



IMITAL: Learned Active Learning Strategy on Synthetic Data

Julius Gonsior¹(✉) , Maik Thiele² , and Wolfgang Lehner¹ 

¹ Technische Universität Dresden, Dresden, Germany
{julius.gonsior,wolfgang.lehner}@tu-dresden.de

² Hochschule für Technik und Wirtschaft Dresden, Dresden, Germany
maik.thiele@htw-dresden.de

Abstract. Active Learning (AL) is a well-known standard method for efficiently obtaining annotated data by first labeling the samples that contain the most information based on a query strategy. In the past, a large variety of such query strategies has been proposed, with each generation of new strategies increasing the runtime and adding more complexity. However, to the best of our knowledge, none of these strategies excels consistently over a large number of datasets from different application domains. Basically, most of the existing AL strategies are a combination of the two simple heuristics *informativeness* and *representativeness*, and the big differences lie in the combination of the often conflicting heuristics. Within this paper, we propose IMITAL, a domain-independent novel query strategy, which encodes AL as a learning-to-rank problem and learns an optimal combination between both heuristics. We train IMITAL on large-scale simulated AL runs on purely synthetic datasets. To show that IMITAL was successfully trained, we perform an extensive evaluation comparing our strategy on 13 different datasets, from a wide range of domains, with 7 other query strategies.

Keywords: Annotation · Active learning · Imitation learning · Learning to rank

1 Introduction

Machine Learning (ML) has found applications across a wide range of domains and impacts (implicitly) nearly every aspect of nowadays life. Still, one of the most limiting factors of successful application of ML is the absence of labels for a training set. Usually, domain experts that are rare and costly are required to obtain a labeled dataset. Thus, to improve the manual label task is a prime object to improve. For example, the average cost for the common label task of segmenting a single image reliably is 6,40 USD¹.

Reducing the amount of necessary human input into the process of generating labeled training sets is of utmost importance to make ML projects possible and

¹ According to scale.ai as of December 2021.

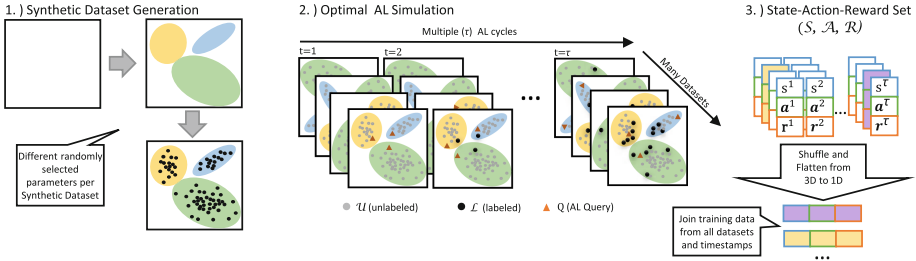


Fig. 1. General overview on the training procedure of IMITAL

scalable. A standard approach to reduce the number of required labels without compromising the quality of the trained ML model, is to exploit Active Learning (AL). The approach consists of an iterative process of selecting exactly those unlabeled samples for labeling by the domain experts that benefit the to-be trained model the most. Given a small initial labeled dataset $\mathcal{L} = \{(x_i, y_i)\}_i^n$ of n samples x_i with the respective labels y_i and a large unlabeled pool $\mathcal{U} = \{x_i\}, x_i \notin \mathcal{L}$, an ML model called *learner* θ is trained on the labeled set. A *query strategy* then subsequently chooses a batch of b unlabeled samples Q , which will be labeled by the human experts and added to the set of labeled data \mathcal{L} . This AL cycle repeats τ times until a stopping criterion is met.

The challenge of applying AL is the almost paradoxical problem to be solved: how to decide, which samples are most beneficial to the ML model, without knowing the label of the samples, since this is exactly the task to be learned by the to-be-trained ML model.

During the past years, many different AL query strategies have been proposed, but to our knowledge, none excels consistently over a large number of datasets and from different application domains. By deliberately focusing on domain-independent AL strategies we aim to shed some light onto this problem. Even though various extensive general survey papers [14, 15, 20] exist, no clearly superior AL strategy has been identified. The results of the individual evaluations in papers with newly proposed AL strategies suggest that current AL strategies highly depend on the underlying dataset domain. Even more interestingly, the naïve baseline of randomly selecting samples often achieves surprisingly competitive results [7, 9, 11, 13].

