Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)
Survey Track

A Comparative Survey: Benchmarking for Pool-based Active Learning

Xueying Zhan[1*], **Huan Liu**[2], **Qing Li**[3] and **Antoni B. Chan**[1]
[1]City University of Hong Kong
[2]Arizona State University
[3]The Hong Kong Polytechnic University
xyzhan2-c@my.cityu.edu.hk, huanliu@asu.edu, csqli@comp.polyu.edu.hk, abchan@cityu.edu.hk

Abstract

Active learning (AL) is a subfield of machine learning (ML) in which a learning algorithm aims to achieve good accuracy with fewer training samples by interactively querying the oracles to label new data points. Pool-based AL is well-motivated in many ML tasks, where unlabeled data is abundant, but their labels are hard or costly to obtain. Although many pool-based AL methods have been developed, some important questions remain unanswered such as how to: 1) determine the current state-of-the-art technique; 2) evaluate the relative benefit of new methods for various properties of the dataset; 3) understand what specific problems merit greater attention; and 4) measure the progress of the field over time.In this paper, we survey and compare various AL strategies used in both recently proposed and classic highly-cited methods. We propose to benchmark pool-based AL methods with a variety of datasets and quantitative metric, and draw insights from the comparative empirical results.

1 Introduction

AL is an effective technique when the training data is insufficient, which selects the most critical instances and queries their labels through the interaction with oracles (annotated by experts or apply crowd-sourcing techniques). AL contrasts with passive learning, where the labeled data are taken at random. The objective in AL is to produce highly-accurate classifiers, ideally using fewer labels than that of passive learning to achieve the same performance [Yang *et al.*, 2013]. AL has many variants according to the sample selection strategy: stream-based AL, membership query synthesis and pool-based AL. All these variants of AL strategies query the oracle for the labels of the points, but differ from each other in the nature of their queries [Settles, 2009]. In this paper, we focus on the category of pool-based AL, which assumes that one has access to a large pool of unlabeled i.i.d data samples, and selects the most informative set of points iteratively until the classifier reaches a certain level of performance, e.g., the classification accuracy or a pre-defined budget is exhausted [Chu *et al.*, 2011].

While many pool-based AL methods have been proposed, relatively less benchmarking and integration of AL techniques have occurred. Some researchers employ AL to improve or solve the data insufficiency problems in their own specific research tasks instead of improving the AL technique itself. The natural isolating effect of research communities may lead researchers to develop new AL methods only within those communities they participate in, which dampens the awareness of effective techniques in other research fields, especially when the method is applied to domain-specific tasks. In many papers, AL algorithms are evaluated on handpicked datasets on which they show major advantage. Although such evaluation shows the benefits of the AL algorithm, it ignores the failure regimes of the algorithms, which are important for understanding and addressing the challenges in AL. For these reasons, it is difficult to determine the current state-of-the-art of pool-based AL, which affects the evaluation of newly proposed AL methods, and obfuscates progress in the field.

Looking at other fields of ML, such as Computer Vision (CV) and Natural Language Processing (NLP), significant research progresses have been made in conjunction with standard benchmark datasets, such as ImageNet, MNIST, Pascal VOC, MSCOCO, GLUE, etc., on which disparate algorithms can be compared in a standard way. In this paper, we propose an AL benchmark, consisting of multiple datasets with various properties, associated evaluation metrics, and experiment protocol. We perform benchmark tests on a variety of AL approaches. We hope that this benchmarking test could bring authentic comparative evaluation for the researchers in AL, providing a quick look at which methods are more effective for those who want to incorporate AL techniques into other research fields, as well as construct a standard benchmark for new AL methods on which fair comparisons can be made.

2 Pool-based AL Techniques

For the sake of generality, we exclude the AL with crowd-sourcing work (labels generated by multiple oracles/human annotators) [Mozafari *et al.*, 2014; Huang *et al.*, 2017], AL with transfer learning or semi-supervised learning [Hoi and Lyu, 2005; Zhao *et al.*, 2013; Guo *et al.*, 2016; Guo *et al.*, 2017] and AL with multi-label classification, regression or ranking tasks [Cai *et al.*, 2013; Mohajer *et al.*, 2017; Reyes *et al.*, 2018]. Since the space of AL algorithms is vast, we

*Contact Author

consider a variety of well-known AL methods that provide representative baselines of current practice. We next review these methods while emphasizing relationships between them and distinguishing traits and possible variants.

2.1 Problem Definition

We consider a general process of pool-based AL for classification task. We have a small initial labeled set $\mathcal{D}_l = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_M, y_M)\}$ and a large unlabeled data pool $\mathcal{D}_u = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, where each instance $\boldsymbol{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in \{0, 1\}$ is the class label of $\boldsymbol{x}_i$ for binary classification, or $y_i \in \{1, ..., k\}$ for multi-class classification. In each iteration, the active learner selects a batch $\mathcal{D}_q$ with size S from $\mathcal{D}_u$, and queries their labels from the oracle. $\mathcal{D}_l$ and $\mathcal{D}_u$ are then updated, and the basic classifier(s) is retrained on $\mathcal{D}_l$. The process terminates when the querying budget B is exhausted.

2.2 Querying Strategy

In terms of AL sampling strategies, pool-based AL approaches/heuristics can be roughly classified into 3 categories [Monarch, 2021]. First, *uncertainty-based sampling* strategies [Lewis and Catlett, 1994] aim to select the unlabeled data samples with lowest confidence (largest uncertainty) for the model to be classified correctly, such as least confidence [Culotta and McCallum, 2005], margin/ratio of confidence [Scheffer *et al.*, 2001], or entropy-based [Dagan and Engelson, 1995]. Second, *diversity/representative sampling* strategies select data that contains diversity information of the data pool to reduce the constraints on the supervised machine learning models from data. e.g., outlier detection [Abe *et al.*, 2006], cluster-based sampling [Dasgupta and Hsu, 2008], representative/density-based sampling [Wang and Ye, 2015]. Third, *advanced/combined* strategies [Shen *et al.*, 2004; Ebert *et al.*, 2012; Li and Guo, 2013; Ash *et al.*, 2019] integrate the advantages of uncertainty-based and diversity-based criteria, and are widely adopted in AL and its applications since they are more adaptable to varying data types.

