Uncertainty Based Active Learning Strategy for Interactive Weakly Supervised Learning through Data Programming

Anonymous Author(s)
Affiliation
Address
email

Abstract

Easy to build and reliable machine learning models are what all data analysts want. Although machine learning is advancing daily, the labeling cost for supervised learning and the black-box nature of machine learning are the main obstacles to its further diffusion. As a method of reducing the labeling cost without increasing the black-box nature, weakly supervised learning, especially data programming, is gaining attention. Its advantage is due to labeling functions that domain experts create based on their knowledge instead of labeling each data point manually. However, data programming alone cannot reduce the actual process cost. This is because domain experts have to carry out a full search in their mind and endlessly implement labeling functions without any insight into what unimplemented labeling functions will be effective. We propose an active learning strategy for interactive weakly supervised learning with labeling functions to solve this problem. The proposed method iteratively presents a small number of highly prioritized data points to be labeled by additional labeling functions considering the uncertainty of predictions. With this method, domain experts need to only implement their knowledge that can be applied to a small number of presented data points as a labeling function. We also verified the effectiveness of this method through a six-class text classification task. The experimental results indicate the effectiveness of the method and its high potential in a machine learning implementation.

1 Introduction

Longstanding technical improvements in machine learning, especially in supervised learning, have had practical results in both academic and industrial fields. Wide applications of machine learning have also highlighted potential problems such as heavy labeling cost and difficulty in interpreting its black-box nature. Solving these problems will lead to a reduction in the resistance to implementing machine learning and further contributions to a broader and deeper field.

A variety of methods using few-shot learning and transfer learning have been proposed as machine-centric methods to solve the labeling-cost problem [10, 22]. It is true that these methods are achieving much success, but the effects of transfer and few-shot are not yet sufficiently understood and may also have some limitations [15, 3]. First, their main scopes are limited to computer vision or natural language processing. The existence of large amounts of labeled data or trained neural networks in similar domains is assumed with these methods. Moreover, they can unintentionally enhance the black-box nature of machine learning. Their accuracy and reliability depend on domain similarity and the distribution of a small number of labeled data. Since the interpretability of these sources is lower than that of a large amount of labeled data in simple supervised learning, the path

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

to interpreting the machine learning results becomes longer and more challenging. Because of these limitations, these methods are not yet widely used in the wild.

Weakly supervised learning is another notable human-centric method for these problems [6]. It does not require domain experts to label each data point directly but indirectly with their knowledge related to the data points. In particular, data programming [16, 2] has gained much attention as a weakly supervised learning method to reduce labeling cost by using human knowledge. Data programming probabilistically labels data points based on the collective knowledge of human-defined labeling functions. Such a method is very useful in many practical situations where domain experts have spent much time on redundant manual labeling without limiting the application field. This is because domain experts use their own implicit rules for the manual labeling process, and these rules are likely to be implemented as labeling functions. Moreover, this human-centric method based on domain experts' knowledge may reduce but never increase the black-box nature of machine learning, unlike the above machine-centric methods. This method only changes the way to assign labels from direct to indirect and makes it possible to understand the simple reasons the labels are assigned from labeling functions.

However, data programming alone does not completely reduce the manual labeling process cost. Domain experts have to perform a full search in their mind for finding their knowledge that can be applied to any data points in a huge pool, and endlessly implement labeling functions. This is attributed to the lack of criteria for which knowledge they should implement as a labeling function to effectively train machine learning models. The missing key is active learning [19, 23]. In traditional supervised learning, active learning iteratively searches and presents unlabeled data points that should be labeled to efficiently improve the accuracy of the model. If the subjects of this search can be replaced with unimplemented labeling functions from unlabeled data points, the workload of domain experts could be reduced sufficiently.

