks maintain a distribution over their weights instead of point estimates. Since exact inference in bayesian neural networks is intractable, we use a variational approximation such as MC dropout (Gal and Ghahramani, 2016). In Bayesian Neural Networks model uncertainty can be measured by an acquisition function like the Batch-BALD acquisition function and label uncertainty

can be measured as the entropy of the labels for the data in our unlabelled pool. In our experiments we repeatedly go through active learning loops, that is we reinitialize the model on the available labelled data and the new data acquired after the labeling procedure. We also keep the dropout masks in MC dropout consistent while sampling from the model.

Our approach is most similar to the Completely Automated Public Turing test to tell Computers and Humans Apart (reCAPTCHA) system Von Ahn et al. (2008). It is a challenge response system used online to determine whether a user is a human or a computer. In it, users are asked to transcribe images into words. The reCAPTCHA system specifically challenges the user with two images. In its most simplest instantiation the system knows the trascribtion of an image, but does not know the trascribtion of the other image. When the user answers the challenge, the system verifies if the user was indeed an human from the known trascribtion and then uses the other answer as the trascribtion of the other image. In further sections we show how this labeling idea can be applied to the framework of active learning in Bayesian Neural Networks.

3.1 Problem Setting

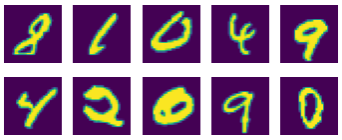
We borrow and modify the problem setup from Kirsch et al. (2019). The Bayesian active learning setup consists of an unlabelled dataset D_{pool}, the current training set D_{train}, a Bayesian model M with model parameters $\omega \sim p(\omega|D_{train})$, and output predictions $p(y|x, \omega, D_{train})$ for data point x and prediction $y \in \{1, ..., c\}$ in the classification case. The model has been trained on D_{train} and the oracle labels a data point in the unlabelled pool $x \in D_{pool}$. The goal is to obtain a certain level of prediction accuracy with the least amount of oracle queries.

At each acquisition step, a batch of data points $\{x_1^*, ..., x_b^*\}$ is selected using the BatchBALD acquisition function which scores the candidate batch of unlabelled data points $\{x_1^*, ..., x_b^*\} \in D_{pool}$ using the current model parameters $p(\omega|D_{train})$:

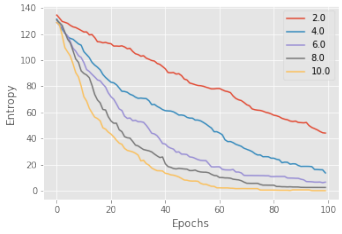
$$\{x_1^*, ..., x_b^*\} = \underset{\{x_1^*,...,x_b^*\} \in D_{pool}}{\arg\max} \sum_{i=1}^{b} I(y_i; \omega | x_i, D_{train}) \tag{1}$$

Here I is the joint entropy between the data points and the model parameters as given in Kirsch et al. (2019). The acquisition function (1) requires us to test each data point using MC dropout to get the posterior for each data point. When the oracle labels on of these points it can be added to the training set. This strategy makes sense when the oracle is always correct. In this paper we relax this assumption, so we need a new strategy to gather labels.

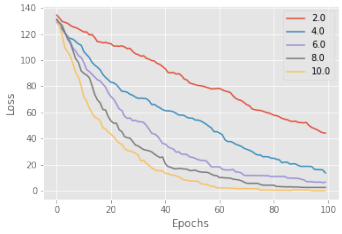
We first select a candidate batch according to (1). Then we run multiple repeated labeling events with different labelers. In each of this event we select some control queries and other candidate queries from the BatchBALD batch. The labelers proficiency is evaluated by comparing their labels with the control queries' labels. This proficiency score is then used while updating the posterior probabilities of the candidate queries. See Figure 1a for an example.



(a) An example candidate query for labeling. Control data points are on the top.