Uncertainty-based. Classical uncertainty-based sampling strategies include: **Uncertainty Sampling (US)** which queries the instances in $\mathcal{D}_u$ that have the least certainty in their predicted label [Lewis and Catlett, 1994], and its variants including Least Confident (LC), Margin-based (M) and Entropy-based (ENT). US has become one of the most frequently used AL heuristics since it is both simple and computationally efficient. However, US only considers the uncertainty of samples and ignores their category distribution, which restricts the quality of sampling [Ye *et al.*, 2016]. **Query-by-Committee (QBC)** uses a committee of models $\mathcal{C} = \{\theta^{(1)}, ..., \theta^{(C)}\}$ (constructed by ensemble methods or various basic classifiers), which are trained on $\mathcal{D}_l$ to predict the labels of $\mathcal{D}_u$, and the ones with largest disagreement are selected for labeling by an oracle [Seung *et al.*, 1992; Settles, 2009]. The disagreement level could be measured by Voting Entropy (VE) or KL divergence.

Other methods aim to reduce the uncertainty in the classifier. **Expected Model Change (EMC)** selects instances that induce the largest change in the classifier (e.g., largest gradient descent) [Cai *et al.*, 2013]. **Expected Error Reduction (EER)** maximizes the decrease of loss by adding new data samples [Settles, 2009]. **Variance Reduction (VR)** regards the most informative data points which minimize the model's variance [Cohn, 1994].

Some AL techniques are designed for specific ML algorithms. Kapoor [2007] proposed an algorithm that balances exploration and exploitation by incorporating mean and variance estimation of the Gaussian Process classifier (**ALGP**). Kremer [2014] proposed a SVM-based AL strategy by minimizing the distances between data points and classification hyperplane (**HintSVM**). These model-driven active learning strategies aim to estimate how strongly learning from a data point influences the current model. **Learning Active Learning (LAL)** is a data-driven approach that uses properties of classifiers and data to predict the potential error reduction [Konyushkova *et al.*, 2017]. However, general uncertainty-based sampling strategies focus more on the benefit obtained by a single point, which might not be robust to outliers. To address this issue, density-weighted methods consider the average similarity between the selected samples and the whole data pool as a weight on the informativeness-based scores. In Section. 4, we adopt density-weighted US (**DWUS**) method.

Diversity/Representative-based. Diversity/representative sampling strategies measure whether an instance well represents the overall pattern of the unlabeled data pool and whether the selected batch maximizes the training utility, by comparing the similarity among data samples. However, this strategy requires querying a large number of instances before reaching the optimal decision boundary, and hence it is not as efficient as the uncertainty-based criterion. A typical method for measuring diversity/representativeness is via clustering [Hsu and Lin, 2015]. Dasgupta [2008] proposed a hierarchical clustering strategy (**Hier**), which uses the cluster information to calculate representativeness. Another typical clustering based method is k-Center (**KCenter**), which finds subset (denote as C) that minimizes the maximum distance of any point to a center [Sener and Savarese, 2017].

Most diversity/representative sampling methods perform better when the number of labeled samples is not sufficient, while uncertainty-based criterion usually overtakes the diversity/representative measure after substantial sampling. The main reason is that the diversity/representative criterion could obtain the entire structure of a database in the beginning stage, but it is insensitive to the data samples that are close to the decision boundary, notwithstanding the fact that such samples are probably more important to the prediction model. In addition, uncertainty-based measure always searches for the "valuable" samples around the current decision boundary, and the optimal decision boundary cannot be found unless a certain number of samples have already been labeled. The uncertainty-based and diversity-based sampling criterion can only guarantee their optimal performance over a period of time in the entire AL processes, and the optimal period differs for each criterion [Zhao *et al.*, 2019].

Advanced/Combined-based. Both uncertainty and diversity based strategies are single criterion-based AL methods, which only consider one optimization goal during the AL process, i.e., select the most informative (maximum uncertainty

Algorithm	Optimization Function	Notation
US (LC)	$x^*_{LC} = \arg\max_x 1 - p_\theta(\hat{y}\|x)$	$\hat{y} = \arg\max_y p_\theta(y\|x)$
US (M)	$x^*_M = \arg\min_x p_\theta(\hat{y_1}\|x) - p_\theta(\hat{y_2}\|x)$	y_1 and y_2 are the first and second most probable class labels
US (ENT)	$x^*_{ENT} = \arg\max_x -\sum_i p(y_i\|x;\theta)\log p(y_i\|x;\theta)$	
QBC (VE)	$x^*_{VE} = \arg\max_x -\sum_i \frac{V(y_i)}{C}\log\frac{V(y_i)}{C}$	$V(\cdot)$ is the voting entropy across the committee of classifiers, $\mathcal{C}$ represents the committee as a whole
QBC (KL)	$x^*_{KL} = \arg\max_x \frac{1}{C}\sum_c KL(P_{\theta(c)}\|\|P_{\mathcal{C}})$	KL – Kullback-Leibler Divergence
EMC	$x^*_{EMC} = \arg\max_x -\sum_i p_\theta(y_i\|x)\|\|\nabla l_x(\theta)\|\|$	
EER	$x^*_{EER} = \arg\min_x \sum_i p_\theta(y_i\|x)(-\sum_{u=1}^{U}\sum_j p_{\theta+}(y_j\|x^{(u)})\log p_{\theta+}(y_j\|x^{(u)}))$	θ^+ refers to the newly trained model after adding new data tuple
VR	$x^*_{VR} = \arg\min_x \tilde{\sigma}^2_o$	$\tilde{\sigma}^2_o$ is the model's variance, which models the learner's squared-loss with respect to the target function.
DWUS	$x^*_{DE} = \arg\max_x \phi_A(x)(\frac{1}{U}\sum_{u=1}^{U} sim(x, x^{(u)}))^\beta$	ϕ indicates the informativeness score generated by aforementioned strategies, $sim(\cdot)$ is the similarity measure.
ALGP	$x^*_{ALGP} = \arg\min_{x_u\in\mathcal{D}_u} \frac{\|\hat{y}_u\|}{\sqrt{\Sigma_u+\sigma^2}}$	This optimization function is based on **GP** classification models, $\hat{y}_u$ and Σ_u are posterior mean and variance, σ^2 is the variance of data distribution
LAL	$x^*_{LAL} = \arg\max_{x\in\mathcal{D}^t_u} g(\phi_t, \psi_x)$	g reflects to a regressor which constructs the relationship between classification state parameter ϕ, data state ψ and loss reduction δ
KCenter	$\min_{C:\|C\|<b}\max_i\min_{j\in C\cup D^l}\Delta(x_i, x_j)$	Δ for pair-wise distance
QUIRE	$x^*_{QUIRE} = \arg\min_{x_s}\hat{\mathcal{L}}(D_l, D_u, x_s)$	$\hat{\mathcal{L}}(D_l, D_u, x_s) = \min_{y_u\in\{\pm1\}^{n_u-1}}\max_{y_s=\pm1}\min_{f\in\mathcal{H}}\frac{\lambda}{2}\|f\|^2_{\mathcal{H}} + \sum_{i=1}^{n} l(y_i, f(x_i))$ (n refers to the size of all data)
Graph	$x^*_{Graph} = \arg\min_x \beta(t)r(U(x)) + (1-\beta(t))r(D(x))$	$r(\cdot)$ refers to ranking, U is the info measure, D is rep measure, β is a time-varying parameter
Margin	$\min_{\alpha:\alpha\in\{0,1\},\alpha^T 1=b}\|\|\frac{1}{n_l+b}(\sum_{j\in L}\Phi(x_j) + \sum_{i\in U}\alpha_i\Phi(x_i)) - \frac{1}{n_u-b}\sum_{i\in U}(1-\alpha_i)\Phi(x_i)\|\|$	α indicates whether the data point is selected or not, $\Phi(\cdot)$ is known as the feature space map from the nonempty compact set ($x\in\mathcal{X}$) to the complete inner product space (i.e., a reproducing kernel Hilbert space (RKHS))
AAL	$x^*_{AAL} = \arg\max_{i\in U} h_\beta(x_i)$	$h_\beta(x) = f(x)^\beta d(x)^{1-\beta}$. $f(x)$ is the uncertainty measure, $d(x)$ is the mutual information based informative density
BMDR	$\min_{\mathcal{D}_q, f}\sum_{\{\mathbf{x},y\}\in\mathcal{D}_l} l(f,\mathbf{x},y) + \sum_{\mathbf{x}_i\in\mathcal{D}_q} l(f,\mathbf{x},\hat{y}) + \lambda\|\|f\|\|^2 + \beta MMD(\mathcal{D}, \mathcal{D}_l\cup\mathcal{D}_q)$	$\mathcal{D} = \mathcal{D}_l\cup\mathcal{D}_u$
SPAL	$\min_{f,\mathbf{w},\mathbf{v}} l(f,\mathbf{w},\mathbf{v}) + \lambda g(\mathbf{v}) + \mu h(\mathcal{D}_l\cup\mathcal{D}_q, \mathcal{D}_u/\mathcal{D}_q) + \gamma\Omega(f)$	$\mathbf{w}$ and $\mathbf{v}$ reflect the informativeness and representativeness of data samples, f is the learning model, $g(\cdot)$ is the self-paced regularizer, h is MMD, $\Omega(f)$ is for controlling the model complexity. l, h, and g are responsible for informativeness, representativeness and easiness, respectively.

