Bidirectional Active Learning: A Two-Way Exploration Into Unlabeled and Labeled Data Set

Xiao-Yu Zhang, *Member, IEEE*, Shupeng Wang, and Xiaochun Yun, *Member, IEEE*

***Abstract*— In practical machine learning applications, human instruction is indispensable for model construction. To utilize the precious labeling effort effectively, active learning queries the user with selective sampling in an interactive way. Traditional active learning techniques merely focus on the unlabeled data set under a unidirectional exploration framework and suffer from model deterioration in the presence of noise. To address this problem, this paper proposes a novel bidirectional active learning algorithm that explores into both unlabeled and labeled data sets simultaneously in a two-way process. For the acquisition of new knowledge, forward learning queries the most informative instances from unlabeled data set. For the introspection of learned knowledge, backward learning detects the most suspiciously unreliable instances within the labeled data set. Under the two-way exploration framework, the generalization ability of the learning model can be greatly improved, which is demonstrated by the encouraging experimental results.**

***Index Terms*— Active learning, bidirectional exploration, generalization performance, noisy data, selective sampling.**

I. Introduction

With the remarkable progress in computer science and artificial intelligence, machine learning is thriving as the computational process of extracting patterns in data and making predictions based on the experience gained from these patterns [1]–[6]. Classic machine learning can be divided into two types: 1) supervised learning and 2) unsupervised learning [7]–[10]. The former aims at inferring a function from the labeled training data, whereas the latter tries to find hidden structure in the unlabeled data. Recent researches indicate that the unlabeled data, when used in conjunction with the labeled data, can produce considerable improvement in learning performance. To make full use of both the labeled and unlabeled data, semisupervised learning has been widely studied as an effective combination of supervised and unsupervised learning [11]–[13]. The cost associated with the labeling process often renders a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. As a result, the available data set typically consists of a small amount of labeled data and a large amount of unlabeled data. In such circumstances, semisupervised learning can be of great practical value.

In practical applications, the instructive human intuition is precious and essential for model training. However, manual labeling is labor intensive and time consuming. How to utilize human effort effectively is the key point of related studies. Traditionally, the instances to be labeled as training data are collected in an unselective way, which are either prepared beforehand or returned randomly as queries. The learning algorithm has no choice but to passively accept and learn from the training data. In contrast, active learning, aiming at achieving the highest learning performance with the least possible labeling effort, is an ideal solution with the ability to implement selective sampling [14]–[16]. The main idea is to actively select the most informative instances with the greatest potential to improve the model. Active learning can be thought of as a special case of semisupervised learning, in which the learning algorithm is able to interactively query the user to obtain outputs at new data. As a result, to achieve a certain learning performance, the number of instances required in active learning can often be much lower than that in traditional methods.

The effectiveness of active learning stems from the in-depth exploration into the unlabeled data set [17], [18], based on which the potential contribution of each unlabeled instance is obtained. However, there is little exploration based on the labeled data set in active learning algorithms studied so far. Once an instance is labeled, as is often the case, its labeled information is accepted and used for training unreservedly. As will be discussed in detail, this widely used one-way manipulation may lead to model deterioration in the presence of noise. The noise may come from either the mislabeled instances or the correctly labeled outliers, both of which have negative impact on the model's generalization ability. There is related work dealing with noisy labels [19]–[22], which places major emphasis on reliable label estimation and achieves remarkable improvement in learning performance. However, the underlying instance flow is still unidirectional. In this paper, we propose a novel bidirectional active learning (BDAL) algorithm based on the reconsideration of knowledge acquisition path, which explores into both the unlabeled and labeled data set in a two-way process. On the

Manuscript received June 26, 2014; revised December 2, 2014, January 26, 2015, and February 3, 2015; accepted February 5, 2015. Date of publication February 26, 2015; date of current version November 16, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61271275 and Grant 61202067, in part by the National High Technology Research and Development Program of China under Grant 2012AA013001, Grant 2013AA013205, and Grant 2013AA01A213, and in part by the Beijing Science and Technology Program under Grant Z131100001113034. *(Corresponding author: Shupeng Wang.)*

The authors are with Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhangxiaoyu@iie.ac.cn; wangshupeng@iie.ac.cn; yunxiaochun@iie.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2015.2401595

one hand, the most informative instances that are most valuable for model update are selected from the unlabeled data set through forward learning. On the other hand, the most suspiciously unreliable instances within the labeled data set that are most responsible for model deterioration are detected via backward learning. Under the framework of BDAL, the generalization ability of the learning model can be greatly improved. Encouraging results are received from experiments conducted on synthetic data, handwritten digit, real-world image, and patent document classification.

The rest of this paper is organized as follows. Section II introduces related work in active learning, which unanimously follows the unidirectional manner in instance flow. In Section III, we present the BDAL as a novel two-way exploration framework. The experimental results are reported and discussed in Section IV. Finally, the conclusion is drawn in Section V.

II. Unidirectional Active Learning

As stated before, active learning is effective to solve machine learning problems where unlabeled data are abundant or easily accessible but labeled data are difficult, time consuming, or expensive to obtain. The key idea behind active learning is that the learning algorithm can achieve greater performance with fewer training data if it is allowed to choose the data from which it learns. The strategy of selective sampling plays a critical role in active learning algorithm, which is formulated by the evaluation of informativeness of unlabeled instances. The task of active learning has been addressed in various ways. Perhaps the simplest and most commonly used strategy is uncertainty sampling [23]–[27], in which the learning algorithm selects the instance about which it is least certain how to label. Similarly, the query-by-committee strategy adopts a voting mechanism by maintaining a committee of competing models which are allowed to vote on the labels [28]–[32]. The instance about which the models most disagree is selected. Another more theoretically motivated selection strategy is based on decision-theoretic approach and selects the instance that would impart the greatest reduction to the model's generalization error if its label was known [33]–[36]. The idea is to estimate the expected error of the model with each unlabeled instances incorporated into the training data set and select the instance with minimal expected future error (or risk). By explicit measuring of expected error, the most informative instance can be captured with higher precision.

Most active learning algorithms proposed so far are unidirectional from the instance point of view. When an instance is labeled and used for model update, it is taken out of the unlabeled data set and added into the labeled data set. This process can be named unidirectional active learning (UDAL), which takes on a one-way path from the unlabeled data set to the labeled data set, as shown in Fig. 1. Ever since being labeled, instances remain in the labeled data set. This is straightforward and seems trivial for further discussion under the strong assumption that nothing goes wrong with either the instances or the labeling results. In practice, however,

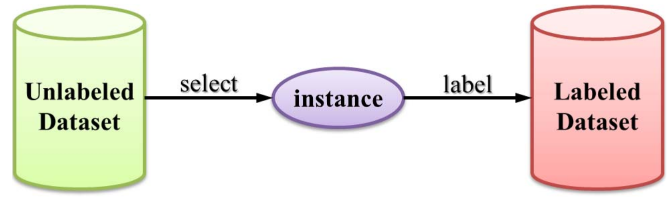


Fig. 1. UDAL.

