

Decisive Supervised Learning

Eileen A. Ni, Da Kuang, and Charles X. Ling

Department of Computer Science
The University of Western Ontario
London, Ontario, Canada
{ani,dkuang,cling}@csd.uwo.ca

Abstract. Traditional active learning selects the most informative (e.g., the most uncertain) example and queries an oracle for the label. However, as more examples are learned in the process, even the most uncertain examples can become certain. In this case, would it be better to make predictions directly and take the consequence if the prediction is wrong, rather than asking the oracle for labels? In this paper, we propose a new learning paradigm. In contrast to the traditional active learning, the learner can obtain true labels not only by querying oracles but also by making predictions and taking the consequence. Under this paradigm, we further propose a novel algorithm named *Decisive Learner* which always chooses the most decisive action (either querying oracles or making predictions) in the learning process. Compared to other typical learners (indecisive learners, traditional active learners, conservative learners), we show empirically that our decisive learner makes fewer mistakes and incurs the smallest total costs in the learning process.

Keywords: active learning; decisive learner; decisive actions;

1 Introduction

In traditional active learning, the learner chooses the most informative example in the unlabeled pool to query the oracle for its label. By doing so, the learner is likely to achieve high accuracy by using few labeled examples. Many variants of active learning [10,13] have been proposed (see reviews in Section 2), among which uncertain sampling [12] is one of the most intuitive and commonly used strategies. In uncertain sampling, the learner selects the most uncertain example which is closest to the decision boundary. By halving the version space, uncertain sampling can make the learning process converge quickly.

Most previous works on active learning assume the true labels can only be obtained by querying oracles. However, in many real world applications, labels can be acquired (revealed) alternatively by making predictions directly and taking the consequence. For example, when a learning system sorts letters by using OCR (optical character recognition) devices of the post office, if the hand-written codes are ambiguous, or too difficult to recognize, they will be passed to the oracles (human) for labels. However, if the OCR can predict accurately the hand-written zip codes, the letter will be sorted and mailed to the recipient directly.

If the prediction is not correct, the letter will be returned and redelivered (cost or consequence of the wrong prediction).

In the above example, the acquired true labels can be given to the learner (OCR system) to further train the predictive model. We can see, besides querying oracles, making predictions provides another way to obtain the true labels in the learning process. In fact, this type of problems is ubiquitous in the real world. In order to effectively deal with the problems, we propose a new learning paradigm, where we consider two actions for acquiring true labels in the learning process. One action is to query oracles (e.g., human or experts). Certain cost has to be paid for each query. The other action is to make predictions (e.g., directly mailing the letter). After the prediction, the true label will be revealed. If the prediction is correct, no cost is paid; otherwise, misclassification cost has to be paid. To determine which action to choose for an example, we should compare the query cost and the misclassification cost. Since the correct action is unknown before each action is taken, we use *expected misclassification cost*.

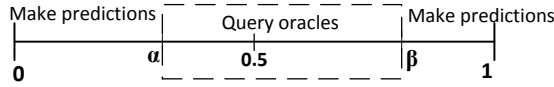


Fig. 1. Probability interval to query oracles. For the examples with the probability in $[\alpha, \beta]$, the learner will query oracles; otherwise, the learner will make predictions.

Based on the querying cost and the expected misclassification cost, we can easily derive a probability interval as $[\alpha, \beta]$ (see more details in Section 3.2) shown in Figure 1. For the examples with the probability falling in the interval, the learner will choose to learn them by querying an oracle for its label; otherwise, the learner will choose to make predictions on them directly. The interval is, in fact, the same as the rejection interval in the classifiers with reject option [6,2], where the learner can reject to make predictions on an example when it is not certain about it. However, unlike our paradigm, it does not make predictions during the learning process.

During the learning process of our new paradigm, if the posterior probability produced by the learner is accurate, then the action taken on each example will be always correct and optimal. Therefore, the action order in the learning process will not matter. When all unlabeled examples are learned, the learner can always learn the same predictive model. However, in reality, the posterior probability may not be very accurate particularly in the beginning of the learning. For those examples close to α or β , it is likely that wrong actions will be taken, which may consequently lead to a higher cost. We call those examples *indecisive examples*, and the actions taken on them *indecisive actions*. We call the examples far away from α and β *decisive examples*, and the actions taken on them *decisive actions*.

