Exploratory Training: When Trainers Learn

Omeed Habibelahian
Oregon State University
habibelo@oregonstate.edu

Rajesh Shrestha
Oregon State University
shresthr@oregonstate.edu

Arash Termehchy
Oregon State University
termehca@oregonstate.edu

Paolo Papotti
EURECOM
papotti@eurecom.fr

ABSTRACT

Data systems often actively present examples and solicit labels from users to learn a target concept, i.e., active learning. Current systems assume that users always provide correct labeling with potentially a fixed and small chance of mistake. Nevertheless, users may have to learn about and explore the underlying data to label examples correctly, particularly for complex target concepts and models. For example, to provide accurate labeling for a model of detecting noisy or abnormal values, users might need to investigate the underlying data to understand typical and clean values in the data. As users gradually learn about the target concept and data, they may revise their labeling strategies. Due to the significance and non-stationarity of errors in this setting, current systems may use incorrect labels and learn inaccurate models from the users. We report the preliminary results of our user studies over real-world datasets on modeling human learning during training the system and layout the next steps in this investigation.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**.

ACM Reference Format:
Omeed Habibelahian, Rajesh Shrestha, Arash Termehchy, and Paolo Papotti. 2021. Exploratory Training: When Trainers Learn. In *SIGMOD*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

SIGMOD, 2022, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Data & AI Systems often present training examples and solicit labels from humans to learn a model, i.e., *active learning* [1, 4, 7, 12, 13]. For instance, data systems often learn data quality constraints and models by presenting chosen examples to the human experts [7]. In each interaction, the system can select an example for labeling based on the training data already given by the user and the effectiveness of the current model. This approach improves the flexibility of choosing examples and may significantly reduce the training data [1].

Due to the sheer volume and complexity of data, data labeling and model training over large datasets may be interactive and exploratory. Users interactively inspect the data to improve their knowledge, refine their inputs, and revise their hypotheses about the target insight based on the results of preceding interactions. Thus, to label examples correctly, users often have to learn about the underlying data, particularly for complex models. Consider an epidemiologist who investigates a new dataset of numerous recent case reports and relevant context information, e.g., locations and environments, about patients to build a model that checks whether a virus is airborne. They may use a labeling system to provide training information for building the ML model. They may initially inform the system that the case report of a certain patient being infected in a movie theater is strong evidence of a virus being airborne. The system may use this information to build a model to predict or find similar cases. But, after some interactions, the epidemiologist may find out that theaters in the patient's region often show movies in open spaces and usually serve food. This new information may make the label for the original case report a false positive. Similarly, to provide accurate labeling for a model of detecting noisy or abnormal values, users may need to explore the data to understand typical and clean values. It is often vital for users to train accurate models fast. It may save many lives if an epidemiologist predicts whether a new virus is airborne soon.

As users explore and interact with the data, their knowledge about the data and the target concept evolves. Thus, they may modify their labeling and training strategies. It is known that human learning in interactive settings is highly non-stationary [3, 5, 10, 11, 14, 15]. For example, they may have a prior belief (distribution) about the properties of positive or negative examples for the target concept based on their background information and expertise. They may continually and gradually revise their belief and in turn labeling strategies after observing new data items, and settle on a relatively fixed strategy after gaining sufficient knowledge about their actions and environment [3, 5, 9, 11, 14]. Due to the huge volume of the data and complexity of the target insight, this learning process may take a relatively long time. Thus, it is challenging for the system to understand or find reliable users' inputs in the interaction to learn an effective model for their target concepts.

Nonetheless, current systems often assume that users have perfect knowledge about the target insight and underlying data. They usually assume that users provide reliable feedback or training data for target insights with potentially a very small and fixed amount of noise [1, 4, 6]. However, as users' knowledge about the data and insight may significantly change during the analysis, it is not clear which inputs are sufficiently reliable for predicting the target insights. Due to the significance and non-stationarity of errors, current systems will use incorrect labels and learn inaccurate models. Moreover, current systems use fixed, stationary, and/or pretrained models of users' exploration and training process to build a model for the target concept [1, 8, 16]. Stationary learners are unduly biased to the initial observations and fail to adapt subsequently.

