ACTIVE LEARNING: AN EXPLORATORY STUDY OF ITS APPLICATION IN R

Andy Wang
Simon Fraser University
Burnaby, BC, Canada
zwa117@sfu.ca

April 25, 2021

ABSTRACT

In many real-world situations of machine learning, unlabeled data are available at low cost, but there is a paucity of labeled data and manually labeling them is expensive and time-consuming. This motivates the development of active learning, in which a learning algorithm can interactively query an oracle (a user or domain expert) to label new data with true labels. In statistics literature, active learning is also called optimal experimental design. Over the past decade, many successful active learning querying techniques have been developed and evaluated to be competitive. However, almost all of them are based on Python environment and only used in computer science industries. In this research project, we conduct an exploratory study of applying active learning in R and discuss its potential application in statistics. We implement some baseline strategies, carry out experiments on a famous machine learning dataset, illustrate their performance, and suggest future directions.

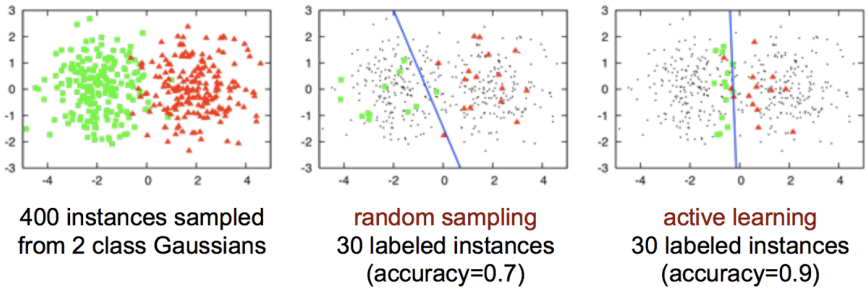
***K*eywords** Active Learning · Classification · Logistic Regression · R · Statistical Learning

1 Introduction

1.1 Why Active Learning in R

As a special case of machine learning, active learning can proactively pose queries to the oracle (e.g., a human annotator) for labels, in order to have the highest impact to train a supervised model iteratively [1]. This also means that active learning can achieve greater accuracy with fewer training labels, if it is allowed to choose the data from which it learns. Hence, active learning has been widely used in many applications including speech recognition, document classification, drug design using recombinant molecules, and protein engineering [2]. Figure 1 illustrates a simple scenario of applying active learning to a classification task. From the third plot, we can see that active learning reaches a good accuracy with a much smaller training set.

Figure 1: Illustration of How Active Learning Works (Source)



Although active learning is attracting more and more attention from industries, most active learning packages are developed in Python and/or Matlab environments, such as *modAL* [3] and *Libact* [4]. It is seldom used in statistics

because of the lack of appropriate packages, which also brings pain for new statisticians to enter this field. As R and R-studio construct a user-friendly platform for statistical learning, we would like to see if R language could similarly make active learning approachable. Meanwhile, we believe that active learning can be an interdisciplinary field to invite statisticians to make contributions. Currently, there are still many open questions and theoretical concerns.

Affiliation with $STAT403$ Besides the motivations mentioned above, this research project is also motivated and affiliated with course materials from $STAT403$ (Intermediate Sampling and Experimental Design):

- **Sampling**. We will introduce two common baseline sampling methods including Random Sampling.
- **Experimental Design**, as active learning is also treated as optimal experimental design in statistics literature.
- **Regression Models**. This project mainly relies on the logistic regression model to do classification tasks on the *Iris* data set (Tutorial 1).

1.2 Experimental Design and Dataset

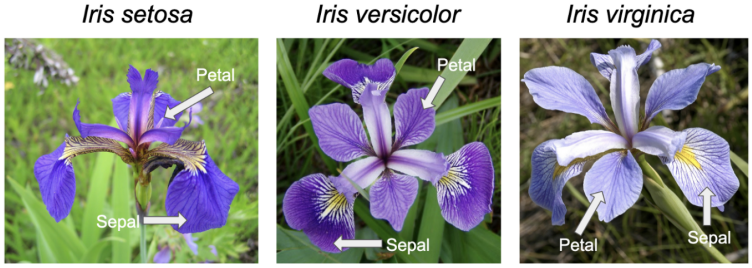
Based on a simple literature search, only one R package called "activelearning" was developed and posted in GitHub before. However, it was obsolete 5 years ago and currently there is no usable R package of active learning. In this case, we will bring our own active learning pipeline from scratch, implement two simple baseline querying strategies, and evaluate the performance using a famous machine learning dataset *Iris*.