At its core, the vast majority of AL strategies rely on the same set of two simple heuristics: *informativeness* and *representativeness*. The first favors samples that foremost improve the classification model, whereas the latter favors samples that represent the overall sample distribution in the feature vector space. Most recent AL strategies add more layers of complexity on top of the two heuristics in their purest form, often resulting in excessive runtimes. This renders many AL strategies unusable in large-scale and interactively operating labeling projects, which are exactly those projects that would benefit the most from “optimal” learning strategies.

We are presenting IMITAL, a novel AL strategy, which at its core is a Neural Network (NN) trained on very large simulated AL episodes with the goal to optimally combine the basic AL heuristics informativeness and representativeness. As it is not practically feasible to enumerate all possible real-world datasets as training data in the simulations, we are approximating them by using synthetic datasets instead. The benefit of synthetic datasets is that we can leverage the knowledge about all the labels to construct an optimal AL strategy, which then serves as training basis for IMITAL. Our work falls therefore under the category of “learning AL strategies”. According to our knowledge, our approach is, in contrast to similar works [9, 11, 13], the first one to solely utilize purely synthetic data to train the strategy. We can present a pre-trained, ready-to-apply AL strategy which can be applied without any further necessary transfer-learning or fine-tuning in any domain.

We start in Sect. 2 by presenting our synthetic datasets simulation process, followed by our Imitation Learning (IL) procedure in Sect. 3. In Sect. 4, we are comparing IMITAL with 7 common AL strategies on 13 real-world datasets and conclude in Sect. 5.

2 Simulating AL on Synthetic Training Data

For the IL training procedure of IMITAL we need an *expert* AL strategy, which the neural network behind IMITAL can learn to imitate. In order to capture the characteristics of “all” possible datasets we pursue the idea by generating initially nearly “infinite” synthetic datasets² and computing an optimal AL strategy on them, leveraging the information about the known full labels for the synthetic datasets.

We construct the nearly-optimal strategy by selecting a batch of those samples for labeling, which will result in the highest accuracy (in the following called *reward*), if they each were added to the set of labeled samples \mathcal{L} . As this process is computationally heavy, we do not consider all possible batches, but perform a *pre-selection* based on a heuristic, which selects a promising and diverse set of the top- k batches. Details of the pre-selection are explained in Sect. 3.

The results of the AL simulation for each AL cycle t for a specific synthetic dataset is a *state-action-reward triple*. The state s is represented as a triple $s = (\mathcal{U}^t, \mathcal{L}^t, \theta^t)$, consisting of the set of unlabeled samples \mathcal{U}^t , the set of labeled samples \mathcal{L}^t , and the state of the learner model θ^t trained on \mathcal{L}^t . The corresponding actions \mathbf{a}_s is a set of the pre-selected queries x , whereas the respective rewards \mathbf{r}_s for each of these actions is a set of rewards r . The optimal choice $Q_s^t \in \mathbf{a}_s$ for the AL cycle t can be easily computed from the given accuracies – the action with the highest future accuracy. This simulation is repeated α -times using different synthetic datasets. The accumulated state-action-reward pairs, denoted as the triple $(\mathcal{S}, \mathcal{A}, \mathcal{R})$, reflect then the input for IL training procedure

² For generating the synthetic datasets the algorithm by [4], which is a runtime efficient method for creating a diverse range of synthetic datasets of varying shape and resulting classification hardness, is used.

of the NN of IMITAL. The whole synthetic data training generation and training of IMITAL has to be done only once, afterwards it is applicable to real-world datasets without any further transfer learning or fine-tuning steps.

3 Training a Neural Network by Imitation Learning

The final step of IMITAL is to use the generated state-action-reward triples $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ for training an NN as AL query strategy. Therefore, we are deploying the ML technique IL [12], where demonstrated expert actions are being replicated by the ML model. The training task for IMITAL is to find patterns in the presented actions.