Table 1: Summary of various AL algorithms: sampling strategies and objective functions.

score) or the most representative data points. The effectiveness of AL could be improved by integrating multiple criteria, that is, constructing the advanced/combined AL strategies that avoid the pitfalls of a single criterion. The combined strategies can be categorized according to the integration pattern [Zhao *et al.*, 2019]:

1) *Serial-form combined* strategies employ each selection criterion sequentially to filter out non-useful samples until the batch size is reached. Shen [2004] proposed an integration strategy by first selecting a subset from $\mathcal{D}_u$ via informativeness scores, then clustering the pre-selected set, taking the clustering centers as final results. This pattern is efficient and flexible since specific querying strategies can be added into the original process. However, its performance relies on the selection of the committee of basic query strategies, and the size of the subsets generated by each component.

2) *Criteria selection* strategies choose one criterion with the highest criterion selection parameter to query samples within one iteration, which could be also called "mix-up AL strategies". Active Learning by Learning (**ALBL**) [Hsu and Lin, 2015] selects data points with probability $q_j(t) = \sum_{k=1}^{K} p_k(t)\phi^k_j(t)$, where p refers to the probability of selecting an AL algorithm.

3) *Parallel-form combined* strategies select samples from multiple querying criteria by using a weighted sum of objectives or other multi-objective optimization methods. The normal practice is to combine two or three criteria measuring informativeness, representativeness and diversity. Parallel-form is the most widely of the combined AL strategies, but the disadvantage is that it depends heavily on how the weights of each criterion are set. **Graph Density (Graph)** is a typical parallel-form combined strategy that balances the uncertainty and representative based measure simultaneously via a time-varying parameter [Ebert *et al.*, 2012]. **Marginal Probability based Batch Mode AL (Margin)** [Chattopadhyay *et al.*, 2013] selects a batch that makes the marginal probability of the new labeled set similar to the one of unlabeled set, via optimization by Maximum Mean Discrepancy (MMD). **Representative Marginal Cluster Mean Sampling (MCM)** first selects the data points in the separating hyperplane generated by SVM and then clusters them to find k centroid data points [Xu *et al.*, 2003]. **QUIRE** queries the most informative and representative data points in each AL iteration [Huang *et al.*, 2010]. **Adaptive Active Learning (AAL)** considers how to adjust the trade-off parameter of each criterion via a self-adjusting mechanism [Li and Guo, 2013]. **Batch-mode Discriminative and Representative AL (BMDR)** [Wang and Ye, 2015] queries a batch of informative and representative examples by minimizing the empirical risk bound of AL. **Self-paced AL (SPAL)** [Tang and Huang, 2019] selects a batch of informative, representative and easy examples by minimizing a well designed objective function.

To facilitate a more intuitive method comparison, we have selected some of these AL methods that can clearly present their optimization goals and summarized them into Table 1.

3 Pool-based AL Benchmark

While many Al methods have been proposed, there still exist fundamental questions worth exploring. In which scenarios are uncertainty or diversity-based querying strategies more suitable? Compared with single-criterion based methods, how well do the advanced/combined sampling strategies foster strengths and circumvent weaknesses? To answer these questions, we construct a large benchmark for pool-based AL,

including datasets, protocol, and metric.

Datasets. Most works employ large number of public general real-life datasets for validating the effectiveness of their AL models on general tasks and several domain-specific datasets for their concerned research field, thereby increasing validity of experimental findings. Synthetic data can also be useful for sanity checks, carefully controlled experiments, and benchmarking. We have studied a large number of papers in the AL literature, and found that there is no uniform set of datasets for evaluation, which potentially leads to the following problems: 1) the disjointed sets of evaluation datasets make it difficult to compare horizontally among various AL methods; 2) When implementing new AL approaches, some datasets are too simple to verify efficacy as the performance is already saturated. Therefore, a unified set of appropriate datasets is required, which could help facilitate meaningful comparisons among methods and benefit progress in AL.

In Table 2, we summarize 35 public datasets that we use in our benchmark[1]. The table shows the source, data properties, imbalance ratio, dimension, size, number of categories and the related literatures that used these datasets.