The Iris flower dataset is a multivariate dataset introduced by Sir Ronald Aylmer Fisher, who is well-known as a statistician and geneticist. Iris contains three class labels: setosa, virginica and versicolor. There are 50 data entries for each class label. Each entry has four features regarding the width and length of petals and sepals.

Figure 2: Morphological Measures of Iris Flowers (Part of the Iris Dataset, Source & License)

Iris setosa				*Iris versicolor*				*Iris virginica*			
Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
5·1	3·5	1·4	0·2	7·0	3·2	4·7	1·4	6·3	3·3	6·0	2·5
4·9	3·0	1·4	0·2	6·4	3·2	4·5	1·5	5·8	2·7	5·1	1·9
4·7	3·2	1·3	0·2	6·9	3·1	4·9	1·5	7·1	3·0	5·9	2·1
4·6	3·1	1·5	0·2	5·5	2·3	4·0	1·3	6·3	2·9	5·6	1·8
5·0	3·6	1·4	0·2	6·5	2·8	4·6	1·5	6·5	3·0	5·8	2·2
5·4	3·9	1·7	0·4	5·7	2·8	4·5	1·3	7·6	3·0	6·6	2·1
4·6	3·4	1·4	0·3	6·3	3·3	4·7	1·6	4·9	2·5	4·5	1·7
5·0	3·4	1·5	0·2	4·9	2·4	3·3	1·0	7·3	2·9	6·3	1·8
4·4	2·9	1·4	0·2	6·6	2·9	4·6	1·3	6·7	2·5	5·8	1·8

Figure 3: Illustration of Iris Class Labels and Features. (Source)



1.3 Research Objectives

In this project, we aim to implement a simple active learning pipeline in R and mainly try to investigate the following three questions:

- How well does each active learning query strategies perform in the Iris experiment?
- What are the insights we can tell from the logistic regression classifier and these query strategies?
- Overall, is R a good platform to carry active learning experiments? What are its strengths and weaknesses compared to Python?

2 Method and Data Cleaning

2.1 Active Learning Criteria

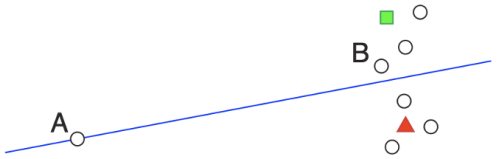
A realistic question is: what kind of sample is the most valuable for query strategies to call? Among various active learning methods, the most commonly used criteria for query functions are based on: *informativeness* and *representativeness* [5]. In detail:

- **Informativeness** measures how much uncertainty a queried sample has when predicting its label. The more uncertain it is, the more information it could potentially bring to change our model. For example, if logistic regression gives a probability $\tilde{0}.5$ on a sample, it means that the model is extremely uncertain on how to predict its label.
- **Representativeness** measures how similar a sample is compared to the other neighbor points. We expect the queried samples to be diverse and bring different information to our model. One example is the cluster representatives, which come from the densest data points.

  Note that **Random Sampling** is also a type of representative strategy. Although randomly selecting sounds a little intuitive, it actually brings some objective representativeness (global diversity; independent and identically distributed) to our classifier [6].

A strategy based on pure informativeness or representativeness may not always work the best. For instance, figure 4 illustrates a situation where pure informativeness (also called: **Uncertainty Sampling** strategy) is poor for classification [1]. Point A was called because it was on the decision boundary and hence, was the most uncertain. However, querying this outlier would not result in more information about the data distribution as a whole. It's better to call point B near the green cluster.

Figure 4: Illustration of Uncertainty Sampling Failure [1]



Usually, recent well-designed active learning strategies consider the trade-off between these two criteria, find a balance and dynamically query different type of samples. Due to the workload of complicated algorithms, in this project, we will only implement, compare, and discover insights from two baseline active learning strategies: **Uncertainty Sampling** (pure informative) and **Random Sampling** (a type of representative).

2.2 Classifier: Logistic Regression

We use logistic regression as the classifier, which is the most widely applied classifier in general and especially outside of machine learning in the applied sciences [7]. Many active learning benchmark surveys consider algorithms built on logistic regression. In part, logistic regression readily provides an estimate of the posterior class probability, which is often exploited in active learning with binary classification problems.