Our research objective is to develop an active learning strategy for interactive weakly supervised learning, in which machines present information about the knowledge that should be incorporated in data programming and domain experts interactively implement labeling functions based on the information. Specifically, machines calculate an uncertainty-based acquisition function for each data point from the probabilistic labeling result of the implemented labeling functions and the output probabilities of the classification model learned on the training dataset created by the implemented labeling functions. The acquisition function is defined as an extension of uncertainty sampling, that is, it is calculated from the difference between the above two types of probabilities and their respective variations. Domain experts then implement labeling functions that can be mostly applied to a small number of presented data points with a high acquisition-function score. This method can reduce the domain experts' task to only the iterative implementation of their implicit knowledge that is useful for efficient model training. In other words, this method reduces the actual labeling process cost for the first time.

The main contribution of this paper is the proposal and verification of an active learning strategy for interactive weakly supervised learning through data programming. This method can reduce the real labeling process cost of domain experts without the limitation of the application field and sacrificing interpretability. We experimented to evaluate our proposed method in a text classification task. The experimental results indicate that our proposed method requires less labeling functions than state-of-the-art methods. This contribution leads to broader machine learning penetration by reducing the cost of and resistance to its deployment.

2 Related work

2.1 Data programming

Data programming is one of the main weakly supervised learning methods [16, 2]. It does not require domain experts to give a label to every single data point but requires them to implement their knowledge as labeling functions. These labeling functions probabilistically label the data points that fit the knowledge together. Of course, labeling functions can also be created from other resources such as the results of crowd-sourcing, distant supervision [14], and so on. Snorkel [17, 18] is a representative Open-Source Software (OSS) for data programming. An example of labeling functions implemented in Python with Snorkel library is shown in Figure 2 in the Experiment section.

One of the important advantages of data programming as weakly supervised learning is that it is acceptable to have some extent of overlapping and inconsistencies between labeling functions [17]. The training dataset in data programming is generated by the collective knowledge of multiple labeling functions. When labeling functions come from a rule that simplifies some of the domain experts' knowledge, a few exceptions can emerge, resulting in mutually contradictory or broken votes. If the votes for a data point are split between a high-precision labeling function and low-precision labeling function, the vote of the higher-precision one should take precedence. Even though the real problem is not so simple because the precision of each labeling function is unobservable, a labeling aggregator estimates the precision of each labeling function by using an unsupervised model from the overall voting results and softly classifies each data point based on the estimated results in data programming [18].

Another important advantage of data programming is its interpretability. Data programming can give a reason a data point is given a label through a labeling function. This does not affect the black-box nature of a classification model, but it gives data points room for interpretation. Therefore, data programming is a method of reducing the labeling cost while improving rather than sacrificing its interpretability.

However, data programming alone does not completely reduce the manual labeling process cost of domain experts. Implementing all the knowledge in their mind as labeling functions might require a rather significant cost unless domain experts can acquire any insight into what kind of knowledge to implement. The desired task is to efficiently implement only their knowledge that is effective for training models.

Some studies are being conducted to reduce the cost of creating labeling functions. BabbleLabble [8] can convert natural language explanations to labeling functions by a simple rule-based semantic parser. Domain experts provide not only a label but an explanation of why they provide that label to the data point in natural language, then machines automatically create a labeling function. This contributes to enabling domain experts to easily perform their tasks, but not to reduce the number of labeling functions to implement. Snuba [24] and GOGGLES [5] can automatically create labeling functions. Snuba can be regarded as a mixed concept of few-shot learning and data programming. This method creates labeling functions automatically and iteratively based on a few manually labeled data points and primitives of many unlabeled data points. In each iteration, machines create simple classifiers as labeling-function candidates based on a few labeled data points and select some that achieve high accuracy on the labeled data points and high coverage on the unlabeled data points. The sampled candidates are added to the labeling function set. GOGGLES can be regarded as a mixed concept of transfer learning and data programming. This method automatically creates labeling functions for image datasets based on a few labeled data points and pre-trained representation learning models such as VGG-16 [20]. Machines first concatenate all labeled data points and unlabeled data points and apply the affinity function extracted from the pre-trained representation learning model to each pair. They then determine the cluster-to-class assignment using a small number of labeled data points. These methods greatly contribute to the automated creation of labeling functions, but the limitations of few-shot learning and transfer learning become apparent again. Thus, domain experts themselves should provide effective knowledge to machines, and the machines should help with that. Active learning [19] is expected to solve this problem in supervised learning.