What is the optimal order of actions in the case where the probability is not perfectly accurate? To tackle the problem under the new paradigm, in this paper, we propose a novel learning algorithm, called *Decisive Learner (DL)*.

DL always takes the most decisive actions and attempts to make as few mistakes on the actions as possible in the learning process. As more examples are learned and the learner becomes more reliable, the indecisive examples may become decisive, thus fewer wrong actions will be taken. We also empirically study the performance of DL on 10 real-world datasets, by comparing it with other typical learners under different cost settings. The experimental results demonstrate that DL has an overall best performance in terms of the total cost.

The main contributions of this work are in two folds.

1. We propose a new learning paradigm, where the learner can take two actions (querying oracles and making predictions). Different from traditional active learning, in addition to querying oracles, the learner can acquire the true labels of the examples by making predictions and taking the consequence.
2. Under the new paradigm, we propose a novel learning algorithm to find the optimal action sequence. As a result, the total cost (querying cost and misclassification cost) is minimized in the learning process.

The rest of the paper is organized as follows. In Section 2, we will review some related work and discuss the difference with our work. In Section 3, we will introduce some preliminary for our problem setting and the new concept of decisive action. Section 4 will talk about the proposed algorithm to select actions. In Section 5, we will experimentally compare our algorithm with other typical learning algorithms. Discussion and future work will be presented in the last section.

2 Related Work

Our paradigm is bridging between the traditional active learning and classification with rejection. It is also similar to two-oracle setting in active learning. In this section, we will discuss the similarities and differences with them.

In traditional active learning, learner tends to choose the most informative example in the unlabeled pool to query oracle for its label. Uncertainty sampling [10] is one of the most intuitive and commonly used active learning strategies. It selects the most uncertain example which is closest to the decision boundary, which is one option in our learner DL.

As we mentioned in Section 1, traditional active learning has only one option to obtain the true labels, which is to query oracles. Most of the previous works assume that there exists at least one oracle who can provide the labels of the examples. In [15], the assumption is that we have one perfect oracle who can correctly give all labels. [14,5] study the setting where the oracle is noisy and may mislabel the examples. [4,14] explore the case where multiple oracles or labelers may contribute to the quality of the labels. We can see in those works querying oracles (human experts or labelers) has been regarded as the only approach to retrieve the true labels.

In our paradigm, we can have two options (actions) to acquire the true labels. Besides querying oracles, the learner can make predictions directly and the true

label will be revealed from the feedback (success or failure). Thus, we need to consider not only the cost to query oracles but also the cost of the wrong predictions. The best learning strategy might not be always selecting the most informative (or uncertain) examples as in the traditional active learning. Our goal is to explore the best learning sequence that minimizes the total cost under this new paradigm.

For the classification with rejection (also known as abstaining classifiers), the learner can reject to make predictions on the uncertain examples. The decision when to reject to make predictions also relies on the ratio between the misclassification cost and the reject cost. [7] studies how to effectively reduce the misclassification rate without considering the cost ratio. However, the existing works on classification with rejection only study how to minimize the cost given a predictive model, while in our paradigm we care more about the learning process that builds the predictive model with the minimal total cost.

In our paradigm, since making predictions can reveal the true labels, it can be regarded as another oracle. However, our paradigm is substantially different from the two-oracle setting [4] in active learning. In our paradigm, the “oracle” (the model for making predictions) is updated with each new labeled example, while the two oracles in the two-oracle setting are static.

3 New Learning Paradigm

In this section, we will first formally define our paradigm setting where the goal is to minimize the total cost during the learning process, and then introduce a new concept named *decisive action* in detail.

3.1 Paradigm Setting

Given a set of labeled data L , a set of unlabeled data U and a learner M learned from L , M is allowed to select examples from U , retrieve the labels from an oracle O and update its model iteratively. Given an example in U , its label can be obtained by taking one of the two actions. The first action is to query the oracle O for its label by paying the querying cost C_q . The second action is to make a prediction (positive or negative) on the example. The consequence of the prediction will reveal the true label. If the prediction is wrong, we have to pay the misclassification cost C_{FN} for false negative and C_{FP} for false positive predictions. For each example with a posterior probability produced by the learner, the action to take can be determined. The goal of this learning problem is to find a proper learning sequence for the examples from U , such that the total cost is minimized.

