Learning Generalizable Batch Active Learning Strategies via Deep Q-networks* (Student Abstract)

Yi-Chen Li[1†], Wen-Jie Shen[1,2†], Boyu Zhang[3], Feng Mao[3], Zongzhang Zhang[1], Yang Yu[1]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2] School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3] Alibaba Group, Hangzhou 310052, China
liyc@lamda.nju.edu.cn, shenwenjie777@gmail.com, {zhangboyu.zby, maofeng.mf}@alibaba-inc.com, {zzzhang, yuy}@nju.edu.cn

Abstract

To handle a large amount of unlabeled data, batch active learning (BAL) queries humans for the labels of a batch of the most valuable data points at every round. Most current BAL strategies are based on human-designed heuristics, such as uncertainty sampling or mutual information maximization. However, there exists a disagreement between these heuristics and the ultimate goal of BAL, i.e., optimizing the model's final performance within the query budgets. This disagreement leads to a limited generality of these heuristics. To this end, we formulate BAL as an MDP and propose a data-driven approach based on deep reinforcement learning. Our method learns the BAL strategy by maximizing the model's final performance. Experiments on the UCI benchmark show that our method can achieve competitive performance compared to existing heuristics-based approaches.

1 Introduction

One way to mine unlabeled data is to query humans for their labels and then apply a supervised learning algorithm. However, querying all the unlabelled data is inefficient and costly. For this reason, batch active learning (BAL) iteratively chooses a batch of the most valuable data points and acquires their labels from humans, enabling better data efficiency with little degradation in model performance. Nevertheless, most current BAL approaches are hand-designed heuristics, such as mutual information maximization (Kirsch, van Amersfoort, and Gal 2019), uncertainty sampling, etc. These heuristics may work well on some tasks, but there exists a disagreement between them and optimizing the model's final performance under query budget constraints. This disagreement leads to their limited generality and effectiveness; sometimes, they perform even worse than random sampling.

This paper proposes a novel approach that learns a more generalizable BAL strategy from data. We first cast BAL into a Markov decision process (MDP). Then we use deep Q-networks (DQN) (Mnih et al. 2015), a state-of-the-art deep reinforcement learning (DRL) algorithm, to learn a BAL strategy where the long-term model performance will serve as the reward signal. Finally, to improve the generality of the learned strategy, we use the domain randomization technique to train a DRL agent on datasets from different domains. Experiments on the general machine learning benchmark UCI (Dua and Graff 2017) show that our method exhibits superior effectiveness and generality compared to existing heuristics-based approaches.

*Corresponding author: Zongzhang Zhang. This work is supported by the NSFC (No. 62276126) and Alibaba Group through Alibaba Research Fellowship Program.

†These authors contributed equally.

2 Our Method

In this section, we introduce our method, BALQ. We will formulate BAL as an MDP and describe the essential DRL components and the algorithm details.

Problem Formulation

We consider the binary classification problem. Denote f the classifier, $D^u = \{x_1^u, x_2^u, \cdots, x_n^u\}$ the unlabeled dataset, and D^l the labeled one, which is empty at start. We are asked to query at most $\mathcal{B}$ samples from D^u to maximize the performance of f. In the batch mode setting, we query k samples every time. After each query, the newly labeled k samples will be added to D^l; then, we retrain f on D^l.

We can formalize BAL as an MDP. At each time step $t, 0 \leq t \leq \mathcal{B}$, the agent perceives state s_t, representing the inner status of the classifier f. Based on s_t, the agent chooses an action a_t, representing the selected unlabeled data point. Then it will receive reward r_t, indicating the goodness of the newly selected data point. We additionally use a buffer to store newly selected points until their number reaches the batch size k.

DRL Components

Here we present our definitions of state, action, and reward. Let $D^{u,t}$, $D^{l,t}$, and $D^{b,t}$ respectively denote the unlabeled dataset, the labeled one, and the set of samples already selected in the current batch, at time step t.

State We want state s_t to represent classifier f. However, f can be any binary classifier model, e.g., a neural network with millions of parameters, where directly compressing them into a low-dimensional embedding is difficult. To this end, we randomly sample m points from D^u before training to form another unlabeled dataset D^s. At time step t, we use f to predict the confidence $\hat{y}_i$ for each point $x_i^s \in D^s$, i.e., the probability of x_i^s belonging to the positive class. We then define state s_t as a vector of sorted $\hat{y}_i$, where $i \in \{1, 2, \cdots, m\}$.

