TODO: Funky active learning pun

Lukas Weber [1]

Abstract

A lot of unlabelled data generally available but annotation is costly. A goal in fields where labelled data is not available is to maximize performance with minimum amount of labeled data instances. Active learning selects instances for labelling based on a selection strategy.

- Objective: Compare active learning query methods using modAL.
- Methods: Experiments with Logistic Regression (MNIST) and a CNN (CIFAR).
- Experiments: Vary initial dataset sizes (small, moderate, large).
- Key findings: Highlight main conclusions about performance in different scenarios.

1. Introduction and Related Work

Supervised learning often requires large, annotated datasets, but labeling data can be costly and time-intensive, especially when expert knowledge is needed. Active learning addresses this challenge of reducing the overall labeling costs through the selection of the most informative samples for annotation based on a query strategy. This approach enables machine learning models to perform well in tasks with limited data and high labeling costs.
The effectiveness of active learning depends heavily on the chosen query strategy, which determines which samples are selected for annotation. Two main frameworks are commonly used:

1. Pool-based active learning: A model is trained on a small labeled dataset and used to evaluate a large pool of unlabeled data. The most informative samples are selected based on a query strategy, labeled by an oracle, and added to the training set. The model is retrained with the updated training set and the process is repeated until a stopping criterion is met.
2. Stream-based active learning: Here, unlabeled data arrives sequentially and the model decides whether to query the label of an incoming instance using an evaluation metric like the uncertainty of its prediction and checking whether a certain threshold is reached. If not, the instance is discarded.

This report focuses on comparing several query strategies for both pool-based and stream-based active learning. Experiments in this paper leverage the modular Active Learning framework (modAL) (Danka & Horvath, 2018), which simplified the implementation of active learning pipelines, includes multiple pre-implemented query strategies and is compatible with models from the Scikit-learn library (Pedregosa et al., 2011). Convolutional Neural Netowrk (CNN) implemented with PyTorch(ADD CITATION) and transformed with skorch (Tietz et al., 2017) to make it compatible with modAL as it is only compatible with sklearn models.
Several previous works have explored and compared different query methods across various domains.
(Schröder et al., 2022) compare various uncertainty-based query strategies in the context of fine-tuning transformer models in text classification tasks. (Zhan et al., 2022) on the other hand compares several query strategies on the image datasets MNIST and CIFAR.
According to (Ueno et al., 2023) most papers concentrate on two image-based datasets: MNIST (Lecun et al., 1998) and CIFAR (Krizhevsky et al., 2009), while their paper evaluates strategies on six different datasets, including medical and visual inspection images. They also examine strategies independent of training processes to minimize biases from early-stage randomness or underfitting. This is shown in an experiment where fully trained models are used to select the most informative samples based on several different query strategies to construct a labeled dataset.
A recurring issue in active learning research is the lack of standardization in experimental setups, as noticed additionally by (Werner et al., 2023). This inconsistency complicates the comparability of existing and novel query strategies. To address this, they propose a benchmark framework for evaluatting active learning strategies across tabular, image and text datasets. This framework uses

*Equal contribution [1]MSc Computer Science, Tübingen, GER. Correspondence to: Lukas Weber <lukas2.weber@student.uni-tuebingen.de>.

Project report for the "Data Literacy" course at the University of Tübingen, Winter 2023/24 (Module ML4201). Style template based on the ICML style files 2023.

robust evaluation protocols to reduce variance and ensure comparability.
In contrast, (Ueno et al., 2023) focus specifically on image-based datasets from diverse domains such as medical or visual inspection images. While (Werner et al., 2023) focuses on a wide range of data-domains including images, text and vector-based datasets, this report focuses on comparing strategies in the image domain. It compares query strategies for both models such as simple logistic regression on MNIST and more complex architectures, like a CNN on CIFAR-10 to study the effect of model complexity on the optimal query strategy.
It is uncertain whether well working query strategies generalize well from simple to more complex models like Neural Networks (Schröder & Niekler, 2020). Understanding this generalizability is crucial for real-world applications where model complexity varies across tasks and datasets.
Furthermore, this report investigates the impact of the initial labeled dataset size, as its impact in combination with model complexity is rarely explored. By examining scenarios with varying amounts of initial labeled data, this work simulates various real-world applications: (1) starting with no labeled data, (2) having a small pool of labeled data, and (3) working with a well-trained model to identify data points for fine-tuning.

