Shattering Distribution for Active Learning

Xiaofeng Cao and Ivor W. Tsang

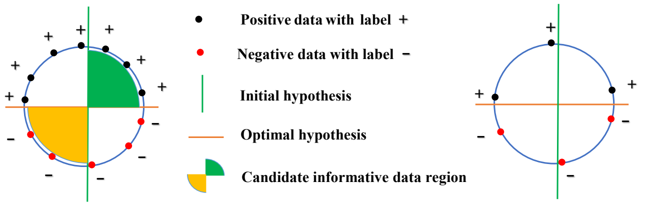
Abstract*—Active learning (AL) aims to maximize the learning performance of the current hypothesis by drawing as few labels as possible from an input distribution. Generally, most existing AL algorithms prune the hypothesis set via querying labels of unlabeled samples and could be deemed as a hypothesis-pruning strategy. However, this process critically depends on the initial hypothesis and its subsequent updates. This paper presents a distribution-shattering strategy without an estimation of hypotheses by shattering the *number density* of the input distribution. For any hypothesis class, we halve the number density of an input distribution to obtain a shattered distribution, which characterizes any hypothesis with a lower bound on VC dimension. Our analysis shows that sampling in a shattered distribution reduces label complexity and error disagreement. With this paradigm guarantee, in an input distribution, a Shattered Distribution-based AL (SDAL) algorithm is derived to continuously split the shattered distribution into a number of representative samples. An empirical evaluation on benchmark datasets further verifies the effectiveness of the halving and querying abilities of SDAL in real-world AL tasks with limited labels. Experiments on active querying with adversarial examples and noisy labels further verify our theoretical insights on the performance disagreement of the hypothesis-pruning and distribution-shattering strategies. Our code: https://github.com/XiaofengCao-MachineLearning/Shattering-Distribution-for-Active-Learning.

***Index Terms*—Active learning, distribution, shattering, adversarial examples, noisy labels.**

I. Introduction

Active learning (AL) [1], leveraging abundant unlabeled data to improve the generalization performance of a classifier, has been widely adopted in various machine learning tasks, such as regression analysis [2], label-scarce classification [3], dynamic data stream processing [4], multi-task learning [5] [6], curriculum learning [7], etc. By employing an AL algorithm, human experts strategically query "highly informative" data [8] to reduce the error rate of the current learning model in different classification tasks. However, a natural question that arises is the following: if we increase the size of the active query set, does the error rate of prediction keep decreasing? Furthermore, can we finally find a hypothesis whose error rate is close to what we desire?

This question has been considered by the agnostic AL community [9], which presents a series of algorithmic paradigms with a fixed or bounded version space [10] covering a possible hypotheses class [11]. Candidates from this class is assigned with a goal of minimizing the queries from the unlabeled pool, where the desired one is with the optimal querying budget. To build a near-optimal querying algorithm in real world, agnostic AL [12] [9] improved the generalization of a realizable-theoretical model with prior labels selected from various distributions and diverse noise conditions [13]. Those generalized AL algorithms involved with pruning the hypothesis set of the version space can be regarded as a hypothesis-pruning strategy. For example, halfspace learning [14] is one AL problem over a unit sphere to explore the theoretical guarantees on error rate and label complexity[1] [14]. Its goal is to learn a halfspace which accurately classifies binary classes (see Fig. 1(a)).


+
−
Positive data with label +
Negative data with label −
Initial hypothesis
Optimal hypothesis
Candidate informative data region

(a) Hypothesis-pruning

(b) Distribution-shattering

Fig. 1. A binary classification issue over a unit sphere with a radius of R, where $+, -$ denote the class labels. (a) The hypothesis-pruning strategy prunes the hypothesis set (reduce the number of candidate hypotheses, i.e., the diameters across the colored regions) via querying data distributed in the colored pool. (b) Distribution-shattering halves the number density of an input distribution w.r.t. $\frac{12}{\pi R^2}$ into a shattered distribution w.r.t. $\frac{6}{\pi R^2}$. Any hypothesis generated from the original distribution is charactered with a lower bound on VC dimension. Thereby, we can find a representation structure that induces a tighter label complexity without estimating the hypothesis.

Fig. 1(a) describes a binary classification task in a two-dimensional sphere (circle) with a uniform distribution and an arbitrary halfspace can generate a linear classifier. To reduce the error rate of the initial hypothesis, an AL algorithm usually samples a number of informative points from the colored candidate pools that can largely update the current classifier.

From the perspective of version space, the querying process is equivalent to searching a subspace that characterizes the same hypothesis with a lower bound on the Vapnik–Chervonenkis (VC) dimension [15] [16]. With each query, the disagreement between the initial and desired hypotheses is expected to shrink. Thus, the disagreement between the initial and optimal hypotheses can be used as a measure to determine the distribution of the candidate hypothesis class in a version space. However, the label complexity of querying unseen samples is sensitive to this measure. That is, a poor initial hypothesis, which is far from the desired hypothesis, results in an increase of their generalized disagreement. The label complexity of querying increases rapidly as well. Therefore, the query samples heavily depend on the initial hypothesis.

Most previous works regarding hypothesis-pruning either

X. Cao and I. W. Tsang are with Australian Artificial Intelligence Institute, University of Technology Sydney, NSW 2008, Australia. This work is supported by ARC Grant DP180100106 and DP200101328. E-mail:xiaofeng.cao.uts@gmail.com, ivor.tsang@uts.edu.au.

Manuscript received xx, xxxx; revised xx, xxxx.

[1] The number of labels requested before achieving a desired error.

makes strong distribution assumptions such as halfspace learning [14], or else it is computationally prohibitive [17]. For any data distribution, [15] removes the hypotheses whose connected edges are labeled with any disagreements larger than a given threshold. Their goal is to decrease the dependence of the initial hypothesis by a group of representative hypotheses. In their work, the version space [10] which includes all feasible hypotheses, is embedded as a graph in a high-dimensional space. After pruning with this graph, any hypothesis in the original version space would be characterized with a lower bound on VC dimension. Then, the upper bound of the label complexity is reduced. However, hypothesis-pruning strategy has the following limitations:

1) performing hypothesis-pruning in the candidate data pool could reduce the influence of the initial hypothesis but it does not completely eradicate its dependence;
2) hypothesis-pruning strategies with the hypothesis class need a special distribution assumption, but it cannot be applied in arbitrary input distributions, though this theoretical description has attracted a lot of attention from researchers.

Therefore, it is desirable to develop a novel shattering strategy which achieves the same goal as the hypothesis-pruning strategy and deals with the input distribution in real-world tasks. To this end, we attempt to bridge the connection between the version space and input distribution.

As discussed in [18], the VC dimension with respect to the optimal hypothesis in the version space affects the number of querying candidate hypotheses, and plays an important role in its distribution description. We propose a fresh proposition that the version space could be shattered by the number density [2] of the input distribution. Then, any hypothesis can be characterized with a lower bound on VC dimension. Especially for any input distribution with a bounded space, the more data located in the input space, the more hypotheses the version space would have. Moreover, *the input distribution induces a natural topology on the version space, and a local hypothesis would easily capture its relevant local distribution [12]*. Hereafter, we would perform the shattering on the number density of the input distribution (see Fig. 1(b)) with the following advantages:

1) it provides theoretical guarantees in relation to reducing the generalized bounds of label complexity and error disagreement as hypothesis-pruning;
2) it breaks the curse of the initial hypothesis;
3) it provides model guidance for distribution-shattering algorithms in real-world AL tasks.

Based on the above insights, this paper generalizes the distribution-shattering strategy in an input distribution. Firstly, we halve the number density of the input distribution to obtain a shattered distribution. We then compare the generalization bounds between the shattered distribution and input distribution on *error disagreement* and *label complexity* for any hypothesis class under arbitrary data distributions. Our theoretical results show that the shattered distribution has lower generalization bounds in terms of the above two properties. Thus, we continuously split the shattered distribution to find a representation structure. This process is guided by a derived algorithm termed Shattered Distribution-based AL (SDAL), which optimizes a group of local sphere centers as representative samples. Based on the analysis of the performance disagreement over hypothesis-pruning and distribution-shattering, we explore a series of scenarios including active querying with a limited labeled set, adversarial examples and noisy labels, where the first scenario is in regard to the poor initial hypothesis, and the last two scenarios are involved with the hypothesis update. The contributions of this paper are described as follows.

- We model the version space and input distribution by number density, which characterizes the generalized capacity of any hypothesis in a natural and direct way.
- We present a theoretical guarantee of the improvement on error disagreement and label complexity for shattering the number density of the input distribution. A derived algorithm named SDAL, which is independent of the initial labeled set and classifier, achieves lower error performance than the hypothesis-pruning algorithms when querying with limited labels, adversarial examples and noisy labels.

The outline of the paper is as follows. The related work is introduced in Section II. Section III presents the main theoretical insights. The theoretical motivation of the hypothesis and distribution is presented in Section IV. The proposed distribution-shattering strategy and the advantages of shattered distribution are reported in Section V. The experimental results are presented in Section VI. Conclusions are drawn in Section VII. (Proofs and discussions are presented in Appendix.)

II. Related Work

The AL algorithms aim to find data which have "highly informativeness or representativeness" for unseen queries. However, the sampled data heavily depend on the labeled set and classifiers. In this section, we introduce hypothesis-pruning from theory to application: Section II.A describes the fundamental theoretical work involved with hypothesis-pruning, and Section II.B reviews a number of AL algorithms that can be generalized with a hypothesis-pruning manner.

A. Hypothesis-pruning in Version Space

To reduce the dependence of the labeled set, AL tries to find a hardness (near-optimal) [19] [20] hypothesis from the hypothesis class in the version space. In this theoretical learning task, learners are given access to a stream of unlabeled data drawn i.i.d. from a fixed distribution. The proposed algorithm paradigms, which have already achieved a dramatic reduction in label complexity, are loosely termed hypothesis-pruning.

Substantial hypothesis-pruning frameworks under various assumptions of classifiers and labeled sets were proposed in past decades. For example, the query by committee algorithm

[2] https://en.wikipedia.org/wiki/Number_density

[21] assumes that a correct Bayesian prior exists on the hypothesis class. To find a desired hypothesis, the committee members vote to eliminate the updated hypothesis with maximal disagreement between them. For any hypothesis class, [22] presents the sufficient and necessary conditions for AL such as classifier setting and initial labeled set, etc. When there exists a perfect separator in classification tasks, any hypothesis-pruning algorithm could directly improve the current classifier in a rapid fashion such as uncertainty evaluation [23], expected error rate change [24], etc. Over this assumption, the learning algorithms do not need to consider the distribution it induces. Any inconsistent hypothesis such as a subsequent hypothesis with higher error can then be pruned by a single or group of querying samples. With the increase in the number of queries, the VC bound of any hypothesis in the candidate hypothesis class would be shrunk continuously, regardless from which distribution this query comes.