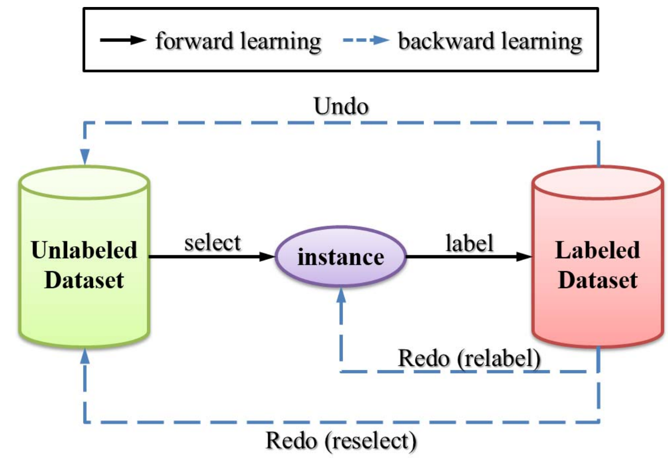


Fig. 2. BDAL.

one can reasonably expect some noise resulting from the data set itself and the oracles (labelers) performing labeling. As a result, in UDAL, the noise in the labeled data set will constantly affect model training and jeopardize the learning performance.

To cope with the noise within the data set, prior research makes use of labels obtained from multiple oracles to average out some of the noise [19], [20]. This type of approach adopts a more complicated labeling mechanism, in which an unlabeled instance can be labeled multiple times before a reliable label is obtained. However, from the instance point of view, it still follows a unidirectional process.

III. Bidirectional Active Learning

We believe that acquiring new knowledge and examining learned knowledge are both indispensable for an effective active learning framework. The former helps to explore the unknown information, while the latter rectifies misconception and refines the obtained knowledge. From the viewpoint of instance flow in learning, this is a two-way process. Unlike UDAL, which embraces the labeled information without any doubt, this two-way manipulation places the labeled instances under inspection before taking them as training data. As a consequence, the suspiciously unreliable instances will be detected and processed for the sake of learning performance.

In this paper, we propose a novel BDAL mechanism to implement the two-way exploration into the unlabeled and labeled data set.

As shown in Fig. 2, the BDAL consists of forward learning and backward learning, which correspond to the acquisition of new knowledge and the introspection of learned knowledge, respectively. In forward learning, informative instances are selected from unlabeled data set and labeled as training data; while in backward learning, the model is further refined by examining the labeled instances and locating the underlying noise.

There is related work on active learning with noisy oracles [21], [22], which does include the idea of repeated labeling. However, this kind of method, what we call multilabeling approach, is essentially different from the BDAL proposed in this paper. First of all, the primary objective of the multilabeling approach is to improve data quality with the collective effort from multiple oracles. As a result, it can be regarded as another unidirectional mechanism for reliable label estimation. Admittedly, the labeled instances are evaluated for selective repeated labeling, but the major concern is to provide an accurately labeled data set. In contrast, BDAL reevaluates the labeled data set not only to confirm the labels themselves but also to provide guidance for the selection of unlabeled instances or the removal of noise, which creates a path from the labeled data set to the unlabeled data set. Moreover, the multilabeling approach is only applicable to noise derived from the mislabeled instances, but cannot deal with the outliers. In BDAL, as will be introduced in detail, various solutions are provided based on the Undo and Redo process to alleviate or even eliminate the negative impact of noise. Last but not least, the multilabeling approach treats all the instance in the same way and selects instances for repeated labeling under a unanimous model, taking the unlabeled instance as a special labeled instance with 0 label. For example, a labeled instance can be represented as a label set: $\{+, -, +\}$, while an unlabeled instance: $\{\}$. In contrast, BDAL analyzes the unlabeled and labeled instances via forward and backward learning separately, making the model update more effectively.

In what follows, we let x denote the input feature of an instance, and $y \in \{1, \ldots, K\}$ denote the class label where K is the number of classes. U and L stand for the unlabeled and labeled data set, respectively. Probabilistic model is used for classification based on the posterior distribution $P(y|x; \theta_L)$ of the label y conditioned on the input x, where θ_L is the model parameter optimized for the corresponding labeled data set L.

A. Forward Learning

As a matter of fact, the forward learning is identical with UDAL, which explores the unlabeled data set and aims at selecting the most informative unlabeled instance to label.

Following the expected error reduction criterion for selective sampling, the most informative instance is the one that maximizes the expected error reduction, or equivalently minimizes the expected entropy over the unlabeled data set

$$x^*_{\mathrm{FL}} = \arg\min_{x \in U} \sum_i P(y_i|x; \theta_L) \sum_{x^u \in U-x} H\left(y^u|x^u; \theta_{L+(x,y_i)}\right) \tag{1}$$

Input		U: unlabeled dataset; L: labeled dataset;
Forward Learning	**(1)**	**Forward instance selecting** Select forward instances x^*_{FL} from U according to (1);
	(2)	**Forward instance labeling** Label x^*_{FL} and obtain (x^*_{FL} , y); Remove x^*_{FL} from U and add it into L: $U = U \setminus \{ x^*_{FL} \}, L = L + \{(x^*_{FL} , y)\}$.
Model Update	**(3)**	**Training** Train a new model based on L.

Fig. 3. Forward learning algorithm.

where $L + (x, y_i)$ stands for the labeled data set with a new instance (x, y_i) labeled, and

$$H(y|x; \theta) = -\sum_i P(y_i|x; \theta) \cdot \log P(y_i|x; \theta) \tag{2}$$

represents the conditional entropy of the label y with respect to the instance x given the model parameter θ.

We name the most informative unlabeled instance selected via forward learning the forward instance. After the selection stage, the obtained forward instance x^*_{FL} is labeled and incorporated into the labeled data set to facilitate model update.

The algorithm of forward learning which consists of selecting and labeling of the forward instance is summarized in Fig. 3.

B. Backward Learning

As discussed above, the backward learning explores the labeled rather than the unlabeled data set. The main idea can be interpreted as detecting and dealing with the most unreliable labeled instance that is most responsible for the deterioration of the classification model. Similarly, the target instance of backward learning can be called the backward instance. Backward learning is comprised of detecting and processing of backward instance.

1) Backward Instance Detecting: In order to detect backward instance, the influence of each labeled instance on the classification model should be evaluated. According to whether we focus on the labeled instance itself or the specific label it takes on, the detection falls into instance-level and label-level methods.

a) Instance-level detecting: From the expected error reduction point of view, the backward instance is the labeled instance without which the expected entropy over unlabeled data set would be minimized. Similar to (1), instance-level detecting can be formulated as follows:

$$x^*_{\mathrm{BL}} = \arg\min_{x \in L} \sum_i P(y_i|x; \theta_{L\setminus(x,y_{i^*})}) \sum_{x^u \in U} H\left(y^u|x^u; \theta_{L\setminus(x,y_{i^*})}\right) \tag{3}$$

where $L\setminus(x, y_{i^*})$ stands for the labeled data set with a specific labeled instance (x, y_{i^*}) excluded.