2.2 Human-in-the-loop machine learning

Human-in-the-loop machine learning is a machine learning framework requiring iterative human interaction for constructing and training models [26]. It has ironically attracted much attention as a practical countermeasure to the doubts and complaints about the reliability and interpretability of machine-only machine learning for full automation. In this framework, humans are expected to make human decisions to avoid bias and mistaking correlation for causality [1], improve accuracy in case a fully automated framework does not result in sufficient accuracy [11], and provide expert knowledge to solving computationally hard problems [9]. Tamr [21] and Magellan [7] are examples of OSS for human-in-the-loop machine learning. Tamr automatically classifies only data points with high confidence and asks humans to classify other data points with low confidence. Magellan supports model learning based on automated tools for debugging processes and step-by-step, end-to-end procedure guides. In short, the goal of human-in-the-loop machine learning is to achieve

knowledge discovery and reliable performance that is impossible or difficult for machines and humans to achieve alone. There are diverse approaches, but it is important to minimize the amount of human labor to ensure reliable and useful results for humans by humans because we are human.

Active learning is one of the main human-in-the-loop strategies to build high-performance models while reducing labeling cost [27]. In each iteration, active learning calculates the acquisition function for each unlabeled data point to sample a few data points to label next, based on the model learned on already labeled data points. The acquisition function is diverse, but the most commonly used acquisition function is the one for uncertainty sampling[28]. Uncertainty sampling is a strategy to sample a few data points with low confidence by using a current classification model based on, for example, a margin of the top two classes and entropy of predicted probabilities. Sampled data points are labeled mainly by domain experts and added to the labeled training dataset from the next iteration. Active learning alone can reduce labeling cost to some extent, but it still requires domain experts to label every single data point, which is redundant and time-consuming.

Only Wang et al. [25] focused on iterative implementation of labeling functions in loops with human intervention. In each iteration, machines sample a data point for which no labeling function can be applied or whose voting results from the labeling functions are broken, and humans create a labeling function that can be applied to the data point. This means that its method does not involve any feedback information from either the labeling aggregator or the subsequent classification model. Moreover, the sampling method is much simpler than active learning. The motivation of this work is excellent, however, its method is not much different from the method to implement a labeling function that can be applied to randomly sampled data points. Especially, this method is not so effective in environments with few implemented labeling functions or their coverage is small, even though these are the very environments where effective labeling function implementations are most needed. This is because there are a large amount of unlabeled data points that fit the sampling conditions. This will be shown in the experimental results described later. Moreover, its main target is binary classification. Although it is extensible to multi-class classification, it is not practical in many problems.

3 Active learning strategy for interactive weakly supervised learning

In this section, we describe our proposed method for reducing labeling cost by reducing the number of labeling functions to implement. With our method, domain experts add a labeling function iteratively by referring to a few prioritized data points derived from acquisition function defined for the data programming framework. This acquisition function is based on uncertainty sampling and is calculated from the two output probability vectors of labeling aggregator in data programming and of the subsequent classification model. Our method enables domain experts' tasks to implement only that knowledge as labeling functions that are effective for efficient training of a classification model considering the insight given by machines.

Figure 1 shows an overview of our proposed method, where $\mathcal{D}$, $\mathcal{L}$, and $\mathcal{F}$ denote the set of data points, label classes, and implemented labeling functions, respectively, and $|\cdot|$ denotes the number of factors in a set. In the simplest generic supervised learning, domain experts annotate the data points $|\mathcal{D}|$ times repeatedly for creating a labeled training dataset. Then, parameters of a classification model are optimized to the dataset, and the model returns the predicted probability matrix $(p^{\mathrm{c}}_{l,d})$, whose d-th column shows the predicted probability vector on labeling functions l for d-th data point. For simplicity, a test dataset is not shown in Figure 1, but of course, the model will be applied to the test dataset.