Since the optimal hypothesis is in respect to the input distribution, some learners generate the distribution under a fixed case, such as [9], [25], etc. Of these, halfspace learning [14] becomes a special setting over a unit sphere with uniform distribution. This problem takes a binary classification issue as an example to study label complexity and error rate change after sampling, where a halfspace is either of the two convex sets into which a hyperplane divides the sphere. The goal is to find the optimal halfspace over a unit sphere. For example, in Fig. 1(a), researchers try to reduce the vector angle θ between the initial and optimal hypotheses as rapidly as possible, in which θ decides the VC dimension of the current hypothesis. Under this training assumption, two methods are presented to reduce the label complexity: (1) halving [26] the volume of the candidate pool to obtain a sparse space, and (2) binary search for halving. By halving, the learner can rapidly reduce the hypothesis capacity of the version space to decrease the label complexity of querying since a part of the hypotheses would be removed. Therefore, the hypothesis-pruning strategy is an effective solution in AL theory. However, most of these AL algorithms either make strong distribution assumptions such as separability, uniform input distribution or are generally computationally prohibitive [12], thus they cannot effectively be applied in AL tasks with input distribution.

B. Generalized Hypothesis-pruning Algorithms

The hypothesis-pruning AL algorithms sample the data which significantly improve the generalization performance of the current learning model. By performing hypothesis-pruning sampling strategies, even non-domain experts no longer annotate such a large amount of unlabeled data. In this subsection, we review some generalized algorithms for the hypothesis-pruning strategy.

In the light of optimizing the querying process, the membership-query strategy [27] tries to find the most distinguished sample in a set by asking subset membership queries. In the structural data space with margins, the AL algorithm optimizes the unlabeled data close to the hyperplane under the support of the SVM theory. These approaches could be loosely approximated as finding the "highly informative" samples [8] [28] [29] to improve the performance of the model it learned. The other sampling criterion is to pick up the representative samples from the unlabeled data pool. With each query, the researchers want to minimize the distribution difference between the labeled and original input sets, such as the experimental design [30] [31] which measures the regression between data and their labels. To strengthen the sampling performance, [32] develops the informativeness and representativeness as a uniform standard with a given evaluation function. Then, a series of AL algorithms which exploit the data with both the characteristics is developed [33].

However, these strategies heavily depend on the categories of classifiers and labeled set. For example, the AL algorithms using the maximum margin of the classification hyperplane need the support of an SVM classifier. Moreover, the influence of the size of the input labeled set is underestimated. In our survey, most researchers usually query sufficient labels as the initial hypothesis, such as [34] [33] [35]. However, in real-world AL tasks, the percentage of labeled data may be less than the expected.

III. Main Theoretical Insights

Section III.A presents the preliminaries for one fundamental learning policy of the hypothesis-pruning strategy, which uses a disagreement coefficient to control the sampling of AL. Specifically, the concept of the sparse hypothesis class that provides the foundation for the distribution-shattering is introduced in Section III.B. Then, Section III.C analyzes the hypothesis-pruning and distribution-shattering AL by halfspace learning and discusses their performance disagreements.

A. Error Disagreement of Hypothesis-pruning

The hypothesis-pruning AL algorithm queries the label of one example based on the empirical rule of error rate difference after assigning a positive or negative label. To describe the basic model of hypothesis-pruning, we present some preliminaries in this section.

Given a data set $\mathcal{X}$ with binary class labels, and $\mathcal{D}$ is the distribution over $\mathcal{X} \times \{\pm 1\}$, we divide $\mathcal{X}$ into two groups: $\mathcal{L}$ and $\mathcal{U}$, in which $\mathcal{L}$ contains the labeled set of $\mathcal{X}$, and $\mathcal{U}$ contains the unlabeled set. Let $\mathrm{err}(\mathcal{X}, \mathcal{L})$ denote the error rate of predicting $\mathcal{X}$ by training the labeled set $\mathcal{L}$, $\{\hat{x}, -1\}$ and $\{\hat{x}, +1\}$ denote the queried data with negative or positive labels, $\hat{x} \in \mathcal{X}$, t denote the tth query, $\mathcal{L}_t$ denote the labeled set in the tth query, and k denote the total number of queries. We present the policy for querying by the error disagreement Δ_{t} [12],

$$\begin{aligned} &|\mathrm{err}(h_{-1}, \mathcal{L}_t \cup \{\hat{x}, -1\}) - \mathrm{err}(h_{+1}, \mathcal{L}_t \cup \{\hat{x}, +1\})| > \Delta_{\mathrm{t}} \\ &\text{s.t.} \quad \{\hat{x}, \pm 1\} \subset \mathcal{U}, t = 1, 2, 3, \cdots, k, \end{aligned} \tag{1}$$

where h_{-1} and h_{+1} denote the classification hypotheses after assuming $\hat{x}$ with a negative and positive label, respectively. By employing this policy, the active learners pick up those data whose error disagreements of $|\mathrm{err}(h_{-1}, \mathcal{L}_t \cup \{\hat{x}, -1\}) - \mathrm{err}(h_{+1}, \mathcal{L}_t \cup \{\hat{x}, +1\})|$ are larger than the given coefficient

Δ_t. If the error disagreement of one data is far greater than the coefficient, it updates the current classification hypothesis significantly. Otherwise, the influence on the current hypothesis of adding the data to $\mathcal{L}$ is insignificant.

The theoretical guarantees for this policy can be expressed in terms of the generalized disagreement coefficient [36] over a fixed assumption. Given a hypothesis class $\mathcal{H}$ over $\mathcal{X}$, let h^* be the optimal hypothesis which satisfies $h^* = \mathrm{arginf}_{\mathrm{h}\in\mathcal{H}}\, \mathrm{err}_\mathcal{D}(h)$, $\nu = \mathrm{err}_\mathcal{D}(h^*)$, and $h(x) \neq h^*(x)$, where $\mathrm{err}_\mathcal{D}(h)$ denotes the error of hypothesis h with respect to distribution $\mathcal{D}$. Let $B(h^*, r)$ [16] be a ball centered with h^*, given a radius r limits the volume of the candidate hypotheses around h^*, we define $B(h^*, r) = \{h' \in \mathcal{H} : \ell(h^*, h') < r\}$, where $\ell(\cdot,\cdot)$ denotes the metrical distance between the two hypotheses. Generally, $\ell(\cdot,\cdot)$ can be generalized as the error disagreement of Eq. (1). Assume there exists a descried error rate ϵ, the generalized disagreement coefficient [15] is defined as the minimum value of θ such that for any r:

$$\theta = \sup\left\{\frac{\mathrm{Pr}_{x\sim\mathcal{D}}[\exists h \in B(h^*, r)]}{r} : r \leq \epsilon + \nu\right\}, \qquad (2)$$

where Pr denotes the probability mass in $B(h^*, r)$ such as the candidate hypothesis disagreement or misclassified data amount.

The generalized types of Pr are typically used in various hypothesis-pruning AL: [22] presents an upper bound of label complexity using maximum disagreement between any hypothesis in $\mathcal{H}$; [15] tights this bound by using the best-in-class error disagreement, etc.

B. Sparse Hypothesis Class

To find the instance with the highest informativeness, the hypothesis-pruning AL algorithms using error disagreement select the data which maximally split $\mathcal{H}$, and then shrink the number of candidate hypothesis. However, the general error disagreements need a linear classifier or fixed distribution and it is only a special metric over hypothesis disagreement. In this section, we study the hypothesis distribution which is independent of the structural assumption of fixed distribution.

Without a given distribution, we assume the hypothesis class is distributed in an unseen graph structure $\mathbb{G}$ and each node denotes a hypothesis. Then, $B(h^*, r)$ denotes a ball centered with h^* and radius r in $\mathbb{G}$. Afterwards, finding a sparse hypothesis class is the most important splitting factor.

Let h_t be the current hypothesis, x_i and x_j be two candidate sampling points in $\mathcal{X}$. Assume h_{t,x_i} and h_{t,x_j} are the updated hypotheses after sampling x_i and x_j, respectively, the disagreement coefficient can be the infimum value of θ'

$$\max_{\mathrm{h}\in\mathrm{B}(\mathrm{h}^*,\mathrm{r})} \ell(h^*, h_{t,x_i}) + \ell(h^*, h_{t,x_j}) \leq 2\theta' r, \forall r > 0, \qquad (3)$$

where h_{t,x_i} is assumed to be the hypothesis with the maximum disagreement to h_{t,x_j} in a given radius setting r. In $B(h^*, r)$ of $\mathbb{G}$, h_{t,x_j} denotes the node which is the farthest from the node of h_{t,x_i}.

Let m be the number of unlabeled data in the candidate pool, the constrained hypothesis relationship set is descried as

$$\mathcal{H}' = \{(h_{t,x_1}, h'_{t,x_1}), (h_{t,x_2}, h'_{t,x_2}), ..., (h_{t,x_m}, h'_{t,x_m})\}, \quad (4)$$

where h'_{t,x_i} denotes the hypothesis that is the furthest from h_{t,x_i}. By employing the hypothesis disagreement function $\ell(\cdot,\cdot)$ of Eq. (3), learners can remove a part of the hypotheses by a margin distance θ'. Then, we obtain a sparse hypothesis class $\mathcal{H}^*$. With this splitting strategy, characterizing any hypothesis in $\mathcal{H}$ with a lower VC bound may be possible. Therefore, the key study of this paper is to prune the original hypothesis class into a sparse structure from the distribution view.

C. Performance Disagreement

Halfspace learning provides a clear visualization to describe the hypothesis relationship. Based on this advantage, in this section, we describe the performance disagreement of the hypothesis-pruning and distribution-shattering AL by halfspace learning. We firstly describe different cases of learning a halfspace over a unit sphere.

Case 1. Halfspace learning. Learning a halfspace c^* [37] [19] in a united sphere is to estimate an unknown vector μ that takes the sphere center as the start point,

$$c^* = \{x \in \mathbb{R} | \langle\mu, x\rangle \geq 0\}, \text{s.t. sign}(\langle x_i, \mu\rangle) \in \{+1, -1\}\,. \quad (5)$$

In this case study, the goal of halfspace learning is to estimate the optimal c^* using the lowest number of queries as possible. However, the label complexity of the unseen sampling process heavily depends on the initial hypothesis. Suppose that the points which could maximize the hypothesis or distribution update are the primary sampling data, we utilize label complexity to observe the difference of hypothesis-pruning and distribution-shattering AL of the halfspace.

To explain the notion of label complexity, we take the label complexity of the passive (random) learning of halfspace as prior knowledge.

Case 1.1. Passive learning of halfspace. Let $\mathcal{D}$ be the distribution over a unit sphere with $1/\epsilon$ data, then the label complexity of passive sampling is $\mathcal{O}(\frac{1}{\epsilon})$.

Let v_t be the vector classifier on the tth query, and θ_t be the angle between v_t and μ, we give the following case studies.

Case 1.2. Hypothesis-pruning AL of halfspace. Let $\mathcal{D}$ be the distribution over a unit sphere with $1/\epsilon$ data, the label complexity of obtaining a lower error rate compared to the initial hypothesis is $\mathcal{O}(\frac{\theta_t}{\pi\epsilon})$. Even using the halving algorithm, the label complexity is $\mathcal{O}(\log\frac{\theta_t}{\pi\epsilon})$.