Experiment protocol. We next describe the experiment protocol for the benchmark. For each dataset, we randomly select 60% of the data for training and the remaining 40% for testing. We select data samples from the training set and evaluate the classification performance on the testing set. In order to reduce the variance of the result (and avoiding results that are just "lucky" splits of the data), we repeat each experiment for 100 trials for dataset of $n < 2000$ and 10 trials for dataset of $n > 2000$, with random splits of the training and testing sets, and report the average testing performance. Note that we set the random seed in each trial so that all AL methods use the same training/testing/initial data in each trial, which ensures a fair comparison among methods. To avoid bias problems, we avoid any dataset-specific tuning or pre-processing.

Evaluation metrics. To evaluate the overall performance, we propose an evaluation metric called *area under the budget curve* (AUBC), which based on the performance-budget curves, computed by evaluating the AL method for different fixed budgets (e.g., accuracy vs. budget – AUBC(acc)). Given the budget curve, the AUBC is calculated by the trapezoid method, and the higher value reflects better performance of the AL strategy under varying budgets. In our experiments, we varying budgets from zero to the size of whole unlabeled data pool. In our experiments, we employ AUBC(acc).

Beam-Search Oracle result. As each dataset is different, the performance of AL methods will vary substantially across datasets based on the data distribution. We compute the "near-optimal AL performance" on each dataset as reference. A full search of all permutations of sample sequences is intractable, and thus we resort to a beam-search method for approximating the optimal sequence of selected samples that maximizes the classification accuracy. Given an initial labeled data pool $\mathcal{D}^{(0)}$, in the first iteration, we select 5 data points $\boldsymbol{x}_i$ that yield the largest test accuracy of the classifier trained on the updated pool $\mathcal{D}_i^{(1)} = \mathcal{D}^{(0)} \cup x_i$. In the second iteration, for each $D_i^{(1)}$, we select another 5 samples $\boldsymbol{x}_j$ that yield largest test accuracy of the classifier trained on the updated pool $\mathcal{D}_{ij}^{(2)} = \mathcal{D}_i^{(1)} \cup x_j$. Now there are 25 pools $\mathcal{D}_{ij}^{(2)}$, from which we select the 5 with largest testing accuracy, to obtain pruned set of 5 pools $\mathcal{D}_k^{(2)}$. The iterations are repeated until the budget is exhausted. Thus, we obtain a *near-optimal* labeling sequence for calculating the AUBC. We denote this method as *Beam-Search Oracle* (BSO), since it uses the test data to optimize the AL sequence. Following the benchmark protocol, we conduct BSO with the same settings as the AL methods.

4 Experiment

In this section we run experiments on our benchmark, comparing 17 aforementioned methods in Section 2 on 35 datasets in Table 2. The main goals of our experiment are to: 1) identify which datasets are more meaningful for evaluating the effectiveness of pool-based AL methods; 2) distinguish high-performance AL methods under multiple datasets.

AL model setup. We use the public implementations of these algorithms: **US**, **QBC**, **HintSVM**, **QUIRE**, **ALBL**, **DWUS** and **VR** are implemented by the libact [Yang *et al.*, 2017]; **Uniform**, **KCenter**, **Margin**, **Graph**, **Hier**, **InfoDiv** and **MCM** are from the Google AL toolbox[6]; **EER**, **BMDR**, **SPAL** and **LAL** are from ALiPy [Tang *et al.*, 2019]. We use **SVM (RBF)** as the basic classifier to test the AL performance.

4.1 Experimental Results

There are a large number of results, as we consider a large number of AL methods and datasets. Thus, we analyze the experimental results at a high level, from the aspects of dataset and method with their different properties.

Dataset aspect. Table 3 summarizes the BSO performance, mean, minimum and maximum performance across 17 AL methods for each dataset. Note that the differences between the actual results and BSO results on some datasets are not as large as one might expect (e.g., *Seeds* and *GCloudb*), since the AUBC metric actually takes the average performance of the whole labeling sequence, while most labeling sequences will converge when the budget is large enough. The performance is saturated on some highly-cited datasets (e.g., *Iris*, *Wine*, *Ionosphere*, *Diabetes*, *Tic-tac-toe*). However, we do not advocate removing these well-worn datasets, since we want our benchmark to contain both easy and hard datasets, so as to test AL methods under different regimes. In Table 3, we underline the datasets that have significant disparities between the BSO results and the best performance of AL methods (i.e., the BSO result is more than one percentage point higher than the best performing AL method), which indicates that there

[1]It is an ongoing process to increase the size since this benchmarking task should become a dynamic and evolving community resource.

