Proceedings of the 53rd Hawaii International Conference on System Sciences | 2020

WeSAL: Applying Active Supervision to Find High-quality Labels at Industrial Scale

Mona Nashaat
University of Alberta
nashaata@ualberta.ca

Aindrila Ghosh
University of Alberta
aindrila@ualberta.ca

James Miller
University of Alberta
jimm@ualberta.ca

Shaikh Quader
IBM Canada
shaikhq@ca.ibm.com

Abstract

Obtaining hand-labeled training data is one of the most tedious and expensive parts of the machine learning pipeline. Previous approaches, such as active learning aim at optimizing user engagement to acquire accurate labels. Other methods utilize weak supervision to generate low-quality labels at scale. In this paper, we propose a new hybrid method named WeSAL that incorporates Weak Supervision sources with Active Learning to keep humans in the loop. The method aims to generate large-scale training labels while enhancing its quality by involving domain experience. To evaluate WeSAL, we compare it with two-state-of-the-art labeling techniques, Active Learning, and Data Programming. The experiments use five publicly available datasets and a real-world dataset of 1.5M records provided by our industrial partner, IBM. The results indicate that WeSAL can generate large-scale, high-quality labels while reducing the labeling cost by up to 68% compared to active learning.

1. Introduction

Machine learning models are deployed in many domains to empower data-driven decisions. However, supervised machine learning models require access to labeled training datasets [1]. Obtaining such labeled data is a major bottleneck in creating learning models, especially with the current popularity of data-greedy methods such as deep learning models that may require millions of labeled data points. As a result, acquiring labeled datasets turns out to be an expensive yet indispensable task in the machine learning pipeline.

Aiming to tackle this challenge, there is ample research [1]–[3] offering solutions to generate labeled training data. Active learning (AL) [2] can be seen as a labeling approach that aims at optimizing labeling cost and classification accuracy. For example, in pool-based AL [2], the learning algorithm iteratively selects the points from a pool of unlabeled points. Since the algorithm queries the user about the most informative points, the resulting model is assumed to achieve better classification performance with fewer labels.

While AL tries to engage human oracles to provide true labels, there is a growing interest in using weak supervision sources [3]. Weak supervision relies on obtaining low-quality, but large-scale training datasets by exploiting cheaper annotating approaches. To integrate training labels from these weak sources, previous studies [1], [4], [5] used generative models [6] to learn the accuracy of such sources and model the true label as a latent variable [4].

However, several questions regarding these approaches remain to be addressed. On the one hand, AL can be expensive with high-dimensional datasets [7]. For instance, the unbalance between the sizes of labeled and unlabeled data can slow the labeling process. Also, previous research [8] indicates that, when dealing with imbalanced data distributions, AL can result in low performance. On the other hand, weak supervision outputs noisy labels which affect the model performance. The uncertainty of the generated labels complicates the process of learning the structure of the generative models [6]. Also, since weak sources often overlap and conflict, debugging these sources can be time-consuming [5].

Therefore, motivated by the shortcomings of these approaches, we present WeSAL, a labeling approach that combines Weak Supervision with Active Learning to create large-scale, high-quality training data. WeSAL extends weak supervision and includes humans-in-the-loop to denoise the weak labels. It tries to overcome the scalability issues of AL by reducing the size of unlabeled pools to only contain conflicting points. Therefore, WeSAL profits from the scalability of weak supervision while economically applies user engagement to enhance labeling accuracy.

Figure 1 illustrates an overview of WeSAL; the approach starts by collecting labels from different weak sources. Although WeSAL can work with any weak supervision sources, we focus on user-defined heuristics

URI: https://hdl.handle.net/10125/63767
978-0-9981331-3-3

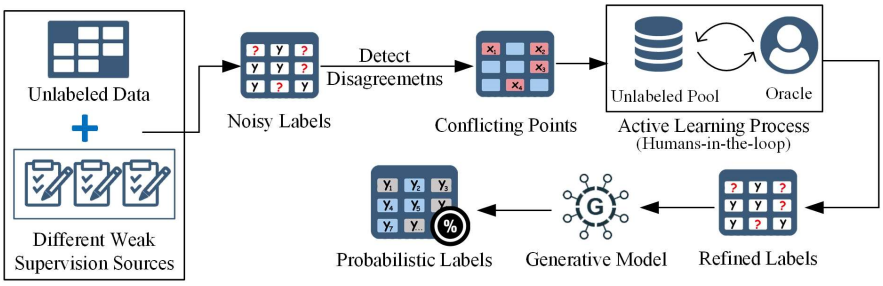

Figure 1. Overview of the proposed method

since they are the most popular methods to generate noisy labels for real-world tasks [4]. Afterward, these labels are examined to create an unlabeled pool. Next, the user is queried about the most informative points. Then, the obtained labels from AL process are used to refine the initial noisy labels. After that, a generative model is used to model the accuracy of the refined heuristics and output probabilistic labels. Finally, these labels are used to train any model to produce predictions for the desired learning task.

To evaluate WeSAL, we compare it with two state-of-the-art techniques, data programming (DP) [1] and AL. The experiments aim at assessing the effectiveness of WeSAL in producing accurate labels in terms of labeling accuracy, labeling budget, and classification performance. The experiments include a sensitivity analysis of the parameters used in the experiments to study their impact on the performance.

The paper is structured as follows: Section 2 discusses the related background. Section 3 presents the proposed method. The experimental results are offered in Section 4. While Section 5 discusses related work; and Section 6 concludes the paper.

2. Background

In this section, we first discuss active learning, then, we overview the data programming paradigm.

