ALWars: Combat-based Evaluation of Active Learning Strategies

Julius Gonsior[1][0000−0002−5985−4348], Jakob Krude[1], Janik Schönfelder[1], Maik Thiele[2][0000−0002−1665−977X], and Wolfgang Lehner[1][0000−0001−8107−2775]

[1] Technische Universität Dresden, Germany
`firstname.lastname@tu-dresden.de`
[2] Hochschule für Technik und Wirtschaft Dresden, Germany
`firstname.lastname@htw-dresden.de`

Abstract. The demand for annotated datasets for supervised *machine learning* (ML) projects is growing rapidly. Annotating a dataset often requires domain experts and is a timely and costly process. A premier method to reduce this overhead drastically is *Active Learning* (AL). Despite a tremendous potential for annotation cost savings, AL is still not used universally in ML projects. The large number of available AL strategies has significantly risen during the past years leading to an increased demand for thorough evaluations of AL strategies. Existing evaluations show in many cases contradicting results, without clear superior strategies. To help researchers in taming the AL zoo we present ALWars: an interactive system with a rich set of features to compare AL strategies in a novel replay view mode of all AL episodes with many available visualization and metrics. Under the hood we support a rich variety of AL strategies by supporting the API of the powerful AL framework ALiPy [21], amounting to over 25 AL strategies out-of-the-box.

Keywords: Active Learning, Python, GUI, Machine Learning, Demo

1 Introduction

Machine learning (ML) is a popular and powerful approach to deal with the rapidly increasing availability of otherwise unusable datasets. Usually, an annotated set of data is required for an initial training phase before being applicable. In order to gain high quality data, these annotation tasks need to be performed by domain experts, who unfortunately dispose of a limited amount of working time and who are costly. The standard approach to reduce human labor cost massively is *Active Learning* (AL). During recent years the amount of proposed AL strategies has increased significantly [3, 6, 7, 9, 11, 12, 20, 24]. Evaluations often show contradicting and mixed results, without any clearly superior strategies [13, 16]. Very often, the strategies struggle even in beating the most naïve baselines e. g. [4,5,9,11,12,22]. Also, most evaluations are based on simple learning curves and only give a glimpse of the possibilities to compare AL strategies. A very important, and often left out, aspect of AL is the time dependency of metrics and visualizations during the iterations of the AL loop. Often, strategies undergo a change during the AL cycles and should therefore not be judged in the light of the final result. We present therefore ALWars, an interactive demo application with a feature-rich *battle mode* to put AL strategies to the test in a novel and time-sensitive simulation replay mode. We included different metrics and visualization methods like the newly proposed *data maps* [19], classification boundaries, and manifold metrics in the comparison. We based our battle mode on top of the annotation web application Etikedi[3] which uses itself the popular AL framework ALiPy [21]. Thereby ALWars can compare over 25 AL strategies[4] out-of-the-box and can be easily extended by additional strategies.

2 Active Learning 101

AL is the process of iteratively selecting those documents to be labeled first that improve the quality of the classification model the most. The basic AL cycle starts with a small labeled dataset $\mathcal{L}$ and a large unlabeled dataset $\mathcal{U}$. In a first step, a learner model θ is trained on the labeled set $\mathcal{L}$. Subsequently, a query strategy selects a set of unlabeled samples $\mathcal{U}_q$ to be annotated by the domain experts. This cycle repeats until the annotation budget is exhausted. Thus, by using a clever AL strategy, many samples that are not adding significant value to the classification model can be left unlabeled, while still achieving the same classification quality. AL strategies often use the confidence of the learner model to select those samples, the model is most uncertain about [10, 14, 18], a query-by-committee approach combining the uncertainty of many learner models [17], or the diversity of the vector space [15]. There are also many more complex strategies that apply for example *Reinforcement Learning* or *Imitation Learning* and use deep neural networks at the core of AL strategies [1–3, 8, 9, 11, 12, 23].