To reduce the error of the initial hypothesis, we need to query the labels of the data distributed between v_t and μ (colored area in Fig. 1). Over a unit sphere with $1/\epsilon$ data, the candidate pool which can reduce the error of the initial hypothesis has $\frac{\theta_t}{\pi\epsilon}$ data. If we use the halving algorithm such as binary search in the candidate pool, the label complexity would be $\mathcal{O}(\log\frac{\theta_t}{\pi\epsilon})$. Different from the hypothesis-pruning

AL, the distribution-shattering AL that requires the unseen sampled data is independent of the initial hypothesis.

Case 1.3. Distribution-shattering AL of halfspace. Let $\mathcal{D}$ be the distribution over a unit sphere with $1/\epsilon$ data, the label complexity of obtaining a lower error rate compared to the initial hypothesis is $\mathcal{O}(1)$.

The above cases compare the sampling policies of the hypothesis-pruning and distribution-shattering AL algorithm over the unit sphere. The performance of the hypothesis-pruning AL strategy heavily depends on the initial hypothesis. In real-world AL tasks, the querying results of AL depend on the input labeled set and updating of the training model. For example, an limited labeled set and misguided model update will degenerate the performance of the subsequent sampling. However, the final estimation on error rate of the proposed distribution-shattering strategy depends on the representation structure of the input distribution. In simple terms, learning the representation structure of the distribution could help to address the limitation of hypothesis-pruning with a certain sampling selection. In a real AL task, the queried samples of any generalized distribution-shattering algorithm will be independent of the input training set.

IV. Hypothesis and Distribution

In Section IV.A, we firstly present the monotonic property of the active query set to show the uncertain error rate change after querying. Then, we discuss the bottleneck of informative AL and describe our splitting rule by representation sampling in Section IV.B. Finally, we discuss the relationship between error rate and number density of input distribution in Section IV.C.

Based on these theoretical analysis, we are motivated to undertake the splitting in input distribution. The goal is to eliminate the hypothesis supervision by learning the structure of the input distribution. Proofs are presented in Appendix.

A. Monotonic Property of the Active Query Set

To observe the error rate change after increasing the size of the active query set, we follow the perceptron training (see Fig. 1(a)) (Dasgupta et al., 2005) to analyze the hypothesis relationship. In our perspective, training the updated hypothesis will result in two uncertain situations: (1) the error rate declines after querying, and (2) the error rate shows negative (or slow) improvement when querying a lot of unlabeled data. Therefore, the monotonic property of the active query set size and error rate are unknown. The following proposition provides a mathematical description for this discovery.

Proposition 1. *The monotonic property of active query set and error rate is unsatisfied or negative. Suppose ϵ_t and ϵ_{t+1} respectively are the error rates of training the active query sets $\mathcal{D}_{\epsilon_t}$ and $\mathcal{D}_{\epsilon_{t+1}}$. There must hold an uncertain probability relationship which satisfies $\Pr(\epsilon_{t+1} \leq \epsilon_t | \mathcal{D}_{\epsilon_t} \subset \mathcal{D}_{\epsilon_{t+1}}) < 1$.*

Proposition 1 describes the first perspective of this paper about the relationship between the performance of the hypothesis and the active query set size. It shows that the probability of reducing the current error rate by increasing the size of the active query set is unpredictable and answers the question that we proposed in the beginning of this paper. In the following, we observe the error rate change by shattering the number density of the candidate pool.

B. Error Rate Change by Shattering Number Density

Following the perceptron training in the unit circle with uniform distribution, we find the error rate grows with the number density of the input distribution. This study also appears in AL of halfspace.

Proposition 2. *Assume $\theta_{t+1} > \theta_t$, we know $\mathrm{err}(\mathcal{D}_{\epsilon_t}) - \mathrm{err}(\mathcal{D}_{\epsilon_{t+1}}) = (\theta_{t+1} - \theta_t)\frac{\mathrm{Den}(B)}{n}$ (w.r.t. the volume of the circle is π), where $\mathrm{Den}(\cdot)$ denotes the number density of the distribution.*

Error rate disagreement denotes the distance between two arbitrary hypotheses. By observing the above propositions, we find that number density affects the hypothesis disagreements. Furthermore, we know the number density roughly decides the VC dimension bound of the optimal hypothesis since $\mathrm{Vcdim}(B) = \sum_{k=1}^{n} \binom{n}{k} = 2^n = 2^{\pi \mathrm{Den}(B)}$. For these two reasons, number density is a direct way to describe the hypothesis distribution in version space. Therefore, we are motivated to shatter the number density of the input distribution to both reduce the VC bound and find a lower label complexity. In addition, we define $\mathrm{Den}(B)$ for the real AL tasks in Section V.C. In the following, we discuss the bottleneck of querying informative samples and present our solution to this issue.

C. Bottleneck of Hypothesis-pruning

In hypothesis-pruning, the generalized algorithm updates the initial hypothesis w.r.t. h_0 into h_ϵ with a desired error ϵ in the original hypothesis class over version space (Fig. 2(a)). The informative samples are the primary querying targets. However, estimating the hypothesis disagreement is challenging. In particular, when the initial hypothesis is set improperly (far from the optimal hypothesis in version space), the path of finding the optimal hypothesis might be difficult. Thus, there exists a bottleneck for the AL sampling by querying informative samples, *i.e., the hypothesis disagreement from the initial hypothesis to the descried hypothesis is uncertain.*

Since the VC dimension greatly affects the path finding process for the optimal hypothesis, splitting the hypothesis class of version space into a sparse structure can alleviate the bottleneck of querying the informative samples. In our assumption, we use distribution-shattering to optimize a group of hypothesis spheres (Fig. 2(b)). Shattering by those sphere centers, the original hypotheses are transformed into a sparse hypothesis class, thereby finding h_ϵ can achieve a lower label complexity than in that original hypotheses (Fig. 2(c)). To implement this proposal, we perform the splitting idea on the input distribution by finding k local balls constrained by the following rules.

Solution. Given $B_{\mathcal{D}}$ is a ball which tightly encloses $\mathcal{D}$, and $\{B_1, B_2, ..., B_k\}$ are the k local split balls with the condition of $\forall i, B_i \subset \mathcal{D}$. Let $\mathrm{Vol}(\cdot), r(\cdot)$ define the volume and radius of the input hypothesis object, respectively. The splitting must satisfy the following conditions: (1) the volume of arbitrary split ball B_i is smaller than that of $B_{\mathcal{D}}$, i.e., $\forall B_i, \mathrm{Vol}(B_i) < \mathrm{Vol}(B_{\mathcal{D}})$, (2) the sum of the volumes of all the split balls B_i is smaller than that of $B_{\mathcal{D}}$, i.e., $\sum_{i=1}^{k} \mathrm{Vol}(B_i) < \mathrm{Vol}(B_{\mathcal{D}})$, (3) the radius of an arbitrary ball is smaller than the radius of $B_{\mathcal{D}}$, i.e., $\forall B_i, r(B_i) < r(B_{\mathcal{D}})$, and (4) the distance between any two local hypothesis balls is bigger than the sum of their radii, i.e., $\ell(c_i, c_j) > r(B_i) + r(B_j)$, where $\ell(\cdot,\cdot)$ denotes the distance between the two inputs, and c_i denotes the center of the ith split ball.

The above splitting rules provide an algorithmic paradigm for distribution-shattering strategy. A generalized algorithm termed SDAL is then presented in Section VI.C.

Remark 1. *The policy of $\ell(c_i, c_j) > r(B_i) + r(B_j)$ is the key of the theoretical solution that avoids overlapping in representations of local hypothesis spheres. It is generalized in the convergence condition w.r.t. Line 15 of SDAL algorithm.*

V. Distribution-shattering for Active Learning

Section V.A explains how to shatter the input distribution from halving to splitting. Using a heuristic greedy selection, we halve the number density of the input distribution to obtain a shattered distribution in Section V.B. Then, we discuss its theoretical advantages in Section V.C. With these guarantees, Section V.D splits the shattered distribution of the input distribution into a certain number of local balls to find a representation structure. Proofs are presented in Appendix.

A. Shattering: From Halving to Splitting

Shattering the input distribution is proposed to eliminate the dependence of the hypothesis. In the last section, halving the number density of the colored candidate pool yields exponential reduce on the label complexity of halfspace learning. To prove the positive help of shattering, we propose to implement the halving algorithm against the input distribution. The theoretical estimations on the generalized label complexity and error rate difference reveal the effectiveness of shattering. If all feasible change can converge uniformly with the shattering percentages, we split the shattered distribution into several representation regions and use their central points as the query samples of AL.

B. Halving Number Density for Shattered Distribution

By sorting the hypothesis disagreement of each pair in $\mathcal{H}'$ of Eq. (4), we use a splitting threshold θ' to halve the number density of the input space under arbitrary data distributions. The cutting rule is: let h_t is centered with its update $\ell(h_{t,x_i}$ on x_i, for any $x_j \in \mathcal{X}$, if $\ell(h_{t,x_i}, h_{t,x_j}) \geq \theta'$, we remove x_j from $\mathcal{X}$. After the cutting, $\mathcal{H}'$ will be reduced to $\mathcal{H}^*$ over a shattered distribution.

In hypothesis class $\mathcal{H}$, the VC dimensions of $\mathcal{H}$ and $\mathcal{H}^*$ can be written as $\mathrm{Vcdim}(\mathcal{H}) := d = \sum_{i=1}^{m} \binom{m}{i} = 2^m$ and $\mathrm{Vcdim}(\mathcal{H}^*) := d' = \sum_{i=1}^{m/2} \binom{m/2}{i} = \sqrt{2}^m$ [38]. Based on these assumptions, let us discuss the advantages of shattered distribution on label complexity and the upper bound of the querying.

Lemma 1. *Label complexity. Let each hypothesis hold for a probability at least $1-\delta$, the label complexity $m(\epsilon, \delta, \mathcal{H}^*)$ is*

$$m(\epsilon, \delta, \mathcal{H}^*) = \frac{64}{\epsilon^2}\left(\frac{1}{\sqrt{2}^{m-2}} \mathrm{In}\frac{12}{\epsilon}\right) + \mathrm{In}\left(\frac{4}{\delta}\right) < m(\epsilon, \delta, \mathcal{H}). \tag{6}$$

Lemma 2. *Upper bound of queries. Following [9], let us assume $0 < \epsilon < 1/2$, $< 0 < \delta < 1/2$, then the AL will make at most $2m(\epsilon, \delta'_{\mathcal{H}^*}, \mathcal{H}^*) < 2m(\epsilon, \delta'_{\mathcal{H}}, \mathcal{H})$ queries, where δ'_H is denoted as $\delta'_H = \frac{\delta}{N(\epsilon,\delta,H)^2+1}$.*

Based on the above discussion, we can easily observe that the values of the two properties of the hypothesis class of the shattered distribution are lower than that of the original hypothesis class since it characterizes any hypothesis with a lower bound on VC dimension.