In a binary classification setting, logistic regression models a posterior probability:

$$P(y_i \mid x_i) = \frac{1}{1 + exp^{-y_i w^T x_i}}$$

where $x_i \in \mathbb{R}^d$ is a training feature vector labeled with $y_i \in \{0, 1\}$ and w is the d-dimensional parameter vector that is determined at training time. During training, we minimize the log-likelihood of the training data $\mathcal{L}$ to learn the model parameter w as follows:

$$min \frac{\lambda}{2} \parallel w \parallel^2 + \sum_{x_i \in \mathcal{L}} log(1 + exp^{-y_i w^T x_i})$$

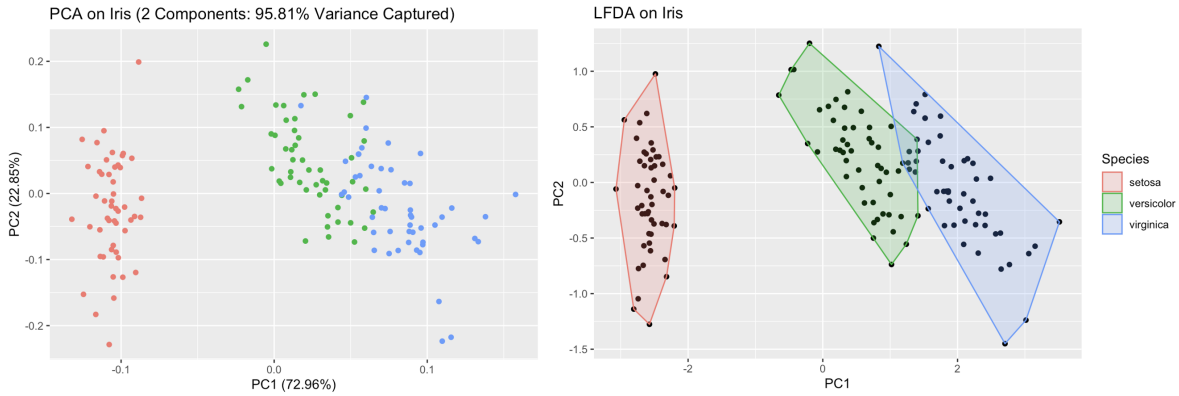
where $\parallel w \parallel^2$ is a regularization term for which λ controls its influence [7].

2.3 Data Preprocessing

In this section, we introduce how we prepare the multivariate Iris data to construct a binary classification problem. In general, the 150*4 data matrix is cleaned to 100*2 for use. The workflow and reasons are explained as follows.

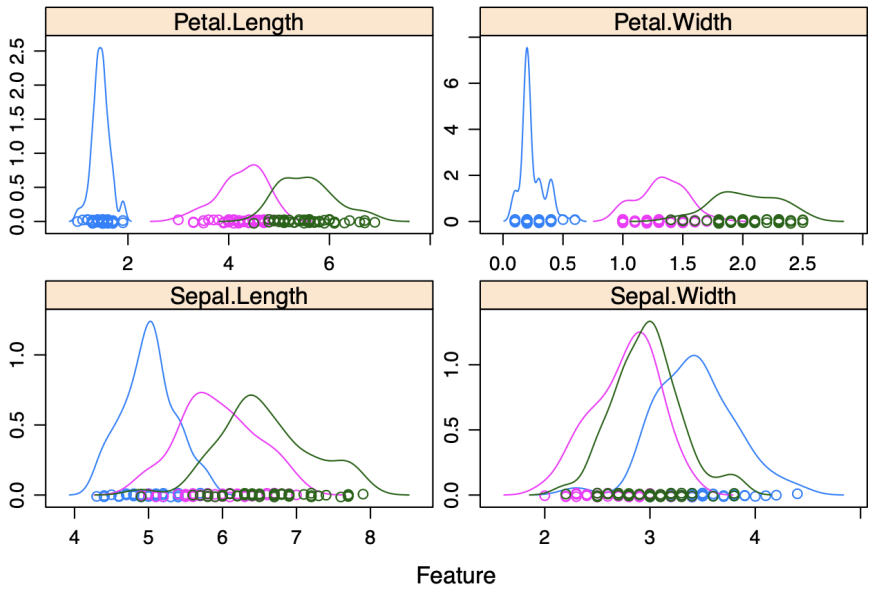
Dimensionality Reduction From the PCA plot Figure 5, 95.81% variance is explained by the first two principal components. Also, it shows a clear separation between the "setosa" group and the other two groups. The LFDA plot also confirms this finding. It will be less meaningful to have two completely separable classes. Hence, we select data entries from "versicolor" (green) and "virginica" (blue) groups to build a binary classification problem.