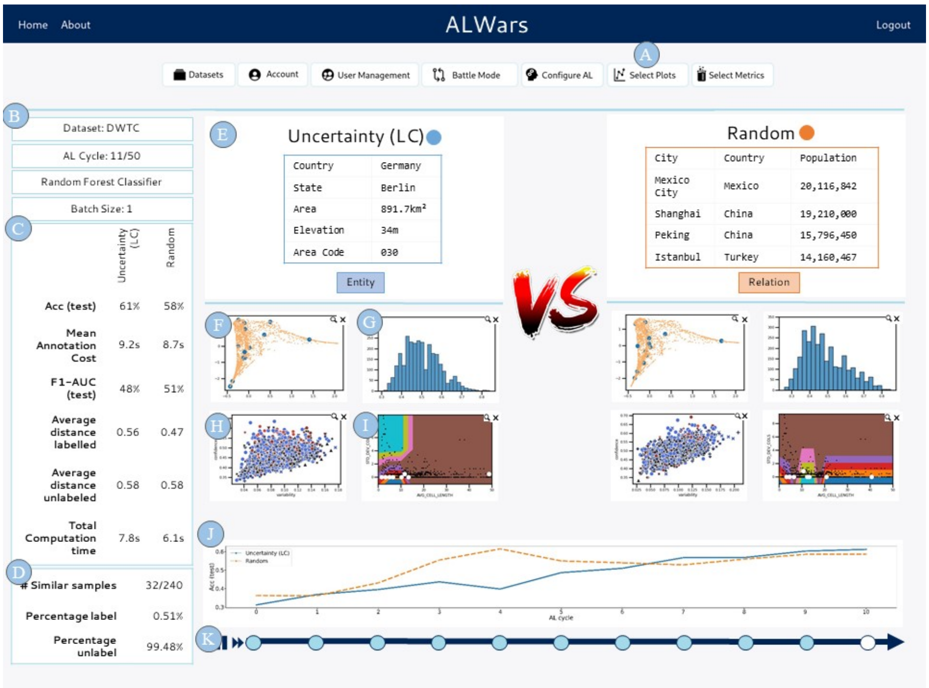
3 Battle Mode

The battle mode enables researchers to compare two AL strategies side-by-side by showing different plots and metrics for each AL cycle separately in a replay simulation. In the following, the possible metrics and visualization tools as well as their relevance to AL research are described, referring to the components shown in the exemplary battle in Figure 1:

Metrics: ALWars displays metrics calculated per each AL strategy (C) as well as metrics computed for both of them (D). The latter ones include the percentage of similar samples annotated by both AL strategies or the percentage of the labeled and unlabeled samples. Metrics calculated for both separately are standard ML metrics such as precision, recall, accuracy, or F1-Score, available for the training and the test dataset. All these metrics are also available in an

[3] https://github.com/etikedi/etikedi

[4] Note that BatchBALD [6] and LAL-RL [9] are, as of now, submitted by us as a Pull-Request to ALiPy, and are not yet part of the upstream AL framework.



Dataset: DWTC

AL Cycle: 11/50

Random Forest Classifier

Batch Size: 1

	Uncertainty (LC)	Random
Acc (test)	61%	58%
Mean Annotation Cost	9.2s	8.7s
F1-AUC (test)	48%	51%
Average distance labelled	0.56	0.47
Average distance unlabeled	0.58	0.58
Total Computation time	7.8s	6.1s

# Similar samples	32/240
Percentage label	0.51%
Percentage unlabel	99.48%

Country	Germany
State	Berlin
Area	891.7km²
Elevation	34m
Area Code	030

City	Country	Population
Mexico City	Mexico	20,116,842
Shanghai	China	19,210,000
Peking	China	15,796,450
Istanbul	Turkey	14,160,467

Fig. 1: Screenshot of ALWars between Uncertainty and Random

AUC-variant, defined as the proportion of the area under the AL learning curve with respect to the optimal learning curve, used as a summary representation of the learning curve. Interestingly, for the displayed example in Figure 1, the Uncertainty strategy is better than Random according to the final test accuracy, but worse according to the AUC-value, as Random performed much better for the early AL cycles. More advanced metrics are the mean annotation cost, the average distance in the vector space across all labeled or all unlabeled samples, the average uncertainty or confidence of the learner model for the training or test samples, or the total computation time of the AL strategies.

Learning Curves: The most common evaluation visualization found in AL papers are learning curves (J). The x-axis, often referred to as time, displays the AL cycles. The y-axis shows ML metrics such as accuracy or F1-score. Optimally, the learning curve goes straight up in the beginning and stays on top, maximizing the area under the curve.

Data Maps: A newly proposed visualization tool for datasets are so-called *Data Maps* [19] (H). In a data map are the mean and the standard deviation of the confidence of the learner model over all AL cycles so far, defined as *confidence* and *variability*, displayed in a scatter plot for all training samples. The percentage of correctness of the predictions of the learner model during all AL cycles is used as color encoding. Data maps can be used to locate three distinct sample regions: *easy-to-learn*, *ambiguous*, and *hard-to-learn* samples. For the displayed battle in Figure 1, it is apparent that the plot for Uncertainty contains more samples to the top left und less to the bottom left than Random, indicating a focus on labeling more easy-to-learn and less hard-to-learn samples.

Vector Space: The often high-dimensional feature vector space can be plotted using either manual selection of two important features, or automatic vector space transformation tools such as *PCA* or *t-SNE* as a 2D-plot (F). Color coded are the labeled and unlabeled samples, as well as the samples, which have been selected in the current AL cycle as $\mathcal{U}_q$. This visualization is useful to understand, if AL strategies focus more on specific regions in the vector space, or evenly distributed, as is the case for both strategies in the example screenshot.