2.1. Active learning

Active learning helps to generate labels with minimum labeling effort [2]. In pool-based AL, a classifier starts with having access to a pool of unlabeled examples, a set of labeled points (the seed), and a test set. Initially, the classifier is trained using the seed. Then, points in the unlabeled pool are ranked, and the most informative points are chosen to query an oracle, then used to train a classifier and evaluate its performance on the test set. Given the new status of the classifier, the points in the unlabeled pool are ranked again, and the process is repeated. AL process stops based on a stopping criterion [2], for example when a target performance is reached. The part that selects the points from the unlabeled pool is the query strategy. Over the past decades, several query strategies are proposed. One of the most effective query strategies is uncertainty sampling [2]. It selects the points about which the classifier is most uncertain. Another query strategy is Query-by-committee [2] which operates similarly as uncertainty sampling, except it uses a committee of classifiers and chooses the points about which the committee members disagree.

Nevertheless, many research articles [9]–[12] point out that AL suffers from many challenges, particularly that AL algorithms are binary methods and do not scale to multi-classification settings [11], [12]. Another problem of AL originates from the complexity of the ranking step [9], [10], especially with large scale unlabeled pools. In these cases, AL becomes an expensive solution. Another study [13] states that training datasets built with AL can contain labels with biased distribution for the chosen model. As a result, we believe that many questions exist regarding the performance of AL when applied to large scale datasets. To address and overcome these issues, WeSAL aims to speed up the ranking procedure and reduce the size of the labeling pool. The solution helps to resolve the unbalance between the labeled and unlabeled data and hence, enhance the scalability of AL. The experiments show that AL annotation costs can be deducted by 36% using the proposed method.

2.2. Weak supervision

In recent years, weak supervision [3] has been gaining popularity in generating labels. In weak supervision, domain experts are asked to provide some form of higher-level, low-quality supervision such as user-defined heuristics. The results of such forms are programmatically generated data which is noisy and contains conflicting labels. As a result, the problem of integrating these diverse sources remains open [1], [5], [6]. DP [1] is a paradigm proposed to integrate labels generated from weak sources. In DP, weak supervision sources are encoded as labeling functions [4], which are arbitrary scripts that translate different weak sources. After applying these functions, DP uses generative models to learn the accuracies of the labeling functions without access to labeled data [4]. DP applies structure

learning techniques to model the true class labels as latent [6]. Finally, the generative model outputs a set of probabilistic training labels which can be used to train any discriminative model.

Depending on high-level supervision, DP generates labels with a noise level that is hard for the end-users to evaluate. Also, the complex structure of the generative model makes it challenging for users to debug its outcome [14]. Therefore, studies [14], [15] have tried to overcome these limitations. One study is Socratic Learning [15], which is a technique to debug generated labels by examining the disagreements between the training data and the generated labels. However, since Socratic Learning is an automated method that does not utilize domain experience in the refinement process, end users may have problems in understanding its decisions [14]. To overcome this lack of explainability, the authors in [14] proposed a visual framework to interpret these decisions. However, the framework does not explain the structure of the generative model, which users often struggle to understand.

Overall, we find that since weak supervision results in noisy conflicting labels, previous studies have exclusively focused on learning the structure of generative models to enhance the labeling quality. However, none of these studies explored the effect of utilizing domain expertise to denoise the output labels. Therefore, in WeSAL, end users are asked to refine the disagreements between the labeling functions by providing labels for the conflicting points. Many researchers [4], [14], [15] have demonstrated that resolving these disagreements enhances accuracy and helps better identify latent subsets in the training data. WeSAL employs domain expertise to perform this task to improve both the labeling quality and help end-users evaluate the accuracy of the weak sources. The experimental results show that WeSAL managed to enhance labeling accuracy by up to 26% when compared to data programming.

3. WeSAL: The proposed method

Let us assume we have a set of unlabeled inputs X of size N denoted as $\{\mathbf{x}_i\}_{i=1}^{N}$ where $\mathbf{x}_i$ represents a set of features describing the i^{th} data point in X, and a set of unknown labels y as $\{y_i\}_{i=1}^{N}$ where $y_i \in \{-1, 1\}$. WeSAL starts by allowing the users to write a group of T labeling functions F denoted as $\{f_j\}_{j=1}^{T}$, where f_j: $\mathbf{X} \rightarrow \{-1, 0, 1\}$. Each labeling function creates a weak label for x_i where 0 describes abstaining. Therefore, the result of applying all functions F to X is a noisy label matrix L where:

$$L_{i,j} = f_j(\mathbf{x}_i) \text{ where } 1 \le i \le N \text{ and } 1 \le j \le T \quad (1)$$

To model the accuracy of the labeling functions, DP [1] forms a generative model G as a factor graph Ø. The graph is encoded using three factors, namely, labeling propensity $Ø^{lab}{}_{i,j}(F, Y) = \mathbf{1}\{f_{i,j} \neq 0\}$, labeling accuracy $Ø^{Acc}{}_{i,j}(F, Y) = \mathbf{1}\{f_{i,j} = y_i\}$, and functions pairwise correlation $Ø^{Corr}{}_{i,j,k}(F, Y) = \mathbf{1}\{f_{i,j} = f_{i,k}\}$ where $j, k \in M$ where M is a set of labeling function pairs (j, k) modeled as dependent [6].

Since these labeling functions rely on imperfect sources, they abstain and conflict with each other. Consequently, WeSAL resolves pairwise disagreements between the labeling functions to increase their accuracy. The pairwise disagreements can be defined as:

$$Ø^{dis}{}_{i,j,k}(F, Y) = \mathbf{1}\{f_{i,j} \neq f_{i,k}\} \text{ where } j, k \in M, i \in N \quad (2)$$

Moreover, WeSAL tries to resolve abstaining situations to increase the coverage of the resulting training labels. The abstaining labels are denoted as:

$$Ø^{abstain}{}_{i,j}(F, Y) = \mathbf{1}\{f_{i,j} = 0\} \quad (3)$$

Next, the proposed method constructs an unlabeled dataset P_U of size U where:

$$P_U \subseteq \mathbf{X}, \forall x_i \in P_U \{x_i | Ø^{dis}{}_{i,j,k}(F, Y) = \mathbf{1}\{f_{i,j} \neq f_{i,k}\} \cup Ø^{abstain}{}_{i,j}(F, Y) = \mathbf{1}\{f_{i,j} = 0\} \quad (4)$$

Therefore, to enhance the accuracy of the labeling functions, WeSAL applies AL to provide true labels and introduce domain experience. The AL component proceeds by choosing points from P_U that are assumed to be beneficial to the classifier. Since P_U represents the conflicting points between the labeling functions, the size of P_U is much smaller than the number of X. Therefore, the ranking time in WeSAL is reduced compared to traditional AL in which all the points in X are ranked at each iteration. Also, as for computational complexity, WeSAL can scale to much larger datasets than traditional active learning since it runs in O(W.U) where W is the number of queries consumed by the AL component in WeSAL and U is the size of P_U.

