Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

ART: An Availability-Aware Active Learning Framework for Data Streams

Benjamin Shickel and **Parisa Rashidi**
University of Florida
Gainesville, FL 32611
shickelb@ufl.edu, parisa.rashidi@ufl.edu

Abstract

Active learning, a technique in which a learner self-selects the most important unlabeled examples to be labeled by a human expert, is a useful approach when labeled training data is either scarce or expensive to obtain. While active learning has been well-documented in the offline pool-based setting, less attention has been paid to applying active learning in an online streaming setting. In this paper, we introduce a novel generic framework called ART (Availability-aware active leaRning in data sTreams). We examine the multiple-oracle active learning environment and present a novel method for querying multiple imperfect oracles based on dynamic availability schedules. We introduce a flexible availability-based definition of labeling budget for data streams, and present a mechanism to automatically adapt to implicit changes in oracle availability based on past oracle behavior. Compared to the baseline approaches, our results indicate improvements in accuracy and query utility using our availability-based multiple oracle framework.

Introduction

In supervised learning settings, a machine learning model is trained using a set of pre-labeled data instances which serve as the ground truth. In practice, these ground truth labels are often nonexistent and expensive to obtain in large quantities. Active learning is a technique for dealing with this issue (Settles 2010), whereby only the most informative instances are labeled by an expert, who is often referred to as an *oracle*. In this context, the most informative instance is typically the one that, once labeled, provides the greatest increase in generalized model accuracy. By adopting this approach, a model can be trained on less labeled data while maintaining similar performance.

Active learning methods are typically characterized by how unlabeled data is processed. In a *pool-based* setting, all unlabeled instances are immediately available as a single offline batch, whereas in a *stream-based* setting, data points arrive sequentially in an online manner. In both settings, individual instances are selected via a sampling strategy and sent to one or more oracles to provide the labels. In this paper, we focus on the stream-based setting.

Traditional active learning settings typically involve a single oracle, but multiple-oracle settings are becoming more common (e.g. crowdsourcing). Nonetheless, existing work on multiple oracle settings tends to focus only on oracle expertise as the determining factor for oracle sampling. However, for real-time streaming data, the schedule and availability of each oracle play an arguably larger role. In this paper, we present a multiple-oracle active learning framework for data streams based on oracle availability, which to our knowledge has yet to be explored.

We refer to our novel active learning framework as ART (Availability-aware active leaRning in data sTreams). In summary, our main contributions are:

- A novel multiple-oracle active learning framework using a probabilistic interpretation of oracle availability, that automatically adapts to implicit changes in oracle querying behavior and schedule to provide maximum query utility while minimizing query cost over the life of the system.
- Reformulating the existing definition of a static query budget to a dynamic, parameter-less quantity that controls query decisions in a more flexible and cost-effective manner based on expected oracle behavior.

Related Work

In traditional active learning environments, oracles are assumed to be *perfect*, meaning they are assumed to (1) always answer a query when asked, and (2) always provide the correct label when they answer. One of the first works to relax these idealistic assumptions was the proactive learning paradigm (Donmez and Carbonell 2008), in which multiple oracles are assumed to possess varying levels of expertise, correctness, and cost. Other work on imperfect oracles include active learning in crowds (Fung 2011), annotator knowledge-based approaches (Yan and Rosales 2012), characterization of unlabeled instances by knowledge category (Fang and Zhu 2014), repeated noisy oracle labeling (Ipeirotis et al. 2014), probabilistic committee-based approaches (Wu et al. 2013), cost-sensitive novice-expert query allocation (Wallace and Small 2011), and importance-weighted ranking of noisy labels (Zhao 2012). While related to our study in terms of modeling multiple fallible oracles, most current research has focused on oracle expertise and modeling query costs based on the content of the query instance.

In contrast, our ART framework addresses the issue of assigning online queries based on oracles' schedules and past availability trends, which to our knowledge has not been previously explored.