Figure 5: PCA and LFDA Plots on Iris



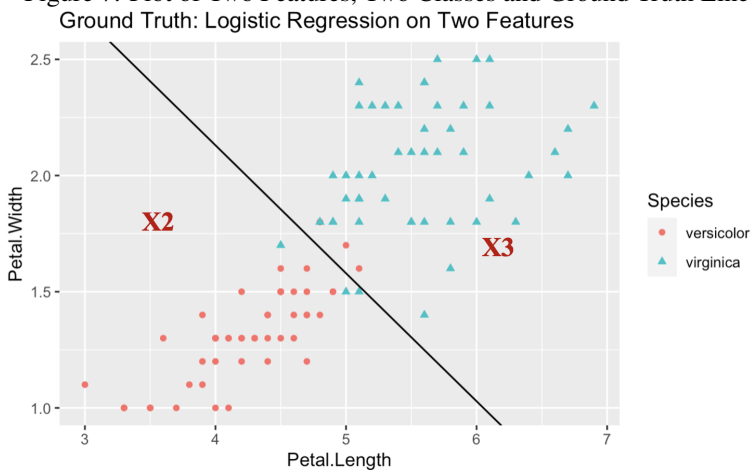
Variable Selection For visualization purposes, we want to select two features and use 2-D plots to illustrate how active learning works. From the feature plot Figure 6, "Petal.Length" and "Petal.Width" are ideal choices because they have moderate intersections for "versicolor" (red) and "virginica" (green) groups.

Figure 6: Feature Plot on Iris



Finally, we have an appropriate preprocessed dataset for active learning. We train a logistic regression classifier on the whole dataset and use this decision boundary as the *ground truth*. Figure 7 shows the plot distribution and we can see that these two groups can not be perfectly separated using a linear decision boundary.

Figure 7: Plot of Two Features, Two Classes and Ground Truth Line

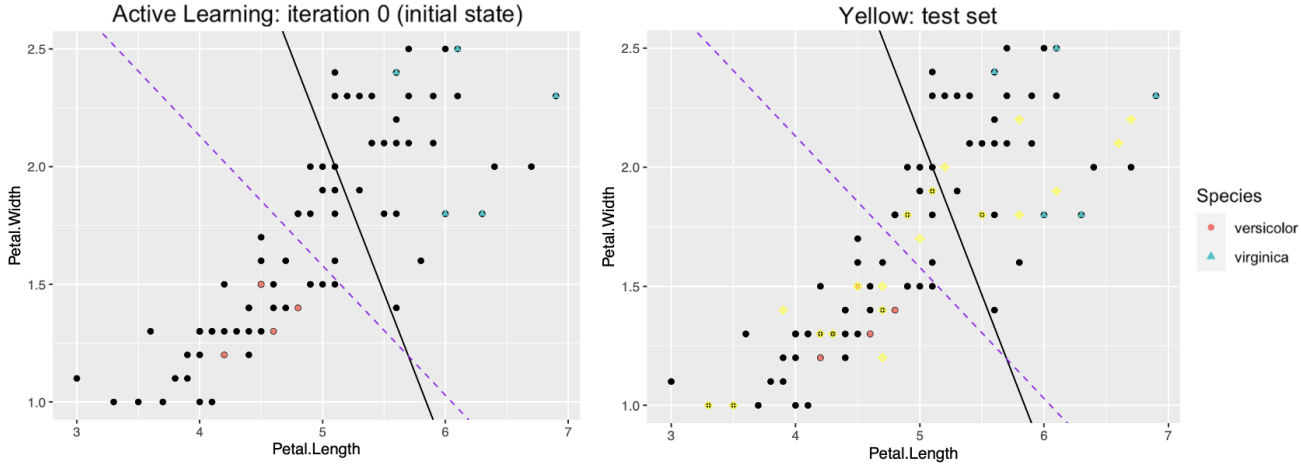


3 Simulations and Experiments

We split these 100 points into two parts: pool and test sets (80:20). Then, we split the pool to be labeled initial training set and unlabeled pool (10:70). Both splits of the dataset are dependent on the random state.