C. Advantages of Shattered Distribution

To observe the advantages of the shattered distribution, we 1) analyze the bounds of error disagreements between the hypotheses with positive or negative labeling assumptions, 2) discuss the upper bound of the error rate by fall-back analysis which requires a change in different assumptions that can hold for the same algorithm, and 3) present the label complexities in η-bounded and v-adversarial noise conditions.

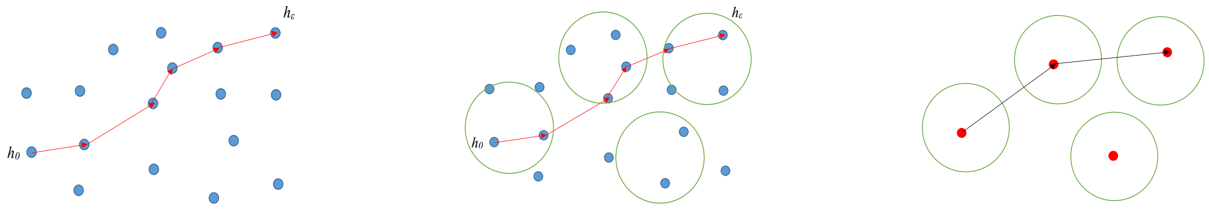
1) Bounds of Error Disagreement in Shattered Distribution: In this learning process, we continue to use the greedy strategy of halving to split the local unit ball $B(h^*, r)$. Before splitting, here we present the halving guarantees of error rate difference on the shattered distribution.

Theorem 1. *Let $\mathcal{D}'$ be the distribution over $\mathcal{H}^*$, $\{h_i, h'_i\} \in \mathcal{H}$, h'_i be furthest from h_i in $\mathcal{H}$, $\mathcal{F}$ be a family of functions $f : \mathcal{Z} \rightarrow \{0, 1\}$, $\mathcal{S}(\mathcal{H}, n)$ be the nth shatter coefficient with infinite VC dimension, $\alpha_t = \sqrt{(4/t)\mathrm{In}(8\mathcal{S}(\mathcal{H}, 2t)^2)/\delta}$, $\mathbb{E}_Z f$ be the empirical average of f over a subset $Z \subset \mathcal{Z} \subset \mathcal{X}$ with probability at least $1 - \delta$. Then, we have $\Delta' = (\mathrm{err}(h_i, \mathcal{D}') - \mathrm{err}(h_i, \mathcal{D})) - (\mathrm{err}(h'_i, \mathcal{D}') - \mathrm{err}(h'_i, \mathcal{D})) \leq 0$.*

Using this lemma, the error rate of the shattered distribution guarantees the decrease. However, it has a relationship with the size of $\mathcal{D}$. To obtain the structure of the version space, we continue to use the halving approach to split $\mathcal{H}$ into k local balls with a fall-back and bounded noises-tolerant guarantees.

2) Fall-back Analysis in Shattered Distribution: Fall-back analysis [12] helps us to observe the upper bound of error rate in the shattered distribution. Before analyzing the fall-back of querying, we need some technical lemmas.

Lemma 3. *[12] With an assumption of normalized uniform, Δ_t could be defined as: $\Delta_t := \beta_t^2 + \beta_t(\sqrt{\mathrm{err}_t(h_{+1})} + \sqrt{\mathrm{err}_t(h_{-1})})$, where $\beta_t = \sqrt{(4/n)\mathrm{In}(8(n^2+n)\mathcal{S}(\mathcal{H}^*, 2n)^2\delta)}$.*



(a) Hypothesis-pruning (b) Local hypothesis spheres (c) Sparse hypothesis class

Fig. 2. The assumption of distribution-shattering with a sparse hypothesis class. Each node denotes one realizable hypothesis, and the lengths of the red lines denote the hypothesis disagreement. Hypothesis-pruning updates the initial hypothesis w.r.t. h_0 into h_ϵ with a desired error ϵ in the original hypothesis class (Fig. 2(a)). Distribution-shattering optimizes a group of local hypothesis spheres (Fig. 2(b)). Shattering by those sphere centers, the original hypotheses are transformed into a sparse hypothesis class (Fig. 2(c), thereby finding h_ϵ can achieve a lower label complexity than in its original hypotheses.

Lemma 4. *With the assumptions of* $\mathrm{err}_t(h_{+1}) - \mathrm{err}_t(h_{-1}) > \Delta_t$, $\mathrm{err}_t(h_{+1}) - \mathrm{err}_t(h_{-1}) > \frac{2\beta_t^2}{1-\beta_t}$ *and it is consistent with the labeled set* $\mathcal{L}_t$ *for all* $t \geq 0$.

With Lemma 4, we then produce the upper bound of error of sampling in a shattered distribution.

Theorem 2. *Assume there exists a hypothesis* h_f *which satisfies* $\mathrm{err}_{D'}(h_f) \leq \mathrm{err}_D(h^*)$. *If the AL algorithm is given by* k *queries with probability of* $1-\delta$, *let* $\nu = \mathrm{err}_{D'}(h^*)$, *the error rate of shattered distribution is at most* $(\sqrt{\nu} + \beta_k)^2$.

From the above analysis, sampling in a shattered distribution can still converge safely. The upper bound of error of sampling in a shattered distribution is further proven to be tighter than sampling in the input distribution without halving. It shows sampling in a shattered distribution may save sampling consumption and a continuous splitting algorithm may further reduce this bound uniformly. Next, let us analyze the bounds of the label complexity in the noise settings.

3) Bounded Noise Analysis of Shattered Distribution: Under the uniform assumption, noises affect the unseen queries. Here we discuss the label complexities of the shattered distribution in η-*bounded* and v-*adversarial* noise settings [13].

Theorem 3. *For some* $\eta \in [0, 1/2]$ *with respect to* μ *(w.r.t. Definition 1), if for any* $x_i \in D'$, $\Pr[Y \neq \mathrm{sign}(\mu \cdot x_i)|X = x_i] \leq \eta$, *we say the distribution of* D' *is* η*-bounded [39]. Under this assumption, (1) there are at most* $\widetilde{\mathcal{O}}(\frac{d'}{(1-2\eta)^3\epsilon})$ *unlabeled data, and (2) the number of queries is at most* $\widetilde{\mathcal{O}}(\frac{d'}{(1-2\eta)^2}\mathrm{In}\frac{1}{2\epsilon})$, *where* $\widetilde{\mathcal{O}}(f(\cdot)) := \mathcal{O}(f(\cdot)\mathrm{ln}f(\cdot))$.

Theorem 4. *For some* $v \in [0, 1]$ *with respect to* μ, *if for any* $x_i \in D'$, $\Pr[Y \neq \mathrm{sign}(\mu \cdot x_i)|X = x_i] \leq v$, *we say the distribution of* D' *is* v*-adversarial noise condition [40]. Under this assumption, (1) there are at most* $\widetilde{\mathcal{O}}(\frac{d'}{2\epsilon})$ *unlabeled data, and (2) the number of queries is at most* $\widetilde{\mathcal{O}}(d'\mathrm{In}\frac{1}{2\epsilon})$.

Compared to the original input distribution, the shattered distribution has lower label complexity since the VC bound of any hypothesis is shattered into a shaper value.

D. Distribution-shattering for AL Tasks

Shattered distribution provides theoretical advantages without special distribution assumptions since number density is independent of arbitrary distribution situation. Therefore, in real-world AL tasks, we firstly halve the number density of the input distribution to learn a shattered distribution via an active scoring strategy. After obtaining the shattered distribution, we split the shattered distribution into k balls via the distribution density. Then, we propose the SDAL algorithm for AL querying.

1) Active Scoring for Halving: Active scoring is used to measure the local representativeness of arbitrary data, in which the score value monotonically grows with the representativeness. By removing some data with the lowest representativeness (i.e., halving the number density of the input distribution), we try to shatter the unlabeled data pool. This reduces the label complexity of the subsequent AL sampling. Here we use the experimental design [30] to finish the operation of halving.

Considering a linear function $f(x) = \mathbf{w}^T x$ from measurements $y_i = \mathbf{w}^T x_i + \xi_i$, where $w \in \mathbb{R}$, and $\xi_i \sim \mathcal{N}(0, \sigma^2)$. The halving algorithm is to optimize a set of $\mathbf{V} = \{(v_1, y_1), (v_2, y_2), ..., (v_m, y_m)\}$ to represent x, where $m = \lfloor n/2 \rfloor$. Therefore, the maximum likelihood estimate of $\mathbf{w}$ is obtained by

$$\underset{\mathbf{w}^*}{\mathrm{argmin}} \left\{ \mathcal{J}(\mathbf{w}) = \sum_{i=1}^{n} (\mathbf{w}^T v_i - y_i) \right\} \tag{7}$$

and the error rate is $e = w - \mathbf{w}^*$, s.t. $\mu(e) = 0, D(e) = \sigma^2 \mathbf{C}_w$, where $\mu(\cdot)$ denotes the mean value of the input variable, $D(\cdot)$ denotes the covariance matrix of the input object, and

$$\mathbf{C}_w = \left(\frac{\partial^2 \mathcal{J}}{\partial \mathbf{w}\mathbf{w}^T} \right)^{-1} = (\mathbf{V}\mathbf{V}^T)^{-1}. \tag{8}$$

Then the average expected square predictive error over $\mathcal{X}$ can be written as

$$\mathrm{E}(y_i - w^*Tx_i) = \sigma^2 + \sigma^2 \mathbf{Tr}(\mathcal{X}^T \mathbf{V}\mathbf{V}^T \mathcal{X}). \tag{9}$$

In order to minimize the average expected square predictive error, we need to minimize $\mathbf{Tr}(\mathcal{X}^T \mathbf{V}\mathbf{V}^T \mathcal{X})$. With mathematical derivations, the minimization issue changes into:

$$\begin{gathered} \underset{\mathbf{V},\mathbf{A}}{\mathrm{argmin}} \sum_{i=1}^{n} ||x_i - \mathbf{V}^T \alpha_i|| + \mu ||\alpha_i||, \\ \mathbf{V} \subset \mathcal{X}, \mathbf{A} = [\alpha_1, \alpha_2, ..., \alpha_n], \end{gathered} \tag{10}$$

where u is the penalty factor of the global optimization.

After mapping the original input space into a non-linear kernel space, we iteratively project the top-($\lfloor n/2 \rfloor$) data with the highest confidence scores to a shattered space[3]. To solve this equation, [30] uses sequential optimization to iteratively select the data with high representativeness in kernel space. In this paper, we follow their results and use the confidence score function to define the representativeness of one data:

$$\begin{aligned} \text{Score}(x_i) &= \frac{||K(\kappa,:)K(:,\kappa)||^2}{K(\kappa,\kappa)+u}, \forall i, \\ \text{s.t. } K &= K - \frac{K(:,\kappa')K(\kappa',:)}{K(\kappa',\kappa')+u}, \end{aligned} \tag{11}$$

where K denotes the kernel matrix of $\mathcal{X}$, κ denotes the sequence position of x_i in $\mathcal{X}$, and κ' denotes the sequence position of the data with the current highest confidence score in $\mathcal{X}$. Generally, sequential optimization costs a time calculation of $O(n^2)$ with a greedy strategy. For a large-scale data set, we can adopt the kernel relevant component analysis trick [41] to reduce the calculation complexity.