Similar to these studies, we also assume imperfect oracles, but we provide a more structured formalization to our assumptions. Namely, we assume that the probability of receiving the correct label from an oracle in a streaming environment is based on three factors: (1) the oracle's *reliability*, (2) the oracle's *availability*, and (3) aspects of the potentially time-variant distribution of incoming data stream instances (also known as concept drift). Most related work has focused on reliability (Donmez and Carbonell 2008; Fung 2011; Yan and Rosales 2012; Wu et al. 2013), which derives models for estimating oracle labeling accuracy based on the specific query posed, i.e., varying levels of oracle expertise. Additionally, other studies have explored effects of concept drift in stream-based active learning. We close the understanding gap by focusing on what has been largely omitted from recent studies: oracle availability. In a time-sensitive setting, where stream instances arrive in real-time and queries are made on demand, accounting for oracle availability is of utmost importance. Thus, we focus purely on this aspect of realistic active learning.

While most active learning studies focus on the pool-based setting, there are several studies exploring the application of active learning to streaming data. Methods for adapting active learning to data streams include minimal variance classifier ensemble techniques (Zhu et al. 2007; 2010), uncertainty-based model reconstruction (Shucheng and Dong 2007), categorizing concept drift (Zhang et al. 2008), interval-based uncertainty sampling (Zliobaite et al. 2011), online optimization of Bayesian linear classifiers (Chu et al. 2011), and sliding window-based density techniques (Ienco, Pfahringer, and Zliobaite 2014). Most of these works focus on the problem of adapting online classifiers to changes in the underlying input distribution over time, i.e., exploring the notion of concept drift. To our knowledge, our ART framework is the first to consider the real-world arrival time of streaming instances and the subsequent optimization of labeling cost by intelligently selecting the most available oracle for each query.

ART Active Learning Framework

Our primary goal is to select an oracle for a given query based on knowledge of each oracle's time-sensitive availability schedule. We assume a probabilistic formulation of receiving the correct answer from a queried instance x to oracle k at time t as the following:

$$P(ans, k, t, x) = P(ans|k, x, t) * P(k|x, t) * P(x|t) \quad (1)$$

where we define $P(ans|k, x, t)$ as the *reliability* of oracle k, $P(k|x, t)$ as the *availability* of oracle k, and $P(x|t)$ as the time-variant input distribution of the data stream, i.e., its concept drift. In our current work, we assume equal expertise among oracles and focus primarily on oracle availability as a function of time. Concept drift is out of scope for this paper and will not be further discussed. However, extending our methods to incorporate concept drift is straightforward. Most previous work in this area has solely dealt with oracle reliability and concept drift, so we bridge the knowledge gap by showing the benefits of designing a schedule-sensitive active learning framework specifically focused on oracle availability.

Active Learning Query Budget

Budget approaches for stream-based active learning typically rely on a predetermined, fixed fraction of the overall data stream that is to be queried (Zliobaite et al. 2011; Ienco, Pfahringer, and Zliobaite 2014). That is, before a query is made, the following budget constraint is typically checked:

$$\frac{n_Q}{n} < B \quad (2)$$

where n_Q is the number of queries that have been made so far, n is the total number of instances seen so far in the data stream, and B is the fixed fraction of the total data stream for which labels are desired. While this works for limiting the total number of queries to a fixed amount, we see several problems with this approach. First, the optimal budget fraction B is difficult to determine prior to processing the entire data stream. Since data streams can be of potentially infinite length, and each query is associated with a cost, this can result in unlimited cost being charged over time. Similarly, this approach tends to incur queries that are spaced relatively uniformly apart from one another, potentially missing out on querying important instances that simply happen to arrive in close proximity. Furthermore, when oracle availabilities are taken into account, this approach to budget could pose queries when no oracles are available to answer them; conversely, when several oracles are available, this approach unnecessarily limits the number of queries made. Thus, we desire a more dynamic alternative that is sensitive to available oracles and adapts system queries accordingly.