Furthermore, we ask users to specify a value for the maximum number of points they are willing to label and set this number as a labeling budget $B_{Labeling}$. Hence, AL process terminates when either all the disagreements are resolved (all data points in P_U are labeled) or the labeling budget is exhausted. Then, the output of AL $(X,Y)_{AL}$ can be described as $\{\mathbf{x}_i, y_i\}_{i=1}^{D}$ where D=min(U, $B_{Labeling}$). WeSAL then uses $(X,Y)_{AL}$ to denoise L as:

$$L_{\text{refined } i,j} = \begin{cases} y_i & \text{if}(x_i, y_i) \in (x, y)_{AL} \\ L_{i,j} & \text{otherwise} \end{cases} \quad j = 1, 2, \dots T \quad (5)$$

Refining the noisy label matrix L increases the empirical probability of the labeling functions f_i and f_j agreeing. The empirical probability can be described as $P_{i,j} = \frac{a}{N}$ where a is the number of agreements between f_i and f_j. Since the refinement process increases a, the empirical probability increases accordingly, and hence, the accuracy of the labeling functions is enhanced.

Then, WeSAL applies a generative model G that uses the refined label matrix $L_{refined}$ to generate a set of

probabilistic labels to train a downstream classifier of choice. G can be formally defined [15] as,

$$G: \pi_{\emptyset}(L_{\text{refined}}, Y) = \frac{1}{Z_{\emptyset}} e^{\emptyset^T L_{\text{refined}} Y} \quad (6)$$

where $Z_{\emptyset}$ is a partition function to guarantee π is a distribution, and Ø represents the average accuracy of the labeling functions [15]. As seen in (6), the generative model learns the accuracy of the labeling functions from their disagreements. Therefore, refining L improves the quality of the final labels. The complete algorithm of the proposed method is shown in Algorithm 1. Although there are other approaches [6], [15] that aim at denoising the generated labels of the DP pipeline, none of these methods have employed domain experience in this process. Therefore, we believe that our approach is the first attempt that tries to include humans in the loop in the form of AL within the weak supervision process.

Algorithm 1 WeSAL, The Proposed Method

Input: Input data set X with unknown labels Y, selected query strategy q for Active learning, labeling budget B_{Labeling}.

Output: Probabilistic labels $y^* = P[y = 1] \in [0,1]$.

```
Write a set of labeling functions F = {F1, F2, ... Ft}
Apply F to X to create a noisy label matrix L
Construct disagreements factor Ø^dis(F, Y)
Construct abstaining labels factor Ø^abstain(F, Y)
Initialize P_U = {}
Loop until i>N
    If Ø^dis_{i,j,k}(F,Y) = 1 then P_U U {x_i}
    If Ø^abstain_{i,j,}(F,Y) = 1 then P_U U {x_i}
    i ← i+1.
End
Initialize (X,Y)_AL= {}
Loop until stopping criterion is met
    Select a point x_i from P_U using q
    query the user to provide a label y_i for x_i
    P_U = P_U - x_i
    (X,Y)_AL = (X,Y)_AL U (x_i, y_i)
    Train classifier using (X,Y)_AL
End
denoise L using (X,Y)_AL to create L_refined
Train generative model G with L_refined to output y*
```

4. Evaluation

The experiments seek to validate two points. First, how accurately can WeSAL generate labels for real tasks. Second, what is the impact of using WeSAL on the labeling cost. To validate the first point, we compare WeSAL to DP [4] and evaluate the performance of the generative and the discriminative models. Also, we report the accuracy of the generated labels. For the second point, we compare WeSAL against AL and report the labeling cost and the performance of the final classifiers. Although there are other labeling approaches [5], [15], [16], the experiments consider active learning and data programming since WeSAL extends these two approaches. However, future work should include evaluations against different labeling methods, such as transfer learning [16]. Also, the primary goal of WeSAL is building better predictive models for various classification tasks. Since training models with accurate labels improves their capability to generalize to unseen observations [1], [4], we report the classification accuracy of the learning models trained with the generated labels.

Table 1. Overview of the datasets. Dim is the dimensionality of the dataset. +/Size is the positive class to the dataset size ratio.

Dataset	Size	Dim.	+/ Size
Renewal Sales	1,354,704	11	73.06
Bank	45,211	17	11.70
News	39,797	61	49.34
Credit Card	30,000	24	22.12
Occupancy Detection	20,560	7	23.10
MNIST	70,000	784	-

4.1. Datasets

We consider generating training labels for real-world tasks over five open-source datasets along with a real business dataset. Summary statistics are provided in Table 1. As for the first dataset, **Renewal Sales** is a business dataset provided by our industrial partner, IBM. The dataset contains more than 1.3 million records of anonymized renewal records describing historical transactions of software subscriptions. The dataset is used in a classification task to predict license cancellations. Another business task is the Bank Marketing dataset **(Bank)** with a classification goal of predicting campaign subscriptions via marketing calls. The default of credit card dataset **(Credit Card)** is used to predict the default payments. The Online News Popularity Dataset **(News)** is a social dataset to predict the level of popularity of online articles. The fifth data is the Occupancy Detection dataset **(Occupancy Detection)**, which represents a binary classification task for room occupancy. These datasets are all publicly available and were downloaded from the UC Irvine Machine Learning Repository[1]. Moreover, to add an example of a multi-classification situation, the **MNIST**[2] dataset is added to the experiments which consists of 70K images of hand-writing digits with ten classes.