2) Splitting by Distribution Density: Implementing splitting in the input distribution by number density has already been proved effective in agnostic distributions (unknown assumptions). However, in $\hat{d}$-dimensional space, calculating the number density of a high dimensional-bounded space is challenging. To approximately generalize number density, we propose to use the exponential value of the distribution density to quickly split the input distribution due to their positive proportional relationship. Here we nearly generalize the number density as

$$\text{Den}(B_i) = \frac{1}{m_i} \sum_{x_j, x_l \in B_i} f^{\hat{d}}(x_j, x_l, h), \tag{12}$$

where $f^{\hat{d}}(\cdot)$ denotes the exponential value of the distribution density, $f(\cdot)$ can be generalized by arbitrary kernel function $\mathcal{K}(\cdot)$ with a bandwidth setting of $\mathcal{K}(\frac{x_j - x_l}{h})$, h denotes the kernel bandwidth, and m_i denotes the data number in B_i. Then, we propose the splitting rule:

$$\min \sum_{B_1, B_2, \ldots, B_k} \sum_{x_j, x_l \in B_i} \frac{1}{m_i} f^{\hat{d}}(x_j, x_l, h). \tag{13}$$

To solve the above minimum optimization problem, we use the (1+ε)-approximation [42] approach to increase the ball radius to make it converge, where ε is set by the empirical threshold.

3) Querying by SDAL: How to query unlabeled data is an important step for AL tasks. In this section, we propose a Shattered Distribution-based AL algorithm (SDAL) to implement the proposed distribution-shattering strategy by following the splitting rule in Section IV.B. The algorithm has two steps. Step 1 (Lines 2 to 10) is to find a shattered distribution which contains the optimal data sequences by the active scoring using Eq. (11). Step 2 (Line 11 to 25) is to solve the optimization of Eq. (13). Finally, the output data of the algorithm are used as the AL queries.

Algorithm 1: SDAL algorithm

```
1  Input: dataset $\mathcal{X}$, radius $r$, approximation ratio $\varepsilon$, number of epochs $T$.
2  while $l = 1 < \lceil n/2 \rceil$ do
3      for $i$=1,2,3...,$n$ do
4          Calculate the score of $x_i$: $\Omega(i) = \frac{||K(\kappa,:)K(:,\kappa)||^2}{K(\kappa,\kappa)+u}$.
5      end
6      Find the sequence $\kappa'$ with the maximum value in $\Omega$:
         $\kappa' = \underset{\text{i}}{\text{argmax}}\ \Omega(i)$.
7      Add $x_i$ to $\mathcal{X}^*$.
8      Update matrix $K = K - \frac{K(:,\kappa')K(\kappa',:)}{K(\kappa',\kappa')+u}$.
9      $l = l + 1$.
10 end
11 Initialize $k$ data points as the ball centers from $\mathcal{X}^*$ using $k$-means.
12 $f_0 = \sum_{B_1,B_2,\ldots,B_k} \sum_{x_j,x_l \in B_i} \frac{1}{m_i} f^{\hat{d}}(x_j, x_l, h)$.
13 while $t = 1 \leq T$ do
14     $f_t = \sum_{B_1,B_2,\ldots,B_k} \sum_{x_j,x_l \in B_i} \frac{1}{m_i} f^{\hat{d}}(x_j, x_l, h)$
15     if $f_t - f_{t-1} \to 0 \Big| \Big| ||c_i - c_j||_2 \leq 2r, \exists i, j$ then
16         break;
17     else
18         Update ball centers $\{c_1, c_2, c_3, \ldots, c_k\}$, where
             $c_i = \frac{1}{m_i} \sum_{x_j \in B_i} x_j$.
19         Update ball radius $r = r(1 + \varepsilon)$.
20         Update $\{B_1, B_2, \ldots, B_k\}$ by new radius setting.
21     end
22     end
23     $t = t + 1$.
24 end
25 Update $c_i$ by their nearest neighbor in $B_i, \forall i < k$.
26 Output: $\{c_1, c_2, c_3, \ldots, c_k\}$.
```

The detailed process is as follows. Lines 2 to 10 iteratively halve the number density of input data set $\mathcal{X}$ by removing a half of the data. The remaining data $\mathcal{X}^*$ with high representativeness denote the data of shattered distribution of $\mathcal{X}$. It reduces the label complexity for the subsequent sampling. In the $(1+\varepsilon)$-approximation, Lines 11 and 12 firstly initialize k balls with the input radius setting. The approximation converges when the balls overlap or the splitting function stops updating (see Line 15). Otherwise, Lines 18 to 20 iteratively update the centers, balls, and radius.

VI. Evaluation and Experiments

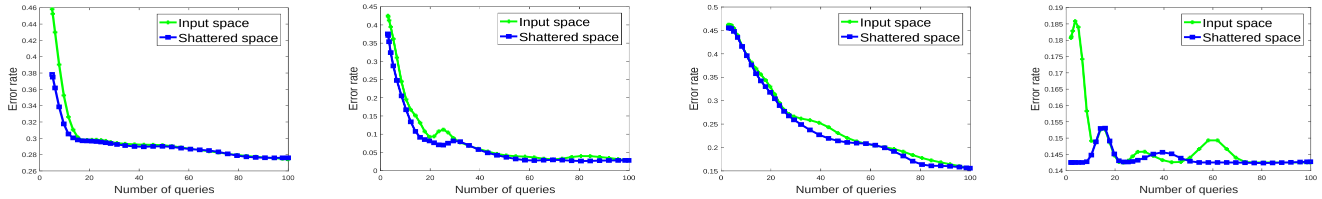
In this section, we investigate the halving and querying performance of the SDAL algorithm on three groups of experiments:

1) comparing the error rates of passive sampling in input and shattered spaces;
2) comparing the optimal error rates of different baselines;
3) comparing the average error rates of different baselines on six real-world datasets, where the datasets used in the querying tests have limited labels.

To defend our theoretical insights on the performance disagreement of hypothesis-pruning and distribution-shattering strategies, we compare their error performance on querying with adversarial examples and noisy labels. In these experiments, the LIBSVM(3.22 version) [43] and convolutional neural network (CNN) are set as the default classification tools. The error rate and mean±std are used as evaluation standards, where error rate is over the entire input set.

There exists two main steps in SDAL algorithm: halving and splitting. In step 1, halving introduces the sequential optimization to score the representativeness of the data, which

[3] Shattered space is a generalization from shattered distribution in real-world.



(a) german (b) iris (c) monk1 (d) vote

Fig. 3. Error rate changes of undertaking passive sampling in input and shattered spaces on different datasets.

relates transductive experimental design (TED). In step 2, splitting uses (1+ε)-approximation to find a group of representative spheres, which is related to Hierarchical clustering-based AL algorithm. We thus select these two approaches as our baselines. GEN is a comprehensive approach that introduces the representative measure in the process of estimating the hypothesis update. It is different with traditional estimation methods. Self-Paced AL is a generalization of hypothesis-pruning that estimates the hypothesis update with error loss and representativeness. Besides this, we present two generalizations of the k-means clustering approaches with different label estimation schemes. The details of these algorithms are described as follows.

- Hiera(Hierarchical clustering-based AL):[44] utilizes the prior knowledge of hierarchical clustering to actively annotate more unlabeled data by an established probability evaluation model, but it is sensitive to cluster structure.
- TED(Transductive Experimental Design): [30] prefers data points that are not only hard to predict but also representative for the rest of the unlabeled pool. It is also called T-optimization.
- GEN(a GENeral active learning framework): [33] pays attention to the data which minimizes the difference between the distribution of the labeled and unlabeled sets.
- k-meansN: updates the final k-means cluster centers into their Nearest neighbors and then queries the labels.
- k-meansA: estimates the label of each final k-means cluster center by rounding the Average label value of its cluster members.
- Self-Paced(Self-Paced active learning): [45] optimizes the least squared loss and maximum mean discrepancy for finding an instance with informativeness and representativeness.
- SDAL(Shattered Distribution-based Active Learning algorithm): the proposed algorithm in this paper.

Note all features of the input data are rescaled into [0,1] before the experiments.

A. *Effectiveness of Halving*

To verify the halving ability of SDAL, we undertake passive sampling in input and shattered spaces to compare their prediction abilities over the input data. The tested datasets are four UCI real datasets: german (1,000 examples), iris (150 examples), monk1 (124 examples), and vote (435 examples). In the experimental process, we undertake passive sampling 10 times to obtain the mean error rate under different querying numbers in the two different spaces. Fig. 3 presents the test results, where LIBSVM follows a parameter setting of [-c 1].

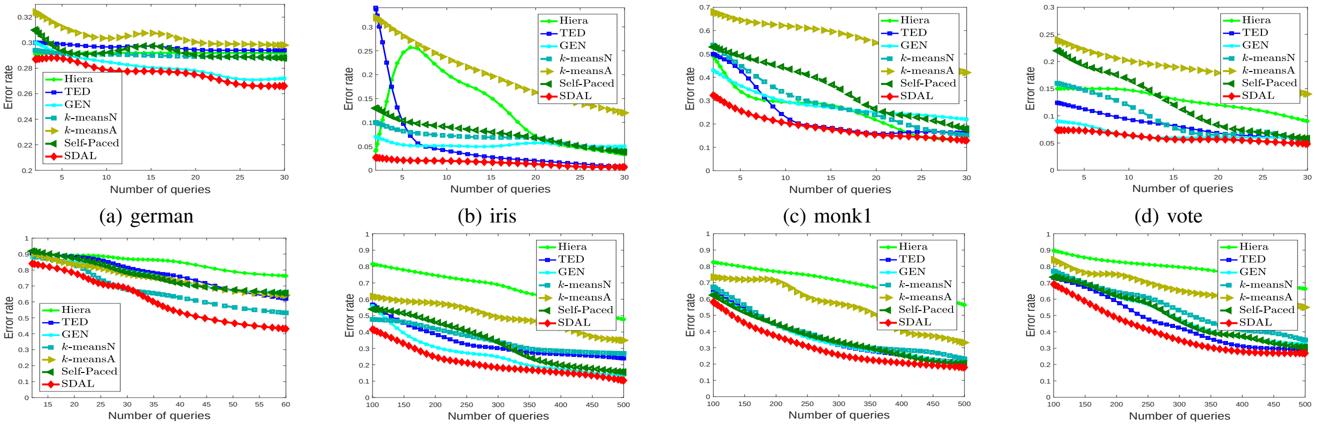
Shattering removes some "low informative points" deriving small influence to training model, thereby querying in a shattered space always has lower error rates than that of the original input space as learning curves in Fig. 3. Assume that there exists p "highly informative points" that determine the final learning model in the input space, with a limited sampling budget k, do not consider the influences of the classifiers and parameter settings, the probabilities of obtaining a descried hypothesis in the two spaces are $\Pr(\mathcal{D}) = \frac{\binom{k}{p}}{\binom{k}{n}}$ and $\Pr(\mathcal{D}') = \frac{\binom{k}{p'}}{\binom{k}{n/2}}$, respectively, where p' denotes the number of the highly-informative points in the shattered space. If $p - p'$ is small enough, $\Pr(\mathcal{D}) < \Pr(\mathcal{D}')$ must hold.