Our solution is to have a separate budget for each oracle, and to frame budget as being inherently tied to oracle availability. If an oracle is highly available, we increase its budget, resulting in more queries; conversely, if an oracle is less available, we decrease its budget. Thus, if we assume an oracle's schedule can be partitioned into n_T distinct time intervals, we have n_T distinct budget values for each oracle - one per interval. Our budget constraint can be written in the following form:

$$\frac{n_Q(k, t)}{n_t} < P(k|t) \quad (3)$$

where $n_Q(k, t)$ is the number of queries oracle k has been sent in the current interval t, n_t is the total number of instances arriving in the current interval t, and $P(k|t)$ is the availability of oracle k at the current interval t. Thus, we provide an upper bound on the number of queries an oracle can answer *in a given interval*. For example, if an oracle's availability is 0.05, it will only be sent queries for a maximum of 5% of the instances arriving in the current interval. ART's approach to budget is much more flexible and dynamic than previous approaches, and adaptively changes based on how likely each oracle is to answer a given query.

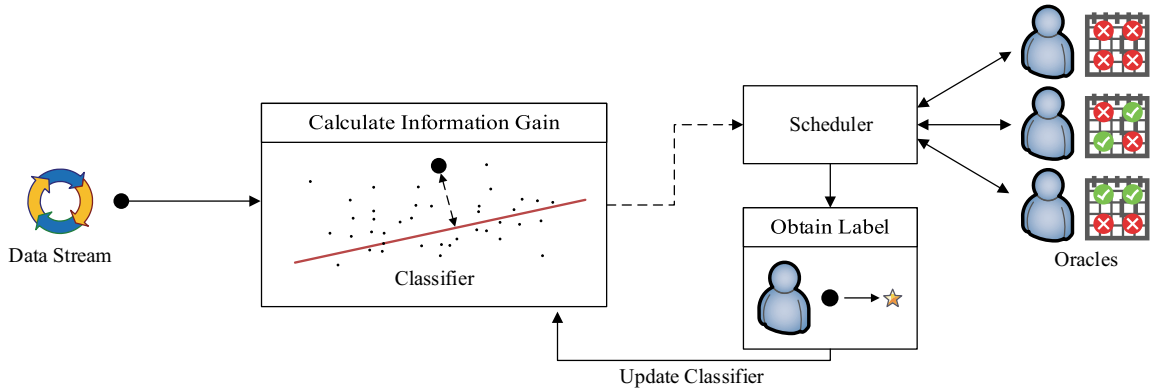

Figure 1: High-level block diagram of the ART framework. When a new data point arrives, the information gain (typically uncertainty) of the instance is compared against the current query threshold. If a query should be made, the most available oracle is chosen for the current time interval and is sent a query. The resulting label is used to incrementally update the classifier.

Problem Formulation

One of ART's primary goals is to minimize the accumulated cost of making queries while simultaneously maximizing the information gained from them. We represent the problem as a cost-benefit tradeoff, where each query is assigned some cost Ψ and expected information gain ϕ. Our primary objective function is the following:

$$\min\left(\Psi(k,t) - \phi(x)\right) \quad s.t. \quad \frac{n_Q(k,t)}{n_t} < P(k|t) \qquad (4)$$

where $\Psi(k,t)$ represents the cost of querying oracle k at time t, and $\phi(x)$ is the expected information gain from receiving instance x's label. As in Equation 3, we also constrain the maximum number of queries per oracle based on their interval availability. We adopt the following probabilistic formulation of the cost of querying oracle k at time t:

$$\Psi(k,t) = 1 - P(k|t) \qquad (5)$$

where $P(k|t)$ is the probability that oracle k will answer a query at time t, if one is asked. We define this probability as an oracle's *availability* at time t. Thus, if an oracle is highly available, the cost to query it will be low; if an oracle is mostly unavailable, the cost to query it will be large. The final objective function can be rewritten in the following form:

$$\max\left(\phi(x) + P(k|t) - 1\right) \quad s.t. \quad \frac{n_Q(k,t)}{n_t} < P(k|t) \qquad (6)$$

where the choice of information function ϕ is largely dataset dependent, and therefore will not be further discussed. In our experiments, ϕ is the entropy of an instance's predicted class label distribution.