With our proposed method, the forward process is derived from data programming. Labeling functions return a vote matrix $(v_{f,d})$, which is labeling function f's voting result for data point d ($v_{f,d} \in \mathcal{L} \cup \{\mathrm{ABSTAIN}\}$). ABSTAIN means that a labeling function does not vote for any label classes. The labeling aggregator returns a probability matrix $(p^{\mathrm{l}}_{l,d})$, whose d-th column shows the estimated probability vector for the d-th data point. The training dataset is generated from this probability matrix. The label of data point d, label d, is set by $\operatorname{argmax}_l p^{\mathrm{l}}_{l,d}$. The classification model returns the predicted probability matrix $(p^{\mathrm{c}}_{l,d})$, as in generic supervised learning.

The active learning model in the backward process is the key component of our method. This model receives two types of probability matrices $(p^{\mathrm{l}}_{l,d})$ and $(p^{\mathrm{c}}_{l,d})$ from the labeling aggregator and

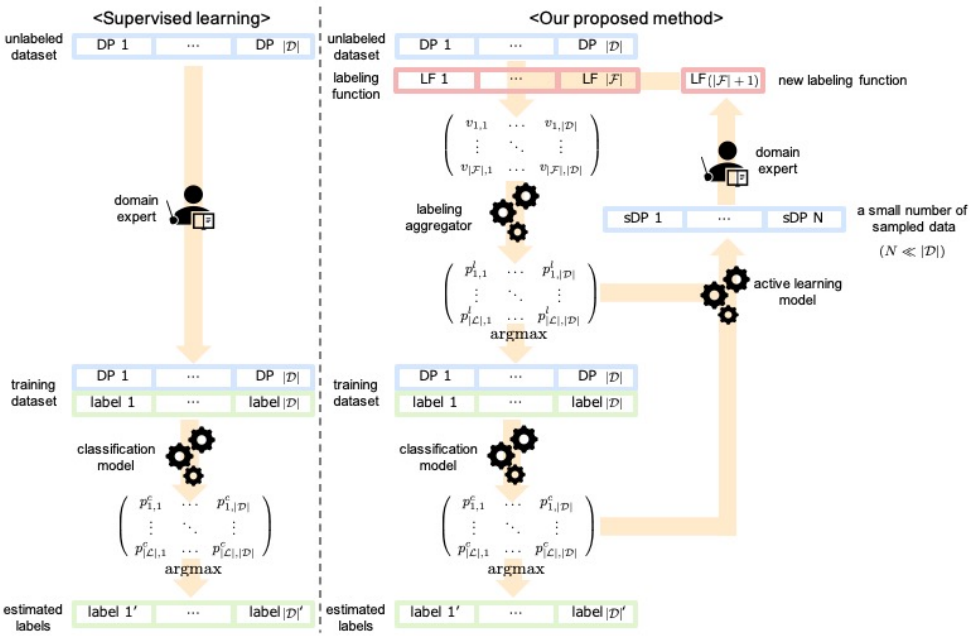

Figure 1: Our proposed method compared with general supervised learning

classification model, respectively, and calculates the priority of each data point to be presented to domain experts. This priority is calculated by a defined acquisition function, which is described later. Domain experts implement a new labeling function by referring to the presented top N data points in the priority ($N \ll |\mathcal{D}|$). If domain experts cannot implement from the data points, they refer to the next top N data points.