As the first step to realize the vision of fast and effective exploratory training, we report our user studies on modeling human learning over real-world datasets. Such a model will enable designing algorithms that learn the target concepts effectively and quickly in exploratory training. We show that human learning in this context accurately follow the well-known Bayesian learning model, which has also been observed to effectively model human learning in other domains, such as economics and image classification [14, 15].

2 USER LEARNING SCHEMES

Human learning is generally categorized as *model-free*, i.e., reinforcement based, and *model-based* learning, i.e., belief learning [5, 14, 15]. In model-free learning, users modify their strategies by directly reinforcing actions. In model-based one, they form hypotheses about the data and actions of the system and update their strategies given the success of their preceding predictions [14, 15].

In the type of interaction we investigate, the system presents a set of unlabeled examples to the user. It is reasonable to assume that the user has some initial hypotheses about the target concept, which they may repeatedly revise after viewing the presented examples in each interaction. Also, in our setting, systems usually interact with domain experts who often have some initial models of the domain. Thus, model-based learning schemes are natural choices for the cases where users are training a system interactively [5, 14, 15].

We focus on mainly the two prominent model-based learning methods and their variations: *hypothesis testing* and *Bayesian learning* [14, 15].

In **hypothesis testing**, the user randomly selects a hypothesis according to how well it explains the observed data and uses it to label examples. For instance, if they view sufficiently many examples with value "NaN" in the interaction, they conclude that "NaN" is not noise and is used by the database designer to represent unknown values. The user frequently evaluates the performance of her hypotheses, rejects the current one if it does not explain sufficient amount of recent data with some tolerance, and picks another hypothesis randomly based on its relative performance [15].

In **Bayesian learning**, the user starts with a prior about different models for the target concept [15]. They then performs Bayesian updating on the probability of choosing each hypothesis given the examples presented to them in each interaction. They picks a hypothesis according to these probabilities and labels examples accordingly.

3 EMPIRICAL STUDY

3.1 Task: Detecting FD Violations

In this paper, we focus on modeling human learning in the context of data quality constraints and discovering violations of those rules. Our work is focused towards modeling the learning process of humans in rule discovery with two primary models: *Bayesian learning* and *Learning based on hypothesis testing*. We address our research questions by analyzing how the user iteratively navigates the space of applicable FDs in several scenarios. In addition to evaluating our models' ability to recreate the participants' thinking process as they hypothesize functional dependencies throughout the interaction, we also evaluate the user's process of identifying and marking FD violations in the data.

We first define functional dependencies and their violations, setup, and the process of interactively labeling data.

3.1.1 Functional Dependency. A **functional dependency (FD)** is a pattern that explains how attributes in a dataset relate to and are dependent on one another. For example, in datasets containing addresses or other location data, an address's city and ZIP code determine the state that location is in. Another example is a person's SSN determining a range of personally identifiable information, such as the person's name, date of birth, and address. These examples can be represented as $(city, zip) \Rightarrow state$ and $SSN \Rightarrow (name, dob, address)$, respectively.

3.1.2 Violation of an FD. If a tuple has one or more cell(s) with values that do not align with the rest of the dataset with respect to an FD, then the tuple is said to violate the FD or referred to as an exception to the FD.

3.2 Experimental Setting

Our empirical study involved a participant group of 20 students at Oregon State University with different background in Computer Science and Data Science. We implemented a user interface that refreshes user on the definition of FDs and their violations, tests their knowledge and presents samples iteratively for interaction and rule specification.

3.2.1 Datasets. We used synthetic variations of two real-world datasets in our experiments: **AIRPORT** and **OMDB**. AIRPORT describes various airports, heliports, and seaplane bases throughout the U.S. state of Alaska whereas OMDB (Open Movie Database) contains information about various English-language movies and TV shows. The exact attributes present in each dataset vary slightly by scenario, but each scenario has around 200 tuples. We utilized 5 total configurations of these two datasets which are shown in table 1.