The contributions of this paper are as follows:

- **Systematic evaluation of active learning query strategies** across different datasets and model complexities, using MNIST with logistic regression and CIFAR-10 with a CNN.
- **Investigation of the inclusence of model complexity** on the effectiveness of query strategies to provide insights into how shallow and deep models respond differently to active learning methods.
- **Analysis of impact of initial labeled dataset size** on the performance of query strategies, simulating possible real-world scenarios such as starting from scratch, having a small labeled dataset or fine-tuning a pretrained model.

Section 2 outlines the used datasets, processing steps and models. Section 3 describes the experimental design, followed by the results and discussion in sections 4 and 5 respectively. The report concludes with key findings in section 6

2. Methods

Several different datasets are used in the following experiments.

2.1. Datasets

MNIST is a dataset consisting of handwritten digits with 60.000 train samples and 10.000 test samples. Digits are size-normalized and centered in a 28×28 image (Lecun et al., 1998). Each digits belongs to one of 10 classes, corresponding from 0 to 9.
Upon loading the dataset, each image is vectorized, i.e. transformed from a 28×28 sized image-matrix to a vector of length 784. Furthermore, each pixel contains a grayscale-value from 0 to 255. Pixel values are normalized to a range between 0 and 1 to provide a value range that works well with logistic regression.

CIFAR-10 is a dataset which consists of 50.000 train- and 10.000 test-images. Each natural image belongs to one of 10 classes. Each class has exactly 5.000 train and 1.000 test images. The image-size is 32×32 pixels. CIFAR preprocessing split. CIFAR-10 includes real-world images and is thus expected to be more complicated to learn.

Both used datasets are balanced, i.e. each class has roughly the same number of samples as any other class. Furthermore, 20% of both train sets are used for 5-fold-cross-validation, resulting in a test set size of 48.000 train images for MNIST and 40.000 train images for CIFAR.

2.2. Multinomial Logistic Regression Model

Sklearn's LogisticRegression module is used to predict digits on MNIST. Multinomial logistic regression predicts a probability for each class instead of a single probability as in Binomial logistic regression.
The model estimates the probability $P(y = i \mid X)$ of a sample belonging to class i using the SoftMax function:

$$P(y = i \mid X) = \frac{\exp(X\beta_i)}{\sum_{j=1}^{k} \exp(X\beta_j)} \tag{1}$$

where β_i are the coefficients for class i. The SoftMax function ensures that probabilities across classes sum to 1.

The model maximizes the likelihood of the observed data:

$$\mathcal{L}(\beta) = \prod_{n=1}^{N} \prod_{i=1}^{k} \left(P\left(y_n = i \mid X_n\right)\right)^{y_{ni}} \tag{2}$$

where N is the number of samples, and y_{ni} indicates if sample n belongs to class i. Should I also explain Cross-Entropy loss? For a multinomial logistic regression model with k classes, the cross-entrop

$$\mathcal{L} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{k} y_{nj} \log\left(P\left(y = j \mid X_n\right)\right)$$

Regularization using the ℓ_2-norm prevents overfitting with a strength $C = 0.01$. The lbfgs-solver, with a maximum iteration of 500, is used for efficient gradient-based

optimization. Early-stopping with a tolerance of 0.001 is used.

2.3. Convolutional Neural Network

The model which is used to learn the classes in CIFAR-10 is a CNN. It comprises of two convolutional layers followed by three fully connected layers.
The first convolutional layer processes the input image with 3 input and $l1\_channels$ output channels, followed by a ReLU activation. The second convolutional layer further processes the output from the first layer with $l1\_channels$ input and $l2\_channels$ output channels, applying ReLU activation, max pooling, and dropout for regularization.
The output from the convolutional layers is flattened and passed through three fully connected layers. The first two fully connected layers are followed by ReLU activations and dropout for regularization. The first fully connected layer has an output size of $l4_input$ and the second layer has an output size of $l5_input$ neurons. The final output layer has 10 neurons, corresponding to the number of classes in the CIFAR dataset.
The model is implemented using PyTorchcite and skorch (Tietz et al., 2017) is used to make it possible to use it with Sklearn, allowing integration with the modAL active learning framework.
Early stopping?
During training, PyTorch's CrossEntropyLoss is used.

2.4. Active Learning

- During active learning with pool-based query strategies in the case of logistic regression, 5 instances are queried at a time for 250 iterations resulting in a labeled pool which is increased by 1.250 samples.
  When training the CNN, 250 instances are chosen for 50 iterations. This enlarges the labeled pool by 12.500 samples.
  In stream-based methods, the number of queries is adapted to match the given numbers and guarantee that the labeled dataset size is the same as in the pool based methods.
- Used active learning query strategies include pool- and stream-based methods. The pool based methods include uncertainty sampling, uncertainty margin sampling and uncertainty entropy sampling. Stream-based active learning is done with uncertainty as the evaluation metric deciding whether to use samples or not. Additionally, random sampling was used to serve as a reference point. maybe explain the query strategies more in detail
- After querying instances and retraining, models are not re-initialized but rather fine-tuned.