The acquisition function is defined as Equation 1 based on an extension of uncertainty sampling.

$$A(d) = \log \mathrm{H}(\boldsymbol{p}_d^{\mathrm{l}}) + \log \mathrm{H}(\boldsymbol{p}_d^{\mathrm{c}}) + \log\left(1 - \cos(\boldsymbol{p}_d^{\mathrm{l}}, \boldsymbol{p}_d^{\mathrm{c}})\right) \tag{1}$$

The first term represents the uncertainty of the prediction probability by the labeling aggregator. The vector $\boldsymbol{p}_d^{\mathrm{l}} (\in [0,1]^{|\mathcal{L}|},\ \Sigma_l p_{l,d}^{\mathrm{l}} = 1)$ denotes the probability vector of data point d, namely the column corresponding to it in $(p_{l,d}^{\mathrm{l}})$, and $H(\cdot)$ denotes the entropy of a probability vector. The uncertainty becomes larger not only when no labeling function can vote or some of them break the vote for d but when only labeling functions with low confidence can vote for d. In these cases, an additional labeling function applicable to d should be implemented. The second term represents the uncertainty of the prediction probability by the classification model. The vector $\boldsymbol{p}_d^{\mathrm{c}} (\in [0,1]^{|\mathcal{L}|},\ \Sigma_l p_{l,d}^{\mathrm{c}} = 1)$ denotes the probability vector of data point d, namely the column corresponding to it in $(p_{l,d}^{\mathrm{c}})$. This term is generally used in the uncertainty sampling for supervised learning. Thus, data point d with a large second term also needs a corresponding labeling function. The last term represents the dissimilarity between the predictions by the labeling aggregator and by the classification model. The function $\cos(\cdot,\cdot)$ denotes cosine similarity between two probability vectors. If both of uncertainty indicators are small but their predicted label classes differ, either model makes wrong predictions. No matter which model predicts wrongly, an additional labeling function applicable to data point d should be implemented, and this is why the acquisition function is described as the sum of logs.

Thanks to the acquisition function, our proposed method is an active learning strategy that makes much use of data programming. First, the labeling function created in each iterative loop with this method can make a larger contribution to training models than one created haphazardly. Second, it is incomparably easier to create a labeling function by referring to a small number of highly prioritized data points than to create one by referring to a large pool because of $N \ll |\mathcal{D}|$. Since similar data points' score close to each other, such data points are more likely to appear in the top N if we do not dare to consider diversity. These advantages make it easier for domain experts to implement effective and general knowledge as labeling functions. As a result, we can expect to reduce the number of labeling functions required for the classification model to achieve a certain accuracy.

Table 1: Basic settings of experiment

# of data points	5452 (training), 500 (test)
# of classes	6
# of LF candidates	42
# of LFs in the initial LF set	6
classification model	bidirectional LSTM
# of data points sampled in an iteration (N)	10

```
def lf_where(x):
    if x.startswith('where'):
        return label_map['LOC']
    else:
        return label_map['ABSTAIN']
```

Figure 2: Example of labeling function for TREC-6

4 Experiment

4.1 Experimental settings

We conducted an experiment in a text classification task involving TREC-6 [12]. Table 1 shows the basic experimental settings. Since the purpose of the experiment was to verify the effectiveness of our proposed method, we made all labeling function candidates based on the rule-based method beforehand and selected one labeling function among the candidates for each loop to eliminate any arbitrariness. The labeling-function candidates are manually created based on the study by Madabushi and Lee [13] and word lists on Li and Roth [12]'s web page[1]. Figure 2 shows an example of a labeling function. This example votes for the LOCATION ('LOC') class if the question starts with "where" and abstains from voting otherwise. The initial labeling function set is assumed to start with 5W1H, including the above example.

We implemented four types of labeling-function-selection (LF-selection) methods, our proposed method, Wang et al. [25]'s method extended to multi-class classification, and two types of random selection methods. With our proposed method, the labeling function to add the next l_{add} is ideally determined from Equation 2 ideally to eliminate any arbitrariness and to the fullest effect.