[2]http://www.keel.es

[3]https://archive.ics.uci.edu/ml/datasets.php

[4]http://openclassroom.stanford.edu

[5]http://www.cs.toronto.edu/ delve/data

[6]https://github.com/google/active-learning

Dataset	Property	IR	(d, n, K)	Source	Related Literature
Appendicitis	Real-life	4.05	(7, 106, 2)	KEEL[2]	[Wang *et al.*, 2019]
Sonar	Real-life	1.14	(60, 108, 2)	UCI[3]	[Chattopadhyay *et al.*, 2013; Hsu and Lin, 2015; Du *et al.*, 2015; Cuong *et al.*, 2016; Tüysüzoğlu and Yaslan, 2018; Wang *et al.*, 2019]
Iris	Real-life	1.00	(4, 150, 3)	UCI	[Chattopadhyay *et al.*, 2013; Du *et al.*, 2015; Wang *et al.*, 2018b; Bernard *et al.*, 2018; Wang *et al.*, 2019]
Wine	Real-life	1.48	(13, 178, 3)	UCI	[Cai *et al.*, 2013; Chattopadhyay *et al.*, 2013; Du *et al.*, 2015; Wang *et al.*, 2018b]
Parkinson	Real-life	3.06	(22, 195, 2)	UCI	[Xiong *et al.*, 2013; Yang and Loog, 2018]
EX8b (linear)	Synthetic	1.00	(2, 206, 2)	ML Course[4]	–
Seeds	Real-life	1.00	(7, 210, 3)	UCI	[Wang *et al.*, 2018a; Wang *et al.*, 2019]
Glass	Real-life	8.44	(9, 214, 7)	UCI	[Mozafari *et al.*, 2014; Wang *et al.*, 2018b; Wang *et al.*, 2018a]
Thyroid	Real-life	5.00	(5, 215, 4)	UCI	[Abe *et al.*, 2006; Wang and Ye, 2015; Wang *et al.*, 2018a; Tang and Huang, 2019]
Heart	Real-life	1.25	(13, 270, 2)	UCI	[Chattopadhyay *et al.*, 2013; Ali *et al.*, 2014; Du *et al.*, 2015; Hsu and Lin, 2015]
Haberman	Real-life	2.78	(3, 306, 2)	UCI	[Azimi *et al.*, 2012; Mozafari *et al.*, 2014; Wang *et al.*, 2019]
Ionosphere	Real-life	1.79	(34, 351, 2)	UCI	[Ali *et al.*, 2014; Du *et al.*, 2015; Cuong *et al.*, 2016; Wang *et al.*, 2018b; Tüysüzoğlu and Yaslan, 2018; Wang *et al.*, 2019]
MUSK (Clean1)	Real-life	1.30	(168, 475, 2)	UCI	[Chattopadhyay *et al.*, 2013; Yang and Loog, 2018; Tang and Huang, 2019]
Breast Cancer	Real-life	1.00	(10, 478, 2)	UCI	[Azimi *et al.*, 2012; Hsu and Lin, 2015; Cuong *et al.*, 2016; Wang *et al.*, 2018a; Yang and Loog, 2018]
Wdbc	Real-life	1.68	(30, 569, 2)	UCI	[Huang *et al.*, 2010; Chen and Krause, 2013; Li *et al.*, 2015; Yang and Loog, 2018]
R15	Synthetic	1.00	(2, 600, 15)	–	[Wang *et al.*, 2018a; Wang *et al.*, 2019]
Statlog (Australian)	Real-life	1.25	(14, 690, 2)	UCI	[Huang *et al.*, 2010; Chen and Krause, 2013; Li *et al.*, 2015; Du *et al.*, 2015; Wang *et al.*, 2018b; Zhao *et al.*, 2019]
Diabetes	Real-life	1.87	(8, 768, 2)	UCI	[Li *et al.*, 2015; Hsu and Lin, 2015; Du *et al.*, 2015; Cuong *et al.*, 2016]
Mammographic	Real-life	1.06	(5, 830, 2)	UCI	[Abe *et al.*, 2006; Mozafari *et al.*, 2014; Krempl *et al.*, 2015; Yang and Loog, 2018]
EX8a (non-linear)	Synthetic	1.00	(2, 863, 2)	ML Course	–
Statlog (Vehicle)	Real-life	1.10	(18, 946, 4)	UCI	[Huang *et al.*, 2010; Chattopadhyay *et al.*, 2013; Huang *et al.*, 2017; Wang *et al.*, 2018b; Zhao *et al.*, 2019]
Tic-Tac-Toe	Real-life	6.79	(9, 958, 2)	UCI	[Huang *et al.*, 2010; Huang *et al.*, 2017; Yang and Loog, 2018; Zhao *et al.*, 2019; Tang and Huang, 2019]
Statlog (German)	Real-life	2.33	(20, 1000, 2)	UCI	[Azimi *et al.*, 2012; Du *et al.*, 2015; Li *et al.*, 2015; Tüysüzoğlu and Yaslan, 2018; Yang and Loog, 2018]
Molecular Biology (Splice)	Real-life	1.07	(61, 1000, 2)	UCI	[Wang and Ye, 2015; Du *et al.*, 2015; Li *et al.*, 2015; Konyushkova *et al.*, 2017; Huang *et al.*, 2017]
Gaussian Cloud Balance	Synthetic	1.00	(2, 1000, 2)	–	[Konyushkova *et al.*, 2017]
Gaussian Cloud Unbalance	Synthetic	2.00	(2, 1000, 2)	–	[Konyushkova *et al.*, 2017]
XOR (Checkerboard2×2)	Synthetic	1.00	(2, 1600, 2)	–	[Konyushkova *et al.*, 2017]
Phishing Websites	Real-life	1.26	(30, 2456, 2)	UCI	[Tang and Huang, 2019]
D31	Synthetic	1.00	(2, 3100, 31)	–	[Wang *et al.*, 2019]
Spambase	Real-life	1.68	(57, 4601, 2)	UCI	[Mozafari *et al.*, 2014; Huang *et al.*, 2017]
Banana	Synthetic	1.23	(2, 5300, 2)	–	[Wang and Ye, 2015; Wang *et al.*, 2019]
Phoneme	Real-life	2.41	(5, 5404, 2)	ELENA Project	[Tang and Huang, 2019]
Texture	Real-life	1.00	(40, 5500, 11)	UCI	[Wang *et al.*, 2019]
Ringnorm	Real-life	1.02	(21, 7400, 2)	Leo Breiman[5]	[Wang and Ye, 2015; Du *et al.*, 2015]
Twonorm	Real-life	1.00	(50, 7400, 2)	Leo Breiman	[Wang and Ye, 2015; Du *et al.*, 2015; Wang *et al.*, 2019]

Table 2: Benchmarking datasets. (d, n, K) are the feature dimension, number of samples, and number of categories. The Imbalance Ratio (IR) is the ratio of the number of samples in the majority class to that of the minority class.

exists enough potential and space for improvement for future research. For the datasets that have no BSO results, we calculate the difference between the average AL performance and random sampling performance (average performance is more than one percentage point higher than RS).

Method aspect. We analyze the results from the method aspect, comparing 17 AL algorithms, as shown in Table 4. Note that among these AL methods, **US**, **QBC**, **HintSVM**, **QUIRE**, **VR**, **ALBL** and **DWUS** do no support batch mode, while **HintSVM** and **ALBL** do not support multi-class classification. Considering the overall performance, **BMDR**, **Hier**, **QBC**, **MCM**, **Graph** perform better (performance gap < 0.3). **QBC** belongs to uncertainty-based sampling strategies and it adopts a committee of basic classifiers which could better adapt to various data situations. **Hier** belongs to diversity-based sampling strategies and besides clustering settings, it also considers reducing the sampling bias during the AL processes. **BMDR**, **MCM**, **Graph** are combined strategies, they show superior effectiveness across datasets (**Margin**, **InfoDiv** are also combined strategies with with overall competitive performances). Considering various types of data properties, we divide the datasets into 5 groups: binary/multi-class, real/synthetic, low/high dimension, small/large scale and balance/imbalance. We also discuss effectiveness of batch mode.

Binary/multi-class view: For binary classification, **LAL**, **QBC** and **Hier** have good performance, while some uncertainty-based methods with single basic classifier, i.e., **HintSVM**, **VR**, **DWUS**, **EER**, show less advantage. The methods that consider the overall data topology yield better performance on multi-class classification tasks (i.e., **BMDR**, **Hier**, **Graph** and **KCenter**), while uncertainty-based sampling methods (e.g., **US**) and the methods that only consider the similarity between the labeled and unlabeled set (e.g., **SPAL**) tend to fall into local optimal.