#	Domain	Attributes	FD
1	Airport	facilityname type manager	Target: *(facilityname, type)* ⇒ *manager* Alternative: *facilityname* ⇒ *(type, manager)*
2	Airport	sitenumber facilityname owner manager	Target: *sitenumber* ⇒ *(facilityname, owner, manager)* Alternative: *facilityname* ⇒ *(sitenumber, owner, manager)*
3	Airport	facilityname owner manager	Target: *manager* ⇒ *owner* Alternative: *facilityname* ⇒ *(owner, manager)*
4	OMDB	title year genre type	Target: *(title, year)* ⇒ *(type, genre)* Alternative: *title* ⇒ *(year, type, genre)*
5	OMDB	title rating type	Target: *rating* ⇒ *type* Alternative: *title* ⇒ *(rating, type)*

Table 1: Scenarios used in the empirical study

3.2.2 Introducing Violations. The BART error generation tool [2] is used to introduce violations to datasets by scrambling values with respect to the target FD as shown in table 1. This type of introduction of target FD violations may further induce violations to other FDs too.

An m:n violation ratio is referred to as the proportion of m target hypothesis violations and n alternative hypothesis violations. This ratio help to reveal if the learning process depends on the quantity of violations present in the samples and determine the FDs associated with violations.

3.2.3 Interactions. In the beginning, the user is presented with schema and asked what they believe is the FD that holds over the entire dataset with the fewest exceptions. In each iteration, a random sample of data is served allowing the user to mark any violations of their hypothesized FD within that sample. After marking violations, they confirm or modify their hypothesized FD. The user must interact for at least 9 such iterations for completion and can continue up to at most 15 iterations.

In order to avoid an high cognitive load on our participants, we focused on simple datasets with few attributes. However, at the same time, they were complex enough to avoid the case of being able to immediately figure out the underlying rule without having to interact with the data.

3.3 Model Configuration

We evaluate both our Bayesian model and our hypothesis testing model in terms of **Mean Reciprocal Rank (MRR)** for an exact match with the participant's submitted hypothesis in the model's top-k suggestions in each iteration. MRR is used to evaluate a process which produces a list of results and it is the average of the reciprocal of the rank of true value. We use the reciprocal of the position of user specified hypothesis in the ranks provided by the models.

In order to evaluate which window for top suggestions better models user behavior, we compare each model's performance with different values of k : $k = 1$, $k = 3$ and $k = 5$, which corresponds with how many FDs are output by the model in each iteration. We therefore refer to our Bayesian and hypothesis testing (HP) model configurations as BAYES-k and HP-k. We also compare each of these models to their respective variants that consider subset and superset relationships in the model's top suggestions rather than only a perfect FD match. In our results, models with this configuration include a "+" in their names.

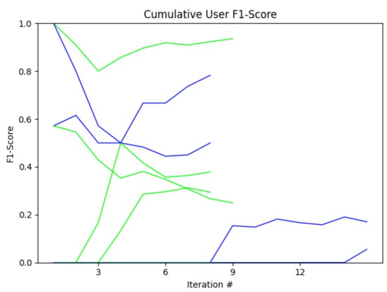
3.4 Experimental Results

3.4.1 Measuring Learning. To quantify learning, we first review the extent to which the f_1-score of the user's hypothesis changes between iterations. Finally, we look for any positive trends in user performance.

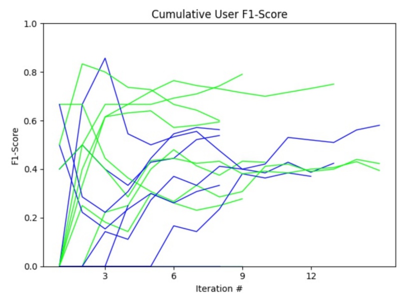
Scenario #	Average change in f_1-score
1	0.1144
2	0.3280
3	0.2301
4	0.2843
5	0.1767

Table 2: Average f_1-score change between any two rounds of user labeling, grouped by scenario

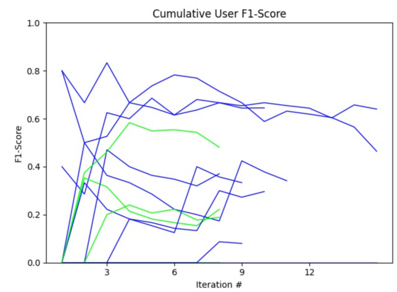
The average change in f_1-score is positive and relatively large as shown in table 2. This suggests that changes in the