Subsequently (Sect. 3.1) we will first explain the IL learning process, followed by the details of the NN input and output encoding (Sect. 3.2), and lastly, the necessary pre-selection process (Sect. 3.3).

3.1 Imitation Learning

For training IMITAL we use Imitation Learning (IL), where an expert demonstrates an optimal strategy, which the neural network behind IMITAL learns to replicate. We use *behavioral cloning* [12] as a variant of IL, which reduces IL to a regular regression ML problem. The desired outcome is a trained strategy returning the optimal action for a given state. We use the state-action-reward set $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ to extract an optimal strategy $\hat{\pi}$, which we are then demonstrating to the to-be-trained network $\hat{\pi}(s) = \operatorname{argmax}_{x \in \mathbf{a}_s, r_x \in \mathbf{r}_s} (r_x)$. For a given state s , the action set \mathcal{A} contains all pre-selected actions \mathbf{a}_s for this state; the reward set \mathcal{R} contains the respective rewards \mathbf{r}_s . As the optimal strategy only contains the optimal actions, it can be used to construct the optimal batch by taking the b -highest actions. In other words: we train a network $\hat{\pi}$ predicting for a given state s and a possible action $x \in \mathbf{a}_s$ – which equals labeling the sample represented by this action – the reward $r_x \in \mathbf{r}_s$. The expected future accuracy is in our case demonstrated by the true reward \hat{r} function as $\hat{r}(s, x) = r_x$.

3.2 Neural Network Input and Output Encoding

Before using the state-action-reward set $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ to train the network predicting the future accuracy, we first transform it into a fixed sized vector representation using feature encoding, and thus making IMITAL dataset agnostic. NNs are limited by the number of the neurons to either a fixed size input, or when using *recurrent* NN to circumvent this limitation, they often suffer from the case of memory loss where the beginning of the input sequence is forgotten due to exploding or vanishing gradients [6]. The last problem occurs more frequently the larger and more length-varying the input is, which is the case for AL. The raw actions set \mathcal{A} may then contain – depending on the number of unlabeled samples – many possible samples, or just a few, varying again drastically. That

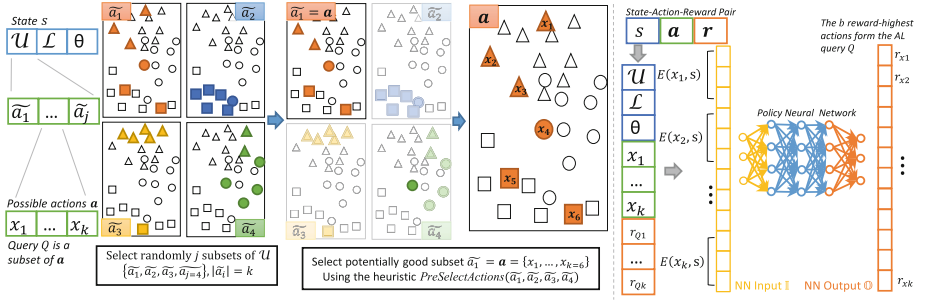


Fig. 2. Pre-selection process and action meaning for IMITAL, example for $j = 4$, $k = 6$, and $b = 3$, and encoding of a state-action-triple

underpins the already mentioned *pre-selection* method, reducing the number of possible actions to a fixed size.

The transformation of the state-action-reward set into a suitable form for the network is called *input* and *output encoding*. Figure 2 displays the general procedure of the encoding to the right. We chose a *listwise* input encoding, where we enter k possible actions $x \in \mathbf{a}_s$, $|\mathbf{a}_s| = k$ at once into the network, in contrast to a *pointwise* encoding, where a single action is entered at a time. This has the benefit of enabling the network to compare each possible action relatively to the others, enabling IMITAL to take batch-aware AL query decisions. Batch-awareness is, as thoroughly explained in [8], a beneficial and desirable property of AL query strategies, meaning that the joint diversity of all samples of the final AL batch Q is taken into account. The input of the network is defined by the vector $\mathbb{I}_s = \{E(x, s) | x \in \mathbf{a}_s\}$, with E being the encoding of the action x . The output $\mathbb{O} = (\hat{r}_1, \dots, \hat{r}_k)$ of the network consists of exactly $|\mathbf{r}_s|$ output neurons, one for each of the predicted accuracies \hat{r}_x for the respective possible actions x . The amount of output neurons equals therefore the amount of pre-selected actions: $|\mathbf{r}_s| = |\mathbf{a}_s| = k$. We use a final softmax layer of k output neurons, each per possible action. The b highest output neurons indicate the samples for the unlabeled query Q .