4.2. Experiments settings

Writing the labeling functions. To compare WeSAL with DP. We use the end-to-end DP framework presented in [4]. To implement the labeling functions,

[1] https://archive.ics.uci.edu/ml/index.php
[2] http://yann.lecun.com/exdb/mnist/

Table 2. Experimental settings

Dataset	Data Programming Settings						Active Learning Settings		
	# Candidates	# Labeling Functions	Labeling Functions Performance				Initial seed	Train set size	Test set size
			Accuracy	Precision	Recall	F1			
Renewal Sales	1,083,763	4	0.75	0.78	0.75	0.76	67,735	839,917	447,052
Bank	36,169	5	0.77	0.78	0.80	0.79	2,260	28,031	14,920
News	31,716	6	0.74	0.82	0.78	0.80	1,989	24,675	13,133
Credit Card	24,001	5	0.67	0.71	0.72	0.72	1,500	18,600	9,900
Occupancy Detection	16,448	7	0.78	0.81	0.78	0.80	1,028	12,747	6,785
MNIST	56,000	5	0.77	0.79	0.69	0.74	3500	43,400	23,100

we focus on threshold-based labeling functions [4], [5] in which the labeling functions assign labels to each data instance or abstain based on values of specific features in the data (e.g., values of client's bill statements may influence their default payment). As for the renewal sales dataset, we consulted a set of sales representatives from IBM to write the labeling functions. As for the other datasets, we relied on pattern matching, which is a consistent approach with best practice found in the literature [4], [5], [14].

Validating the labeling functions. Also, to only accommodate high accuracy sources, we used a set of labeled data (gold labels) to develop labeling functions. We calculated the empirical accuracy of the labeling functions concerning the gold labels. Also, we set an accuracy threshold of 60% and only included the functions that exceed this threshold. Table 2 shows the experiments settings. As for the DP settings, the table shows, the number of candidates (records) for which labels are generated, the number of labeling functions, and the evaluation buckets (Accuracy, Precision, Recall, and F1 measure) for the labeling functions.

Active Learning settings. Similarly, we compare WeSAL against different sampling techniques of AL, namely uncertainty sampling (UNC), query-by committee (QBC), and random sampling (RAND). The results of AL experiments are averaged over ten runs. The general settings used in AL experiments are illustrated in Table 2. For each dataset, the table shows the seed, the initial size of X_{train}, and the size of the test set D_{test} used to evaluate the classifier.

Also, to decide on the stopping criteria for AL, we examined the learning curves and stopped the process when the classifier performance shows no improvement with additional iterations [17]. We use $\lambda = 0.0001$ as a threshold of performance differences and stop the experiments when the mean of performance differences does not exceed λ for a successive number of iterations. Moreover, to use the same conditions throughout the experiments, we use the number of labels required to satisfy the performance stability condition as the labeling budget $B_{Labeling}$ for the proposed method.

4.3. Experiments results

In this section, we present the results of comparing WeSAL to DP and AL.

4.3.1. WeSAL vs. DP. First, we compare WeSAL to DP using the same labeling functions. Table 3 shows the results in terms of performance of the generative and the discriminative models. Reporting the performance of the discriminative models assesses the effect of the improved labeling accuracy to the performance of the learning models. To avoid measurement bias, we report a wide range of performance measures. As for the generative model, we report Precision (P), Recall (R), and F1 measure (F1). We calculate the same measures for the discriminative model along with Matthews correlation coefficient (MCC). The table also shows the labeling accuracy, which is calculated as the ratio of the number of correct labels to the size of the training set.

The results show that, with regard to the generative model, WeSAL achieved higher performance in all tasks. Since the generative model performance depends on the labeling functions, this empirically proves the effectiveness of WeSAL in enhancing the accuracy of the labeling functions. WeSAL managed to improve the

Table 3. Data programming results

Dataset	WeSAL								Data Programming							
	Generative Model			Labeling Accuracy	Discriminative Model				Generative Model			Labeling Accuracy	Discriminative Model			
	P	R	F1		P	R	MCC	F1	P	R	F1		P	R	MCC	F1
Renewal Sales	0.94	0.88	0.91	0.84	0.89	0.90	0.90	0.89	0.87	0.75	0.81	0.68	0.86	0.75	0.78	0.80
Bank	0.89	0.82	0.85	0.77	0.87	0.86	0.87	0.86	0.64	0.71	0.67	0.61	0.84	0.74	0.77	0.79
News	0.87	0.80	0.83	0.59	0.88	0.97	0.96	0.92	0.75	0.73	0.74	0.49	0.85	0.92	0.89	0.88
Credit Card	0.85	0.77	0.81	0.37	0.88	0.73	0.75	0.80	0.83	0.71	0.77	0.34	0.87	0.65	0.71	0.74
Occupancy Detection	0.94	0.81	0.87	0.75	0.90	0.94	0.95	0.92	0.82	0.78	0.80	0.67	0.87	0.83	0.84	0.85
MNIST	0.88	0.93	0.90	0.59	0.88	0.95	0.95	0.91	0.73	0.74	0.73	0.51	0.84	0.83	0.84	0.83

Table 4. Active learning results

Dataset	WeSAL					Active Learning				
	P	R	MCC	Accuracy	# queried instances	P	R	MCC	Accuracy	# queried instances
Renewal Sales	0.98	0.98	0.91	0.98	73,320	0.98	0.96	0.84	0.95	125,988
Bank	0.79	0.91	0.82	0.97	2,151	0.71	0.7	0.66	0.93	3,364
News	0.93	0.95	0.85	0.92	4,374	0.89	0.9	0.8	0.90	13,818
Credit Card	0.75	0.84	0.73	0.90	12,958	0.73	0.8	0.67	0.91	12,958
Occupancy Detection	0.75	0.98	0.81	0.94	7,283	0.72	0.82	0.7	0.90	11,855
MNIST	0.92	0.97	0.92	0.95	2,452	0.88	0.95	0.84	0.92	3,472

F1 score of the generative model by 27% and 23% in the Bank and MNIST datasets respectively. The reason for this improvement is that since the quality of the labeling functions were good (0.79 and 0.74 as F1 (Table 2)), the labeling budget was effectively spent to resolve the disagreements between the functions, and hence improve the overall performance. Moreover, WeSAL surpassed DP in discriminative model performance within all datasets. Since providing accurate data to the discriminative model improves its capability to generalize to unseen observations, this proves that WeSAL enhances the quality of the learning models.