$$l_{\mathrm{add}} = \underset{l \in \mathcal{L}_{cand}}{\mathrm{argmax}} \sum_{d \in \mathcal{D}_{\mathrm{vote}}(l)} A(d), \tag{2}$$

where $\mathcal{L}_{cand}$ denotes the set of labeling-function candidates that have not been added, and $\mathcal{D}_{\mathrm{vote}}(l)$ denotes the set of data points that are ranked in top N in terms of acquisition function A and do not abstain from voting by labeling function l. However, creating a labeling function while adding the values of acquisition functions is too complicated for humans. Therefore, we simplified it to the following two steps. The labeling-function candidate whose $\mathcal{D}_{\mathrm{vote}}(l)$ contains the most data points in $\mathcal{L}_{cand}$ is selected as l_{add}. If there is more than one corresponding labeling function, the one that has a data point with a larger acquisition function in $\mathcal{D}_{\mathrm{vote}}(l)$ is selected. With the extended method of Wang et al. [25], data points are prioritized as the first if they are abstained by all implemented labeling functions and as the second if they are voted by some labeling functions but the votes are broken. The top N data points are selected (in the same priority, they are selected randomly), and the labeling function candidate whose $\mathcal{D}_{\mathrm{vote}}(l)$ contains the most data points in $\mathcal{L}_{cand}$ is selected as l_{add}. If there is more than one corresponding labeling function in the same priority, l_{add} is selected randomly from the corresponding labeling functions. The two types of random sampling methods are random (DP) and random (LF). Random (DP) selects N data points randomly and the labeling function candidate, which can be applied to most of the data points in the N data points, is selected as l_{add}. Random (LF) selects l_{add} directly and randomly. Since these comparison methods include random elements, we validated five times with different seeds with each method and evaluated them by their mean and standard deviation. The evaluation indicator is the macro F-measure calculated using the classification model applied to the test dataset for each loop.

[1] https://cogcomp.seas.upenn.edu/Data/QA/QC/

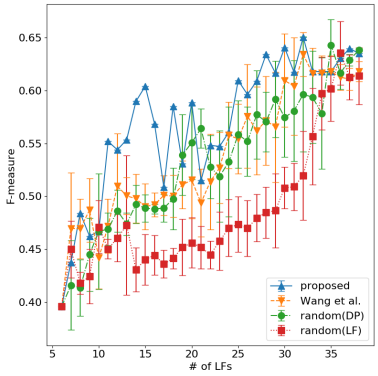

Figure 3: F-measures of each iterative loop on each LF-selection method

Table 2: Sum of F-measure improvements on each LF-selection method

proposed	Wang et al. [25]	random (DP)	random (LF)
5.72	4.65	4.50	2.96

Table 3: The number of LFs that are required to always achieve representative F-measure values

F-measure	proposed	Wang et al. [25]	random (DP)	random (LF)
0.4	7	7	7	7
0.5	11	22	19	30
0.6	27	30	35	35

4.2 Experimental results

Figure 3 shows the experimental results, F-measure values of each iterative loop on each LF-selection method. The F-measure increased more quickly in the order of our proposed method, Wang et al. [25], random (DP), and random (LF) as an overall trend. Table 2 shows the quantitative evaluation result indicating this overall trend. This table shows the sum of (achieved F-measure minus the initial F-measure) of each plot on each method. This metric indicates the overall performance of how higher F-measure can be achieved with fewer labeling functions. A larger value means better performance under the same setting as the Area Under the Curve (AUC) for Receiver Operating Characteristic (ROC) curve. These results indicate that not only our proposed method is superior to other methods, but Wang et al. [25] and random (DP) are almost neck and neck as described in Related Work section.

At last, Table 3 shows the number of labeling functions that are required to always achieve F-measure ≥ 0.4, 0.5, and 0.6. 'Always' means that F-measure never decrease from 0.4, 0.5, and 0.6 even though further additional labeling functions are added (since the addition of labeling functions may lead to lower F-values). This result indicates that our proposed method can reduce the number of labeling functions to implement.

4.3 Comparison with conventional manual labeling

As a further evaluation, we also implemented two types of data-point-selection (DP-selection) methods, entropy-based uncertainty sampling (active) and random sampling (random). Data points are selected and are labeled directly as general active or supervised learning. The purpose is to compare our proposed method with conventional manual labeling to each data point. In other words, we evaluated the labors required in traditional labeling methods to reach an accuracy comparable to that achieved by our proposed method. The evaluation indicator is the same as the above evaluation using labeling functions. This comparison is only reference data because the comparison is strongly influenced by the quality of each labeling function and the size of training dataset, but the results can represent the potential of our proposed method. The number of data points selected for each loop is set to 10 also to align with N in the above methods. The initial data points were selected randomly.