Availability

An oracle's availability is fundamentally related to their daily schedule. Since oracles are unique individuals each with distinct commitments, working hours, and preferred query answering times, each oracle will have a distinct probability of answering a given query for some particular point in time. Our ART framework is general enough to allow for these schedules to exist in multiple forms, but in all cases, each oracle's schedule is divided into n_T time-based intervals, each containing a distinct availability $P(k|t)$. Table 1 shows a motivational example. Initial values of availability are self-reported (or perhaps imported from a calendar as the fraction of free time per interval), and will automatically adapt over time to implicit changes in oracle behavior.

Interval	1	2	3	4	5	6	7	8
Availability	0.9	0.2	0.25	0	0.55	0.7	0.3	0.45

Table 1: An example of an oracle's schedule for 8 hourly time intervals, representing a typical work day. An interval encompasses the time from the end of the previous interval until the start of the next interval. Availability is defined as the probability that an oracle will answer a query during a particular period of time.

Adapting to Schedule Changes

As the data stream progresses, oracle availabilities are likely to change; some oracles might become more available during certain time intervals, while other oracles might become less available. To account for this *schedule drift*, we automatically adapt oracle availabilities based on their querying behavior. After a query is posed to an oracle, we update their availability according to the following:

$$P(k|t) = \alpha I(x) + (1-\alpha)P(k|t) \qquad (7)$$

where $I(x)$ is an indicator function taking the value 1 if the query was answered and 0 if the query was not answered, and α is a parameter defining how quickly the system adapts to changes in oracle behavior. In a particular setting, if oracle schedules are expected to frequently change, a large α value will ensure a rapid availability adjustment. Similarly, if schedules are expected to remain relatively stationary, a smaller α will prevent overcompensation of the result of a single query.

Our final ART active learning framework is shown in Algorithm 1. A high-level block diagram is also shown in Figure 1. For each stream instance x_t, the expected information gain $\phi(x_t)$ is compared against the query threshold θ to determine if the instance should be queried. ART can function with any choice of ϕ, such as entropy or maximum margin (uncertainty sampling).

Algorithm 1: ART framework

Input: Data stream $\mathcal{X}$, set of oracles $\mathcal{K}$, initial availabilities $P(k|t)$ for each oracle/interval combination, adaptation parameter α, query threshold θ, threshold adjusting step s, information function ϕ

foreach *instance* $x_t \in \mathcal{X}$ **do**
  if *new interval* **then**
    $n = 0$
    $n_Q(k) = 0 \ \forall \ k \in \mathcal{K}$
  end
  if $\phi(x_t) < \theta$ **then**
    queryAnswered $\longleftarrow$ False
    $\mathcal{Q} \longleftarrow k \in \mathcal{K}$ s.t. $\frac{n_Q(k)}{n} < P(k|t)$
    while *!queryAnswered &&* $|\mathcal{Q}| > 0$ **do**
      $k^* \leftarrow \arg\max_{k \in \mathcal{Q}} [P(k|t)]$
      Request label y_t from oracle k^*
      $n_Q(k^*) \leftarrow n_Q(k^*) + 1$
      $n \leftarrow n + 1$
      if Label received **then**
        Update classifier with (x_t, y_t)
        $P(k^*|t) \leftarrow \alpha + (1 - \alpha)P(k^*|t)$
        queryAnswered $\leftarrow$ True
      else
        $Q \leftarrow Q - k^*$
        $P(k^*|t) \leftarrow (1 - \alpha)P(k^*|t)$
      end
    end
    $\theta \leftarrow \theta * (1 + s)$
  else
    $\theta \leftarrow \theta * (1 - s)$
  end
end

Experiment Setup

We test our ART framework with three publicly available classification datasets: *20 Newsgroups* (text classification), *Digits* (image classification), and *Letters* (image classification). All datasets are summarized in Table 2. We simulate a stream-based setting by sending single instances at random real-world arrival times for all experimental settings. We initially train our classifiers on 1% of the total dataset prior to starting the simulated data stream, and we hold out 10% of the data for cross-validating each model.