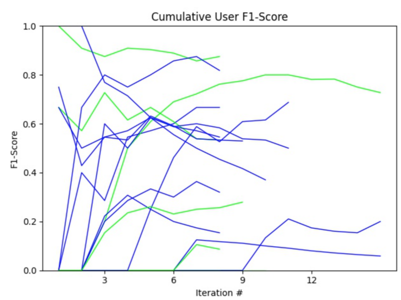
(a) Scenario 1



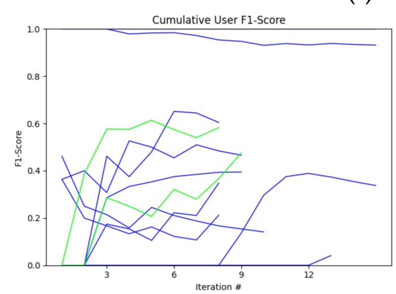
(b) Scenario 2



(c) Scenario 3



(d) Scenario 4



(e) Scenario 5

Figure 1: Each participant's change in labeling performance, measured by f1-score. Blue lines represent interactions with a 3:1 violation ratio. Green lines represent interactions with a 3:2 violation ratio

user's hypothesis from iteration to iteration are likely not due to simple noise and error, but is rather due to the user actually learning and updating their hypothesis.

Next, we analyze the users' labeling activity across each interaction to evaluate how their hypothesis-forming process evolves as they are exposed to more data and continue interacting with them. The result in **Figure 1** reveals the user's ability, to detect and mark violations of the target FD, improving as the interaction proceeds.

3.4.2 Evaluating Learning Models. Now that we can see user performance improve over time, we turn to our models to measure their ability to replicate the users' hypotheses over the course of the interaction. We use reward based on *MRR* metric, specifically on n (the ranking of the FD in the output), to evaluate the models.

$$reward_{MRR} = \begin{cases} \frac{1}{n}, & \text{if the model's output includes user's FD} \\ 0, & \text{otherwise} \end{cases}$$

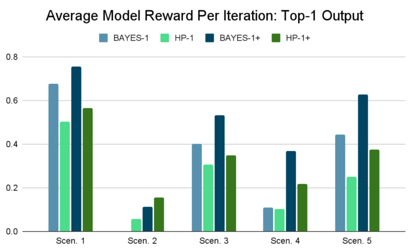
In comparison to the hypothesis testing model, our Bayesian model acquires better modeling performance with higher reward per iteration on average, in almost all the scenarios and value of k (**Figure 2**). As the value of k increases, the performance of all model configurations improve as expected, as this gives the model more opportunities to return an FD that matches a user's hypothesis.

Our empirical results show that as humans interact with more data, their ability to accurately label violations in the data and specify qualitative hypothesis of data rule increases. This is due to the user iteratively learning and refining their understanding of the data. Furthermore, the user's learning process follows a probabilistic Bayesian learning model when the model's measure of success is its ability to perfectly match the user's hypothesis. However, when the measure of success is loosened to accept variations of the user's hypothesis as viable matches, the hypothesis testing model also mimics user behavior fairly well.

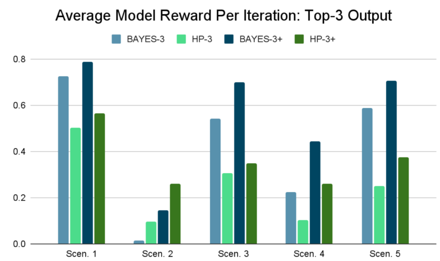
4 LIMITATIONS & CHALLENGES

Generalizable Modeling of Human Learning. To validate further our assumptions on the process and model of human learning, we plan to perform additional user studies on other tasks. We also plan to investigate the characteristics of the target concept, e.g., difficulty of the concept, that may cause users to prefer certain learning method(s) or hyper-parameters, e.g., tolerance in hypothesis testing.

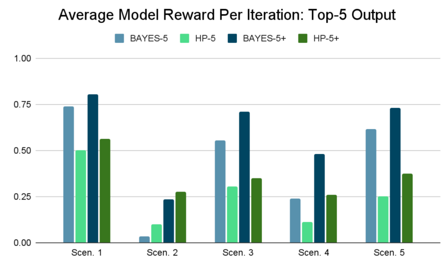
Collaborate With Learning Humans. We plan to design algorithms to accurately learn the target concept from a learning trainer. This approach views exploratory training as the interactive collaboration user and data/AI system, for achieving the common goal of learning users' desired concept using the fewest possible interactions.