Equation (3) is analogous with (1) except that the backward instance is selected from the labeled data set. In this case, the second summation over the unlabeled data set in (3) is constant with respect to the first summation over various labels y_i of the labeled instance x, and thus it can be taken out of the first summation. As a result, (3) can be rewritten as follows:

$$\begin{aligned} x_{\mathrm{BL}}^* &= \arg\min_{x \in L} \sum_i P\left(y_i | x; \theta_{L \backslash (x, y_{i^*})}\right) \sum_{x^u \in U} H\left(y^u | x^u; \theta_{L \backslash (x, y_{i^*})}\right) \\ &= \arg\min_{x \in L} \sum_{x^u \in U} H\left(y^u | x^u; \theta_{L \backslash (x, y_{i^*})}\right) \sum_i P\left(y_i | x; \theta_{L \backslash (x, y_{i^*})}\right) \\ &= \arg\min_{x \in L} \sum_{x^u \in U} H\left(y^u | x^u; \theta_{L \backslash (x, y_{i^*})}\right). \end{aligned} \tag{4}$$

The second equality of (4) follows from the commutability of the two summations discussed above; the third equality follows from the fact that

$$\sum_i P\left(y_i \middle| x; \theta_{L \backslash (x, y_{i^*})}\right) = 1. \tag{5}$$

To locate the backward instance, (4) questions the reliability of the labeled instances, and tries to answer the questions whether and to what extent the model would be improved if we left a certain labeled instance unlabeled (or unused).

b) Label-level detecting: There is an alternative way to perform backward learning, on the label level rather than instance level. The key point is to find the most suspiciously mislabeled instance. If it adopted a new label other than the current one, the expected entropy over the unlabeled data set would be minimized. The formulation is as follows:

$$\begin{aligned} x_{\mathrm{BL}}^* &= \arg\min_{x \in L} \frac{1}{Z} \sum_{i \neq i^*} P\left(y_i | x; \theta_{L \backslash (x, y_{i^*})}\right) \\ &\quad \times \sum_{x^u \in U} H\left(y^u | x^u; \theta_{L | (x, y_i)}\right) \end{aligned} \tag{6}$$

where Z plays the role as a normalization coefficient

$$\begin{aligned} Z &= \sum_{i \neq i^*} P(y_i | x; \theta_{L \backslash (x, y_{i^*})}) \\ &= 1 - P(y_{i^*} | x; \theta_{L \backslash (x, y_{i^*})}) \end{aligned} \tag{7}$$

and $L|(x, y_i)$ stands for the labeled data set with instance x labeled as y_i.

Different from (4), (6) questions the label of each labeled instance, and tries to answer the questions whether and to what extent the model would be improved if a labeled instance was labeled otherwise.

2) Backward Instance Processing: Both instance-level and label-level methods are effective for detecting backward instance. After that, measures should be taken to eliminate negative effects of the backward instance. According to whether the backward instance is retained in the training data set, backward learning can be divided into two manners: 1) the Undo process and 2) the Redo process.

a) Undo: The main idea of Undo process is to eliminate the negative influence of the backward instance by excluding it from the training data set. As the saying goes, less is more. The existence of backward instance will constantly jeopardize the model performance. In this case, a training data set without the backward instance, even if the number of labeled instances is less, can be more helpful for model update. As a result, undoing the influence of backward instance is an effective solution. During the Undo process, the training data set simply rolls back from L to $L \backslash (x_{\mathrm{BL}}^*, y_i)$.

Since the Undo process discards the backward instance as a whole, it is suitable for instances detected via instance-level method.

b) Redo: The main idea of Redo process is to validate the label of the backward instance by labeling it again or sampling within its neighborhood. The corresponding implementation methods are named relabel and reselect, respectively. Compared with the seemingly radical Undo process, the Redo process looks much more moderate by giving a second chance to either the backward instance or the instance nearby for rectification.

1) *Redo (Relabel):* In the relabel process, the backward instance x_{BL}^* is returned to be labeled for the second time. If the new label is identical with the original one, the backward instance will be treated as a new instance and incorporated into training data set as a copy of the original one. That is to say, the labeled data set will contain two identical instances. It is similar to the idea of boosting [37]–[40] in that the probably mislabeled instance will be paid closer attention and given higher weight in the cost function. If the label of the backward instance is altered after relabeling, we simply replace the original label with the new one. In this way, the labeling error is rectified.
2) *Redo (Reselect):* There is actually no additional instance incorporated in either the Undo or relabel process. If we want to check the reliability of backward instance and meanwhile obtain new information, we can conduct reselect process by sampling a new instance to label which is the nearest neighbor of the backward instance x_{BL}^*. The probability is high that an instance shares the same label with its nearest neighbor. If the label of the nearest neighbor turns out to be identical with that of x_{BL}^*, the confidence in both instances is strengthened. Otherwise, incorporation of the nearest neighbor can alleviate the influence of x_{BL}^*, and rectify the classification model to some extent.

Since the Redo process pays more attention to the label of backward instance, it naturally matches up with label-level detecting method.

3) Backward Learning Algorithm: As a combination of backward instance detecting and processing, the algorithm of backward learning is summarized in Fig. 4.

C. Serial and Batch Mode

In UDAL, and similarly forward learning of BDAL, the choice of serial or batch mode is a tradeoff between effectiveness and efficiency [41]–[46]. The comparison of serial and batch mode is shown in Fig. 5. The serial mode labels a single forward instance each time and updates the model right away. In serial mode, the model can start making progress immediately after the labeling of each new instance. However, the challenge confronted by serial mode is the overwhelming computational complexity resulting from frequent training, since every labeling is accompanied by model training.

Input	U: unlabeled dataset; L: labeled dataset; T_{BL}: choice of backward learning strategy, T_{BL} = {Undo, Redo: re-label, Redo: re-select}.
Backward Learning	**(1) Backward instance detecting** • In case of T_{BL} = Undo, · Detect backward instance (x^*_{BL} , y_{OL}) from L according to instance-level detection (4); • In case of $T_{BL} \in$ {Redo: re-label, Redo: re-select}, · Detect backward instance (x^*_{BL} , y_{OL}) from L according to label-level detection (6); **(2) Backward instance processing** • In case of T_{BL} = Undo, · Remove x^*_{BL} from L: $L = L \setminus \{(x^*_{BL}, y_{OL})\}$; • In case of T_{BL} = Redo: re-label, · Re-label x^*_{BL} and obtain (x^*_{BL} , y_{RL}); · If $y_{RL} = y_{OL}$, add (x^*_{BL} , y_{RL}) as a copy of (x^*_{BL} , y_{OL}): $L = L + \{(x^*_{BL}, y_{RL})\}$; · If $y_{RL} \neq y_{OL}$, replace (x^*_{BL} , y_{OL}) with (x^*_{BL} , y_{RL}): $L = L \setminus \{(x^*_{BL}, y_{OL})\} + \{(x^*_{BL}, y_{RL})\}$; • In case of T_{BL} = Redo: re-select, · Select the nearest neighbor $x^*_{BL\_RS}$ of x^*_{BL} from U; · Label $x^*_{BL\_RS}$ and obtain ($x^*_{BL\_RS}$, y_{RS}); · Remove $x^*_{BL\_RS}$ from U and add it into L: $U = U \setminus \{x^*_{BL\_RS}\}$, $L = L + \{(x^*_{BL\_RS}, y_{RS})\}$.
Model Update	**(3) Training** Train a new model based on L.

Fig. 4. Backward learning algorithm.

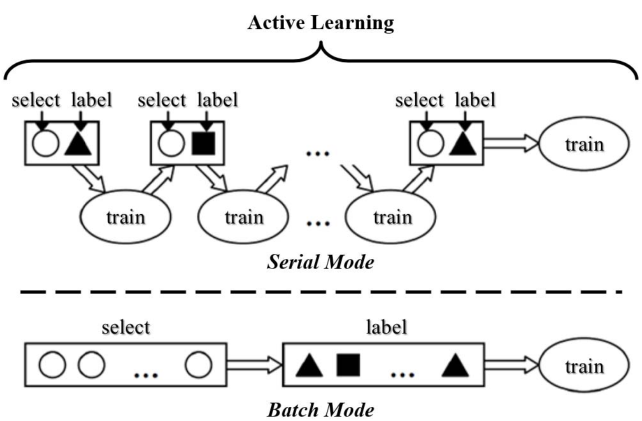


Fig. 5. Comparison of serial and batch modes.