Dataset	No. Instances	No. Features	No. Classes
20 Newsgroups	11314	101631	20
Letters	20000	16	26
Digits	1797	64	10

Table 2: Summary of the datasets used in experimentation. Individual instances were processed incrementally via a simulated data stream.

For all experiments, we use an online logistic regression classifier that is incrementally trained via stochastic gradient descent. For our base measure of information gain, we take the entropy of predicted class probability distribution.

We run all simulations with 5 oracles for clear visualization of results, but the fundamental concepts of our framework apply equally well for any number of oracles. For each experimental trial, we generate a random schedule of n_T initial availabilities in the range [0,1] for each oracle. In order to provide more naturally interpretable results, in our experiments we fix the number of time intervals n_T to be 8 (i.e., a typical work day), where each time interval can be interpreted as a one-hour period. However, in practice our framework can be used with any number of repeating intervals. Based on dataset-dependent experimental feedback, we choose an optimal query threshold adjustment step of 0.01.

In order to simulate an oracle's probability of responding to a query, we store two availability tables for each oracle: the *observed* interval availabilities, and the *actual* interval availabilities. Observed availabilities are the quantities known to the system, and are based on the empirical results of previous queries. Actual availabilities are the true probabilities that an oracle will answer a query in a given interval, and is hidden to the system. At the start of each simulation, both availability tables are initialized to the same random values, however, over time the actual availabilities are expected to become out of sync as oracle schedules implicitly drift. We simulate this schedule drift by sampling a *drift parameter* $\zeta \sim N(0, \lambda)$ for each oracle at the start of each trial, where λ is a simulation parameter controlling the overall level of schedule drift. At the beginning of each cycle of intervals as the simulation progresses, we sample a value $\rho \sim N(P, \zeta)$ for each oracle, where P is the oracle's *actual* availability, and update as $P = P * \rho$. Thus, some oracles will have schedules which are changing quickly, while others will have minuscule change.

We ran experiments with 10 different drift parameters ζ in range [0.001, 10] and 13 different adaptation rates α in range [0.1, 0.9]. For each parameter configuration, we ran 60 simulations and averaged the results.

Results

In the following experiments, we compare our ART framework to the baseline approach of randomly sampled oracles, which does not account for schedule availability. We begin by demonstrating the percentage of successful queries for the *20 Newsgroups* dataset in Figure 2 (complete results are shown in Table 3). We compare the effects of schedule drift, i.e., how fast oracle schedules are implicitly chang-

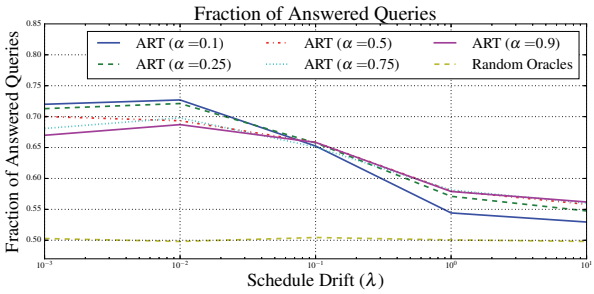

Figure 2: Percentage of successfully answered queries for the *Digits* dataset experiments. The ART framework always results in a higher fraction of answered queries. As schedules become more in flux, models with a larger α value adapt to these changes faster, resulting in the best query utility.