3.2 Choice of Actions

For an example x in U , how to choose the action depends on the costs of the two possible actions. Given the probability of being positive $p(1|x)$ predicted by the

learner, the expected misclassification cost will be $P(1|x) \times C_{FN}$ if $0 \leq P(1|x) \leq 0.5$, or $(1 - P(1|x)) \times C_{FP}$ if $0.5 < P(1|x) \leq 1$, where C_{FN} and C_{FP} are the costs for false negative and false positive. If the expected misclassification cost is less than the cost of querying the oracle, we should make the prediction directly; otherwise, we should query the oracle for the label. Therefore, we can calculate the values of α and β in Figure 1 of Section 1: $\alpha = C_q/C_{FN}$ and $\beta = 1 - C_q/C_{FP}$. Without loss of generality, we transform Figure 1 into Figure 2 and look close into the region of $[0.5, 1.0]$. Instead of $P(1|x)$, the horizontal axis in Figure 2 changes to $P(d|x)$. Here, d is the prediction (0 or 1) made by the learner, which depends on the higher probability of $P(0|x)$ and $P(1|x)$. If $P(0|x) > 0.5$, then $d = 0$; otherwise, $d = 1$. In the following, we will introduce two concepts that are related to the choice of actions.

The first concept we will introduce is *action boundary*. In Figure 2, due to the similarity, we only look into the threshold $\beta = 1 - C_q/C_{FP}$ for $P(d|x)$, instead of two thresholds α and β in Figure 1. For examples with $0.5 \leq P(d|x) < \beta$, we should query the oracle; while for examples with $\beta \leq P(d|x) \leq 1$, we should make predictions. We call the threshold β *action boundary*. The position of β is not necessarily in the center of the axis, instead it relies on the cost of querying the oracle and the misclassification cost. If the oracle is too expensive to query, the value $1 - C_q/C_{FP}$ will be small, and thus β will be closer to 0.5. If the wrong prediction is costly, β will be closer to 1.

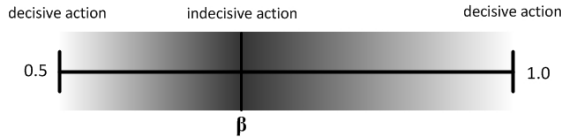


Fig. 2. Illustration for action boundary as well as decisive and indecisive action. The horizontal axis represents $P(d|x)$.

The correct choice of the actions depends on the accuracy of the posterior probability $P(d|x)$. Here, the accurate probability means that the probability is perfectly calibrated [16]. Different classifiers have different calibration behaviors. For example, the scores produced by *naive bayes*, *decision tree* and *SVM* are not calibrated, while *bagged decision tree* and *random forest* can produce calibrated probabilities [11,16]. In addition, the insufficiency of training data may also lead to the inaccuracy of the probability, particularly in the beginning of the learning process when the learner does not observe enough examples. If a classifier can produce perfectly calibrated probability $P(d|x)$, then the action taken on each example in U will always be the correct choice.

The second concept, in terms of the choice of actions, is *Indecisive* and *Decisive Actions*. The inaccuracy of the posterior probabilities $P(1|x)$ will easily lead to the wrong choice of the actions, particularly in the boundary area close to β as shown in Figure 2. For the boundary examples, the learner is not sure which

action to take. Therefore, we call those boundary examples *indecisive examples*, and the actions taken on them *indecisive actions*. For the examples far away from the action boundary β (approaching 0.5 or 1), the learner is more certain about which action to take and the actions taken on them will be less likely to be mistaken. Hence, we call those examples *decisive examples*, and their actions *decisive actions*¹. In fact, there is no clear threshold to distinguish decisive and indecisive actions. In Figure 2, we use the darkness to demonstrate the decisiveness of the actions. The darker the color of the area, the less decisive the actions taken in the area. We can see that the decisiveness of the actions gradually decreases from β to the two ends (0.5 or 1). In Figure 2, the decisiveness of actions looks similar to the uncertainty of examples, but they are different. The probabilities of uncertain examples are close to 0.5, while the probabilities of decisive actions varies within $[0.5, 1]$ depending on C_q and C_{FP} .