Low/high dimension view: We observe that the high-dimensional data are much more difficult than low-dimensional data, i.e., the performance gap of HD are twice as large as LD for most AL methods. It shows that testing on high-dimensional data sets yields more representative evaluations of AL methods. Compared with other AL methods, data with higher dimension does not have much impact on **QBC**.

Data scale view: Uncertainty-based sampling strategies with single basic classifier, e.g., **US**, **VR**, **DWUS** and **EER** have severe performance drop when the data scale become larger. In contrast, the diversity-based and combined strategies that integrate diversity/representative criterion (e.g., **Hier**, **InfoDiv**) perform even better on large-scale datasets, since these strategies capture the structure of the data.

Real/synthetic view: For the synthetic datasets, the methods that consider diversity measure, e.g., **Hier** and **KCenter**, achieve good performance. This is because in the simulated datasets, data are generated by certain well-defined rules, and therefore the data structure is easy to be captured by diversity/representative-based strategies. In contrast for real-life datasets where the data topology is more cluttered and complex, combined strategies help to select data points that have higher impact to the predictions and are more representative of the entire data pool.

Data balance/imbalance view: **QBC** and **LAL** both well handle various data types (**QBC** employs multiple classifiers and **LAL** learns properties of various classifiers and data), and

Dataset	RS	BSO	Avg	Best		Worst	
Appendicitis	0.836	0.881	0.844	0.859	EER	0.826	DWUS
Sonar	0.617	0.830	0.755	0.775	LAL	0.732	HintSVM
Iris	0.835	0.932	0.912	0.945	BMDR	0.870	US
Wine	0.858	0.946	0.942	0.967	BMDR	0.889	EER
Parkinsons	0.840	0.865	0.845	0.858	QBC	0.829	HintSVM
Ex8b	0.866	0.924	0.890	0.909	SPAL	0.864	HintSVM
Seeds	0.862	0.922	0.912	0.921	BMDR	0.905	VR
Glass	0.387	0.474	0.446	0.473	KCenter	0.393	DWUS
Thyroid	0.696	0.705	0.708	0.728	EER	0.693	DWUS
Heart	0.808	0.848	0.787	0.830	InfoDiv	0.718	DWUS
Haberman	0.727	0.751	0.727	0.735	BMDR	0.720	QUIRE
Ionosphere	0.901	0.933	0.909	0.931	LAL	0.884	HintSVM
Clean1	0.649	0.871	0.805	0.838	LAL	0.747	HintSVM
Breast	0.954	0.961	0.957	0.963	SPAL	0.953	DWUS
Wdbc	0.952	0.973	0.955	0.965	LAL	0.940	EER
R15	0.877	0.954	0.925	0.978	QBC	0.761	QUIRE
Australian	0.846	0.878	0.845	0.853	KCenter	0.820	DWUS
Diabetes	0.736	0.784	0.741	0.751	KCenter	0.691	EER
Mammographic	0.819	0.844	0.815	0.825	MCM	0.798	EER
Ex8a	0.838	0.873	0.836	0.864	Hier	0.804	QUIRE
Vehicle	0.567	0.598	0.486	0.575	BMDR	0.372	SPAL
Tic-tac-toe	0.870	0.873	0.870	0.872	EER	0.865	QUIRE
German	0.726	0.783	0.737	0.744	QBC	0.720	DWUS
Splice	0.806	0.871	0.791	0.821	QBC	0.679	EER
GCloudb	0.893	0.901	0.886	0.897	Graph	0.868	HintSVM
GCloudub	0.942	0.963	0.929	0.954	QBC	0.864	EER
Checkerboard	0.978	0.992	0.943	0.986	KCenter	0.902	VR
Phishing	0.926	–	0.939	0.945	LAL	0.923	Graph
D31	0.582	0.922	0.805	0.950	KCenter	0.634	QUIRE
Spambase	0.685	–	0.877	0.919	QBC	0.685	DWUS
Banana	0.895	–	0.847	0.893	Hier	0.784	QUIRE
Phoneme	0.822	–	0.823	0.831	QBC	0.802	HintSVM
Texture	0.666	–	0.918	0.973	Hier	0.617	DWUS
Ringnorm	0.976	–	0.949	0.978	LAL	0.800	DWUS
Twonorm	0.976	–	0.975	0.976	KCenter	0.972	DWUS

Table 3: AL performance by AUBC(acc) from the dataset aspect. We present the Random Sampling (RS) performance, BSO results, average (Avg) performance of each dataset across 17 AL methods, and the best (Best) and worst (Worst) performing AL methods. Symbol "–" indicates that BSO results are not completed yet.

thus well handle imbalanced data. Generally, all AL methods perform better on data imbalance situations, which indicates the effectiveness of AL methods. During AL processes, data samples that belong to the minority classes are more informative and thus are selected first.

Effectiveness of batch mode: Batch-mode AL aims to reduce the computational cost (re-training the classifiers) by selecting multiple samples in each round. Typically a diversity measure is included to encourage selection of different points in the batch – otherwise, without a diversity measure the same or similar points will be selected in a batch, which reduces the performance when compared with selecting 1 sample at time. The diversity measure in batch-mode AL works if the AL performance does not degrade with increasing batch sizes. To evaluate its effectiveness, we compare the performance with batch size of 1 (the basic setting) and batch sizes of 2, 5 and 10 (see "Batch" column in Table 3). Except for **LAL**, most batch mode AL methods have only a slight performance drop when applying larger batch size.

5 Discussion and Conclusion

We began this paper by noting the continued and frequent use of simple methods (e.g., **US**, **QBC**, etc.) to solve AL related tasks, despite the fact that many more sophisticated methods have been proposed. We are interested in exploring how much benefit does these sophisticated methods could offer. Furthermore, when a limited number of datasets and baselines are presented (and likely carefully selected to show advantage