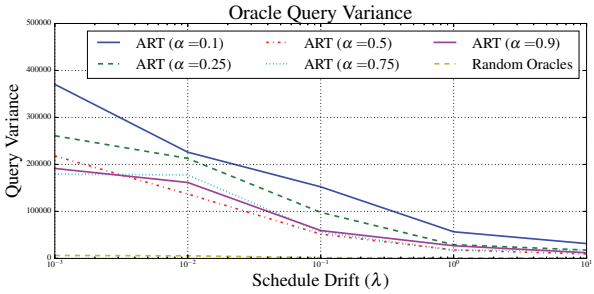

Figure 3: Query variance among oracles for the *Letters* dataset experiments. When oracles are randomly chosen, queries are more evenly spread (smaller variance). The ART framework queries the most available oracles, resulting in larger query variance (which is the desired behavior given oracle schedules.) As schedules are more in flux, the variance becomes more in line with random oracles.

ing, between both methods and experiment with various values of α to observe how well our framework adapts to implicit changes in oracle behavior. For all datasets, the ART framework yields a much larger fraction of queries answered than the baseline for all settings of alpha and schedule drift. We notice that when schedules are relatively stationary, i.e. small latent drift parameter, the smallest α values yield the best performance, and vice versa.

Because we sample oracles based on the availability criterion, the most available oracles end up receiving the most queries. In most situations, this is ideal, since we do not wish to send queries to oracles who are unlikely to provide an answer. However, depending on the application, it may be desirable to achieve a more even spread of queries. In Figure 3, we show the variance of total query counts among all oracles at the end of each simulation for varying amounts of schedule drift and adaptation rates for the *Letters* dataset. Most configurations of ART result in higher query variance than a random oracle sampling scheme, however, as schedules become more in flux, ART yields similar query variances to the random oracle baseline.

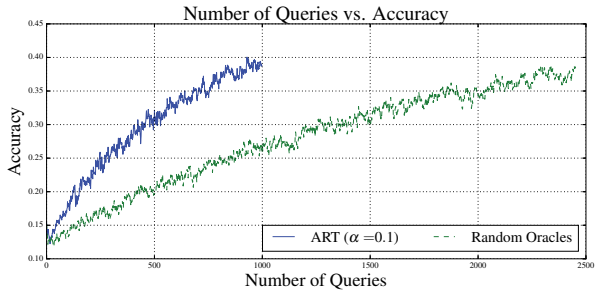

Figure 4: Classifier accuracy as a function of the number of queries posed to oracles for the 20 Newsgroups experiments under low schedule drift settings. The random oracle baseline induces more failed query attempts than ART, which do not contribute to the incremental training of the classifier.

We plot the number of queries vs. classifier accuracy for the *20 Newsgroups* dataset in Figure 4 for the minimal schedule drift setting ($\zeta = 0.001$, $\alpha = 0.1$). ART shows significant accuracy benefits over the random oracle approach. Because querying random oracles results in many more failed query attempts, the ART framework results in higher accuracy for the same number of queries.

In Table 3, we present results for the fraction of successful queries, total queries, final classifier accuracy, and oracle query variance between our ART framework and the baseline approach of randomly sampled oracles. For each model, we ran experiments with 10 different schedule drift parameters ζ in the range [0.001, 10]. We show the two extreme cases in Table 3, which includes what we define as the *low drift* setting ($\zeta = 0.001$) and the *high drift* setting ($\zeta = 10.0$). Logical drift range thresholds were chosen experimentally.

From Table 3, it is apparent that the ART framework has marked improvement over a random sampling scheme for fraction of successful queries and final accuracy, both objective measures. Additionally, the ART framework results in fewer total queries, which translates to less accumulated query cost for higher accuracy. It is interesting to note that the query variance is always lower when randomly sampling oracles, i.e., queries are more spread out amongst all oracles. While this could potentially be a positive for the baseline approach, as it prevents the same oracle from receiving multiple consecutive queries, it fails to account for oracle availability, and could request labels from unresponsive oracles.