Based on the decisiveness of action, in the next section, we will propose a new learning strategy to minimize the total cost.

4 Decisive Learner

The key issue to the learning problem defined in Section 3 is how to correctly determine the learning sequence on the examples in U such that we can minimize the total cost. In the following, we will propose a novel learning strategy that selects examples from the most decisive to the most indecisive.

4.1 Algorithm

We design a novel learner called *Decisive Learner (DL)*. The basic idea of DL is that the decisive examples take precedence to be selected for learning since the actions taken on them are more likely to be correct, and the indecisive examples will be left for learning later. As we mentioned in Section 1, when more examples are observed by the learner, the model built will become more accurate, the indecisive examples may become decisive, and consequently actions will be less likely to be mistaken. There are two types of decisive examples: examples (close to 0.5) to query the oracle and examples (close to 1) to make predictions. Both of the two types examples are beneficial for the learner. The examples with probabilities close to 0.5 can help the learner achieve high accuracy with few examples. Direct prediction on the examples with probabilities close to 1 makes good use of the current learner and is likely to obtain the true labels without any cost. Moreover, those (certain) examples can make the learner more robust. Therefore, in our algorithm, we take advantage of both types of examples by alternating them.

Specifically, the algorithms of decisive learner can be decomposed into the following four steps.

¹ In this paper, for each selected example an action will be taken, thus we regard taking actions and selecting examples equivalently.

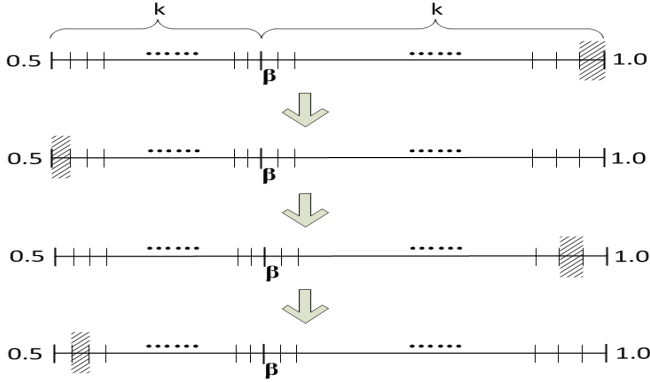


Fig. 3. Illustration of the learning process for the decisive learner (DL). DL learns the examples in the intervals (shaded) alternately on the two sides of the action boundary β , gradually approaching β . The actions are taken from the most decisive to the most indecisive.

1. **Splitting Probability Interval:** Each of the probability ranges $[0.5, \beta)$ and $[\beta, 1.0]$ is split into k equal intervals. Each example in U falls in an interval according to the posterior probability $P(d|x)$ predicted by the current learner.
2. **Selecting the Starting Interval:** The learner chooses the most decisive interval which is furthest from the action boundary β to start the learning.
3. **Learning in an Interval:** The current learner checks if there is any example in U located in the current interval. If yes, we select the most decisive example in the current interval, acquire its label by taking the corresponding action and update the learner, and then repeat step 3; otherwise, we go to step 4.
4. **Alternating Interval:** If all examples in U are learned, we terminate the learning; otherwise, the learner selects an interval on the other side of β as the next interval, and then go back to step 3 to learn examples in it.

From the algorithm of DL, it is clear that the learner always learns the most decisive examples and attempts to make as few mistakes on the actions as possible. This feature ensures that DL achieves a good performance in terms of the total cost.

5 Empirical Studies

In this section, we will empirically study the performance of DL in terms of the total cost. We will compare it with other three typical learners.

5.1 Datasets and Cost Ratios

We will evaluate the performance of DL on 10 UCI datasets [1], including abalone, adult-census, anneal, credit-g, diabetes, nursery, sick spambase, splice and waveform, with the size ranging from 898 to 32561.