In contrast, the batch mode labels a batch of instances, i.e., the forward instances that rank top N_{FL} according to (1), before taking a training step, where N_{FL} denotes the number of instances in a batch. Compared with serial mode, batch mode is relatively more suitable for practical applications because of the restrained use of training. The problem that obsesses the batch mode is redundancy. Since the instances are selected all at once, the correlation is inevitably neglected. As a result, the instances within a batch can be highly correlated and, on the whole, provide rather limited information for model update.

To cope with the problem of redundancy in batch mode and meanwhile retain the advantage of efficiency, a grouped batch method is adopted as a simple solution. First, the instances are clustered into $M(M > N_{\mathrm{FL}})$ groups. Then within each group, a single forward instance is selected. Finally, among the selected M forward instances, the top N_{FL} ranked ones that minimize the expected entropy in (1) come into a batch. Since the selected instances are derived from different groups, redundancy within the batch can be significantly reduced. When M equals to the total number of unlabeled instances, the grouped batch method is reduced to the conventional batch mode. In the case of $N_{\mathrm{FL}} = 1$, the grouped batch method is equivalent to the serial mode.

It is also the case with backward learning. For the sake of effectiveness and efficiency, the same grouped batch method can be used to select a batch of backward instances before model training. The difference is that a single backward instance is selected for each of the M groups according to (4) or (6) rather than (1), and the top N_{BL} ranked ones form a batch of backward instances.

D. Algorithm

BDAL is the combination of forward learning and backward learning, in which both the unlabeled and labeled data are explored. In BDAL, forward and backward learning are performed within one process, the algorithm of which is shown in Fig. 6.

As shown in Fig. 6, the level of user involvement is controlled by the input N_L, which is the total number of instances for manual labeling. With a specific N_L given, the algorithm can operate at limited cost of human labeling effort, and various strategies can be compared rather fairly under the same user interference. As for the evaluation of the number of backward instances N_{BL}, it can either be preset heuristically or determined based on the comparison of expected entropy between forward and backward instances.

At initializing stage, the number of forward instances to be selected and consequently labeled in forward learning, i.e., N_{FL}, is evaluated according to the choice of backward learning type. During the Undo process, the backward instances are simply discarded from the labeled data set and no extra effort is required from the user. Therefore, the entire labeling amount N_L is assigned to N_{FL}. On the contrary, the Redo process, whether it is implemented in the relabel or reselect manner, demands the user to label N_{BL} instances. In this case, forward and backward learning should share the labeling amount N_L. As a result, N_{FL} is the amount that remains after N_{BL} is subtracted from N_L.

At learning stage, forward and backward learning explore the unlabeled and labeled data set simultaneously and prepare a new training data set for further update. In relabel process of backward learning, Diff (S_{BL}, $S_{\mathrm{BL_{RL}}}$) stands for the instances labeled differently in data set S_{BL} and $S_{\mathrm{BL_{RL}}}$.

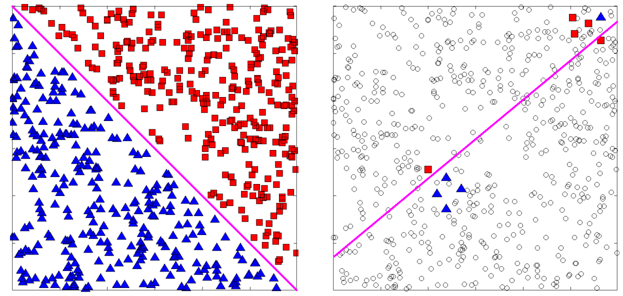
At model update stage, the classification model is retrained using the newly obtained L as training data set.

E. Discussion

BDAL focuses on improving the generalization ability of the model. As a result, not only the unlabeled instances are selectively labeled, but also the labeled instances are

Input	U: unlabeled dataset; L: labeled dataset; N_L: total number of instances to be labeled; N_{BL}: number of backward instances to be selected in backward learning ($N_{BL} < N_L$); T_{BL}: choice of backward learning strategy, T_{BL} = {Undo, Redo: re-label, Redo: re-select}.
Initializing	N_{FL}: number of forward instances to be selected in forward learning; • If T_{BL} = Undo, $N_{FL} = N_L$; • Else $N_{FL} = N_L - N_{BL}$.
Forward Learning	**(1) Forward instance selecting** Select a batch of N_{FL} forward instances from U and obtain a dataset $S_{FL} \subset U$; **(2) Forward instance labeling** Label forward instances in S_{FL}; Remove S_{FL} from U and add it into L: $U = U \setminus S_{FL}$, $L = L + S_{FL}$.
Backward Learning	**(3) Backward instance detecting** Select a batch of N_{BL} backward instances from L and obtain a dataset $S_{BL} \subset L$; **(4) Backward instance processing** • In case of T_{BL} = Undo, · Remove backward instances in S_{BL} from L: $L = L \setminus S_{BL}$; • In case of T_{BL} = Redo: re-label, · Re-label backward instances in S_{BL} and obtain a new dataset S_{BL_RL}; · Incorporate the instances with identical labels as copies of the original ones, and replace the instances with different labels with new ones: $L = L + S_{BL\_RL} \setminus \text{Diff}(S_{BL}, S_{BL\_RL})$; • In case of T_{BL} = Redo: re-select, · Select the nearest neighbors of backward instances in S_{BL} from U and obtain a new dataset $S_{BL\_RS} \subset U$; · Label instances in S_{BL_RS}; · Remove S_{BL_RS} from U and add it into L: $U = U \setminus S_{BL\_RS}$, $L = L + S_{BL\_RS}$.
Model Update	**(5) Training** Train a new model based on L.

Fig. 6. BDAL algorithm.



(a) (b)

Fig. 7. 2-D synthetic data set. ■: positive instance. ▲: negative instance. ○: unlabeled instance. –: classifier. (a) Real data distribution and the optimal classifier. (b) Initial training data and the initial classifier.

selectively detected for potential noise. Here, the noise may stem from either the mislabeled instances or the instances that are correctly labeled but have negative impact on the model's generalization ability, also known as the outliers. For the mislabeled instances, both the Undo and Redo processes of backward learning are reasonable solutions for rectifying or reducing the importance of the labeling errors. For the outliers, backward learning also makes good sense. As we know, the outliers undermine the model's performance with the risk of overfitting, even though their labels are correct. The Undo process eliminates the negative influence of outliers at the cost of slight shrinkage in the scale of training data set. In the Redo process, the backward instances are placed under double check, during which the labels of outliers will be confirmed rather than changed since they are correctly labeled. For implementation, the confirmed outliers will never be selected again as backward instances to prevent endless loop. In this way, the influence of outliers will naturally fade away with the expansion of the labeled data set. It is worth noticing that BDAL places key emphasis on promoting generalization ability of the model rather than denoising. Therefore, whether the noise results from mislabeled instances or outliers is not the major concern of BDAL. As long as an instance is suspiciously responsible for model deterioration when its verification becomes far more urgent than the incorporation of new labeled instances, backward learning will be triggered to alleviate the negative impact.

As a matter of fact, the problem of improving generalization can be addressed by regularization as well, which is a process of introducing additional penalty terms into the cost function to prevent overfitting. Regularization can be seen as an implicit approach to achieve noise tolerance to some extent. As will be shown in the experiments, the model with regularization still suffers from performance degradation when the noise is quite influential. In contrast, BDAL copes with the potential noise in an explicit way based on the exploration into both the unlabeled and labeled data set directly. As a result, the negative impact brought about by the noisy instances can be alleviated or eliminated more effectively. On one hand, regularization works at the model training stage; on the other hand, BDAL mainly focus on the instance selection stage. In this sense, regularization can be incorporated into BDAL framework so as to jointly improve the model's generalization performance implicitly and explicitly.