We start the active learning process by training an initial logistic regression classifier on these 10 points. Figure 8 shows the result of iteration 0 (test set is marked as yellow points on the right figure). We can see that the decision boundary deviates from the ground truth (Figure 7) a lot, which is reasonable because we only start with a small set.

Figure 8: Active Learning Initial State (iteration 0; with and without test set marked)



In each iteration of active learning, we query a point based on different query strategies, add it to the training set and re-train the logistic regression classifier. For **Uncertainty Sampling**, we select the point with least confidence, which is the point nearest to the current decision boundary. For **Random Sampling**, we query the point randomly from the unlabeled pool.

3.1 Active Learning: Uncertainty Sampling

Figure 9 illustrates how the decision boundary changes from iterations 1 to 3, querying by the uncertainty sampling strategy. The blue diamond symbol marks which points are queried in each iteration.

We can see that the predicted line changes a lot from the initial queried samples. This conforms to the logic of uncertainty sampling, as we expect it to bring the most information and dramatically change our model. However, we

also notice that some queried samples may not actually be useful. For example, the second queried sample makes the prediction boundary become worse and drops the classification accuracy.

Figure 9: Active Learning (Uncertainty) Iterations 1 to 3

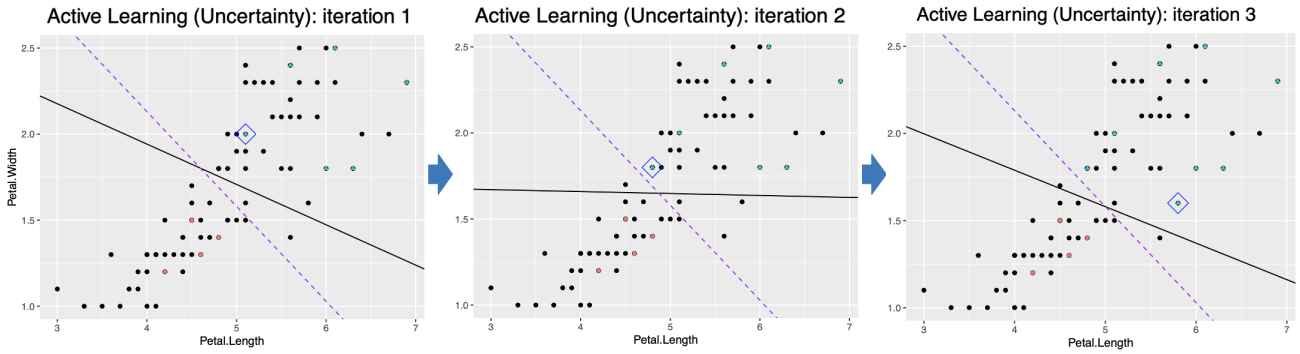
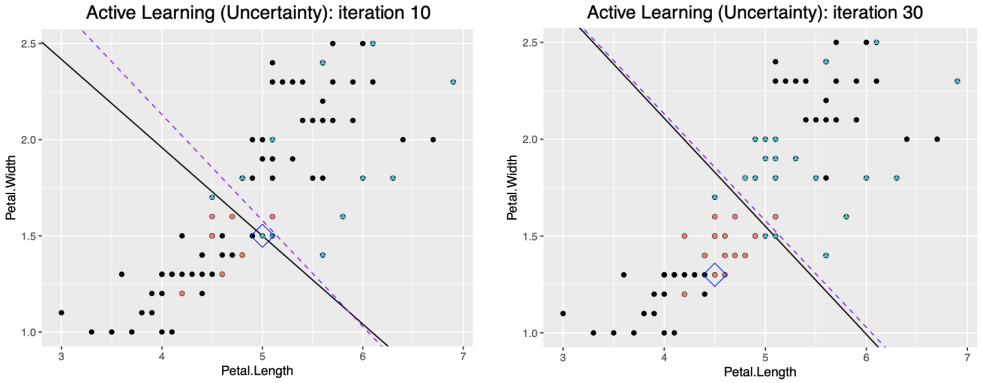


Figure 10 shows the active learning process in iterations 10 and 30. In iteration 10, the decision boundary of logistic regression is close to the ground truth. In iteration 30, the central, critical and most informative points are all queried and the prediction boundary is almost fixed. We use a smaller training set to achieve a pretty good classification model.