To be more comprehensive, our evaluation will be conducted under different cost ratios (the misclassification cost C_{FN} and C_{FP} over the oracle querying cost C_q). As different values of C_{FN} and C_{FP} will not affect the comparison results, we let $C_{FN} = C_{FP} = C_m$, and choose three cost ratios, $C_m/C_q = 2.5$, $C_m/C_q = 4$ and $C_m/C_q = 10$. Since $\beta = 1 - C_q/C_m$, the corresponding action boundary β are 0.6, 0.75 and 0.9. The reason we choose the minimum cost ratio as 2.5 is that we should make sure $C_m/C_q \geq 2$; otherwise, for any $P(d|x)$, $C_m \times (1 - P(d|x)) < 2 \times C_q \times 0.5 < C_q$, meaning that making predictions is always better than querying oracles regardless of $P(d|x)$ and thus we do not need to conduct our research.

5.2 Other Learners

In our experiments, DL will be compared with other three typical learners with different learning sequence under our learning paradigm. We will give a brief introduction of them in this subsection.

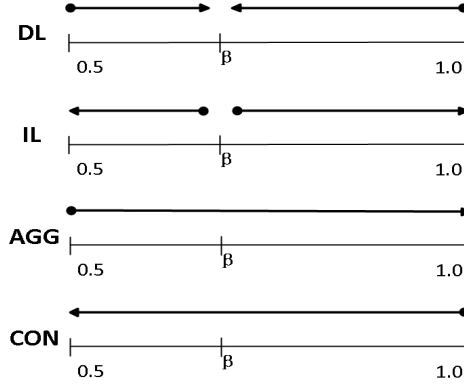


Fig. 4. Illustration of the learning process for four learners. DL is the decisive learner. IL is the indecisive learner, opposite of DL. AGG is the aggressive learner which is the traditional uncertain sampling. CON is the conservative learner which learns in the same sequence as self-directed learning.

1. **Indecisive Learner:** The first learner is named indecisive learner (IL), which takes the opposite learning sequence of DL (see Figure 4 for illustration). It always selects the most indecisive examples in the learning process. Specifically, it has the same interval splitting strategy as DL and also alternates the intervals on both sides of β . The only difference is that it starts from the closest interval to β where examples are indecisive. It can be expected that the learner will make many mistakes on actions, especially at the beginning of the learning process.
2. **Aggressive Learner:** Aggressive learners (AGG) are those that choose the most challenging example (i.e., the example that is least certain by the current learner), to learn in each iteration. It is the same as *uncertainty sampling*

[15] in the traditional active learning. It always gives preference to the example that is closest to the decision boundary and queries the oracle for its label.

3. **Conservative Learner:** In contrast to the aggressive learner, conservative learner (*CON*) always exploits the data it can predict well and tries to make as few mistakes on the predictions as possible. Thus, it prefers to learn the examples with the posterior probability $P(d|x)$ close to 1. The learning sequence of conservative learner is the same as self-directed learning [9].

Furthermore, for all the four learners (DL, IL, AGG and CON), we use *bagged decision tree* as the base classifier, due to its well-calibrated posterior probability as we mentioned in Section 3.2.

5.3 Experimental Setting

For each of the 10 UCI datasets, we randomly select 100 examples as the labeled set, and use it to train the initial classifier for each of the four learners. The rest of the examples belong to the unlabeled set. For the four learners mentioned in Section 5.2, we calculate the total cost (the misclassification cost and the cost of querying the oracle) spent in the entire learning process. The less the cost, the better the learner. We run the four learners on the 10 datasets under the three cost ratios (Section 5.1) for 10 times. Friedman test and Wilcoxon signed-rank test will be chosen to statistically test the difference of the total cost of the four learners. It should be noted that after learning all the unlabeled examples, the four learners should have the same predictive model, since the model is built on the same set of examples.

5.4 Statistic Testing Methods

The total cost in each repeat can be affected by the initial split of the dataset, thus the cost may have large variance in different repeats and even the data itself may not be normally distributed. In this case, Friedman test can be a reasonable choice for our statistic testing, since it uses the ranks of the data rather than their raw values to calculate the statistic. Friedman test has been widely used to test whether there is a statistically significant difference between a group of values [3,8]. If significant difference exists in the group, we still need a post-hoc test on different pairs of groups to report their statistical difference. Wilcoxon signed-rank is a commonly used post-hoc test following Friedman test [3,8], thus we will use it in our experiment.