As for the labeling accuracy, WeSAL achieved better values than DP in all datasets. In some problems such as the Bank dataset, WeSAL improved the labeling accuracy by 26% when compared to DP. Alternatively, in the credit card dataset, WeSAL achieved a relatively small enhancement of 9%. The reason behind that is the low accuracy of the labeling functions used in the credit card dataset. Therefore, WeSAL could only resolve a small portion of the conflicts, and hence, could not achieve a significant accuracy boost. Overall, WeSAL managed to enhance labeling accuracy by an average of 18% when compared to DP.

4.3.2. WeSAL vs. AL. In this part, we compare WeSAL to AL. First, to determine the labeling budget for WeSAL, we applied three query strategies to the datasets. Figure 2 shows the learning curves using UNC, QBC, and RAND query strategies. The learning curves illustrate the relationship between the number of queried points and classifier accuracy. Since the curves show that UNC achieved the highest accuracy in all the datasets, we report the evaluation metrics obtained by WeSAL and UNC in Table 4. Similar to the experiments with DP, we report the performance of the learning models to assess the influence of the generated labels to the underlying classification tasks. The table also shows the number of queried instances required to obtain the equivalent accuracy values.

The table depicts that WeSAL achieved better MCC values in all the problems with the highest improvements in the Bank dataset of 24% comparing to AL. Also, the results show that WeSAL did not need to use the labeling budget assigned by AL in most of the problems. Since the size of P_U is much smaller than the size of X_{train}, WeSAL managed to resolve all the disagreements between the labeling functions without exceeding $B_{Labeling}$. For example, while AL needed to label 12% of the training dataset in the Bank dataset, the size of P_U only represents 8% of X_{train}, hence a decrease ratio of 36% in labeling cost. Similarly, WeSAL managed to decrease the labeling cost in Renewal Sales and Occupancy Detection datasets by 42% and 39% respectively. The only dataset in which WeSAL exceeded the assigned budget is the credit card dataset. The reason for the increased labeling cost is due to the low accuracy labeling functions in this task, which result in a large number of disagreements that surpassed the assigned labeling budget. We, however, find this point agrees with our conclusion of the importance of utilizing domain experience in the labeling process by designing labeling functions with high accuracy.

The results also attest that WeSAL outperformed AL in both precision and recall in all the problems. WeSAL managed to enhance the precision values achieved by AL by 10% and 4% in the Bank and the MNIST datasets. As for the recall values, WeSAL improved the performance of the machine learning models in all the problems with the highest enhancements in the Bank and the Occupancy Detection datasets by 30% and 20% respectively. Overall, the results empirically prove that training models using labels generated by WeSAL results in remarkably improved performance, while reducing the labeling cost on real classification tasks.

4.4. Sensitivity analysis of the experimental parameters

In this section, we report the outcomes of the experiments under alternative assumptions of the parameters of the experiments.

4.4.1. Sensitivity analysis of the parameter λ. We stop AL process once the arithmetic mean of performance differences for a number of iterations is less than a predefined threshold λ=0.0001. We also utilized the number of annotations required by AL as the labeling budget $B_{Labeling}$ in WeSAL. Therefore, to observe the effect of the parameter λ on the performance of both AL

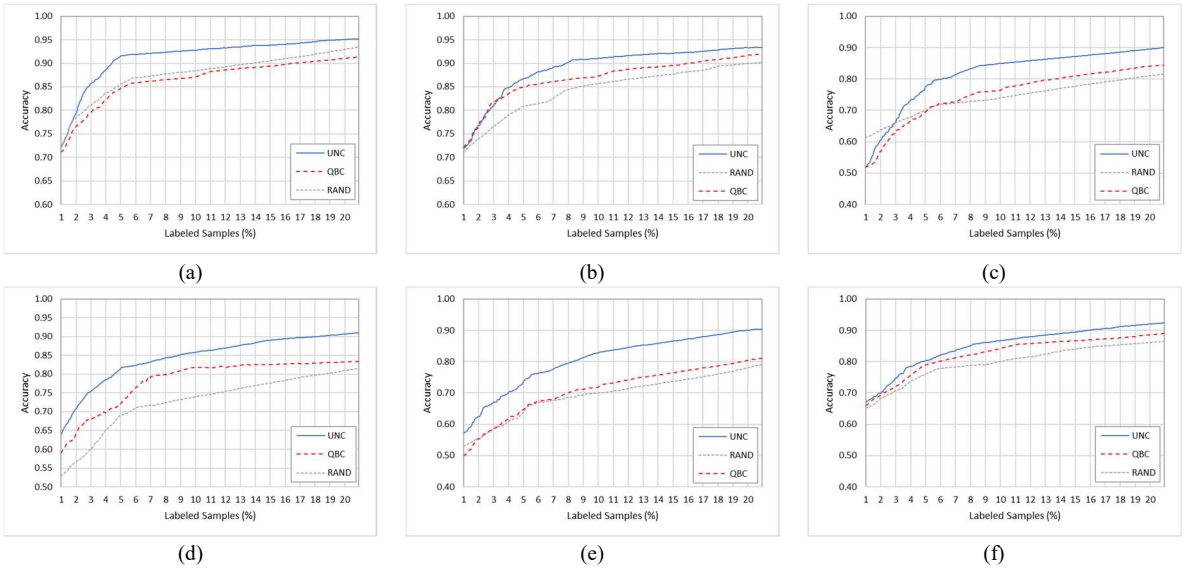

Figure 2. Learning curves of active learning for (a) renewal sales dataset (b) bank dataset (c) news dataset (d) credit card dataset (e) occupancy detection dataset (f) MNIST dataset