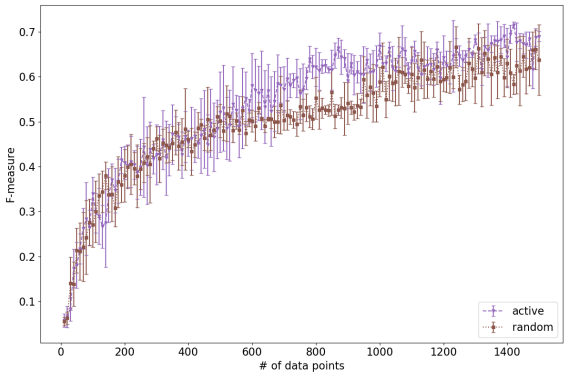

Figure 4: F-measures of each iterative loop on each DP-selection method

Table 4: Comparison of our proposed method and each DP-selection method

F-measure	proposed (# of LFs)	active (# of DPs)	random (# of DPs)
0.4	7	240	250
0.5	11	570	750
0.6	27	1200	1430

Figure 4 shows the results of the data point selection (DP-selection) setting, and Table 4 lists the numbers of labeling functions in our proposed method and data points in both DP-selection methods that are required to always achieve F-measure $\geq$ 0.4, 0.5, and 0.6. It is clear that on a simple average, an implementation of one labeling function is equivalent to labeling 30 to 50 data points. These results indicate that our proposed method can significantly reduce the burden of domain experts' redundant labeling tasks. Additionally, we can confirm the effectiveness of active learning in supervised learning by comparing active and random.

4.4 Limitation

Our proposed method is very effective for domain experts who introduce machine learning as shown in the experimental results. However, an issue remains that actual effectiveness depends somewhat on the ease of implementing labeling functions. It is also true that there is a demand to reduce labeling cost for building a machine learning model because it is difficult to articulate the domain expert's knowledge for some problems. Varma et al. [24] and Das et al. [5], which are referred to in Related Work section, are already working on this issue by using an automation approach for creating labeling functions. We believe that human intervention is still essential for creating reliable low-resistance machine learning models, so combining these ideas for automation with our proposed method and incorporating their benefits is our future work.

5 Conclusion

We proposed and evaluated an uncertainty-based active learning strategy for interactive weakly supervised learning to reduce labeling cost without sacrificing interpretability of data. Our proposed method is a human-in-the-loop method that presents a small number of highly prioritized data points to humans based on the acquisition function derived from uncertainties of the labeling aggregator and the subsequent classification model and requires humans to iteratively implement their knowledge applicable to the highly prioritized data points as labeling functions. With this method, domain experts' redundant and time-consuming labeling process can be replaced by the minimum required implementation of labeling functions by referring to a small number of highly prioritized data points. The experimental results verify the effectiveness of our proposed method in reducing the required number of labeling functions. For future work, we will reduce the cost of creating labeling functions without losing the interpretability of these functions by incorporating the benefits of our method and methods of automatically creating labeling functions.

References

[1] T. Allen, M. Chen, J. Goldsmith, N. Mattei, A. Popova, M. Regenwetter, F. Rossi, and C. Zwilling. Beyond theory and data in preference modeling: Bringing humans into the loop. In *Proc. of ADT*, page 3–18, 2015.

[2] S. Bach, B. He, A. Ratner, and C. Ré. Learning the structure of generative models without labeled data. In *Proc. of ICML*, page 273–282, 2017.

[3] W. Chen, Y. Liu, Z. Kira, Y. Wang, and J. Huang. A closer look at few-shot classification. In *Proc. of ICLR*, 2019.

[4] N. Das, S. Chaba, R. Wu, S. Gandhi, D. Chau, and X. Chu. Goggles: Automatic image labeling with affinity coding, 2020. arXiv.