5.5 Comparative Results

Table 1 demonstrates the average total costs of the four learners on the 10 UCI datasets, under three cost ratios. In order to evaluate the statistical difference, we also calculate the ranking of the four learners based on Wilcoxon signed-rank test. The ranking is calculated by the following steps. For each row in the table,

we first sort the four learners by their average costs ascending. Then we compare the learner with the smallest mean to the one with the largest mean. If there is no significant difference, all the learners will be ranked as 1; otherwise, we continue to compare the smallest mean with the second largest mean, until all the learners have been compared or no significant difference is found. In the next round, we will compare the second smallest mean with the largest mean, and repeat the same step. The process iterates until all learners are ranked.

In Table 1, the rank of each learner is presented in the bracket next to the average total cost. The four rows in the bottom of Table 1 present the average rank of the four learners over all the 10 datasets under the cost ratio (2.5, 4 and 10) respectively, as well as the overall average rank over 30 rows in the table. We can see clearly that DL is top ranked in 27 out of the total 30 comparisons and has the lowest average rank 1.1 over the 30 rows. The average costs on all the 10 datasets are also presented in Table 1, and we can see that DL is much better than the other three learners. Both the rank and the average costs illustrate that DL has the overall best performance in terms of the total cost.

Theoretically, IL is supposed to have the poorest performance since it always selects the most indecisive actions and is expected to make many mistakes. However, we observe that it is not exactly the case in Table 1. Overall, the average rank (2.4) of IL is slightly better than that (2.5) of AGG. A closer look reveals that under the cost ratio 4 and 10, IL indeed has the lowest rank among the four learners. The slight superiority of IL to AGG is due to the fact that IL performs much better than AGG when the cost ratio is 2.5. Under the cost ratio 2.5, β ($1-1/2.5=0.6$) is relatively close to 0.5. IL starts learning from the examples with probability around 0.6 while AGG from the examples with probability close to 0.5. Although the examples selected by IL are not as informative as those selected by AGG, they are still useful for the learner. Furthermore, IL can even save more costs by directly making predictions on the examples to the right side of β , as the expected cost of making predictions on those examples is lower than the cost to query the oracle ($(1 - P(d|x)) \times C_m < C_q$, where $P(d|x) > \beta$).

From the average rank in Table 1, we can also observe that DL is more likely to have better performance when the cost ratio is high, as its average rank increases when cost ratio becomes higher. It means when the misclassification cost is much higher than the querying cost, it is safer and more desirable to use DL as the learner.

6 Conclusion

This paper proposed a new learning paradigm where the learner is able to take two possible actions (querying oracles and making predictions) to acquire true labels for new examples. The new paradigm is different from the traditional active learning where oracles are the only source to obtain true labels. Under the new learning paradigm, we proposed a novel learning algorithm named decisive learner (DL), which always selects the most decisive examples and makes as few mistakes on the actions as possible in the learning process. In the experiments,

Table 1. Statistical comparisons between the four learners in terms of the cost. Each cell shows the average cost and its rank (in the bracket) of a specific learner on a dataset under a cost ratio. The rank is calculated by a statistical test named Wilcoxon signed-rank. The cells in bold represent the winner(s) in the corresponding row.