Action We use action a_i^t, where $i \in \{1, 2, \cdots, |D^{u,t}|\}$, to represent the candidate unlabeled data point x_i^u at time step t. Action a_i^t consists of three components, i.e.,

- $L_i^u = \sum_{x_j^u \in D^{u,t}} d(x_i^u, x_j^u)/|D^{u,t}|$,
- $L_i^l = \sum_{x_j^l \in D^{l,t}} d(x_i^u, x_j^l)/|D^{l,t}|$,
- $L_i^b = \sum_{x_j^b \in D^{b,t}} d(x_i^u, x_j^b)/|D^{b,t}|$.

Here, d is a distance measure and we use the cosine distance in this paper. We thus define $a_i^t = (L_i^u, L_i^l, L_i^b)$, which measures the average distances of point x_i^u to all unlabeled, labeled, and in-batch points.

Reward The agent receives a reward $r_t = -1$ every time step t until we exhaust our budget or the classifier f reaches a pre-specified performance metric, such as accuracy. Such a reward definition will prompt the agent to select the most valuable data points as soon as possible since doing so returns the maximum cumulative rewards.

We have now successfully instantiated the BAL as a finite-horizon MDP with deterministic transition, on which we can apply any suitable DRL algorithm. In this paper, we use DQN, a well-known DRL algorithm. Following LAL (Konyushkova, Sznitman, and Fua 2018), our DQN receives state and action as input and outputs a scalar value representing how good the currently considered point is. We train the DQN agent on datasets from different domains to improve the learned strategy's generality.

3 Experiments

Experimental Setup We compare our method BALQ with random sampling, uncertainty sampling, and BatchBALD (Kirsch, van Amersfoort, and Gal 2019), a state-of-the-art BAL algorithm. Uncertainty sampling and random sampling are two commonly used active learning strategies, but they are unsuitable for batch mode settings. We, therefore, take the k most uncertain samples (or randomly select k samples) as a batch in each query. We use the *Mushroom* dataset from the general machine learning benchmark UCI (Dua and Graff 2017) to test the performance of all algorithms, with AUC as the metric. All algorithms use the logistic regression classifier. For BALQ, we set $k = 3$, the target AUC to be 0.98, and train a DQN on eight datasets, including *Adult*, *Australian*, *Breast Cancer*, *Diabetis*, *Flare Solar*, *German*, *Heart*, *Waveform*, and *Wdbc*. The learned DQN will be transferred directly to the *mushroom* dataset. For a fair comparison, we use 10-fold cross-validation to obtain their average performance. The result is shown in Figure 1.

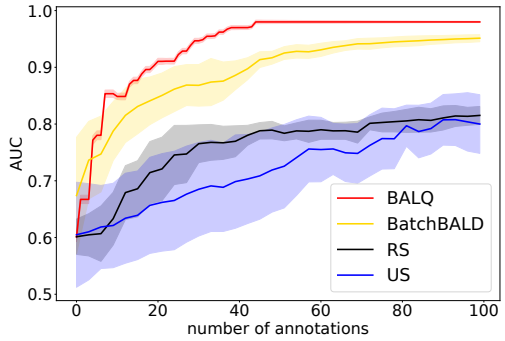

Figure 1: AUC curves of all methods on the *Mushroom* dataset from the benchmark UCI. Shaded areas are their corresponding 95% confidence intervals.

Benchmark Results From Figure 1, we see that BALQ performs comparably better than all baseline methods. Notably, the 95% confidence intervals of all baseline methods are relatively wide, meaning they perform unstable on random dataset partitions. In contrast, BALQ performs much more stable and quickly selects the most valuable data points, showing superior generality and effectiveness. See the supplementary materials[1] for more details.

4 Conclusion and Future Work

We propose BALQ, a data-driven approach to learning BAL strategies. Compared with previous works, BALQ shows many advantages. One is that we can mine BAL strategies from all existing labeled datasets. Although it shows impressive performance, BALQ is our first step. There are many promising directions for future work, and we list one below. Currently, BALQ uses a zero-shot paradigm, i.e., the learned strategy is directly transferred to the test dataset. However, fine-tuning it on the target domain may improve its performance. Overall, we hope that our work will inspire more relevant research.

References

Dua, D.; and Graff, C. 2017. UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed: 2022-08-27.

Kirsch, A.; van Amersfoort, J.; and Gal, Y. 2019. BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Advances in Neural Information Processing*, 7024–7035. Vancouver, BC, Canada.

Konyushkova, K.; Sznitman, R.; and Fua, P. 2018. Discovering general-purpose active learning strategies. arXiv:1810.04114.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M. A.; Fidjeland, A.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

[1] http://www.lamda.nju.edu.cn/liyc/files/BALQ_sup.zip