A single action represents an unlabeled sample $x \in \mathcal{U}$. The input encoding function $E(x, s)$ defines on what basis the network can make the AL query strategy decision. We use the state $s = (\mathcal{U}, \mathcal{L}, \theta)$ to calculate the encoding, which is a 5-tuple consisting of multiple parts, the individual functions will be explained in the following paragraphs $E(x, s) = (u_1(x, \theta), u_2(x, \theta), u_3(x, \theta), dl(x, \mathcal{L}), du(x, \mathcal{U}))$. The complete network input vector \mathbb{I} consists then of 5-times k values, an encoded 5-tuple for each unlabeled sample x out of the set of possible actions $\mathbf{a} \mathbb{I}_s = \{E(x_1, s), \dots, E(x_k, s)\}, x \in \mathbf{a}_s$. As mentioned in the beginning, a good AL query strategy takes informativeness as well as representativeness into account. Informativeness is derived by $u_i(x, \theta)$, a function computing the uncertainty of the learner θ for the i -th most probable class for the sample $x \in \mathcal{U}$, given the probability of the learner $P_\theta(y|x)$ in classifying x with the label y :

$$u_i(x, \theta) = \begin{cases} P_\theta \left(\left(\operatorname{argmax}_{y,i} P_\theta(y|x) \mid x \right) \right), & \text{if } i \leq C \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$\operatorname{argmax}_{\cdot,i}$ denotes the i -th maximum argument, and C the number of classification classes.

For representativeness we compute $dl(x, \mathcal{L})$ and $du(x, \mathcal{U})$, the first denoting the average distance to all labeled samples, the latter the average distance to all unlabeled samples $dl(x, \mathcal{L}) = \frac{1}{|\mathcal{L}|} \sum_{x_l \in \mathcal{L}} d(x, x_l)$, $du(x, \mathcal{U}) = \frac{1}{|\mathcal{U}|} \sum_{x_u \in \mathcal{U}} d(x, x_u)$, where $d(x_1, x_2)$ is an arbitrary distance metric between the points x_1 and x_2 . We use the Euclidean distance for small feature vector spaces, and recommend using the cosine distance for high-dimensional feature vector space. Both feature encoding functions represent the most raw, unprocessed forms of informativeness and representativeness, as the neural network should learn necessary transformations of the feature vector space.

3.3 Pre-selection

Instead of considering all possible actions, we pre-select promising actions, whose individual samples have the largest diversity and whose individual samples are the furthest away from each other, similar to [8]. The pre-selection fulfills two objectives: first and foremost, we can present a fixed amount of actions to the network, and secondly it keeps the runtime of the simulations within a processable range. A positive side effect of the fixed-size input of the network is the low and static runtime of IMITAL, which is almost independent of the size of the dataset. The effect is especially apparent with very large datasets.

We start the pre-selection by drawing randomly j possible actions $\{\tilde{a}_1, \dots, \tilde{a}_j\}$, with each \tilde{a} being a subset of \mathcal{U} . After that we use a heuristic to select the top- k most promising actions \mathbf{a} out of the random ones. The pre-selection process is illustrated in Fig. 2 at the left side.

We are using a heuristic to filter out potentially uninteresting actions. By calculating the average distance to the already labeled samples of all the samples in each possible action set \tilde{a} and select the action set \mathbf{a} having the highest average distance: $\mathbf{a} = \operatorname{argmax}_{\tilde{a}} \sum_{x \in \tilde{a}} dl(x, \mathcal{L})$, where \tilde{a} contains k unlabeled samples. Thus, we are ensuring that we sample evenly distributed from each region in the sample vector space during the training process. We compute the heuristic for j random possible batches, instead of all possible subsets of \mathcal{U} .