(a) k=1



(b) k=3



(c) k=5

Figure 2: Average reward per iteration for each learning model with k

REFERENCES

[1] Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S. Yu. 2014. Active Learning: A Survey. In *Data Classification: Algorithms and Applications*, Charu C. Aggarwal (Ed.). CRC Press, 571–606. http://www.crcnetbase.com/doi/abs/10.1201/b17320-23

[2] Patricia C. Arocena, Boris Glavic, Giansalvatore Mecca, Renée J. Miller, Paolo Papotti, and Donatello Santoro. 2015. Messing up with BART: Error Generation for Evaluating Data-Cleaning Algorithms. *Proc. VLDB Endow.* 9, 2 (oct 2015), 36–47. https://doi.org/10.14778/2850578.2850579

[3] Colin F. Camerer, Teck-Hua Ho, and Juin Kuan Chong. 2004. Behavioural Game Theory: Thinking, Learning and Teaching. In *Advances in understanding strategic behaviour : game theory, experiments, and bounded rationality*. Palgrave Macmillan, 120–180.

[4] Kyriaki Dimitriadou, Olga Papaemmanouil, and Yanlei Diao. 2014. Explore-by-example: an automatic query steering framework for interactive data exploration. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, Curtis E. Dyreson, Feifei Li, and M. Tamer Özsu (Eds.). ACM, 517–528. https://doi.org/10.1145/2588555.2610523

[5] Samuel J. Gershman, Eric J. Horvitz, and Joshua B. Tenenbaum. 2015. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science* 349, 6245 (2015), 273–278. https://doi.org/10.1126/science.aac6076 arXiv:https://science.sciencemag.org/content/349/6245/273.full.pdf

[6] Daniel Golovin, Andreas Krause, and Debajyoti Ray. 2010. Near-Optimal Bayesian Active Learning with Noisy Observations. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1* (Vancouver, British Columbia, Canada) *(NIPS'10)*. Curran Associates Inc., Red Hook, NY, USA, 766–774.

[7] Jian He, Enzo Veltri, Donatello Santoro, Guoliang Li, Giansalvatore Mecca, Paolo Papotti, and Nan Tang. 2016. Interactive and Deterministic Data Cleaning. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, Fatma Özcan, Georgia Koutrika, and Sam Madden (Eds.). ACM, 893–907. https://doi.org/10.1145/2882903.2915242

[8] Rui Li, Rui Guo, Zhenquan Xu, and Wei Feng. 2012. A prefetching model based on access popularity for geospatial data in a cluster-based caching system. *International Journal of Geographical Information Science* 26, 10 (2012), 1831–1844.

[9] Ben McCamish, Vahid Ghadakchi, Arash Termehchy, Behrouz Touri, and Liang Huang. 2018. The Data Interaction Game. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. ACM, New York, NY, USA, 83–98. https://doi.org/10.1145/3183713.3196899

[10] Yael Niv. 2009. The Neuroscience of Reinforcement Learning. In *ICML*.

[11] Y Niv. 2009. Reinforcement learning in the brain. *The Journal of Mathematical Psychology* 53, 3 (2009), 139–154.

[12] Burr Settles. 2009. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison. http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf

[13] Burr Settles. 2012. *Active Learning*. Morgan & Claypool Publishers.

[14] Joshua B. Tenenbaum. 1999. Bayesian Modeling of Human Concept Learning. In *Advances in Neural Information Processing Systems 11*, M. J. Kearns, S. A. Solla, and D. A. Cohn (Eds.). MIT Press, 59–68. http://papers.nips.cc/paper/1542-bayesian-modeling-of-human-concept-learning.pdf

[15] H Peyton Young. 2004. *Strategic learning and its limits*. OUP Oxford.

[16] Ugur Çetintemel, Mitch Cherniack, Justin DeBrabant, Yanlei Diao, Kyriaki Dimitriadou, Alexander Kalinin, Olga Papaemmanouil, and Stanley B. Zdonik. 2013. Query Steering for Interactive Data Exploration. In *CIDR*.