and the proposed method, the experiments were repeated with various values for λ. Figure 3.a shows the accuracy values reported by AL with values of λ = 0.001, 0.0001, 0.00001. Likewise, depending on the number of annotations consumed for each λ, the parameter $B_{Labeling}$ in WeSAL was adjusted accordingly. Table 5 shows, for each value of λ, in each dataset, the size of the initial unlabeled pool X_{train}, the number of queried labels at the end of the AL process as a percent of the size of X_{train} (AL Cost %). As for WeSAL, the size of P_U is assumed to be much smaller than the size of X_{train}. To highlight this point, the table shows the size of P_U as a percent of the size of X_{train} (P_U%) and the value of $B_{Labeling}$. Additionally, Figure 3.b shows the accuracy levels achieved by WeSAL for each value of $B_{Labeling}$.

As Figure 3.b depicts, choosing a larger value for λ may result in missing useful generalizations and force AL process to stop early [18]. This was the case for the news, credit card, and occupancy detection datasets as setting λ =0.001 reduced the classifier accuracy in AL by 14%, 7%, and 27% respectively when compared to the performance achieved with λ =0.0001 (Figure 3.a). Also, setting λ to a small value may enhance the performance but at the risk of wasting annotation effort. However, the figure shows no significant performance enhancement with λ=0.00001. Overall, the results show that the initial choice of λ =0.0001 was valid since, in most of the datasets, it succeeded to catch the elbow values in the learning curves after which the performance changes become notably smaller.

Moreover, Figure 3.b shows that for most of the datasets, changing λ does not impose a big difference in the performance of WeSAL. The reason behind that, as mentioned before, is since the size of P_U is less than the size of X_{train}, the cost of annotating all the points in P_U may have an upper bound of a value less than the predefined $B_{Labeling}$. This was the case in almost all the datasets. For example, in the bank, and the news datasets, WeSAL managed to fully annotate P_U with $B_{Labeling}$ corresponding to λ =0.0001 and 0.00001. On the other hand, in datasets such as the credit card and the occupancy detection datasets, having a value of λ=0.001 suppressed the performance of WeSAL since the AL

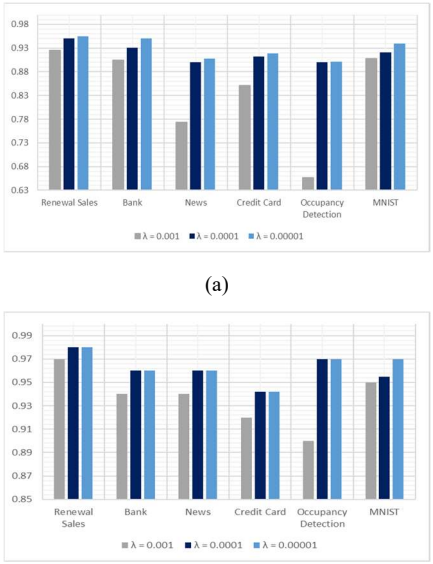

Figure 3. Accuracy values for (a) the classifiers in AL (b) the discriminative models in WeSAL with changing values of λ = 0.001, 0.0001, 0.00001

Table 5. Values of the experiments' parameters with different values of λ

Dataset	λ	Active Learning		WeSAL	
		Size of X_{train}	AL Cost %	P_U %	$B_{Labeling}$
Renewal Sales	0.001	839,917	7%	19%	61594
	0.0001		15%		125988
	0.00001		23%		195981
Bank	0.001	28,031	6%	8%	1682
	0.0001		12%		3364
	0.00001		40%		11306
News	0.001	24,675	16%	18%	3948
	0.0001		56%		13818
	0.00001		88%		21796
Credit Card	0.001	18,600	26%	72%	4836
	0.0001		70%		12958
	0.00001		83%		15438
Occupancy Detection	0.001	12,747	8%	57%	1020
	0.0001		93%		11855
	0.00001		97%		12365
MNIST	0.001	43,400	6%	6%	2459
	0.0001		8%		3472
	0.00001		68%		29657

component could only resolve a portion of the disagreements. This results in reducing the performance by 2% and 7% in the credit card and occupancy detection datasets respectively when compared to the performance achieved with λ =0.0001 (Figure 3.b). Nevertheless, WeSAL still managed to achieve better results than AL in these two datasets. Overall, the results illustrated in Figure 3 show that the proposed method managed to achieve better performance than active learning with all variation of λ in all the datasets.

4.4.2. Sensitivity analysis of labeling functions. To estimate the effect of changing the accuracy of the labeling functions, we repeat the experiments in Section 4.3.1 using sets of labeling functions with varying levels

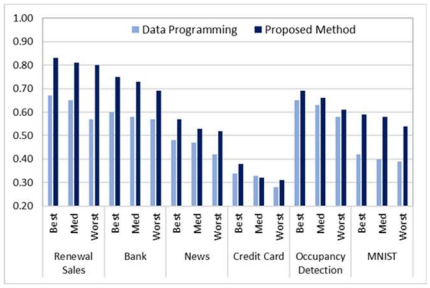

Figure 4. Labeling accuracy of DP and WeSAL with different labeling functions

of accuracy. For each dataset, we create three sets of labeling functions, namely LF_{Best}, $LF_{Mediocre}$, and LF_{Worst} by sampling the best, mediocre, worst three labeling functions from the original set (Table 2). The overall accuracy and F1 measures for each set are reported in Table 6, along with the performance of the discriminative model of both WeSAL and DP.