4 Evaluation

The goal of IMITAL is to learn a domain-independent AL strategy from synthetic datasets, which combines the strength of both the basic informativeness and the representativeness heuristics. For evaluation, we are comparing IMITAL therefore with 7 AL strategies on 13 real-world datasets from varying domains.

4.1 Experiment Details

The datasets are from the UCI ML Repository [1] with varying sample size, feature size, and application domain, similar to the evaluations of [7, 9, 13]. As an additional larger dataset the table classification dataset DWTC [2] was also included. For the experiments we started with a single random sample of each class, and ran the AL loop with a batch size b of 5 for 25 cycles, or until all data was labeled. We repeated this for 1,000 times with varying initial labeled samples to generate statistically stable results. As learner model θ a simple NN with 2 hidden layers and 100 neurons each was used. The datasets were split randomly into a 50 % train and 50 % test evaluation set.

As evaluation metric we used the *area-under-the-curve* (AUC) of the learning curve, as has been also done recently in the AL survey by Chan et. al. [20]. This makes it easy to calculate the mean of 1,000 times repeated experiments. Similarly to [5] we are further normalizing the AUC values by the maximum possible AUC value – a rectangle of 100% F1-Scores for each time step – to additionally enable comparisons across datasets.

The training of IMITAL is highly parallelizable, as the generation of the synthetic datasets and the respective AL simulation may run completely in parallel. For a full training of IMITAL with the best parameters we needed 100,000 computation jobs, resulting in a set of 1,000,000 state-action pairs as training data. In total, $\sim 1\text{M}$ CPU-hours were needed for all experiments conducted for this paper, including testing out different NN and IL configurations, and training the final version of IMITAL. For the final version of IMITAL we set the parameter of the simulated AL cycle τ to 10, the pre-sampling parameter k to 20 and j to 10 during training, and 2 during application, as this suffices for a trained IMITAL. The batch size was fixed to a standard value of 5 for the used UCI datasets.

4.2 Comparison with Other Active Learning Strategies

Our evaluation compares 7 AL strategies against our AL strategy, IMITAL. The results are shown in Table 1. Each displayed value is the mean of F1-AUC values for the 1,000 repeated runs. As the percentages are often quite similar, we additionally included the ranks. The displayed percentages are rounded, but the ranks are computed on the complete numbers, which can lead to different ranks for otherwise equally rounded displayed percentages.

We included Least Confidence (LC) and Uncertainty Entropy (Ent) [17], the two most common and basic variants of the informativeness heuristic, where greedily the most uncertain samples based on the classification probability of the learner model are selected for labeling. The Graph Density (GD) strategy [3] was added as a pure representativeness heuristic-based strategy which solely focuses on sampling evenly from the vector space. BatchBALD [8] is a popular AL strategy which works well for computer vision deep neural networks. Querying Informative and Representative Examples (QUIRE) [7] is a computationally

Table 1. F1-AUC-scores (%) for different AL query strategies, mean for 1,000 repeated experiments each, including the ranks and the ranked mean. Empty cells indicate no calculable results within the maximum runtime window of seven days.