[5] N. Das, S. Chaba, R. Wu, S. Gandhi, D. Horng Chau, and X. Chu. Goggles: Automatic image labeling with affinity coding. In *Proc. of SIGMOD*, page 1717–1732, 2020.

[6] G. Goh, C. Siegel, A. Vishnu, and N. Hodas. Using rule-based labels for weak supervised learning: A chemnet for transferable chemical property prediction. In *Proc. of KDD*, page 302–310, 2018.

[7] Y. Govind, P. Konda, P Suganthan, P. Martinkus, P. Nagarajan, H. Li, A. Soundararajan, S. Mudgal, J. Ballard, H. Zhang, A. Ardalan, S. Das, D. Paulsen, A. Saini, E. Paulson, Y. Park, M. Carter, M. Sun, G. Fung, and A. Doan. Entity matching meets data science: A progress report from the magellan project. In *Proc. of SIGMOD*, page 389–403, 2019.

[8] B. Hancock, P. Varma, S. Wang, M. Bringmann, P. Liang, and C. Ré. Training classifiers with natural language explanations. In *Proc. of ACL*, pages 1884–1895, 2018.

[9] A. Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016.

[10] A. Li, T. Luo, Z. Lu, T. Xiang, and L. Wang. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In *Proc. of CVPR*, pages 7205–7213, 2019.

[11] G. Li. Human-in-the-loop data integration. *Proc. VLDB Endow.*, 10(12):2006–2017, 2017.

[12] X. Li and D. Roth. Learning question classifiers. In *Proc. of COLING*, page 1–7, 2002.

[13] H. Madabushi and M. Lee. High accuracy rule-based question classification using question syntax and semantics. In *Proc. of COLING*, pages 1220–1230, 2016.

[14] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proc. of ACL*, pages 1003–1011, 2009.

[15] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Proc. of NeurIPS*, pages 3347–3357. 2019.

[16] A. Ratner, C. Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *Proc. of NeurIPS*, pages 3567–3575. 2016.

[17] A. Ratner, S. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. *Proc. VLDB Endow.*, 11(3):269–282, 2017.

[18] A. Ratner, B. Hancock, J. Dunnmon, F. Sala, S. Pandey, and C. Ré. Training complex models with multi-task weak supervision. *Proc. of AAAI*, 33:4763–4771, 2019.

[19] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of ICLR*, 2015.

[21] M. Stonebraker, G. Beskales, A. Pagan, D. Bruckner, M. Cherniack, S. Xu, V. Analytics, I. Ilyas, and S. Zdonik. Data curation at scale: The data tamer system. In *Proc. of CIDR*, 2013.

[22] Q. Sun, Y. Liu, T. Chua, and B. Schiele. Meta-transfer learning for few-shot learning. In *Proc. of CVPR*, pages 403–412, 2019.

[23] F. Tang. Bidirectional active learning with gold-instance-based human training. In *Proc. of IJCAI*, pages 5989–5996, 2019.

[24] P. Varma and C. Ré. Snuba: Automating weak supervision to label training data. *Proc. VLDB Endow.*, 12(3):223–236, 2018.

[25] B. Wang, A. Ratner, S. Mussmann, and C. Ré. Interactive programmatic labeling for weak supervision. In *Proc. of KDD adj.*, 2019.

[26] D. Xin, L. Ma, J. Liu, S. Macke, S. Song, and A. Parameswaran. Accelerating human-in-the-loop machine learning: Challenges and opportunities. In *Proc. of DEEM*, pages 1–4, 2018.

[27] K. Yang, J. Ren, Y. Zhu, and W. Zhang. Active learning for wireless iot intrusion detection. *IEEE Wireless Communications*, 25(6):19–25, 2018.

[28] J. Yuan, X. Hou, Y. Xiao, D. Cao, W. Guan, and L. Nie. Multi-criteria active deep learning for image classification. *Knowledge-Based Systems*, 172:86 – 94, 2019.