F. Summary

As a two-way exploration into both the unlabeled and labeled data set, BDAL makes better use of the instances compared with UDAL. In BDAL, forward learning expands the training data set with instances from unlabeled data set that are most valuable for model improvement, meanwhile backward learning refines the training data set through validation of underlying noise within the labeled instances. In this framework, the whole data set is fully explored, and the model is optimized efficiently.

IV. Experiments

In order to test the effectiveness of the proposed BDAL, we apply it to the classification of synthetic data, handwritten digits, real-world images, and patent documents, respectively.

A. Synthetic Data Classification

A two-class synthetic data set is prepared for binary classification. As shown in Fig. 7(a), the data set contains

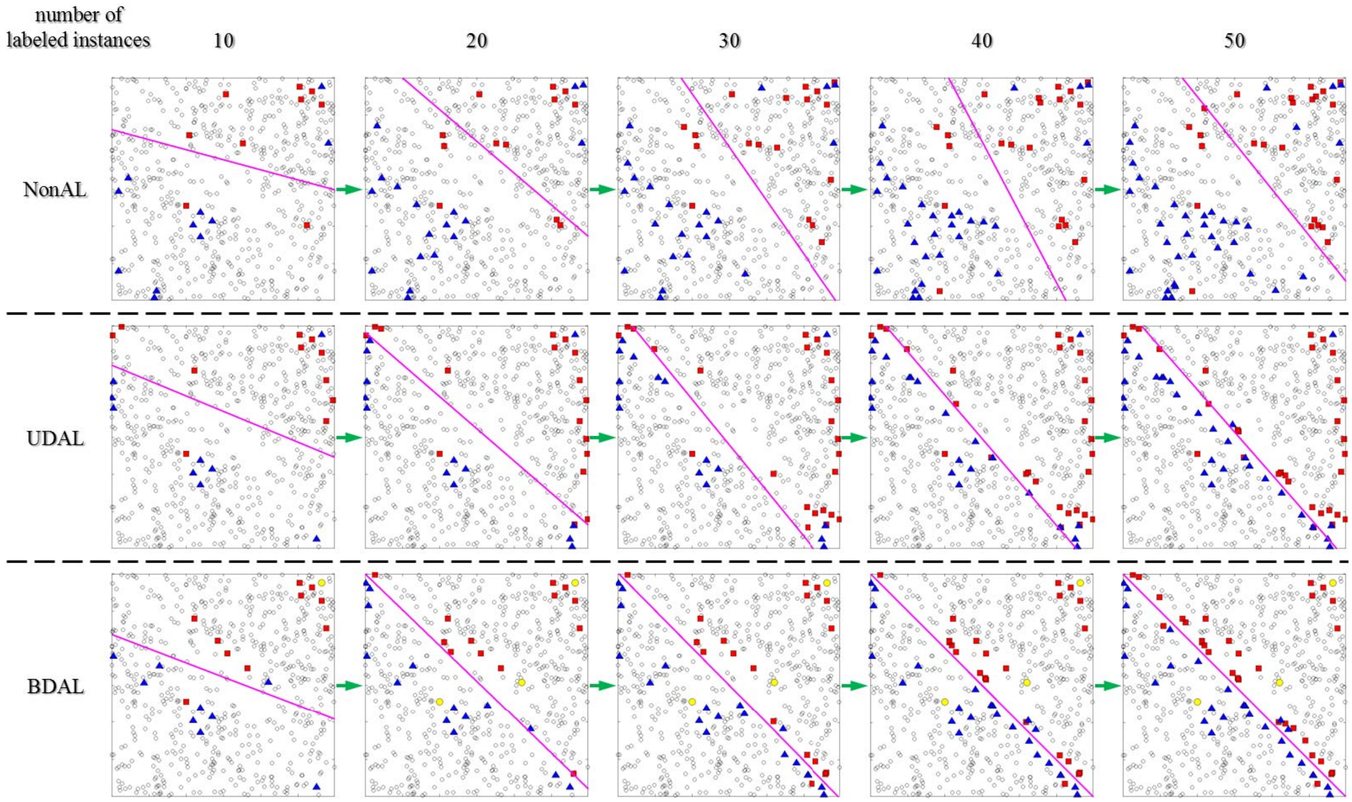


Fig. 8. Synthetic data classification results with N (N = 10, 20, 30, 40, and 50) instances selected and labeled for model update. ■: positive instance. ▲: negative instance. ○: unlabeled instance. ●: backward instance in BDAL. –: classifier.

410 instances which are randomly generated points in the 2-D space. These instances fall into two classes, 205 for each class, and they can be perfectly separated by a linear classification boundary. Initially, as shown in Fig. 7(b), all the instances are unlabeled except 10 randomly selected ones used as training data. Among the 10 initial training instances, two are mislabeled, based on which an initial classifier is obtained. Regularized logistic regression is employed for the construction of binary classification models.

In order to update the current model, more instances are required to be labeled for further training. Active learning is used for effective model learning. In the experiment, UDAL and BDAL, both in serial mode, are compared. For backward learning of BDAL, Undo strategy is used. Nonactive learning (NonAL) method with training instances selected randomly is use as baseline. To simulate noise in the labeling process, an instance is mislabeled with probability 0.1. Note that in all the three methods, regularization is incorporated in the objective functions for model optimization.

Comparison of synthetic data classification performance is shown in Fig. 8, where the same number of instances are labeled in different methods.

1) As can be seen from Fig. 8, in NonAL, the randomly selected instances are not informative enough for model update. As a result, even after 50 instances (with 10% noise) are labeled, the corresponding classifier is still far from satisfactory.
2) In contrast, active learning can actively query the most informative instances. Therefore, better performance can be achieved with less instances used. However, because of the existence of noise, the classification model of UDAL is constantly affected and held back from further improvement. Although regularization is incorporated, the negative impact of noise cannot be completely eradicated. To be specific, in UDAL, with labeled instances increasing from 30 to 50, the model update is rather limited and the bias derived from noise can hardly be eliminated.
3) In BDAL, the most informative instances are selected via forward learning, meanwhile the suspicious noise is detected and subsequently discarded using backward learning. In this way, the learning performance can be remarkably improved. As shown in Fig. 8, after only 20 instances labeled (with three backward instances discarded), the trained classifier is approximately optimal. It is worth noting that the three most influential noisy instances with high classification certainty are precisely detected and removed from the training data set. As for the remaining noisy instances which are close to the classification boundary, their negative impact can be safely ignored. Based on this observation, we reveal an interesting intuitive conclusion without rigorous proof. For forward learning the instances with low classification certainty are usually informative, while for backward learning the potential noisy instance with high certainty are the valuable ones.

To sum up, BDAL outperforms NonAL and UDAL by achieving the best performance with the minimum instances labeled.