B. *Optimal Error of Querying*

The experiments on halving have shown that the shattered space could have a better passive sampling performance compared to the original input space. It provides a guarantee for performing AL querying by the distribution-shattering strategy in a shattered space. However, most of the AL work require the supervision from a labeled set. To run these hypothesis-pruning algorithms in a warm start, we set the size of the initial training set as the class category via randomly selecting one datum from each class of the input datasets. Because these AL algorithms always show negative performance when the start labeled set is insufficient, we minimize the influence of the labeled set by tunning their best parameters (related tunning is described in Section V.B). Under different settings on the querying numbers, we collected the their optimal prediction results by initializing the start labeled set 100 times.

Fig. 4 presents the error rate curves of the five AL approaches on different tested datasets these being german, iris, monk1, vote, and four subsets of the *letter* data set. Note that A-T denotes the instances of letter A to T. The classifier toolbox is LIBSVM that follows a parameter setting of [-c 1]. Although we have maximized the model performance of the hypothesis-pruning AL algorithms, the SDAL algorithm is still better than others in terms of optimal error.

To analyze the paradigm differences of these algorithms, we begin the discussions: (1) The idea of Hieral is active annotation. It depends on the cluster assumption from version space. Classification ability of it in unstructured datasets such as the subsets of *letter* thus is unstable. This makes the recorded



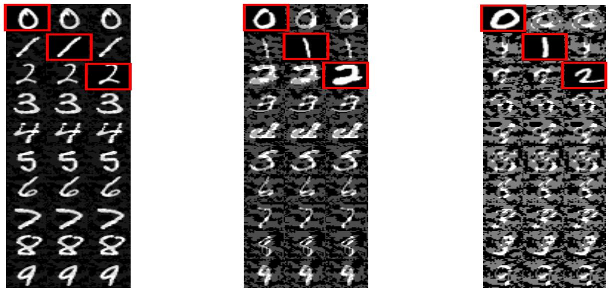
(a) german (b) iris (c) monk1 (d) vote

(e) A-T (f) A-L (g) A-P (h) A-Z

Fig. 4. The error rate performances of the seven AL approaches on the active learning test. (a)-(d) are four UCI datasets; (e)-(k) are that on the selected sub datasets of *letter*, where the class number of them are 12, 16, 20, and 26.

Table I
THE STATISTICAL RESULTS (MEAN±STD IN %) OF ERROR OF DIFFERENT AL BASELINES ON SIX REAL-WORLD DATASETS

Datasets	Class Number	Algorithms	Number of queries								
			100	200	300	400	500	600	700	800	900
Phishing	2	Hiera	49.6±3.0	45.0±7.2	42.5±2.3	38.5±1.7	33.2±2.1	22.6±1.4	19.3±1.2	18.0±1.1	14.6±0.7
		TED	39.0±1.9	39.1±0.9	34.9±0.3	34.1±0.6	31.2±0.7	28.5±0.5	27.9±0.5	18.6±0.5	13.8±0.8
		GEN	47.4±3.6	45.4±2.2	38.6±3.8	32.8±2.9	31.7±2.1	22.6±3.2	19.8±3.6	16.8±2.1	14.5±2.2
		k-meansN	37.0±0.1	**36.1±1.2**	34.9±0.1	32.1±0.1	30.2±0.5	27.5±0.0	25.9±0.4	16.6±0.3	14.8±0.7
		k-meansA	58.0±0.6	56.3±0.8	52.2±0.1	52.1±1.1	50.7±1.0	47.6±1.2	45.4±1.5	42.1±0.8	40.2±0.4
		Self-Paced	47.4±2.7	44.4±2.8	41.2±2.8	36.8±3.4	35.7±2.8	35.6±2.9	22.8±2.8	17.8±3.2	15.5±2.8
		SDAL	**36.5±1.7**	36.5±1.2	**30.4±2.8**	**30.0±2.8**	**27.4±2.2**	**19.1±1.7**	**16.5±1.7**	**12.6±1.8**	**11.2±2.1**
Satimage	6	Hiera	22.1±0.9	19.9±0.6	18.9±1.1	18.5±1.0	18.4±1.1	18.4±2.3	17.3±1.2	16.7±1.0	15.9±0.6
		TED	20.4±0.7	19.6±0.1	18.4±0.4	17.8±0.6	17.6±0.2	17.4±0.2	17.2±0.1	16.8±0.3	16.0±0.1
		GEN	21.9±4.0	20.1±2.7	18.5±0.5	18.4±0.3	18.1±1.2	18.0±1.5	17.8±0.7	16.5±0.9	16.4±2.0
		k-meansN	22.2±1.1	22.0±0.1	19.8±0.7	18.7±1.2	18.2±0.9	17.5±0.8	17.0±0.7	17.1±0.0	17.2±0.1
		k-meansA	34.2±1.0	32.0±1.1	30.8±1.2	28.5±1.2	26.3±1.1	25.4±0.0	22.8±0.8	19.6±1.0	19.2±0.9
		Self-Paced	24.7±2.8	23.1±2.7	20.5±0.5	18.2±0.3	17.3±1.2	17.8±1.5	17.1±0.7	16.4±0.9	16.2±2.0
		SDAL	**18.4±1.5**	**17.5±1.2**	**17.4±2.3**	**16.8±0.1**	**16.4±1.3**	**15.1±2.2**	**14.9±1.2**	**14.1±1.3**	**14.1±1.9**
MNIST	10	Hiera	51.2±2.7	46.0±1.7	37.3±2.4	21.3±2.8	20.1±2.3	11.9±2.2	9.6±1.5	9.3±1.3	9.0±1.0
		TED	63.3±1.2	40.7±2.3	21.5±3.2	21.7±0.8	**8.9±0.5**	8.3±0.9	8.3±0.2	8.2±0.5	7.8±0.6
		GEN	57.3±5.7	50.7±1.9	30.1±1.6	20.7±1.3	14.9±1.6	11.0±0.6	9.2±0.6	8.1±1.4	8.0±0.1
		k-meansN	65.7±0.7	52.7±1.3	32.4±1.2	29.8±0.2	16.4±0.3	12.5±0.2	11.6±0.1	10.7±0.1	7.8±0.4
		k-meansA	82.6±1.2	75.4±1.1	64.4±0.8	57.8±0.6	46.7±0.6	42.2±0.4	34.7±0.2	28.9±0.4	26.3±0.5
		Self-Paced	78.6±4.3	54.8±2.7	42.2±3.2	35.5±2.1	26.4±2.3	22.6±3.7	10.6±1.9	9.3±1.7	9.4±0.9
		SDAL	**44.2±2.4**	**37.0±2.8**	**19.8±3.8**	**11.5±1.9**	9.0±1.2	**8.1±0.9**	**7.9±0.8**	**7.7±0.6**	**7.6±0.3**



(a) ε=0.1 (b) ε=0.3 (c) ε=0.5

Fig. 5. Illustration of the produced adversarial examples by FGSM with different perturbation parameters, where the marked examples are clean data without feature perturbations.

error rates of Hieral be higher than that of other approaches, although we have increased the test number. Moreover, active annotation has a negative influence on the subsequent querying once the clustering result is not correct as its error rate curves in Fig. 4(a). In other words, actively annotating the labels of a given budget have to undertake the positive or negative influences of pre-clustering. (2) TED tends to select those points with large norms, which might be hard to predict, but they do not best represent the whole data set. Also, the noises or low informative data are sampled in its querying process. So the reported classification results are good but not the best. (3) GEN always presents disappointing results at the beginning of training in all the tested datasets. Its error rate declines rapidly with the increase of the number of queries. The reason is that the established objective function prefers the data located at the center area of classes, which does not reflect the whole class structure well. (4) The performance of k-meansN is at middle level amongst all compared baselines because of the intuitive cluster structures of the tested datasets. While the error rate cannot decrease rapidly as other baselines. Besides this, the performance of k-meansA presents the worst performance of this group of experiments since averaging the labels of the cluster members cannot provide a correct estimation. (5) The performance of Self-Paced AL optimizes the hypothesis update with a constrain on distribution representation. When the initial hypothesis is set improperly, the update will lead to an biased selection or random index. Thus, the performance of it is similar with GEN. (6) Compared to the above algorithms, the SDAL algorithm halves the number density of the data distribution into a shattered distribution, which removes most of the redundant points. The remaining points, which represent the local data distributions, help the learner to obtain the structure of the original data distribution. In the reported error rate curves, this represented structure shows effective sampling guidance when the number of queries is insufficient.

C. Average Error of Querying

The optima error of querying reflects the best sampling performance of different AL algorithms. To tightly analyze their performance dependency on the initialized hypothesis (labeled set), this section presents their average error rates on three UCI datasets namely Phishing (11,055 examples), Satimage (4,435 examples), and one handwritten digit dataset MNIST (60,000 examples). [4] Parameter settings are:1) vary the pruning budget of Hieral from 100 to 1000 with a step of 100; 2) kernel bandwidth parameter of TED is set as σ=1.8, then vary the kernel ridge regression λ from 0.01 to 1 with a step of 0.01; 3) vary the trade-off parameter of Self-Paced from 1 to 1000 with a step of 10; 4) vary the paced learning parameter from 0.01 to 1 with a step of 0.01; 4) number of queries are set as the clustering number of k-meansA and k-meansN; 5) for SDAL, the used kernel in the sequential optimization is RBF, where the hyper parameter h is set as 1.8, and the hyper parameter μ is set as $10e$-4, we then vary the ball radii from 0.01 to 0.51 with a step of 0.05 and ε from 0.01 to 0.51 with a step of 0.05. To run Hieral, TED, GEN and Self-Paced algorithms, we select one datum with label from each class of the six datasets respectively as their initialized labeled sets. The classifier toolbox is LIBSVM with following parameter settings: 1) [-c 1 -g 25] for $N \leq 600$, [-c 1 -g 20] for $N > 600$ on Phishing, 2) [-c 1] on Satimage, 3) [-c 2] for $N \leq 300$ and [-c 4 -g 0.0015 -r 91.1 0.001] for $N > 300$ on MNIST, where N denotes the number of queries. The diver settings make the derived errors of AL process decrease slowly but finally achieve the optimal; better observation on learning changes by adding perturbations of classifier. The mean and standard deviation (std) errors of the that algorithms on these datasets are reported in Table I with the results showing that SDAL significantly outperforms the others indicated in bold.

As shown in Table I, (1) on all settings of the querying numbers, the SDAL algorithm achieves the lowest error rates over other baselines; (2) with the experience setting on parameters, all baselines achieve an average error below 0.5 after querying 300 data from the unlabeled data; (3) the SDAL algorithm produces significantly less errors when the numbers of querying are less than 600, benefiting from the representative structure of the input space; (4) for Hieral, TED, GEN and Self-Paced, the initial selection of the labeled set greatly affects their subsequent sampling; (5) on all settings, all algorithms obtain an average error below 0.3 after querying 600 data from the unlabeled data; (6) with an increase of the querying percentages, the differences between each algorithm begin to narrow since the number of their overlapped data increases. Therefore, we conclude that our proposed SDAL algorithm, an approach derived from distribution-shattering strategy, breaks the curse of the initial hypothesis.