Figure 10: Active Learning (Uncertainty) Iterations 10 and 30



3.2 Active Learning: Random Sampling

We perform a similar experiment using the random sampling strategy. The initial state does not change. However, which point to query in each iteration is totally different from uncertainty sampling. Figure 11 shows the active learning process in iterations 1 to 3 and figure 12 is iterations 10 and 30.

We can see that the points are selected in random (blue diamond points) and the decision boundaries do not change too much. Even in iteration 10, the prediction boundary still looks far from the ground truth. Apparently, random sampling converges slower than uncertainty sampling and takes more time to reach the same level of classification accuracy.

3.3 Insights: Uncertainty vs. Random

Clearly, random sampling does not guarantee to be optimal because it does not pay any attention to whether the queried instance offers new information to the trained classifiers. As it is arbitrary, fairly simple and fast to implement, random sampling usually serves as the most common baseline that is used for benchmarking different querying strategies. The question is, *is uncertainty sampling always better than random sampling?*

In fact, random sampling is a fairly competitive strategy and **no active learning method can prevent performing worse than random sampling**. Although it is commonly believed that there should exist active learning techniques that perform *at least* as well as random sampling on average, and should in most circumstances outperform random sampling [8]. However, this is frequently contradicted by empirical evaluations from real-world settings. Interestingly, the

Figure 11: Active Learning (Random) Iterations 1 to 3

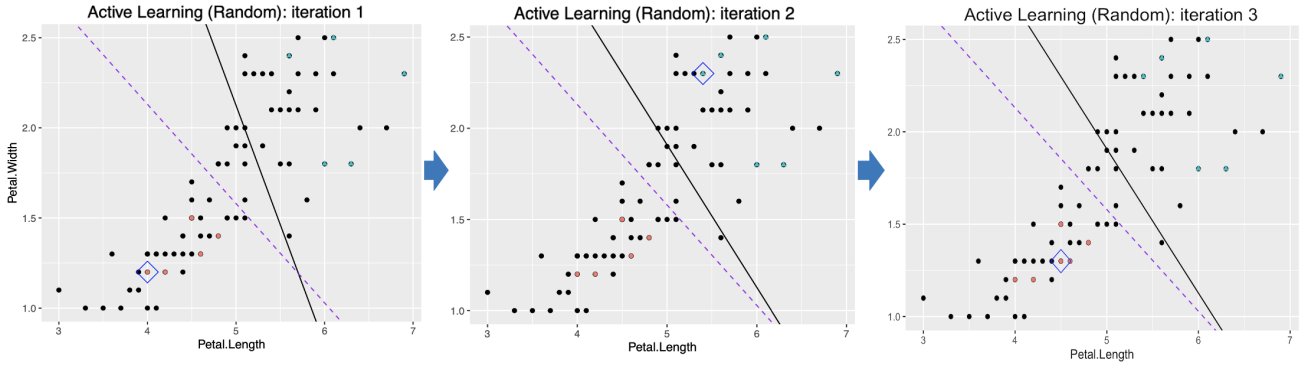
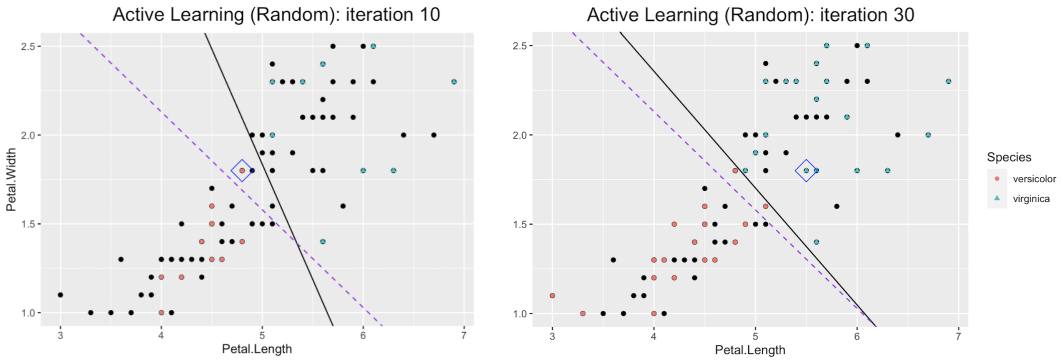


Figure 12: Active Learning (Random) Iterations 10 and 30



performance of random sampling in reality can often be competitive to the well-defined heuristic methods, depending on different experiments raised by several papers [7][9].