TABLE I
ALGORITHMS COMPARED IN EXPERIMENTS

Algorithm	Description
NonAL	Non-active learning
UDAL	Unidirectional active learning
BDAL1	Bidirectional active learning with undo strategy
BDAL2	Bidirectional active learning with re-label strategy
BDAL3	Bidirectional active learning with re-select strategy

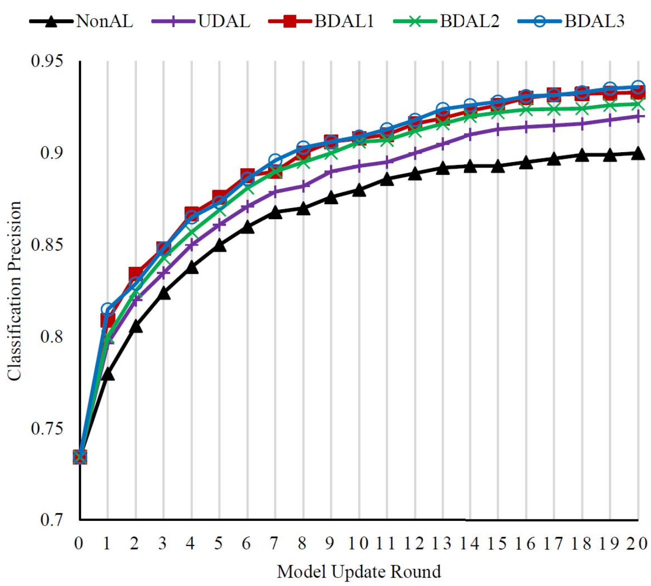


Fig. 9. Handwritten digit classification results.

B. Handwritten Digit Classification

The Mixed National Institute of Standards and Technology database (MNIST) data set [47] of handwritten digits contains a test data set of 10 000 instances. It is a subset of a larger set available from National Institute of Standards and Technology database. Each digit has been size-normalized and centered in a fixed-size image with 28 × 28 pixels. It is a good data set for researchers who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting. In the data set, 2000 instances are randomly picked out and used as a separate test data set, and the remaining 8000 instances are used for model learning.

The vector corresponding to the 28 × 28 pixels is used directly as input feature. Softmax regression, also known as multiclass logistic regression, is adopted to classify MNIST images.

In the data set, 100 images are randomly picked as initial training data for classifier construction. For model update, 100 images are labeled each time with the noise rate at 5%. NonAL, UDAL, and various implementations of BDAL, all in batch mode, are compared, which are listed in Table I. For BDAL, N_{BL} is preset as $N_{\mathrm{BL}} = 3$.

Throughout 20 rounds of model update, the results of handwritten digit classification, averaged over 20 runs, are shown in Fig. 9. The precision of backward instance detection in BDAL is compared in Table II.

TABLE II
PRECISION OF MISLABELED INSTANCE DETECTION

Algorithm	# backward instances	# noisy instances	Detection precision
BDAL1	60	47.20	78.67%
BDAL2	60	44.35	73.92%
BDAL3	60	47.05	78.42%

The analysis is as follows, where > stands for outperforms.

1) *UDAL > NonAL:* Compared with NonAL which has no control over the selection of training data, active learning can actively select the most informative instances with the greatest potential to improve the classification model. As a result, with the same number of instances labeled, classification performance can be boosted more significantly using active learning.
2) *BDAL > UDAL:* In UDAL, the unlabeled data set is explored, while the labeled data set is used without discrimination. In the presence of noise, the learning performance is inevitably undermined. On the contrary, BDAL can effectively combine the exploration into the unlabeled and labeled data set. Beside forward learning which is identical with UDAL, backward learning alleviates or eliminates the negative impact of noise and further refines the model.
3) No significant difference is observed among various implementations of BDAL. Under a relatively high detection precision (at ∼75%) of noise, both Undo and Redo strategy received satisfactory learning performance.

C. Image Classification

The data set is a subset selected from the Corel image CDs. In the data set, there are 50 categories with different semantic meanings, such as car, ship, human, and so on. Each category contains 100 images, thus there are altogether 5000 images. The separate test-set contains 1000 images and the remaining 4000 images are used for model learning.

Color and texture features are employed to represent images. The color features consist of 125-D color histogram and 6-D color moment in RGB. The texture features are extracted using three-level discrete wavelet transformation, and the mean and variance averaging on each of 10 subbands form a 20-D vector. Regularized softmax regression is used for classification.

In the experiments, the first 10 images of each category, 500 in all, are picked out as initial training data for classification model construction. For model update, 500 images are labeled each time with the noise rate at 5%. Algorithms listed in Table I are compared, and N_{BL} is preset as $N_{\mathrm{BL}} = 10$ for BDAL.

The results of image classification, averaged over 50 runs, are shown in Fig. 10. The precision of backward instance detection in BDAL is compared in Table III.

Similar comparison results are received, i.e., BDAL > UDAL > NonAL. Under a lower detection

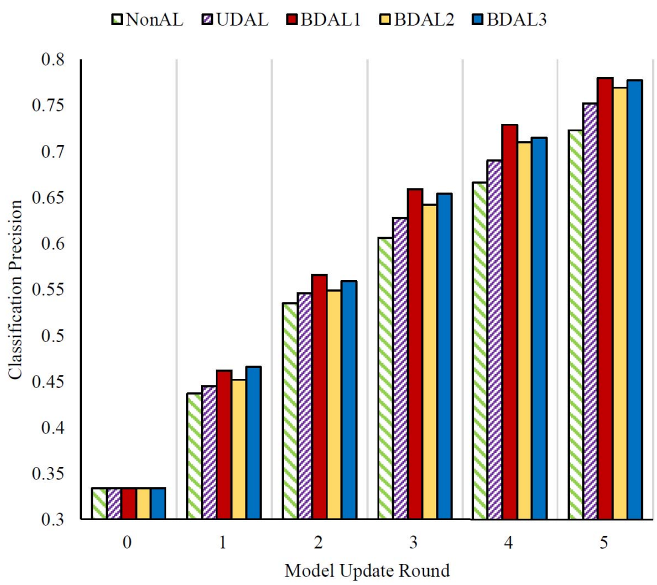


Fig. 10. Image classification results.

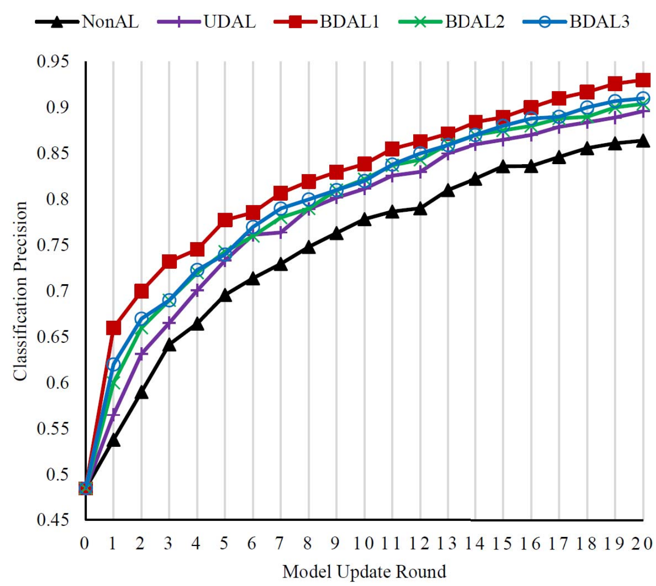


Fig. 11. Patent document classification results (with 5% mislabeling).

TABLE III

PRECISION OF MISLABELED INSTANCE DETECTION

Algorithm	# backward instances	# noisy instances	Detection precision
BDAL1	50	34.84	69.68%
BDAL2	50	33.22	66.44%
BDAL3	50	33.18	66.36%

TABLE IV

PRECISION OF MISLABELED INSTANCE DETECTION

Algorithm	# backward instances	# noisy instances	Detection precision
BDAL1	60	38.10	63.50%
BDAL2	60	35.35	58.92%
BDAL3	60	34.45	57.42%

precision (<70%) of mislabeled instance, BDAL still outperforms UDAL. This indicates that the backward instances detected by backward learning are not only mislabeled instances but also outliers that jeopardize the model's generalization performance. Via Undo or Redo process, the negative impact of backward instances can be mitigated.