	IMITAL	LC	QBC	Ent	Rand	GD	BatchBALD	QUIRE
abalone	21.2 (2)	19.3 (5)	19.6 (4)	17.8 (6)	21.3 (1)	15.6 (7)	21.1 (3)	11.2 (8)
adult	54.5 (1)	53.5 (4)	54.1 (2)	53.5 (3)	51.8 (5)	47.9 (7)	51.3 (6)	
australian	83.9 (1)	83.8 (2)	83.8 (3)	83.8 (2)	83.0 (5)	83.6 (4)	79.8 (6)	71.5 (7)
BREAST	94.0 (3)	94.4 (1)	94.4 (2)	94.4 (1)	92.8 (4)	91.6 (5)	90.9 (6)	84.6 (7)
DWTC	69.3 (1)	65.3 (5)	65.9 (4)	63.4 (6)	67.8 (2)	52.8 (7)	66.1 (3)	50.1 (8)
fertility	88.2 (2)	87.8 (3)	87.7 (4)	87.8 (3)	87.0 (5)	88.2 (1)	86.8 (6)	86.8 (7)
flags	57.5 (1)	55.5 (6)	55.6 (5)	54.7 (7)	55.8 (4)	56.6 (2)	55.9 (3)	43.7 (8)
german	74.5 (2)	74.2 (4)	74.2 (5)	74.2 (4)	74.3 (3)	75.5 (1)	74.1 (6)	71.5 (7)
glass	68.9 (1)	67.6 (2)	67.4 (3)	66.6 (6)	67.3 (4)	66.3 (7)	67.1 (5)	40.6 (8)
heart	79.0 (1)	78.8 (3)	78.8 (5)	78.8 (3)	78.8 (4)	78.9 (2)	78.3 (6)	71.4 (7)
ionos	88.9 (1)	88.6 (2)	88.5 (3)	88.6 (2)	88.0 (5)	88.2 (4)	82.9 (6)	53.5 (7)
wine	95.2 (1)	94.8 (2)	94.6 (4)	94.6 (3)	94.4 (5)	94.3 (6)	93.5 (7)	84.9 (8)
zoo	93.7 (1)	93.3 (2)	92.9 (6)	93.2 (3)	93.1 (4)	92.8 (7)	93.1 (5)	92.7 (8)
mean %	74.5 (1)	73.6 (3)	73.7 (2)	73.2 (5)	73.5 (4)	71.7 (7)	72.4 (6)	58.7 (8)
mean (r)	1.38	3.15	3.85	3.77	3.92	4.62	5.23	7.54

expensive combination of both heuristics, and Query-by-committee (QBC) [16] a combination of the uncertainty of multiple learner models [10]³.

IMITAL learns a combination of the two heuristics informativeness and representativeness. For the datasets FERTILITY, FLAG, GERMAN, and HEART GD is much better than LC. This is an indication that on these datasets a pure informativeness heuristic is challenged the most, whereas for the other strategies LC still seems to be the safest bet as a general-purpose AL strategy. IMITAL successfully learned to combine the best of both strategies, which can be especially seen by the superior performance on the datasets. QBC achieved quite competitive results, but at the cost of almost twice as high running cost than IMITAL due to the expensive retraining of multiple learner models instead of a single one. The good results from the original QUIRE and BatchBALD paper could not be reproduced by us. Additionally, the runtime of QUIRE was so high that not even one AL experiment finished within seven days. The pre-selection of IMITAL with our used parameters means that IMITAL always considers a fixed amount of 40 unlabeled samples during each AL iteration, making it 10 times faster than even the second fastest LC strategy, which has to consider all unlabeled samples.

We also performed a significance test to prove that IMITAL is not only by chance but indeed statistically sound better than the competitors. We used a Wilcoxon signed-rank test [19] with a confidence interval of 95% to calculate the proportional win/tie/losses between IMITAL and each competing strategy.

³ We used for all strategies the implementations from the open-source AL framework ALiPy [18].

For each of the 1,000 starting points, we took the F1-values of all the 25 AL iterations⁴ of the two strategies to compare. Our null hypothesis is that the mean of both learning curves is identical. If the null hypothesis holds true we count this experiment repetition as a tie, and otherwise as a win or loss depending on which strategy performed according to the better mean. Due to lack of space we are omitting the table with the results of all datasets, but overall, IMITAL won at least 35% more often compared to each strategy than lost against them. It also has to be noticed that the majority of the direct comparisons resulted in a tie with a total amount of 55%.

5 Conclusion

We presented a novel approach of training a universally applicable AL query strategy on purely synthetic datasets by encoding AL as a listwise learning-to-rank problem. For training, we chose IL, as it is cheap to generate a huge amount of training data when relying on synthetic datasets. Our evaluation showed that IMITAL successfully learned to combine the two basic AL heuristics informativeness and representativeness by outperforming both heuristics and other AL strategies over multiple datasets of varying domains. In the future, we want to include more requirements of large ML projects into the state-encoding of IMITAL to make it more applicable.

Acknowledgements. This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) and the European Social Funds (ESF) within the “Innovations for Tomorrow’s Production, Services, and Work” Program (funding number 02L18B561) and implemented by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.