Classification Boundaries: In addition to the 2D representation of the vector space the classification boundaries of the learner model can be included as a surface plot overlay in an additional plot (I). This plot is useful to analyze in depth how the learner model behaves regarding specific features.

Uncertainty Histogram: Similar to the classification boundaries plot, the uncertainty or confidence of the learner model can be displayed as a histogram for the training or test set (G). For the displayed example in Figure 1 the Uncertainty strategy leads to an overall slightly more confident learner model indicated by the flatter histogram in contrast to the Random strategy.

4 Demo Walkthrough

At the beginning, the visitors of ALWars are requested to select two AL strategies, to upload the evaluation dataset (if not already present on the server), to set common AL configuration options like the AL batch size, the learner model, the amount of AL cycles to simulate, the train-test split ratio, or to configure the desired plots and metrics (A). After the simulation is finished, the visitors of the demo are presented with the screenshot displayed in Figure 1. At its core the researchers can see the samples of $\mathcal{U}_q$ (E). Next to them are different plots and metrics about the current state of the AL strategies to be found. Using the timeline slider at the bottom (K) the users can navigate through the AL cycles of the simulation. The plots can be maximized to get a more detailed look at them, or they can be reconfigured to display f. e. different features. The used dataset and the current AL cycle are displayed to the left (B).

5 Conclusion

ALWars enables fellow AL researchers to gain a deep and novel understanding on how AL strategies behave differently over the course of all AL cycles by displaying metrics and visualizations separately for each AL cycle. This leads to unique and more detailed time-sensitive evaluations of AL strategies, helping researchers in deciding which AL strategies to use for their ML projects, and opening the door for further improved AL strategies.

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.

References

1. Bachman, P., Sordoni, A., Trischler, A.: Learning algorithms for active learning. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 301–310. PMLR, International Convention Centre, Sydney, Australia (06–11 Aug 2017)
2. Fang, M., Li, Y., Cohn, T.: Learning how to active learn: A deep reinforcement learning approach. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 595–605. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). https://doi.org/10.18653/v1/D17-1063
3. Gonsior, J., Thiele, M., Lehner, W.: Imital: Learning active learning strategies from synthetic data (2021)
4. Hsu, W.N., Lin, H.T.: Active learning by learning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. p. 26592665. AAAI'15, AAAI Press (2015)
5. 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)
6. 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)
7. Kirsch, A., Rainforth, T., Gal, Y.: Active learning under pool set distribution shift and noisy data. arXiv preprint arXiv:2106.11719 (2021)
8. Konyushkova, K., Sznitman, R., Fua, P.: Learning active learning from data. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30, pp. 4225–4235. Curran Associates, Inc. (2017)
9. Konyushkova, K., Sznitman, R., Fua, P.: Discovering general-purpose active learning strategies. arXiv preprint arXiv:1810.04114 (2018)
10. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: SIGIR '94. pp. 3–12. Springer London (1994)
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 (Volume 1: Long Papers). pp. 1874–1883. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1174
12. Pang, K., Dong, M., Wu, Y., Hospedales, T.: Meta-learning transferable active learning policies by deep reinforcement learning. arXiv preprint arXiv:1806.04798 (2018)

13. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X., Wang, X.: A survey of deep active learning. arXiv preprint arXiv:2009.00236 (2020)
14. Scheffer, T., Decomain, C., Wrobel, S.: Active hidden markov models for information extraction. In: Hoffmann, F., Hand, D.J., Adams, N., Fisher, D., Guimaraes, G. (eds.) Advances in Intelligent Data Analysis. pp. 309–318. Springer Berlin Heidelberg (2001)
15. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach (2018)
16. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648 (2010)
17. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. p. 287294. COLT '92, Association for Computing Machinery, New York, NY, USA (1992). https://doi.org/10.1145/130385.130417
18. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
19. Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N.A., Choi, Y.: Dataset cartography: Mapping and diagnosing datasets with training dynamics. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 9275–9293. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.746
20. Tang, Y.P., Huang, S.J.: Self-paced active learning: Query the right thing at the right time. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 5117–5124 (Jul 2019). https://doi.org/10.1609/aaai.v33i01.33015117
21. Tang, Y.P., Li, G.X., Huang, S.J.: Alipy: active learning in python. arXiv preprint arXiv:1901.03802 (2019)
22. Wang, Z., Ye, J.: Querying discriminative and representative samples for batch mode active learning. ACM Trans. Knowl. Discov. Data **9**(3) (Feb 2015). https://doi.org/10.1145/2700408
23. Woodward, M., Finn, C.: Active one-shot learning. arXiv preprint arXiv:1702.06559 (2017)
24. Zhang, M., Plank, B.: Cartography active learning. arXiv preprint arXiv:2109.04282 (2021)