Method	All	B	M	LD	HD	SS	LS	R	S	BAL	IMB	Batch
QBC	**0.024**	**0.023**	0.039	**0.023**	**0.030**	**0.028**	**0.015**	0.026	**0.016**	**0.028**	**0.015**	n/a
LAL	0.032	**0.020**	0.091	0.033	0.041	0.030	0.044	**0.025**	0.057	0.044	**0.016**	-0.010
US	0.039	0.033	0.058	0.038	0.047	0.033	0.049	0.032	0.069	0.047	0.024	n/a
HintSVM	0.048	0.048	n/a	0.043	0.067	0.049	0.047	0.045	0.064	0.058	0.036	n/a
VR	0.051	0.049	0.061	0.046	0.093	0.048	0.060	0.050	0.064	0.061	0.035	n/a
EER	0.051	0.051	0.052	0.040	0.117	0.042	0.074	0.047	0.061	0.055	0.034	**0.000**
DWUS	0.063	0.060	0.076	0.055	0.108	0.050	0.112	0.062	0.068	0.068	0.053	n/a
Hier	**0.024**	**0.024**	0.028	**0.021**	**0.039**	0.029	**0.014**	**0.025**	**0.016**	**0.025**	**0.021**	-0.003
KCenter	0.031	0.035	**0.025**	0.026	0.060	0.030	0.033	0.033	0.021	0.033	0.026	-0.002
BMDR	**0.022**	0.027	**0.011**	**0.016**	0.063	**0.021**	0.029	**0.024**	**0.017**	**0.023**	0.021	**0.000**
Graph	0.027	0.029	**0.026**	0.024	0.047	0.030	0.023	0.029	0.018	**0.028**	0.025	-0.003
MCM	0.029	0.029	0.035	0.026	0.043	**0.028**	0.030	0.027	0.039	0.032	0.024	-0.003
ALBL	0.030	0.030	n/a	0.027	0.042	0.036	**0.021**	0.030	0.027	0.035	0.024	n/a
InfoDiv	0.031	0.027	0.057	0.029	**0.040**	0.033	0.028	0.031	0.035	0.035	0.024	-0.004
Margin	0.032	0.029	0.058	0.031	**0.040**	0.033	0.030	0.032	0.036	0.036	0.025	-0.003
SPAL	0.051	0.029	0.124	0.052	0.066	0.044	0.104	0.045	0.078	0.067	0.024	**0.001**
QUIRE	0.062	0.048	0.099	0.059	0.088	0.051	0.097	0.048	0.109	0.080	0.029	n/a

Table 4: AL performance by AUBC(acc) from the method aspect. We present the average performance difference between the best AL/BSO and the AL method, i.e., $\delta_i = \max(BSO, a_1, \cdots, a_{17}) - a_i$, where a_i is the AUBC for the i-th method. **Smaller values indicate better AL performance.** We consider various data properties: the overall (All) performance, B/M is **B**inary/**M**ulti-class ($K = 2$, $K > 2$), LD/HD is **L**ow-**D**imension/**H**igh-**D**imension ($d < 50$, $d \geq 50$), SS/LS is **S**mall-**S**cale/**L**arge-**S**cale ($n < 1000$, $n \geq 1000$), R/S is **R**eal-life/**S**ynthetic, BAL/IMB is **BAL**ance/**IMB**alance ($IR < 1.5$, $IR \geq 1.5$). For methods that support batch-mode, the average performance drop when increasing batch sizes to 2, 5, and 10 is reported in the last column (Batch). In each column, the top-3 methods are bolded: **1st**, **2nd**, **3rd**. "n/a" indicates that the category (e.g., M) is not applicable to the related AL method.

of the proposed method), how to quantify the superiority of the proposed AL model is also a crucial problem.

When implementing new methods, we need to validate the proposed AL method against a variety of AL approaches on a large set of representative datasets, to ensure its improvement and generality. The degree of empirical diversity observed is larger than we expected since we only aggregated the existing AL methods and public datasets together into a larger benchmark, yet we obtain a variety of datasets with different properties: small/large scale, low/high dim, binary/multi-class, balance/imbalance, etc.

We have observed that some single criterion based methods (e.g., **QBC**, **Hier**) in the benchmark tests are often superior to many combined strategies. Moreover, in the scenarios where batch-mode AL is used to accelerate the learning process, the methods that integrate multiple criteria (e.g., **BMDR**) also perform well on our benchmark tests. However, compared with single criterion based methods, we do not observe much impressive improvements from the more sophisticated methods (e.g., **SPAL**, **QUIRE**). It is likely that some methods have overfit certain datasets at the expense of performance on other datasets. Thus, horizontal comparisons between AL methods on a common benchmark are necessary to ensure overall progress of the field.

We will continue to collect more datasets and AL methods for our benchmark, since better benchmarking tests help us to understand where the improvements come from.

Acknowledgments

This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11215820).

References

[Abe *et al.*, 2006] Naoki Abe, Bianca Zadrozny, and John Langford. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 504–509, 2006.

[Ali *et al.*, 2014] Alnur Ali, Rich Caruana, and Ashish Kapoor. Active learning with model selection. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[Ash *et al.*, 2019] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[Azimi *et al.*, 2012] Javad Azimi, Alan Fern, Xiaoli Zhang-Fern, Glencora Borradaile, and Brent Heeringa. Batch active learning via coordinated matching. *arXiv preprint arXiv:1206.6458*, 2012.

[Bernard *et al.*, 2018] Jürgen Bernard, Matthias Zeppelzauer, Markus Lehmann, Martin Müller, and Michael Sedlmair. Towards user-centered active learning algorithms. In *Computer Graphics Forum*, volume 37, pages 121–132. Wiley Online Library, 2018.

[Cai *et al.*, 2013] Wenbin Cai, Ya Zhang, and Jun Zhou. Maximizing expected model change for active learning in regression. In *2013 IEEE 13th International Conference on Data Mining*, pages 51–60. IEEE, 2013.

[Chattopadhyay *et al.*, 2013] Rita Chattopadhyay, Zheng Wang, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Batch mode active sampling based on marginal probability distribution matching. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):1–25, 2013.

[Chen and Krause, 2013] Yuxin Chen and Andreas Krause. Near-optimal batch mode active learning and adaptive submodular optimization. *ICML (1)*, 28(160-168):8–1, 2013.

[Chu *et al.*, 2011] Wei Chu, Martin Zinkevich, Lihong Li, Achint Thomas, and Belle Tseng. Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 195–203, 2011.

[Cohn, 1994] David A Cohn. Neural network exploration using optimal experiment design. In *Advances in neural information processing systems*, pages 679–686, 1994.

[Culotta and McCallum, 2005] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751, 2005.

[Cuong *et al.*, 2016] Nguyen Viet Cuong, Nan Ye, and Wee Sun Lee. Robustness of bayesian pool-based active learning against prior misspecification. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Dagan and Engelson, 1995] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, 1995.

[Dasgupta and Hsu, 2008] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. ACM, 2008.

[Du *et al.*, 2015] Bo Du, Zengmao Wang, Lefei Zhang, Liangpei Zhang, Wei Liu, Jialie Shen, and Dacheng Tao. Exploring representativeness and informativeness for active learning. *IEEE transactions on cybernetics*, 47(1):14–26, 2015.