3. Experiments

Several experiments are conducted with the two model-dataset combinations to research different scenarios for which active learning can be used.

- Experiment 1: Starts with a minimal amount of data. In both cases these are 10 samples.
- Experiment 2: Starts with 15% of the train dataset.
- Experiment 3: Starts with 80% of the train dataset.

In experiment 1, a random sample is chosen for each class in the datasets.
In each experiment, the random samples

For reproducibility purposes, the same random seed 42 was used to initialize all random number generators and model weights to ensure that same configurations lead to the same outcomes, even when training models in succession.

For each query method one model is trained. (I did not perform multiple runs) Before starting the active learning process, each model is pre-trained using the samples in the initial labeled dataset for X epochs. In the case of pool-based query methods, 5 samples are queried for 250 iterations when training with samples from MNIST and 250 samples are queried for 50 iterations when training with instances from CIFAR.

- Batch size for CNN is 128.
- Evaluation metrics. Metrics for evaluation (Accuracy, train error, confusion matrix, ...)
- How are the splits (initial pool, initial training data) generated
- initial labeled pool sizes

Scenarios

- describe experiments and their settings and purposes

4. Results

Quantitative results: Tables or plots comparing the query strategies across scenarios. Also maybe key perforamnce differences (accuracy vs number of labeled examples)

5. Discussion

Qualitative analysis: Why certain strategies perform better or worse in specific settings. Trends between simple models and complex models

Impact of initial dataset size

Most methods require knowledge about the balance of the dataset and might be disadvantageous to use in case of a disbalanced dataset. In real world scenarios, this is not known. Both used datasets are balanced so the impact of query strategies on imbalanced datasets was not tested.
Only image-based datasets were tested. Not vector- or text-based.

Only one network is trained for each query method. This could be improved upon when training multiple models.

6. Conclusion

Summary of findings and insights (maybe that a strategy works best with small datasets while another one is robust on bigger ones)
Implications: practical recommendations?
Future work? (More diverse datasets, exploring additional strategies, applying in real-world tasks)

7. Cool ideas

- Show the typical framework of active learning in a figure in the introduction
- Accuracy vs. fraction of data size in relation to the datset (percentage of used data points)

8. Good formulations

From the active learning survey: Entropy is an information-theoretic measure that represents the amount of information needed to "encode" a distri- bution. As such, it is often thought of as a measure of uncertainty or impurity in machine learning.

9. Comparison to the other papers

Previous papers do not: Is everything here now mentioned previously? if yes, then it can be deleted.

- (Schröder et al., 2022): Does not compare across different sizes of the pool, and for simple models.
- (**?**): Does not compare impact of different initial train set size and strategies on shallow classifiers.
- According to (Ueno et al., 2023): check if true: Does not compare different sizes of the initial starting sets as well as how simple and more complex perform with the query methods
- (Werner et al., 2023): They focus on a wide range of domains, this paper only compares methods in the image domain. I compare strategies for a simple logistic regression model and a complex CNN across different starting sizes of the annotated dataset.

References

Danka, T. and Horvath, P. modal: A modular active learning framework for python. 2018. URL https://arxiv.org/abs/1805.00979.

Krizhevsky, A. et al. Learning multiple layers of features from tiny images. 2009.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.

Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Schröder, C. and Niekler, A. A survey of active learning for text classification using deep neural networks, 2020. URL https://arxiv.org/abs/2008.07267.

Schröder, C., Niekler, A., and Potthast, M. Revisiting uncertainty-based query strategies for active learning with transformers, 2022. URL https://arxiv.org/abs/2107.05687.

Tietz, M., Fan, T. J., Nouri, D., Bossan, B., and skorch Developers. *skorch: A scikit-learn compatible neural network library that wraps PyTorch*, July 2017. URL https://skorch.readthedocs.io/en/stable/.

Ueno, S., Yamada, Y., Nakatsuka, S., and Kato, K. Benchmarking of query strategies: Towards future deep active learning, 2023. URL https://arxiv.org/abs/2312.05751.

Werner, T., Burchert, J., and Schmidt-Thieme, L. Towards comparable active learning, 2023. URL https://arxiv.org/abs/2311.18356.

Zhan, X., Wang, Q., hao Huang, K., Xiong, H., Dou, D., and Chan, A. B. A comparative survey of deep active learning, 2022. URL https://arxiv.org/abs/2203.13450.