It is still an open question During the past decade, most research papers focused on proposing new, fancy and balanced active learning techniques. Whereas limited work attempts to look at why random sampling is practically so competitive. In active learning field, it is still a challenging question to find a *safe*, yet effective active learning method.

Insights from our experiment There are many interesting insights that we can tell from this simple experiment. The first thing we notice is that, different active learning methods change the model's behavior in different ways. For example, the hypotheses that we would expect are:

- A pure informative (uncertainty) queried sample could dramatically change the classification model (i.e., logistic regression) and predictions on the test set (Figure 9). However, it also has a chance to only bring a slight change, no change at all (Figure 4, uncertainty failure), or even some negative change (Figure 9, iteration 2).
- A pure representative queried sample cannot change the predictions dramatically, but will mostly bring gradual and small changes (Figure 11 and 12). It has a smaller probability of bringing no change.

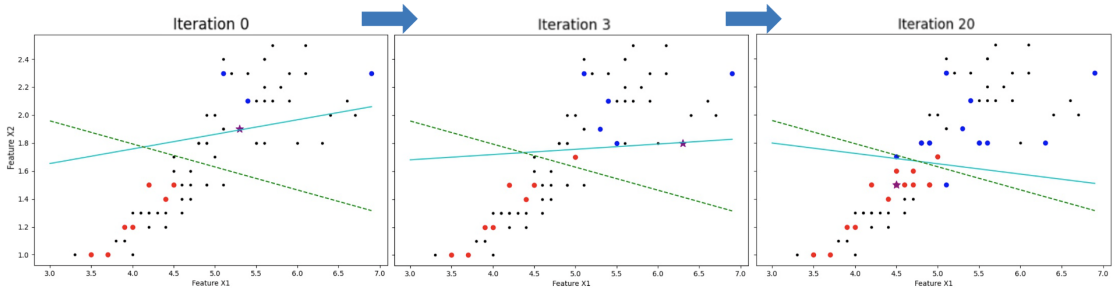
Our hypotheses help explain why random sampling has a chance to go beyond expectation: sometimes gradual improvements might be better than taking risks, which means that *some degree of randomization might be necessary when querying a sample*. This has been observed and implemented in some research. For example, this paper [10] finds that selecting the top-ranked query will not always be advantageous for heuristic active learning algorithms. Instead, they choose an item at random from the top $p\%$ from the pool.

3.4 Active Learning: R vs. Python

R and Python are two widely used open-source programming languages. They both hold a large community and new libraries are added continuously to their respective catalog. Python provides a smoother approach to learn data science while R is mainly used for statistical analysis. Unfortunately for active learning, almost all of the packages are currently based on Python environment and only used in computer science industries.

We also implemented a similar active learning workflow in Python. It takes around 1/4 fewer amount of code in Python, owe to its conciseness in list and *Pandas.dataframe* operations. Not surprisingly, there is no big difference from the final results and insights. Also, *ggplot2* and *matplotlib* both make visualization easy for us. Figure 13 illustrates how the cyan predicted boundary moves closer to the green truth boundary under uncertainty sampling iterations.

Figure 13: Example of Active Learning in Python



Although our experiment is simple and based on a toy dataset, implementing these active learning techniques in R does not look that brutal compared to Python. In a KDD2018 workshop "Active learning and transfer learning at scale with R and Python" [11], the authors also used both R and Python to deliver two excellent examples of active learning. They claimed that, "our exercises will emphasize approaches to interoperability between these languages so that we may use both environments toward a common goal".

So *why not R*? This question is kind of explicable if we go back to the origin of our inference: active learning is still a growing field without safe techniques built. There is no guarantee to be better than random sampling, and how to explain and use randomness is still an open question. Most developed methods claim to get over this constraint, but usually are shown to contradict when using different real-world datasets and/or classification models to test in Python. In this case, it is less necessary to implement the same experiment in R to get frustrating support.

On the other hand, we still see that there is a great potential to introduce active learning in R as well as further invite statisticians to contribute. Currently, there is no usable active learning packages in R. Implementing some representative active learning methods and bringing them to the R community can be a great topic for statistics master and Ph.D. students to contribute. From our project, we have not seen it to be significantly difficult. Meanwhile, as active learning belongs to optimal experimental designs, we believe experienced statisticians are able to help with the open questions in active learning, such as the random sampling issue we talked about in Section 3.3. Active learning is still a growing field and there is an exciting prospect of interdisciplinary collaborations.