The results show that discriminative model in WeSAL achieves better performance in all the problems. The table also illustrates that using a smaller number of labeling functions affects the coverage of the training set, and hence, negatively influences the discriminative models. However, WeSAL tries to address abstaining situations by providing true labels to improve the coverage. Also, the results show that some LF_{Worst} sets have low accuracy levels close to the accuracy threshold such as the credit card dataset. As a result, the MCC values of DP and WeSAL decreased by 27% and 14% respectively compared to the MCC levels obtained using the original set (Table 3). However, WeSAL managed to achieve better performance than DP since it enhances the accuracy of these labeling functions by resolving some of their disagreements.

Table 6. Performance of DP and WeSAL with different sets of labeling functions

Datasets	LFs Sets	Labeling functions		WeSAL (Discriminative Model)				DP (Discriminative Model)			
		Accuracy	F1	P	R	MCC	F1	P	R	MCC	F1
Renewal Sales	LF_{Best}	0.80	0.78	0.88	0.90	0.90	0.89	0.85	0.73	0.75	0.79
	$LF_{Mediocre}$	0.76	0.79	0.85	0.89	0.87	0.87	0.82	0.70	0.71	0.76
	LF_{Worst}	0.71	0.77	0.81	0.89	0.81	0.85	0.79	0.61	0.68	0.69
Bank	LF_{Best}	0.84	0.76	0.84	0.86	0.85	0.85	0.83	0.70	0.75	0.76
	$LF_{Mediocre}$	0.78	0.79	0.76	0.81	0.80	0.78	0.80	0.69	0.73	0.74
	LF_{Worst}	0.70	0.81	0.73	0.8	0.79	0.76	0.77	0.65	0.72	0.70
News	LF_{Best}	0.79	0.79	0.86	0.90	0.92	0.88	0.82	0.90	0.88	0.86
	$LF_{Mediocre}$	0.73	0.82	0.82	0.88	0.90	0.85	0.80	0.86	0.85	0.83
	LF_{Worst}	0.69	0.81	0.79	0.84	0.85	0.81	0.79	0.85	0.81	0.82
Credit Card	LF_{Best}	0.72	0.73	0.90	0.89	0.86	0.89	0.85	0.60	0.69	0.70
	$LF_{Mediocre}$	0.67	0.71	0.88	0.85	0.81	0.86	0.83	0.59	0.62	0.69
	LF_{Worst}	0.63	0.70	0.86	0.80	0.78	0.83	0.80	0.57	0.52	0.67
Occupancy Detection	LF_{Best}	0.85	0.79	0.88	0.85	0.90	0.86	0.86	0.82	0.80	0.84
	$LF_{Mediocre}$	0.77	0.79	0.87	0.83	0.86	0.85	0.84	0.81	0.76	0.82
	LF_{Worst}	0.70	0.85	0.81	0.79	0.82	0.80	0.83	0.78	0.71	0.80
MNIST	LF_{Best}	0.81	0.72	0.85	0.87	0.91	0.86	0.83	0.80	0.84	0.81
	$LF_{Mediocre}$	0.79	0.75	0.82	0.87	0.91	0.84	0.80	0.79	0.80	0.79
	LF_{Worst}	0.75	0.74	0.80	0.8	0.88	0.80	0.78	0.75	0.77	0.76

We also report the labeling accuracy achieved using each set of labeling functions. The results are illustrated in Figure 4 and show that WeSAL maintained its superiority of generating more accurate labels than DP in all the problems. Overall, the results depict that reducing the accuracy and the coverage of the labeling functions deteriorate the discriminative model performance. However, the experiments show that WeSAL manages to outperform DP since it injects the domain expertise to resolve the abstaining situations (increase the coverage) and refine the disagreements between the labeling functions (enhance the accuracy).

5. Related work

WeSAL utilizes weak supervision with AL to create large training datasets. Therefore, we surveyed research [3], [19], [20] that employs weak supervision to label datasets. For example, Hickson et al. [19] propose unsupervised clustering method to classify objects using unlabeled data. Another research [3] investigates information retrieval by modeling weak sources as noisy channels and tries to learn accurate signals. Xu et al. [20] design a solution that employs weak labels and learns to semantically segment images. Although all these approaches use weak supervision sources, unlike WeSAL, none of them tried to enhance the accuracy of the resulting labels using domain experience.

Focusing on enhancing the labels quality, other research [1], [4], [5], [21], [22] attempt to denoise weak supervision sources. For example, in [21], an end-to-end system is proposed for multi-task learning, which learns the accuracy of weak sources. Also, Wu et al. in [22] provide a programming model to convert domain experience to a form of supervision to train knowledge base construction systems. Moreover, authors in [5] present a system that creates heuristics automatically and uses generative models to denoise them. Although all these efforts have employed the idea of generative models to denoise the imperfect sources of labels, none of them have investigated the process of refining the input to the generative model using active learning.

On the other hand, there is ample research [23]–[26] that looks into enhancing the scalability of AL. For instance, authors in [23] investigate the annotation cost for AL in real situations and propose a cost-sensitive tree sampling algorithm to reduce the annotation effort. Another recent study [24] applies AL to the social media domain to identify malicious contents. Although the results show that the proposed technique achieves respectable classification accuracy, the method is only applicable to shortlisted textual/link-based posts and validated using a set of datasets with a maximum size of 32k records. Addressing the problem of classifying new classes, researchers in [25] provide an approach that combines Support Vector Machines with clustering to learn new classes. The approach aims at reducing the annotation cost by optimizing the number of iterations that AL requires. Other research in [26] studies the problem of applying AL to large datasets for multi-class classifications tasks and proposes a new query selection criterion to enable hierarchical expansion of candidates. However, in contrast to our approach, the approaches [23], [24], [26] are validated using a group of synthetic and real-world datasets varying in size with a maximum of 100k records. The algorithm in [23] used a set of 12 datasets from the UCI Repository with a maximum size of 32k records. Hence, the applicability of these methods is not guaranteed for large real-world datasets.

Furthermore, several approaches [27], [28] are proposed which integrate AL with weak supervision. Authors in [27] explored both AL and weak supervision as ways to use model assertion to specify constraints on model outputs. Alternatively, authors in [28] apply AL to multiple instance classification where data are weakly labeled. Nevertheless, unlike the proposed method, neither of these approaches tries to reduce the labeling cost while improving the scalability of the output labels.