[Ebert *et al.*, 2012] Sandra Ebert, Mario Fritz, and Bernt Schiele. Ralf: A reinforced active learning formulation for object class recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3626–3633. IEEE, 2012.

[Guo *et al.*, 2016] Yuchen Guo, Guiguang Ding, Yuqi Wang, and Xiaoming Jin. Active learning with cross-class knowledge transfer. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Guo *et al.*, 2017] Yuchen Guo, Guiguang Ding, Yue Gao, and Jungong Han. Active learning with cross-class similarity transfer. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[Hoi and Lyu, 2005] Steven CH Hoi and Michael R Lyu. A semi-supervised active learning framework for image retrieval. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 302–309. IEEE, 2005.

[Hsu and Lin, 2015] Wei-Ning Hsu and Hsuan-Tien Lin. Active learning by learning. In *Twenty-Ninth AAAI conference on artificial intelligence*, 2015.

[Huang *et al.*, 2010] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. In *Advances in neural information processing systems*, pages 892–900, 2010.

[Huang *et al.*, 2017] Sheng-Jun Huang, Jia-Lve Chen, Xin Mu, and Zhi-Hua Zhou. Cost-effective active learning from diverse labelers. In *IJCAI*, pages 1879–1885, 2017.

[Kapoor *et al.*, 2007] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[Konyushkova *et al.*, 2017] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from data. In *Advances in Neural Information Processing Systems*, pages 4225–4235, 2017.

[Kremer *et al.*, 2014] Jan Kremer, Kim Steenstrup Pedersen, and Christian Igel. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4):313–326, 2014.

[Krempl *et al.*, 2015] Georg Krempl, Daniel Kottke, and Vincent Lemaire. Optimised probabilistic active learning (opal). *Machine Learning*, 100(2):449–476, 2015.

[Lewis and Catlett, 1994] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994.

[Li and Guo, 2013] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–866, 2013.

[Li *et al.*, 2015] Chun-Liang Li, Chun-Sung Ferng, and Hsuan-Tien Lin. Active learning using hint information. *Neural computation*, 27(8):1738–1765, 2015.

[Mohajer *et al.*, 2017] Soheil Mohajer, Changho Suh, and Adel Elmahdy. Active learning for top-k rank aggregation from noisy comparisons. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2488–2497. JMLR. org, 2017.

[Monarch, 2021] R. Monarch. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Manning Publications, 2021.

[Mozafari *et al.*, 2014] Barzan Mozafari, Purna Sarkar, Michael Franklin, Michael Jordan, and Samuel Madden. Scaling up crowd-sourcing to very large datasets: a case for active learning. *Proceedings of the VLDB Endowment*, 8(2):125–136, 2014.

[Reyes *et al.*, 2018] Oscar Reyes, Carlos Morell, and Sebastián Ventura. Effective active learning strategy for multi-label learning. *Neurocomputing*, 273:494–508, 2018.

[Scheffer *et al.*, 2001] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318. Springer, 2001.

[Sener and Savarese, 2017] Ozan Sener and Silvio Savarese. A geometric approach to active learning for convolutional neural networks. *arXiv preprint arXiv*, 1708:1, 2017.

[Settles, 2009] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[Seung *et al.*, 1992] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294, 1992.

[Shen *et al.*, 2004] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 589. Association for Computational Linguistics, 2004.

[Tang and Huang, 2019] Ying-Peng Tang and Sheng-Jun Huang. Self-paced active learning: Query the right thing at the right time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5117–5124, 2019.

[Tang *et al.*, 2019] Ying-Peng Tang, Guo-Xiang Li, and Sheng-Jun Huang. ALiPy: Active learning in python. Technical report, Nanjing University of Aeronautics and Astronautics, January 2019.

[Tüysüzoğlu and Yaslan, 2018] Göksu Tüysüzoğlu and Yusuf Yaslan. Sparse coding based classifier ensembles in supervised and active learning scenarios for data classification. *Expert Systems with Applications*, 91:364–373, 2018.

[Wang and Ye, 2015] Zheng Wang and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):1–23, 2015.

[Wang *et al.*, 2018a] Xiaoqian Wang, Yijun Huang, Ji Liu, and Heng Huang. New balanced active learning model and optimization algorithm. In *27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*, 2018.

[Wang *et al.*, 2018b] Zengmao Wang, Xi Fang, Xinyao Tang, and Chen Wu. Multi-class active learning by integrating uncertainty and diversity. *IEEE Access*, 6:22794–22803, 2018.

[Wang *et al.*, 2019] Min Wang, Ying-Yi Zhang, and Fan Min. Active learning through multi-standard optimization. *IEEE Access*, 7:56772–56784, 2019.

[Xiong *et al.*, 2013] Sicheng Xiong, Javad Azimi, and Xiaoli Z Fern. Active learning of constraints for semi-supervised clustering. *IEEE Transactions on Knowledge and Data Engineering*, pages 43–54, 2013.

[Xu *et al.*, 2003] Zhao Xu, Kai Yu, Volker Tresp, Xiaowei Xu, and Jizhi Wang. Representative sampling for text classification using support vector machines. In *European conference on information retrieval*, pages 393–407. Springer, 2003.

[Yang and Loog, 2018] Yazhou Yang and Marco Loog. A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83:401–415, 2018.

[Yang *et al.*, 2013] Liu Yang, Steve Hanneke, and Jaime Carbonell. A theory of transfer learning with applications to active learning. *Machine learning*, 90(2):161–189, 2013.

[Yang *et al.*, 2017] Yao-Yuan Yang, Shao-Chuan Lee, Yu-An Chung, Tung-En Wu, Si-An Chen, and Hsuan-Tien Lin. libact: Pool-based active learning in python. Technical report, National Taiwan University, October 2017.

[Ye *et al.*, 2016] Zhipeng Ye, Peng Liu, Jiafeng Liu, Xianglong Tang, and Wei Zhao. Practice makes perfect: an adaptive active learning framework for image classification. *Neurocomputing*, 196:95–106, 2016.

[Zhao *et al.*, 2013] Lili Zhao, Sinno Jialin Pan, Evan Wei Xiang, Erheng Zhong, Zhongqi Lu, and Qiang Yang. Active transfer learning for cross-system recommendation. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[Zhao *et al.*, 2019] Yu Zhao, Zhenhui Shi, Jingyang Zhang, Dong Chen, and Lixu Gu. A novel active learning framework for classification: using weighted rank aggregation to achieve multiple query criteria. *Pattern Recognition*, 93:581–602, 2019.