4 Limitations and Future Work

Our empirical experiment and evaluation in R only include two of the most common active learning strategies (i.e., uncertainty sampling and random sampling) and only one classifier (i.e., logistic regression). There are numerous other active learning methods (e.g., query-by-committee [12], Expected Error Reduction [13], etc.) and numerous other classifiers (e.g., support vector machines, naive bayes, etc.) that we do not include in our study. Nonetheless, our comparison of two baseline techniques on a simple dataset reveals interesting insights on the active learning criteria.

Future work involves implementing more active learning methods and testing using different classifiers, which leads to many interesting avenues for further exploration. Another direction is to conduct experiments on different and high-dimensional datasets to exploit insights from the structure of data. Finally, it may be possible to extend the classification idea here to active learning for regression using different models.

5 Conclusion

In this research, we develop a simple active learning pipeline in R with two baseline methods, inspired by the background that almost all existing active learning methods are based on Python environment. For starters, we visualize and illustrate how different active learning methods work and how the classifier improves in each querying iteration. Moreover, our experiment indicates that different methods can change logistic regression's behavior differently. Inspired by these insights, we make several hypotheses on the randomization purpose in active learning and hope to raise potential awareness for future research. In spite of these observations, we present some promising directions for developing active learning in R community and leave some open questions to encourage statisticians to make interdisciplinary contributions.

6 Acknowledgements

I would like to show my gratitude to the course, $STAT403$ (Sampling and Experimental Design; by Prof. Liangliang Wang) for setting an open topic for me to explore active learning, which is honestly more interesting than I imagined. The main exploration about why randomness is so competitive is inspired by the randomization principle in experimental design.

I would also like to acknowledge the related courses: $CMPT419$ (Machine Learning; by Prof. Greg Mori) for a fresh introduction of Active Learning, $STAT452$ (Statistical Learning; by Prof. Tom Loughin) and $CMPT459$ (Data Mining; by Prof. Martin Ester), both for giving me a solid background in semi-supervised learning. I am immensely grateful to the great learning experience these two years at Simon Fraser University, especially during this difficult pandemic situation.

References

[1] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52, 07 2010.

[2] Isabelle Guyon, Gavin Cawley, Gideon Dror, and Vincent Lemaire. Results of the active learning challenge. *Journal of Machine Learning Research - Proceedings Track*, 16:19–45, 01 2011.

[3] Tivadar Danka and Peter Horvath. modAL: A modular active learning framework for Python. available on arXiv at `https://arxiv.org/abs/1805.00979`.

[4] Yao-Yuan Yang, Shao-Chuan Lee, Yu-An Chung, Tung-En Wu, Si-An Chen, and Hsuan-Tien Lin. libact: Pool-based active learning in python. Technical report, National Taiwan University, 10 2017. available as arXiv preprint `https://arxiv.org/abs/1710.00379`.

[5] S. Huang, R. Jin, and Z. Zhou. Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):1936–1949, 2014.

[6] Maria Ramirez-Loaiza, Manali Sharma, Geet Kumar, and Mustafa Bilgic. Active learning: an empirical study of common baselines. *Data Mining and Knowledge Discovery*, 31, 03 2017.

[7] Yazhou Yang and Marco Loog. A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83:401–415, 2018.

[8] Y. Freund, H. Seung, E. Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 2004.

[9] Gavin C. Cawley. Baseline methods for active learning. In Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov, editors, *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pages 47–57, Sardinia, Italy, 16 May 2011. JMLR Workshop and Conference Proceedings.

[10] Dominic Mazzoni, Kiri L. Wagstaff, and Michael C. Burl. Active learning with irrelevant examples. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 695–702, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[11] John-mark Agosta, Olga Liakhovich, Horton Robert, Mario Inchiosa, and et al. Active learning and transfer learning at scale with r and python. *SIGKDD 2018*, Apr 2018.

[12] Naoki Abe and Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. pages 1–9, 01 1998.

[13] Nicholas Roy and Andrew Mccallum. Toward optimal active learning through sampling estimation of error reduction. *Proceedings of the 18th International Conference on Machine Learning*, 08 2001.