D. Patent Document Classification

Patent, as a document accessible to the general public, is an important form of intellectual property containing rich structured content regarding technological innovations. The analysis of patents is a widely used method to discover inventive activity and output over different fields, regions, and time, and reveal trends in science and technology [48], [49]. As the basis for patent analysis, patent classification is indispensable for effective management of patents and in-depth exploration of valuable information.

Patent documents come from the Innography database [50]. About 5000 patents on electric automobile are collected as the data set, all of which are classified manually by domain experts into five classes, i.e., battery, battery management, motor, motor control, and vehicle control unit. In the data set, 1000 randomly selected patents constitute a separate test-set and the remaining patents are used for model learning.

Among the patents, 5484 terms are extracted as raw text feature. The weight of a term within a patent is calculated with term frequency–inverse document frequency. Principal component analysis is then used for dimension reduction, arriving at a 150-D feature vector. Likewise, regularized softmax regression is used to cope with the multiclass classification task.

In the data set, 50 patents are randomly picked as initial training data for classifier construction. For model update, 100 patents are labeled each time with the noise rate at 5%. The same algorithms are compared as listed in Table I, and similarly, N_{BL} is preset as $N_{\mathrm{BL}} = 3$ for BDAL.

Compared with MNIST and Coral data, classification of patents is highly dependent on sophisticated domain knowledge, and thus is a much more challenging task.

Results of patent document classification, averaged over 20 runs, are shown in Fig. 11. The precision of backward instance detection in BDAL is compared in Table IV.

The comparison results are similar to those in Section IV-C, i.e., BDAL > UDAL > NonAL.

Surprisingly, even when the detection precision of mislabeled instance is relatively low (<65%), BDAL still outperforms UDAL. It indicates that BDAL focuses on the improvement in generalization performance rather than data denoising. The instance that jeopardizes the model's generalization ability, no matter it is mislabeled or an outlier, will be detected and processed.

Among BDAL, different strategies perform differently. The Undo strategy achieves the best performance, while the Redo strategy is less effective. This is because in the Undo strategy,

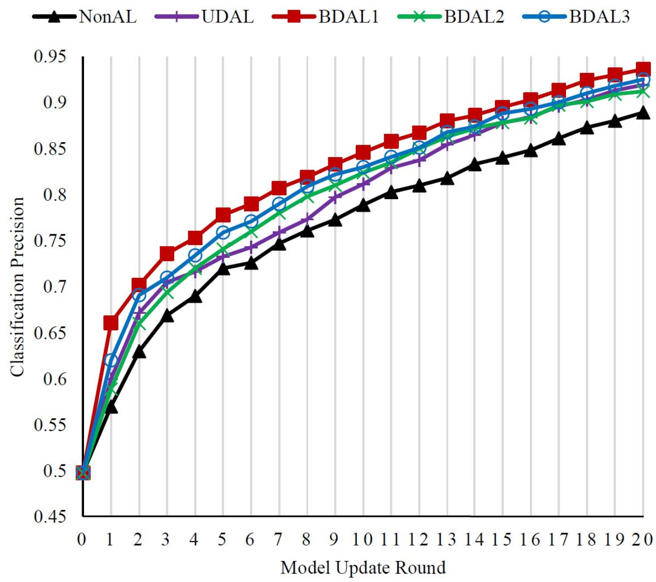


Fig. 12. Patent document classification results (without mislabeling).

all the newly incorporated instances are the most informative ones coming from forward learning. While in the Redo strategy, some of the newly labeled instances are derived from the reevaluation of backward instances. When detection precision of mislabeled instance is not high enough, a large proportion of the backward instances are outliers. Relabeling an outlier or reselecting around an outlier will most probably add a similar instance with identical label into the training data set, thus the information gain is lower than redoing a mislabeled instance. Therefore, the performance of the Redo strategy will be limited on outliers.

To verify the above viewpoint, we carry out an additional experiment without mislabeling. In this case, all the backward instances will be outliers which are correctly labeled but have negative impact on the model's generalization performance. Experimental results, averaged over 20 runs, are shown in Fig. 12.

As can be expected, similar comparison results, BDAL > UDAL > NonAL, are observed. Among BDAL, the Redo strategy receives slightly improvement compared with UDAL, while the Undo strategy achieves the best performance. It is indicated that the Undo strategy is more effective than the Redo strategy to cope with the noise derived from outliers.

To sum up, BDAL with Undo strategy can achieve the best performance especially for the sophisticated classification tasks.

V. Conclusion

In this paper, a BDAL algorithm is proposed for effective model learning with abundant unlabeled data and limited labeling effort. The contributions of this paper can be summarized as follows. First, an in-depth study is carried out on active learning which, compared with NonAL, arrives at a better balance between effectiveness and efficiency by means of human–computer interaction. Second, a two-way learning process is proposed based on a refined learning model, which successfully integrates the acquisition of new knowledge and the introspection of learned knowledge. With the exploration into both the unlabeled and labeled data set, learning model with higher generalization performance can be achieved. Moreover, clustered batch mode is adopted to cope with the problem of redundancy in batch mode while retaining the advantage of efficiency. The advantage of BDAL is demonstrated based on comprehensive comparison on classification tasks on synthetic data, handwritten digits, images, and patent documents.

In the future, we will focus on the extension of BDAL. Compared with the classification problem in this paper where each instance can only has one label, multilabel classification allows an instance to be attached with multiple labels. For multilabel classification, 2-D active learning [51], [52] is an effective algorithm for the selection of the most informative instance-label pairs. The combination of bidirectional and 2-D active learning is a challenging but promising task. Furthermore, noise stemming from the instances and the oracles should be further analyzed, so that different strategies of backward learning can be adopted adaptively according to the corresponding noise model. Last but not least, fast approximation algorithms will be studied for the promotion of computational efficiency.

Acknowledgment

The authors would like to thank the Associate Editor and the reviewers for their valuable comments and advice.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
[2] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Burlington, MA, USA: Elsevier, 2011.
[3] X. Zhang, C. Xu, J. Cheng, H. Lu, and S. Ma, "Effective annotation and search for video blogs with integration of context and content analysis," *IEEE Trans. Multimedia*, vol. 11, no. 2, pp. 272–285, Feb. 2009.
[4] R. He, W.-S. Zheng, T. Tan, and Z. Sun, "Half-quadratic-based iterative minimization for robust sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 261–275, Feb. 2014.
[5] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Two-stage nonnegative sparse representation for large-scale face recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 35–46, Jan. 2013.
[6] R. He, Z. Sun, T. Tan, and W.-S. Zheng, "Recovery of corrupted low-rank matrices via half-quadratic based nonconvex minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 2889–2896.
[7] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.
[8] G. M. James, "Variance and bias for general loss functions," *Mach. Learn.*, vol. 51, no. 2, pp. 115–135, 2003.
[9] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognit.*, vol. 44, no. 7, pp. 1540–1551, 2011.
[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer-Verlag, 2009, pp. 485–585.
[11] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
[12] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. San Rafael, CA, USA: Morgan & Claypool, 2009.
[13] R. G. F. Soares, H. Chen, and X. Yao, "Semisupervised classification with cluster regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1779–1792, Nov. 2012.
[14] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, pp. 129–145, Mar. 1996.
[15] P. Campigotto, A. Passerini, and R. Battiti, "Active learning of Pareto fronts," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 506–519, Mar. 2014.