6. Conclusions

In this paper, we present a new method for generating massive labeled data. The proposed method applies weak supervision with active learning to incorporate users while profiting from the scalability of weak supervision. The method starts with collecting noisy labels from high-level inputs. Then, it refines these labels by resolving the conflicts between the inputs using active learning. To evaluate the proposed method, we applied it to a real case within our industrial partner, IBM, to generate labels for a large-scale dataset of more than 1.3 million records along with five real-world classification tasks. The empirical results show that the proposed method outperforms weak supervision by up to 18% in labeling accuracy. The method also achieves better results than active learning while reducing the labeling accuracy by up to 36%.

7. References

[1] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré, "Data Programming: Creating Large Training Sets, Quickly," in *Advances in Neural Information Processing Systems*, 2016, pp. 3567–3575.

[2] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowledge and Information Systems*, vol. 35, no. 2, pp. 249–283, 2013.

[3] H. Zamani and W. B. Croft, "On the theory of weak supervision for information retrieval," in *ACM International Conference on Theory of Information Retrieval*, 2018.

[4] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: rapid training data creation with weak supervision," *VLDB Endowment*, vol. 11, pp. 269–282, 2017.

[5] P. Varma and C. Ré, "Snuba: automating weak supervision to label training data," *VLDB Endowment*, vol. 12, 2018.

[6] S. H. Bach, B. He, A. Ratner, and C. Ré, "Learning the Structure of Generative Models without Labeled Data," *ArXiv170300854 Cs Stat*, 2017.

[7] G. V. Cormack and M. R. Grossman, "Scalability of Continuous Active Learning for Reliable High-Recall Text Classification," in *ACM International Conference on Information and Knowledge Management*, 2016.

[8] H. Yu, X. Yang, S. Zheng, and C. Sun, "Active Learning from Imbalanced Data: A Solution of Online Weighted Extreme Learning Machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 4, 2019.

[9] E.-C. Huang, H.-K. Pao, and Y.-J. Lee, "Big active learning," in *IEEE International Conference on Big Data*, Boston, MA, USA, 2017, pp. 94–101.

[10] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-Effective Active Learning for Deep Image Classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2017.

[11] P. Jain and A. Kapoor, "Active learning for large multi-class problems," in *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 762–769.

[12] S. Ertekin, J. Huang, L. Bottou, and L. Giles, "Learning on the border: active learning in imbalanced data classification," in *ACM conference on information and knowledge management*, Lisbon, Portugal, 2007, pp. 127–136.

[13] M. E. Ramirez-Loaiza, M. Sharma, G. Kumar, and M. Bilgic, "Active learning: an empirical study of common baselines," *Data Mining and Knowledge Discovery*, vol. 31, no. 2, pp. 287–313, 2017.

[14] P. Varma, D. Iter, C. De Sa, and C. Ré, "Flipper: A Systematic Approach to Debugging Training Sets," in *Workshop on Human-In-the-Loop Data Analytics*, 2017.

[15] P. Varma, B. He, D. Iter, P. Xu, R. Yu, C. D. Sa, and C. Ré, "Socratic Learning: Augmenting Generative Models to Incorporate Latent Subsets in Training Data," *ArXiv161008123 Cs Stat*, 2017.

[16] M. Liu, W. Buntine, and G. Haffari, "Learning How to Actively Learn: A Deep Imitation Learning Approach," in *Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, 2018, pp. 1874–1883.

[17] G. Beatty, E. Kochis, and M. Bloodgood, "The Use of Unlabeled Data Versus Labeled Data for Stopping Active Learning for Text Classification," in *IEEE International Conference on Semantic Computing*, 2019, pp. 287–294.

[18] M. Bloodgood and K. Vijay-Shanker, "A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping," in *Conference on Computational Natural Language Learning*, Boulder, Colorado, 2009, pp. 39–47.

[19] S. Hickson, A. Angelova, I. Essa, and R. Sukthankar, "Object category learning and retrieval with weak supervision," *ArXiv Prepr. ArXiv180108985*, 2018.

[20] J. Xu, A. G. Schwing, and R. Urtasun, "Learning to Segment Under Various Forms of Weak Supervision," in *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015.

[21] A. Ratner, B. Hancock, J. Dunnmon, R. Goldman, and C. Ré, "Snorkel MeTaL: Weak Supervision for Multi-Task Learning," in *the Second Workshop on Data Management for End-To-End Machine Learning*, New York, NY, USA, 2018.

[22] S. Wu, L. Hsiao, X. Cheng, B. Hancock, T. Rekatsinas, P. Levis, and C. Ré, "Fonduer: Knowledge Base Construction from Richly Formatted Data," in *International Conference on Management of Data*, Houston, USA, 2018, pp. 1301–1316.

[23] Y.-L. Tsou and H.-T. Lin, "Annotation cost-sensitive active learning by tree sampling," *Machine Learning*, 2019.

[24] S. D. Bhattacharjee, W. J. Tolone, and V. S. Paranjape, "Identifying malicious social media contents using multi-view Context-Aware active learning," *Future Generation Computer Systems*, vol. 100, pp. 365–379, 2019.

[25] L. F. S. Coletta, M. Ponti, E. R. Hruschka, A. Acharya, and J. Ghosh, "Combining clustering and active learning for the detection and learning of new image classes," *Neurocomputing*, vol. 358, pp. 150–165, 2019.

[26] W. Fu, M. Wang, S. Hao, and X. Wu, "Scalable Active Learning by Approximated Error Reduction," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2018, pp. 1396–1405.

[27] D. Kang, D. Raghavan, P. Bailis, and M. Zaharia, "Model assertions for debugging machine learning," in *NeurIPS MLSys Workshop*, 2018.

[28] M. Carbonneau, E. Granger, and G. Gagnon, "Bag-Level Aggregation for Multiple-Instance Active Learning in Instance Classification Problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1441–1451, May 2019.