Conclusion

In this paper, we introduced our novel ART framework for multiple-oracle online active learning that utilizes real-time schedule and availability information to minimize query cost and maximize query efficiency. ART shows significant improvement over baseline methods that do not factor in oracle availability, arguably the most important factor in time-sensitive active learning. We also demonstrated how online active learning systems can be improved by automatically adapting to implicit changes in oracle availability, resulting

Value	Drift	Model	*20 Newsgroups*	*Digits*	*Letters*
Fraction Successful Queries	Low	Random Oracles	0.495	0.503	0.498
		ART	**0.731**	**0.720**	**0.742**
	High	Random Oracles	0.501	0.503	0.499
		ART	**0.586**	**0.562**	**0.554**
Total Queries	Low	Random Oracles	9952.8	915.17	9225.63
		ART	**6722.4**	**641.9**	**6180.1**
	High	Random Oracles	9784.6	910.9	9092.9
		ART	**8331.0**	**812.9**	**8187.8**
Final Accuracy	Low	Random Oracles	0.264	0.467	0.189
		ART	**0.388**	**0.641**	**0.271**
	High	Random Oracles	0.422	0.803	0.330
		ART	**0.460**	**0.840**	**0.342**
Query Variance	Low	Random Oracles	7990.8	123.3	6298.1
		ART	342746.6	2695.3	370453.8
	High	Random Oracles	950.5	84.7	887.5
		ART	15557.7	745.8	12109.1

Table 3: Summary of results for our ART framework vs. the random oracle baseline for both low ($\zeta = 0.001$) and high ($\zeta = 10.0$) schedule drift environments.

in a dynamic and utility-aware query budget.

In future work, we plan to explore the effects of concept drift, oracle reliability, and instance difficulty, as they pertain to time-sensitive active learning and oracle availability in real-world experiments. Furthermore, we intend to explore in greater depth the influence of initial availability settings as they pertain to automatic schedule adaptation.

References

Chu, W.; Zinkevich, M.; Li, L.; Thomas, A.; and Tseng, B. 2011. Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 195–203.

Donmez, P., and Carbonell, J. G. 2008. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM Conference on Information and Knowledge Mining*, 619–628.

Fang, M., and Zhu, X. 2014. Active learning with uncertain labeling knowledge. *Pattern Recognition Letters* 43:98–108.

Fung, G. 2011. Active learning from crowds. In *Proceedings of the 28th International Conference on Machine Learning*, ICML '11, 1161–1168.

Ienco, D.; Pfahringer, B.; and Zliobaite, I. 2014. High density-focused uncertainty sampling for active learning over evolving stream data. In *Proceedings of the 3rd International Workshop on Big Data Mining, JMLR W&CP*, volume 36, 133–148.

Ipeirotis, P. G.; Provost, F.; Sheng, V. S.; and Wang, J. 2014. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery* 28(2):402–441.

Settles, B. 2010. Active learning literature survey. Technical report.

Shucheng, H., and Dong, Y. 2007. An active learning method for mining Time-Changing data streams. *Intelligent Data Analysis* 11(4):401–419.

Wallace, B., and Small, K. 2011. Who should label what? Instance allocation in multiple expert active learning. In *Proceedings of the SIAM International Conference on Data Mining*, 176–187.

Wu, W.; Liu, Y.; Guo, M.; Wang, C.; and Liu, X. 2013. A probabilistic model of active learning with multiple noisy oracles. *Neurocomputing* 118:253–262.

Yan, Y., and Rosales, R. 2012. Active learning from multiple knowledge sources. In *International Conference on Artificial Intelligence and Statistics*.

Zhang, P.; Zhang, P.; Zhu, X.; Zhu, X.; Shi, Y.; and Shi, Y. 2008. Categorizing and mining concept drifting data streams. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 812–820.

Zhao, L. 2012. Importance-weighted label prediction for active learning with noisy annotations. In *2012 21st International Conference on Pattern Recognition*, 3476–3479.

Zhu, X. Z. X.; Zhang, P. Z. P.; Lin, X. L. X.; and Shi, Y. S. Y. 2007. Active learning from data streams. In *Seventh IEEE International Conference on Data Mining*, 757–762. Ieee.

Zhu, X.; Zhang, P.; Lin, X.; and Shi, Y. 2010. Active learning from stream data using optimal weight classifier ensemble. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 40(6):1607–1621.

Zliobaite, I.; Bifet, A.; Holmes, G.; and Pfahringer, B. 2011. MOA Concept Drift Active Learning Strategies for Streaming Data. In *Second Workshop on Applications of Pattern Analysis*, 48–55.