Towards Comparable Active Learning

February 23, 2023

1 Introduction

1.1 Contribution

2 Related Work

3 Overview

3.1 Problem Description / Delineation

Basic Classification
We assume a dataset $\mathcal{D} := (\mathcal{U}, \mathcal{L})$ consisting of a large pool of unlabeled data $\mathcal{U}$ and a small set of labeled data called the seed set with $|\mathcal{L}| \ll |\mathcal{U}|$. For the purpose of this work, only fully labeled datasets are used and the labels for $\mathcal{U}$ are suppressed until they are selected for labeling. For evaluation purposes we also assume a held-out test set $(x_{test}, y_{test}) \in (\mathbb{R}^M, \mathbb{R}^C)$ We consider only classification problems, hence the instances of a dataset have the form $x \in \mathbb{R}^M$ for $\mathcal{U}$ and $(x, y) \in (\mathbb{R}^M, \mathbb{R}^C)$ for $\mathcal{L}$. To perform classification, a model $\hat{y}_\theta := \mathbb{R}^M \to \mathbb{R}^C$ is used. To fit the model, it is parameterized by θ and subjected to loss $l(y, \hat{y}_\theta(x)) := \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}$. For this work, categorical cross-entropy (CE) is used. For evaluating classification performance, we use Accuracy

Pool-based AL with single instances (non-batch setting)
Active learning is defined as sequentially removing single instances $x_i \in \mathcal{U}_i$; $\mathcal{U}_{i+1} := \mathcal{U}_i / \{x_i\}$, requesting their label y_i and adding them to the labeled pool $\mathcal{L}_{i+1} := \mathcal{L}_i \cup (x_i, y_i)$ until a fixed budget B is exhausted $i := 0 \ldots B$. After each added instance the classification model is retrained according to section 4.3 and its performance is measured on a held-out test set. The quality of an active learning algorithm is evaluated by an "anytime" protocol that incorporates classification performance at every iteration, not just the final performance after the budget is exhausted. We employ the normalized area under the accuracy curve (AUC):

$$\text{auc}(\mathcal{U}, \mathcal{L}, \hat{y}, B) := \frac{1}{B} \sum_{i=1}^{B} \text{Acc}(y_{test}, \hat{y}_i(x_{test})) \tag{1}$$

Where $\hat{y}_i$ is the retrained classification model after the i-th instance was selected.

Framing AL as RL
We define the active learning process as an adapted reinforcement learning loop $(S, A, \tau, \Omega, \omega)$ where an environment iteratively will expose a state $s \in S$ to an agent Ω, which will choose actions $a \in A$. For each iteration i the environment samples a subset of size τ of unlabeled instances $x_i \underset{\tau}{\sim} \mathcal{U}_i$, constructs the state $s_i := \omega(x_i)$ and presents it to the agent to select an action $a_i := \Omega(s_i)$. The action a_i is the index of the selected instance in u_i out of all possible indices $A := [0 \ldots \tau]$. This process is repeated B times $i := [0 \ldots B]$.

Algorithm 1 Active Learning

```
Require: $\mathcal{U}$                                   ▷ Unlabeled Pool
Require: $\tau$                                          ▷ Unlabeled Sample Size
Require: $\Omega$                                        ▷ AL Agent
Require: $\omega$                                        ▷ Environment State function
 1: $\mathcal{L}^{(0)} \leftarrow \text{seed}(\mathcal{U})$                ▷ Create the initial labeled set
 2: $\mathcal{U}^{(0)} \leftarrow \mathcal{U}$
 3: for $i := 0 \dots B$ do
 4:     $u^{(i)} \underset{\tau}{\sim} \mathcal{U}^{(i)}$
 5:     $s^{(i)} \leftarrow \omega(u^{(i)})$
 6:     $a^{(i)} \leftarrow \Omega(s^{(i)})$                    ▷ $a^{(i)}$ is an index inside of $u^{(i)}$
 7:     $y^{(i)} \leftarrow \text{oracle}(u_a^{(i)})$              ▷ $u_a^{(i)}$ is shorthand for $u_{a^{(i)}}^{(i)}$
 8:     $\mathcal{L}^{(i+1)} \leftarrow \mathcal{L}^{(i)} \cup \{(u_a^{(i)}, y^{(i)})\}$
 9:     $\mathcal{U}^{(i+1)} \leftarrow \mathcal{U}^{(i)} \setminus \{u_a^{(i)}\}$
10:     $\text{acc}^{(i)} \leftarrow \text{Retrain}(\mathcal{L}^{(i+1)})$   ▷ $\text{Retrain}(\mathcal{L}^{(i+1)})$ is shorthand for $\text{Retrain}(\mathcal{L}^{(i+1)}, \mathcal{L}^{\text{test}}, \hat{y}_\theta, e^{\max})$
11: end for
12: return $\frac{1}{B}\sum_{i=1}^{B} \text{acc}^{(i)}$
```

Algorithm 2 Retrain

```
Require: $\mathcal{L}$                                   ▷ Labeled Pool
Require: $\mathcal{L}^{\text{test}}$                              ▷ Labeled Test Data
Require: $\hat{y}_\theta$                                  ▷ Classification Model
Require: $e^{\max}$                                  ▷ Maximum Epochs
 1: $\text{loss}^* \leftarrow \infty$
 2: for $i := 0 \dots e^{\max}$ do
 3:     $\theta_{i+1} \leftarrow \theta_i - \eta \nabla_\theta \ell(\mathcal{L}, \hat{y}_\theta)$
 4:     $\text{loss}_i \leftarrow \ell(\mathcal{L}^{\text{test}}, \hat{y}_\theta)$
 5:     if $\text{loss}_i < \text{loss}^*$ then
 6:         $\text{loss}^* \leftarrow \text{loss}_i$
 7:     else
 8:         Break
 9:     end if
10: end for
11: return $\text{Acc}(\mathcal{L}^{\text{test}}, \hat{y}_\theta)$
```

4 Methodology

4.1 Classification Model

The classifier is constructed according to two kinds of information. The general class of model (Dense, Convolutional, Attention, ...), and the configuration of the model (number of layers, size of each layer, ...). The model class and exact configuration is determined by the dataset, i.e. tabular datasets will prescribe a dense model. If special capabilities of the model are needed (i.e. Monte-Carlo Dropout), an extension of the given model class can be provided to the framework.

To ensure comparability between models, the model's configuration should not be changed or an additional evaluation of the new configuration should be conducted to compare the baseline expressivity.

4.2 State Space

Since every AL agent needs a different state space our environment exposes a callback-function that gives the agent full control over how the state is constructed.

The callback includes the following information:

- The current sample of IDs that point to the presented unlabeled datapoints
- The entire labeled dataset $\mathcal{L}$
- The entire unlabeled dataset $\mathcal{U}$
- A histogram of labeled points per class (count)
- The available budget
- Number of added datapoints $|\mathcal{L}| - |\mathcal{S}|$
- The initial validation accuracy and current validation accuracy
- The current classification model including all model weights
- The current optimizer including it's full state

Every agent needs to implement this callback that transforms the given information into a state tensor that will be directly consumed by the agent to make it's prediction.

4.3 Training the Classifier

4.4 Evaluation

5 Ablation Studies

- Weird drop of performance for multiples of batch size (drop_last in DataLoader)
- Reduction of the test set for speed

References