The authors are grateful to the Center for Information Services and High Performance Computing [Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH)] at TU Dresden for providing its facilities for high throughput calculations.

References

1. Dua, D., Graff, C.: UCI machine learning repository (2017)
2. Eberius, J., Braunschweig, K., Hentsch, M., Thiele, M., Ahmadov, A., Lehner, W.: Building the Dresden web table corpus: a classification approach, pp. 41–50, December 2015
3. Ebert, S., Fritz, M., Schiele, B.: Ralf: A reinforced active learning formulation for object class recognition. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3626–3633 (2012). <https://doi.org/10.1109/CVPR.2012.6248108>

⁴ As the exact p-values of the Wilcoxon signed-rank test are only computed for a sample size of up to 25, and for greater values an approximate – in our case not existent – normal distribution has to be assumed, we decided to stop our AL experiments after 25 iterations.

4. Guyon, I.: Design of experiments of the nips 2003 variable selection benchmark. In: NIPS Workshop on Feature Extraction and Feature Selection, vol. 253 (2003)
5. Guyon, I., Cawley, G., Dror, G., Lemaire, V.: Results of the active learning challenge. *J. Mach. Learn. Res. Proc. Track* **16**, 19–45 (2011)
6. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
7. Huang, S.j., Jin, R., Zhou, Z.H.: Active learning by querying informative and representative examples. In: Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., Culotta, A. (eds.) *Advances in Neural Information Processing Systems*, vol. 23, pp. 892–900. Curran Associates, Inc. (2010)
8. Kirsch, A., v. Amersfoort, J., Gal, Y.: BatchBALD: efficient and diverse batch acquisition for deep bayesian active learning. In: NIPS, vol. 32, pp. 7026–7037. Curran Associates, Inc. (2019)
9. Konyushkova, K., Sznitman, R., Fua, P.: Discovering general-purpose active learning strategies. arXiv preprint [arXiv:1810.04114](https://arxiv.org/abs/1810.04114) (2018)
10. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Croft, B.W., van Rijsbergen, C.J. (eds.) *SIGIR 1994*, pp. 3–12. Springer, London (1994). https://doi.org/10.1007/978-1-4471-2099-5_1
11. Liu, M., Buntine, W., Haffari, G.: Learning how to actively learn: a deep imitation learning approach. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia (Volume 1: Long Papers)*, pp. 1874–1883. Association for Computational Linguistics, July 2018. <https://doi.org/10.18653/v1/P18-1174>
12. Michie, D., Camacho, R.: Building symbolic representations of intuitive real-time skills from performance data. In: *Machine Intelligence*, vol. 13, pp. 385–418. Oxford University Press (1994)
13. Pang, K., Dong, M., Wu, Y., Hospedales, T.: Meta-learning transferable active learning policies by deep reinforcement learning. arXiv preprint [arXiv:1806.04798](https://arxiv.org/abs/1806.04798) (2018)
14. Ren, P., et al.: A survey of deep active learning. *ACM Comput. Surv. (CSUR)* **54**(9), 1–40 (2021)
15. Settles, B.: Active learning literature survey. *Computer Sciences Technical Report 1648* (2010)
16. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, New York, NY, USA*, pp. 287–294. COLT 1992, Association for Computing Machinery (1992). <https://doi.org/10.1145/130385.130417>
17. Shannon, C.E.: A mathematical theory of communication. *Bell Syst. Tech. J.* **27**(3), 379–423 (1948)
18. Tang, Y.P., Li, G.X., Huang, S.J.: ALiPy: active learning in Python. arXiv preprint [arXiv:1901.03802](https://arxiv.org/abs/1901.03802) (2019)
19. Wilcoxon, F.: Individual comparisons by ranking methods. *Biometrics Bull.* **1**(6), 80–83 (1945)
20. Zhan, X., Liu, H., Li, Q., Chan, A.B.: A comparative survey: benchmarking for pool-based active learning. In: *IJCAI*, pp. 4679–4686, August 2021. <https://doi.org/10.24963/ijcai.2021/634>, survey Track