| | Cost ratio | DL | IL | AGG | CON |
|--------------|------------|----------------|---------------|----------------|---------------|
| abalone | 2.5 | 889(1) | 955(2) | 1031(3) | 891(1) |
| | 4 | 1081(1) | 1313(3) | 1238(2) | 1341(3) |
| | 10 | 1461(1) | 1520(2) | 1410(1) | 1723(3) |
| adult-census | 2.5 | 4803(1) | 4904(2) | 5511(3) | 4813(1) |
| | 4 | 5893(1) | 7178(3) | 6887(2) | 7102(3) |
| | 10 | 8097(1) | 9228(3) | 9121(2) | 9199(2) |
| anneal | 2.5 | 66(1) | 75(2) | 88(2) | 58(1) |
| | 4 | 102(1) | 124(2) | 124(2) | 101(1) |
| | 10 | 130(1) | 210(3) | 194(3) | 167(2) |
| credit-g | 2.5 | 268(1) | 282(2) | 308(3) | 271(1) |
| | 4 | 331(1) | 397(3) | 350(2) | 396(3) |
| | 10 | 375(1) | 400(2) | 380(2) | 445(3) |
| diabetes | 2.5 | 195(1) | 204(2) | 211(2) | 194(1) |
| | 4 | 234(1) | 262(2) | 241(1) | 274(2) |
| | 10 | 290(1) | 294(1) | 285(1) | 309(1) |
| nursery | 2.5 | 273(1) | 276(1) | 385(3) | 285(2) |
| | 4 | 349(1) | 436(3) | 497(4) | 391(2) |
| | 10 | 610(1) | 879(3) | 802(2) | 764(2) |
| sick | 2.5 | 84(1) | 93(1) | 93(1) | 83(1) |
| | 4 | 122(1) | 143(2) | 132(1) | 137(2) |
| | 10 | 217(1) | 258(2) | 254(2) | 270(2) |
| spambase | 2.5 | 416(3) | 364(1) | 628(4) | 389(2) |
| | 4 | 554(1) | 704(3) | 891(4) | 623(2) |
| | 10 | 753(1) | 1436(3) | 1415(2) | 1335(2) |
| splice | 2.5 | 391(2) | 322(1) | 625(3) | 333(1) |
| | 4 | 557(1) | 673(2) | 802(3) | 535(1) |
| | 10 | 688(1) | 1094(2) | 1077(2) | 1055(3) |
| waveform | 2.5 | 640(1) | 624(1) | 940(2) | 614(1) |
| | 4 | 847(1) | 959(2) | 1101(3) | 971(2) |
| | 10 | 1039(1) | 1503(3) | 1341(2) | 1952(4) |
| average cost | overall | 1058(1) | 1237(2) | 1278(3) | 1234(2) |
| average rank | 2.5 | 1.3 | 1.5 | 2.6 | 1.2 |
| | 4 | 1 | 2.5 | 2.4 | 2.1 |
| | 10 | 1 | 2.4 | 1.9 | 2.4 |
| | overall | 1.1 | 2.1 | 2.3 | 1.9 |

we demonstrated the outstanding performance of DL in reducing the total cost, compared to three other typical learning strategies under the same learning paradigm. The proposed learning algorithm (decisive learner) can be applied to various real-world applications, such as optical character recognition and online advertising.

References

1. Newman, D., Asuncion, A.: UCI machine learning repository (2007)
2. Bartlett, P., Wegkamp, M.: Classification with a reject option using a hinge loss. *The Journal of Machine Learning Research* 9, 1823–1840 (2008)
3. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, 1–30 (2006)
4. Donmez, P., Carbonell, J.G.: Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008*, pp. 619–628. ACM, New York (2008)
5. Du, J., Ling, C.X.: Active learning with human-like noisy oracle. In: *Proceedings of the 2010 IEEE International Conference on Data Mining, ICDM 2010*, pp. 797–802. IEEE Computer Society, Washington, DC (2010)
6. Du, J., Ni, E., Ling, C.: Adapting cost-sensitive learning for reject option. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pp. 1865–1868. ACM (2010)
7. Fumera, G., Roli, F., Giacinto, G.: Reject option with multiple thresholds. *Pattern Recognition* 33(12), 2099–2101 (2000)
8. García, S., Herrera, F.: An Extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for all Pairwise Comparisons. *Journal of Machine Learning Research* 9, 2677–2694 (2008)
9. Goldman, S.A., Sloan, R.H.: The power of self-directed learning. *Mach. Learn.* 14(3), 271–294 (1994)
10. Lewis, D., Gale, W.: A sequential algorithm for training text classifiers. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3–12. Springer-Verlag New York, Inc. (1994)
11. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: *Proceedings of the 22nd International Conference on Machine Learning, ICML 2005*, pp. 625–632. ACM, New York (2005)
12. Settles, B.: Active learning literature survey. *Sciences New York* 15(2) (2010)
13. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1070–1079. Association for Computational Linguistics (2008)
14. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get another label? improving data quality and data mining using multiple, noisy labelers. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008*, pp. 614–622. ACM, New York (2008)
15. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* 2, 45–66 (2002)
16. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 694–699. ACM (2002)