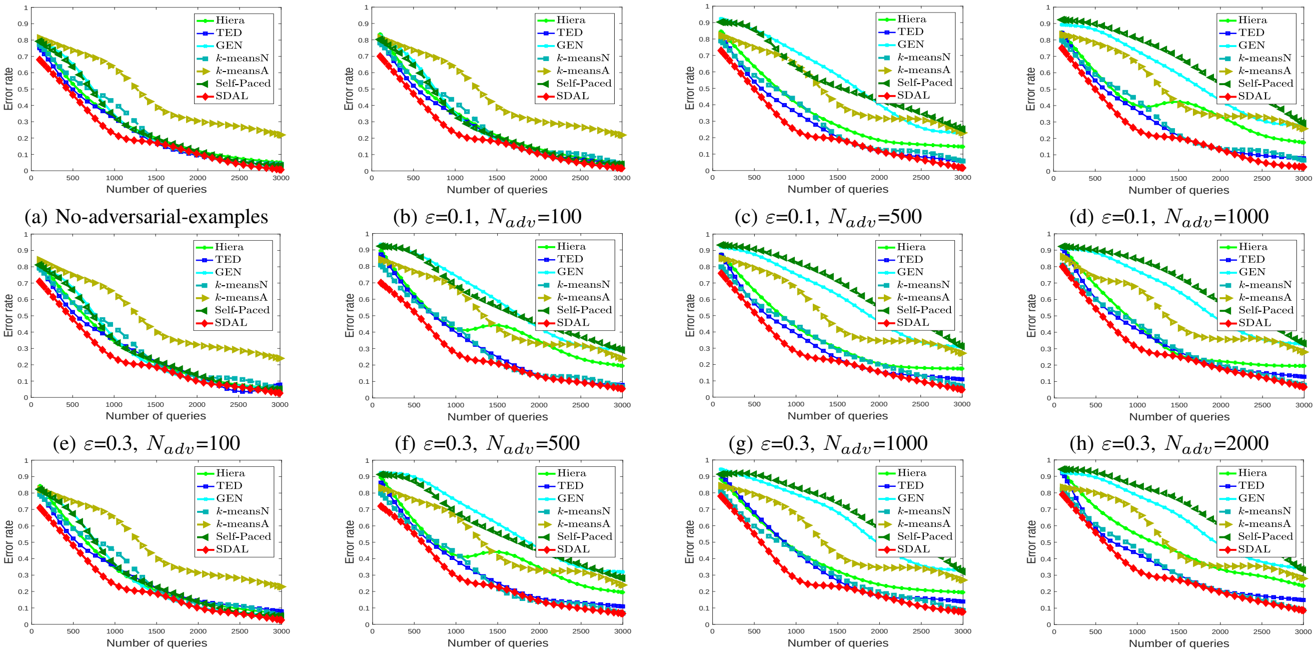
D. Querying with Adversarial Examples

In the machine learning community, the training models may misclassify the adversarial examples [46] generated from the distribution of the correctly classified examples. The degradation of the performance in supervision training, caused by adversarial examples, is already not a mystery: the adversarial perturbation affects the precision of the features. In particular, the linear models are vulnerable to adversarial perturbation, such as regression and SVM models. In our study, the general hypothesis-pruning AL strategies which need the support of the classifiers preferably pick up the adversarial examples. The underlying reason is that the adversarial examples make disagreement between the current and subsequent models more obvious than the examples without perturbation (clear data). Therefore, active querying with adversarial examples significantly describes the performance disagreement of hypothesis-pruning and distribution-shattering AL strategies, and further defends our theoretical insights.

The experiments are tested on the MNIST dataset and we respectively generate 9,000 adversarial samples by the Fast Gradient Sign Method (FGSM) [46] attack method under different perturbation parameter ε: 0.1, 0.3, 0.5. For each parameter, such as ε= 0.1, we randomly choose 1,000 legitimate images from the MNIST test dataset, and 100 images for each class. For each image, we generate 9 adversarial samples with different labels. For example, for an image with label 0, we generate 9 adversarial samples with labels 1 to 9. Fig. 5 presents a group of illustrations of adversarial examples, where each illustration marks three clean examples. To intuitively observe the influence of the adversarial examples in AL querying, we use the 9,000 examples with ground truth labels as the unlabeled set of AL and the 9,000 data with misclassified labels as the adversarial set. The features are extracted by the LeNet model and the classification model is CNN. To accelerate the experiments, we adjust the umber of epochs: 1) epoch=1 for $N{\leq}2000$, 2) epoch=5 for $2000{\leq}N{<}2500$, 3) epoch=20 for $N{>}2500$, where N denotes the number of queries. This way defers the decrease of error rate that benefits the observation on the influences of subsequent perturbations from adversarial examples. Parameters of baselines follow their best tunning in Section VI.C.

In a dynamic view, we add a different number of adversarial examples to see the error change of different algorithms in the querying process. Before the querying test, we randomly select 20 data from the training set as the initial (start) labeled set for the hypothesis-pruning AL algorithms including Hiera, TED, GEN and Self-Paced. Fig. 6 draws the error rate change of predicting the labels of the entire training set under different settings on the perturbation parameter, number of added adversarial examples (N_{adv}), and the number of queries.

[4] http://yann.lecun.com/exdb/mnist/

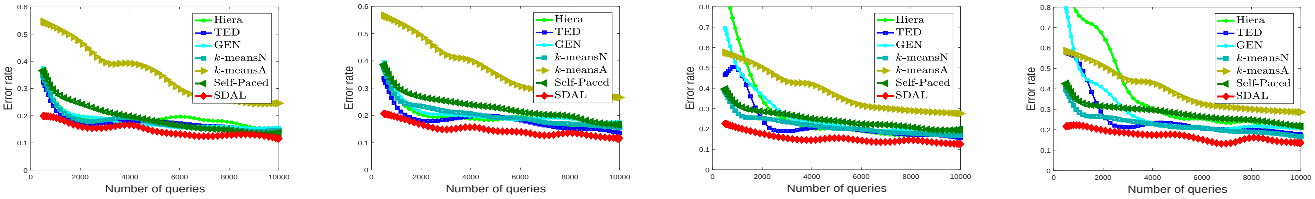


(a) No-adversarial-examples (b) ε=0.1, N_{adv}=100 (c) ε=0.1, N_{adv}=500 (d) ε=0.1, N_{adv}=1000

(e) ε=0.3, N_{adv}=100 (f) ε=0.3, N_{adv}=500 (g) ε=0.3, N_{adv}=1000 (h) ε=0.3, N_{adv}=2000

(i) ε=0.5, N_{adv}=100 (j) ε=0.5, N_{adv}=500 (k) ε=0.5, N_{adv}=1000 (l) ε=0.5, N_{adv}=2000

Fig. 6. The performance of error rate on AL querying with adversarial examples, where the adversarial examples are produced by FSGM with different perturbation parameter settings.



(a) N_{noi}=0 (b) N_{noi}=100 (c) N_{noi}=500 (d) N_{noi}=1000

Fig. 7. The performance of error rate on AL querying with noisy labels.

With the dynamic views on Fig. 6(a) to 6(d), 6(e) to 6(h), and 6(i) to 6(l), we find that the three hypothesis-pruning AL algorithms significantly degenerate their error rates. Because the added adversarial examples misclassify the classifier using fraudulent labels, they significantly affect the update of the training model largely. By mixing more adversarial examples into the training set, the current training model has a greater chance to select the adversarial examples. However, our proposed SDAL algorithm which utilizes the distribution-shattering strategy is not sensitive to the classifier. Thus, its error rates only slightly reduce when querying the same number of unlabeled data, even adding more adversarial examples. In another view of setting different perturbation parameters, i.e., from the comparison of Fig. {6(b), 6(e), 6(i)}, {6(c), 6(f), 6(j)}, {6(d), 6(g), 6(k)}, and {6(h), 6(l)}, we find the error rates of these hypothesis-pruning AL algorithms also reduce significantly with an increase of ε.

To tighten the above analysis, Table II calculates the mean and std values of querying 1,000 legitimates when varying the number of adversarial examples with different ε. By observing the statistical results, we can clearly find that SDAL presents a slight change on error rate even when adding a different number of adversarial examples or setting different perturbation parameters. However, the estimation of hypothesis update on an adversarial example is highly-skewed than a clear example. It then leads to sensitive perturbations for GEN and Self-Paced algorithms. Moreover, the approaches involved with representative examples such as Hiera, TED, k-meansN, and k-meansA also present small perturbations.

E. Querying with Noisy Labels

In many learning issues, the cost of obtaining the ground truth labels is expensive. A group of good annotation results on the unlabeled set is difficult to obtain due to manual error or simply a lack of precision of the original data [47]. This also makes the queried labels in AL noisy. When hypothesis-pruning querying meets the noisy labels, these examples will generate an unprepared perturbation for the estimation of model change of a hypothesis-pruning AL. Further, querying with noisy labels zooms the performance disagreement of the hypothesis-pruning and distribution-shattering AL. Therefore, the experiment results can be a group of evidence to defend our theoretical insights.

We firstly collect the Fashion-Mnist dataset [5]. With a similar

[5] https://github.com/zalandoresearch/fashion-mnist

Table II
THE STATISTICAL RESULTS (MEAN±STD IN %) OF AL WITH ADVERSARIAL EXAMPLES OF DIFFERENT ALGORITHMS (ALG DENOTES ALGORITHMS)