(b) Entropy decreases as a the number labels per data point increases



(c) Loss converges quickly when more labels are used per data point

Figure 1

3.2 Updating from labelers feedback

The oracle is imperfect. So for updating the posterior probabilities of the labels of a candidate point we can use the Bayes Rule. It allows us to add a prior to the labels and the confidence in the labelers labels. Using the Bayes rule

$$P(C|D) = \frac{P(D|C)P(C)}{\sum_{C \in \boldsymbol{C}} P(D|C)P(C)} \tag{2}$$

where $\boldsymbol{C}$ is the set of all the labels of C and D is the proficiency of the labeler. When multiple labelers have labeled a candidate data point, its posterior is updated using this method.

Consider the case of a binary classification problem. If the prior is uniform on the binary labels and the labeler has the worst proficiency (the worst proficiency value is 0.5 which means that the labeler randomly chooses a label), the posterior calculated using (2) has the desired effect that the label uncertainty or entropy never decreases. However if the prior is not uniform or the labeler is slightly better than labeling at random, entropy of the labels will decrease with every repeated label as shown in Figure 1b.

3.3 Modeling proficiency of the oracle

In the most crudest sense, a labeler can be thought of as a classification model. The proficiency of the labeler then can be modeled by their accuracy. This works well in practice. In our experiments though we use the cross entropy between the oracle's labels and the predictions of the model as the proficiency score. It has the additional benefit of taking into consideration the uncertainty of the labels while finding proficiency. This allows us to use data from the pool set as control data points for whom we might not know the true labels while creating a candidate query.

$$p = H(y_i, \hat{y}_c) = \sum_{i=1}^{c} y_i log \frac{1}{\hat{y}_c} \tag{3}$$

3.4 Extending BatchBALD algorithm

In this labeling approach we do not restrict ourselves to acquiring labels for a batch once before retraining our model. Here we get a new candidate batch from our pool dataset and gather repeated labels for it until they are below a certain entropy. After that we add it to our training set and run the active learning loop, that is we reinitialize the model using the original training set and the new acquired data after repeated labeling.

After the new model is learned, we then rerun the MC dropout to get a new candidate batch using the BatchBALD acquisition function and repeating the process. Overall this approach allows us to control the quality of the labeled data that goes into training. If the current batch is not satisfactorily labeled, it can be discarded without affecting the learning process. In this way the active learner can build the model with high confidence from data that has less label uncertainty.

4. Experiments

From the analysis shown in Von Ahn et al. (2008), 67.87% of the words required only two human responses to be considered correct, 17.86% required three, 7.10% required four, 3.11% required five, and only 4.06% required six or more transcribtions. Just like reCaptcha we were interested for finding out how repeated labels affect accuracy of the candidate data points. We chose to measure accuracy using entropy in the labels and log loss. We measured entropy because we are using the proficiency of the labeler to update the posterior of the labels, so a working experiment should show entropy decreasing as we get similar labels. If we get different labels for the same data point it might mean that the particular data point is ambiguous for our labelers.

After each labeling event we calculate the entropy on the labels for all candidate data points. Figure 1b shows how the entropies of the candidate batch goes down as we query more labelers. As the size of the queries increase the entropy decreases faster. Similar trends are seen in Figure 1c which shows the log loss on the candidate batch.

5. Conclusion

We show how the BatchBALD algorithm proposed in Kirsch et al. (2019) can be extended to use human in the loop for labeling, especially when the human is not an oracle and can make mistakes. We propose a new querying technique where the labelers are shown control queries along with candidate queries from the BatchBALD algorithm. Their proficiency is determine from their labels on the control queries and it is used to update the posteriors of the candidate query data points. Using this approach allows us to use labeled data points with confidence before adding them to the training set and rerunning the MC dropout pipeline to get a new candidate batch that is best for retraining the learner.

References

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Gaurav Gupta, Anit Kumar Sahu, and Wan-Yi Lin. Learning in confusion: Batch active learning with noisy oracle, 2019.

Panagiotis G Ipeirotis, Foster Provost, Victor S Sheng, and Jing Wang. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*, 28(2):402–441, 2014.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7026–7037, 2019.

Emmanouil Antonios Platanios, Maruan Al-Shedivat, Eric Xing, and Tom Mitchell. Learning from imperfect annotations, 2020.

Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622, 2008.

Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321 (5895):1465–1468, 2008.