[16] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, “Active learning with drifting streaming data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 27–39, Jan. 2014.

[17] B. Settles, “Active learning literature survey,” Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1648, 2010.

[18] E. Lughofer, “Hybrid active learning for reducing the annotation effort of operators in classification systems,” *Pattern Recognit.*, vol. 45, no. 2, pp. 884–896, 2012.

[19] P. Smyth, “Bounds on the mean classification error rate of multiple experts,” *Pattern Recognit. Lett.*, vol. 17, no. 12, pp. 1253–1257, 1996.

[20] R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng, “Cheap and fast—But is it good?: Evaluating non-expert annotations for natural language tasks,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Honolulu, HI, USA, Oct. 2008, pp. 254–263.

[21] V. S. Sheng, F. Provost, and P. G. Ipeirotis, “Get another label? Improving data quality and data mining using multiple, noisy labelers,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Las Vegas, NV, USA, Aug. 2008, pp. 614–622.

[22] P. Donmez, J. G. Carbonell, and J. Schneider, “Efficiently learning the accuracy of labeling sources for selective sampling,” in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Paris, France, Jun. 2009, pp. 259–268.

[23] D. D. Lewis and J. Catlett, “Heterogeneous uncertainty sampling for supervised learning,” in *Proc. 11th Int. Conf. Mach. Learn.*, New Brunswick, NJ, USA, Jul. 1994, pp. 148–156.

[24] D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” in *Proc. 17th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Dublin, Ireland, Jul. 1994, pp. 3–12.

[25] T. Scheffer, C. Decomain, and S. Wrobel, “Active hidden Markov models for information extraction,” in *Proc. 4th Int. Conf. Adv. Intell. Data Anal.*, Cascais, Portugal, Sep. 2001, pp. 309–318.

[26] A. Culotta and A. McCallum, “Reducing labeling effort for structured prediction tasks,” in *Proc. 20th Nat. Conf. Artif. Intell. (AAAI)*, Pittsburgh, PA, USA, Jul. 2005, pp. 746–751.

[27] B. Settles and M. Craven, “An analysis of active learning strategies for sequence labeling tasks,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Honolulu, HI, USA, Oct. 2008, pp. 1070–1079.

[28] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, Pittsburgh, PA, USA, Jul. 1992, pp. 287–294.

[29] I. Muslea, S. Minton, and C. A. Knoblock, “Selective sampling with redundant views,” in *Proc. 17th Nat. Conf. Artif. Intell. (AAAI)*, Austin, TX, USA, Jul. 2000, pp. 621–626.

[30] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” in *Proc. 17th Int. Conf. Mach. Learn.*, Stanford, CA, USA, Jun. 2000, pp. 999–1006.

[31] P. Melville and R. J. Mooney, “Diverse ensembles for active learning,” in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 2004, pp. 584–591.

[32] P. Melville, S. M. Yang, M. Saar-Tsechansky, and R. Mooney, “Active learning for probability estimation using Jensen–Shannon divergence,” in *Proc. 16th Eur. Conf. Mach. Learn.*, Porto, Portugal, Oct. 2005, pp. 268–279.

[33] N. Roy and A. McCallum, “Toward optimal active learning through sampling estimation of error reduction,” in *Proc. 18th Int. Conf. Mach. Learn.*, Williamstown, MA, USA, Jun. 2001, pp. 441–448.

[34] X. Zhu, J. Lafferty, and Z. Ghahramani, “Combining active learning and semisupervised learning using Gaussian fields and harmonic functions,” in *Proc. ICML Workshop Continuum Labeled Unlabeled Data*, Washington, DC, USA, Aug. 2003, pp. 58–65.

[35] Y. Guo and R. Greiner, “Optimistic active learning using mutual information,” in *Proc. 20th Int. Joint Conf. Artif. Intell.*, Hyderabad, India, Jan. 2007, pp. 823–829.

[36] R. Moskovitch, N. Nissim, D. Stopel, C. Feher, R. Englert, and Y. Elovici, “Improving the detection of unknown computer worms activity using active learning,” in *Proc. 30th Annu. German Conf. AI*, Osnabrück, Germany, Sep. 2007, pp. 489–493.

[37] R. E. Schapire, “The strength of weak learnability,” *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.

[38] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[39] R. E. Schapire and Y. Singer, “Improved boosting algorithms using confidence-rated predictions,” *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.

[40] P. M. Long and R. A. Servedio, “Random classification noise defeats all convex potential boosters,” *Mach. Learn.*, vol. 78, no. 3, pp. 287–304, 2010.

[41] K. Brinker, “Incorporating diversity in active learning with support vector machines,” in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, USA, Aug. 2003, pp. 59–66.

[42] S. Tong and E. Chang, “Support vector machine active learning for image retrieval,” in *Proc. 9th ACM Int. Conf. Multimedia*, Ottawa, ON, Canada, Sep. 2001, pp. 107–118.

[43] T. Zhang and F. Oles, “The value of unlabeled data for classification problems,” in *Proc. 17th Int. Conf. Mach. Learn.*, Stanford, CA, USA, Jun. 2000, pp. 1191–1198.

[44] J. Cheng and K. Wang, “Active learning for image retrieval with Co-SVM,” *Pattern Recognit.*, vol. 40, no. 1, pp. 330–334, 2007.

[45] X. Zhang, J. Cheng, H. Lu, and S. Ma, “Weighted Co-SVM for image retrieval with MVB strategy,” in *Proc. IEEE Int. Conf. Image Process.*, San Antonio, TX, USA, Sep. 2007, pp. IV-517–IV-520.

[46] X. Zhang, “Interactive patent classification based on multi-classifier fusion and active learning,” *Neurocomputing*, vol. 127, pp. 200–205, Mar. 2014.

[47] *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist, accessed Jan. 31, 2014.

[48] S. Rozhkov and L. Ivantcheva, “Scientometrical indicators of national science & technology policy based on patent statistics data,” *World Patent Inf.*, vol. 20, nos. 3–4, pp. 161–166, 1998.

[49] C.-H. Wu, Y. Ken, and T. Huang, “Patent classification system using a new hybrid genetic algorithm support vector machine,” *Appl. Soft Comput.*, vol. 10, no. 4, pp. 1164–1177, 2010.

[50] *The Innography Database*. [Online]. Available: http://www.innography.com, accessed Oct. 31, 2010.

[51] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang, “Two-dimensional multilabel active learning with an efficient online adaptation model for image classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1880–1897, Oct. 2009.

[52] X. Zhang, J. Cheng, C. Xu, H. Lu, and S. Ma, “Multi-view multi-label active learning for image classification,” in *Proc. IEEE Int. Conf. Multimedia Expo*, Cancún, Mexico, Jun./Jul. 2009, pp. 258–261.

Xiao-Yu Zhang (M’09) received the B.S. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 2005, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently an Assistant Professor with the Institute of Information Engineering, Chinese Academy of Sciences. His current research interests include machine learning, data mining, and big data analysis.

Dr. Zhang was a recipient of the Silver Prize in 2009 Microsoft Cup IEEE China Student Paper Contest, and the 2009 Outstanding Student Scholarship of Baosteel Education Fund.

Shupeng Wang received the M.S. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 2004 and 2007, respectively.

He is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His current research interests include big data management and analytics, and network storage.

Xiaochun Yun (M’12) received the M.S. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1993 and 1998, respectively.

He is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. He has authored over 200 papers in refereed journals and conference proceedings. His current research interests include network and information security, and big data analysis.