ε	Alg	Number of added adversarial examples (N_{adv})				
		0	100	500	1,000	2,000
0.1	Hiera	29.3±27.3	31.4±28.7	37.4±7.3	42.4±5.5	39.0±7.1
	TED	26.8±26.7	29.0±7.5	30.5±7.9	32.0 ±7.9	35.7±7.0
	GEN	30.2±31.6	31.3±10.3	56.6±8.1	58.5±6.7	60.1±6.0
	k-meansN	30.0±1.0	31.3±0.8	32.5±0.5	33.5±0.9	34.1±1.2
	k-meansA	47.9±1.4	31.3±0.7	49.7±0.6	51.6±1.0	53.1±1.3
	Self-Paced	30.6±28.5	31.3±8.7	56.3±9.3	66.0±8.3	70.1±7.2
	SDAL	23.6±24.6	24.5±6.2	26.0±6.6	26.7±6.9	28.9±5.9
0.3	Hiera	29.3±27.3	31.2±8.4	44.7±5.9	39.8±7.3	41.2±8.0
	TED	26.8±26.7	29.9 ±7.6	34.3 ±8.9	34.6±7.9	37.6±8.1
	GEN	30.2±31.6	32.0±9.8	59.8 ±7.1	61.2±6.5	62.3±6.2
	k-meansN	30.0±1.00	30.2±1.3	31.3±0.9	35.7±0.6	36.3±0.8
	k-meansA	47.9±1.4	49.2±1.2	31.3±0.6	52.7±0.7	52.5±1.1
	Self-Paced	30.6±28.5	59.5±12.5	67.6±9.3	58.5±8.9	68.7±11.4
	SDAL	23.6±24.6	25.4 ±6.2	28.0±5.9	29.2±6.5	31.9±6.7
0.5	Hiera	29.3±27.3	31.9±8.4	44.7±5.9	41.9±6.9	49.4±6.1
	TED	26.8±26.7	31.3±6.8	35.0±7.8	39.0±9.0	38.8±7.6
	GEN	30.2±31.6	32.0±9.8	61.6±6.5	63.8±6.4	63.8±6.0
	k-meansN	30.0±1.0	33.3±1.3	33.5±0.9	36.4±0.5	37.2±0.8
	k-meansA	47.9±1.4	49.9±1.2	50.7±0.7	52.6±0.6	53.2±1.1
	Self-Paced	30.6±28.5	33.2±12.5	59.0±9.3	68.8±8.9	70.1±9.7
	SDAL	23.6±24.6	25.4 ±6.2	29.6 ±7.2	30.9±6.7	33.3±6.3

experiment setting, we respectively revise the original labels of the first 10, 500, 1000 data with noisy labels such as revising the label '0' to '1'. Fig. 7 describes the error rate change of adding a different number of noisy labels (N_{noi}), where the classifier also is a CNN model following Section VI.D, and parameters of baselines follow their best tunning in Section VI.C. In the drawn curves, the noisy examples have a negative influence on AL querying since they may misclassify a lot of unlabeled data after adding them in to the labeled set. Thus, they are also picked up as the primary sampling objects in the estimation of the model chance policy of hypothesis-pruning AL methods. However, the distribution-shattering approach avoids the perturbations. Only if the percentage of the noisy labels are large, the influence on the SDAL algorithm is obvious. Besides this, GEN shows a biased selection with the noisy setting. The noise perturbation to it is the most sensitive among the compared baselines. The others keep clear perturbation but not so series as GEN. The inherent reason follows the analysis of Section VI.D.

F. Calculation Complexity

The proposed SDAL algorithm (Algorithm 1 on Page 9) has two steps: halving and splitting, where Lines 2-10 describe the halving process using Eq. (11), and Lines 11-24 split the shattered distribution into k geometrical balls using Eq. (13). Generally, the halving step costs a calculation complexity of $\mathcal{O}(n^3)$ and the splitting step costs a time complexity of $\mathcal{O}(nk)$. Therefore, the total calculation complexity of SDAL algorithm is $\mathcal{O}(n^3)$. For any generalized hypothesis-pruning algorithm, estimating the hypothesis update needs to retrain the classification models, which results an uncertain calculation complexity. For example, GEN and Self-Paced algorithms repeatedly train a SVM model to select the samples which can maximize the error update, in the experiments. Generally, sampling k data will retrain and repredict the classifier kn' times, where n' denotes the unlabeled data number that is usually close to n. Then, the calculation complexity is almost $\mathcal{O}(kn^3)$ to $\mathcal{O}(kn^4)$ since SVM costs a calculation complexity of $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$. In addition the two generalized k-means algorithms approximately cost $\mathcal{O}(kn)$. The TED approach costs $\mathcal{O}(n^2)$ due to a greedy selection. The Hierarchical clustering-based AL costs $\mathcal{O}(n^3)$ due to the pre-clustering.

VII. CONCLUSION

AL algorithms provide strong theoretical guarantees on supervision sampling under fixed distribution and noise conditions. However, the label complexity bounds of the general hypothesis-pruning methods heavily depend on the initial hypothesis. This generates a challenging gap between the theoretical guarantee and application performance of AL algorithms.

To bridge this gap, we propose a distribution-shattering strategy from a theoretical perspective of number density. With lower generalization error and label complexity in the shattered distribution, we implement the proposed theoretical strategy against an arbitrary distribution by the SDAL algorithm in real-world querying tasks. Based on these theoretical analyses, empirical evaluation, and experiment results, we conclude that the hypothesis-pruning AL strategies degenerate their performance when querying with limited labels, adversarial examples, and noisy labels since they heavily depend on the initial labeled set and classifier. However, the proposed distribution-shattering strategy only presents slight perturbations in these querying scenarios.

REFERENCES

[1] B. Settles, "Active learning literature survey," *Computer Sciences Technical Report, University of Wisconsin, Madison*, 2009.

[2] D. Wu, "Pool-based sequential active learning for regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1348–1359, 2018.

[3] Z. Qiu, D. J. Miller, and G. Kesidis, "A maximum entropy framework for semisupervised and active learning with unknown and label-scarce classes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 4, pp. 917–933, 2016.

[4] S. Mohamad, A. Bouchachia, and M. Sayed-Mouchaweh, "A bi-criteria active learning algorithm for dynamic data streams," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 74–86, 2016.

[5] A. Harpale and Y. Yang, "Active learning for multi-task adaptive filtering," in *International Conference on Machine Learning*, 2010.

[6] M. Fang, J. Yin, L. O. Hall, and D. Tao, "Active multitask learning with trace norm regularization based on excess risk," *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3906–3915, 2017.

[7] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman, "Teacher-student curriculum learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[8] A. K. McCallumzy and K. Nigamy, "Employing em and pool-based active learning for text classification," in *International Conference on Machine Learning*. Citeseer, 1998, pp. 359–367.

[9] M.-F. Balcan, A. Beygelzimer, and J. Langford, "Agnostic active learning," in *International Conference on Machine Learning*. ACM, 2006, pp. 65–72.

[10] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, 1994.

[11] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," in *Measures of Complexity*. Springer, 2015, pp. 11–30.

[12] S. Dasgupta, D. J. Hsu, and C. Monteleoni, "A general agnostic active learning algorithm," in *Neural Information Processing Systems*, 2008, pp. 353–360.

[13] S. Yan and C. Zhang, "Revisiting perceptron: Efficient and label-optimal learning of halfspaces," in *Neural Information Processing Systems*, 2017, pp. 1056–1066.

[14] A. Gonen, S. Sabato, and S. Shalev-Shwartz, “Efficient active learning of halfspaces: an aggressive approach,” *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2583–2615, 2013.

[15] C. Cortes, G. DeSalvo, M. Mohri, N. Zhang, and C. Gentile, “Active learning with disagreement graphs,” in *Proceedings of the 36th International Conference on Machine Learning, International Conference on Machine Learning 2019, 9-15 June 2019, Long Beach, California, USA*, 2019, pp. 1379–1387.

[16] S. Dasgupta, “Two faces of active learning,” *Theoretical Computer Science*, vol. 412, no. 19, pp. 1767–1781, 2011.

[17] G. Brightwell and P. Winkler, “Counting linear extensions is# p-complete,” in *Proceedings of the twenty-third annual ACM Symposium on Theory of Computing*. ACM, 1991, pp. 175–181.

[18] M.-F. Balcan, S. Hanneke, and J. W. Vaughan, “The true sample complexity of active learning,” *Machine Learning*, vol. 80, no. 2-3, pp. 111–139, 2010.

[19] L. Chen, S. H. Hassani, and A. Karbasi, “Near-optimal active learning of halfspaces via query synthesis in the noisy setting.” in *Association for the Advancement of Artificial Intelligence*, 2017, pp. 1798–1804.

[20] D. Golovin, A. Krause, and D. Ray, “Near-optimal bayesian active learning with noisy observations,” in *Advances in Neural Information Processing Systems*, 2010, pp. 766–774.

[21] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, “Selective sampling using the query by committee algorithm,” *Machine Learning*, vol. 28, no. 2-3, pp. 133–168, 1997.

[22] S. Dasgupta, “Coarse sample complexity bounds for active learning,” in *Neural Information Processing Systems*, 2006, pp. 235–242.

[23] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, “Multi-class active learning by uncertainty sampling with diversity maximization,” *International Journal of Computer Vision*, vol. 113, no. 2, pp. 113–127, 2015.

[24] N. Roy and A. McCallum, “Toward optimal active learning through monte carlo estimation of error reduction,” *International Conference on Machine Learning*, pp. 441–448, 2001.

[25] A. Beygelzimer, D. J. Hsu, J. Langford, and T. Zhang, “Agnostic active learning without constraints,” in *Neural Information Processing Systems*, 2010, pp. 199–207.

[26] S. Hanneke, “Teaching dimension and the complexity of active learning,” in *International Conference on Computational Learning Theory*. Springer, 2007, pp. 66–81.

[27] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” in *Proceedings of the Fifth Annual workshop on Computational Learning Theory*. ACM, 1992, pp. 287–294.

[28] C. Tosh and S. Dasgupta, “Diameter-based active learning,” *International Conference on Machine Learning*, 2017.

[29] Y. Gal, R. Islam, and Z. Ghahramani, “Deep bayesian active learning with image data,” *International Conference on Machine Learning*, 2017.

[30] K. Yu, J. Bi, and V. Tresp, “Active learning via transductive experimental design,” in *International Conference on Machine Learning*, 2006, pp. 1081–1088.

[31] L. Zhang, C. Chen, J. Bu, D. Cai, X. He, and T. S. Huang, “Active learning based on locally linear reconstruction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2026–2038, 2011.

[32] S.-J. Huang, R. Jin, and Z.-H. Zhou, “Active learning by querying informative and representative examples,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 10, pp. 1936–1949, 2014.

[33] B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao, “Exploring representativeness and informativeness for active learning,” *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 14–26, 2017.

[34] D. Cai and X. He, “Manifold adaptive experimental design for text categorization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 707–719, 2012.

[35] L. Shi and Y.-D. Shen, “Diversifying convex transductive experimental design for active learning.” in *International Joint Conferences on Artificial Intelligence*, 2016, pp. 1997–2003.

[36] S. Hanneke, “A bound on the label complexity of agnostic active learning,” in *International Conference on Machine Learning*. ACM, 2007, pp. 353–360.

[37] I. M. Alabdulmohsin, X. Gao, and X. Zhang, “Efficient active learning of halfspaces via query synthesis.” in *Association for the Advancement of Artificial Intelligence*, 2015, pp. 2483–2489.

[38] X. Cao, I. W. Tsang, and G. Xu, “A structured perspective of volumes on active learning,” *arXiv:1807.08904*, 2018.

[39] P. Massart, É. Nédélec *et al.*, “Risk bounds for statistical learning,” *The Annals of Statistics*, vol. 34, no. 5, pp. 2326–2366, 2006.

[40] P. Awasthi, M. F. Balcan, and P. M. Long, “The power of localization for efficiently learning linear separators with noise,” in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*. ACM, 2014, pp. 449–458.

[41] I. W. Tsang, P.-M. Cheung, and J. T. Kwok, “Kernel relevant component analysis for distance metric learning,” in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 954–959.

[42] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, “Core vector machines: Fast svm training on very large data sets,” *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 363–392, 2005.

[43] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[44] S. Dasgupta and D. Hsu, “Hierarchical sampling for active learning,” in *International Conference on Machine Learning*, 2008, pp. 208–215.

[45] Y.-P. Tang and S.-J. Huang, “Self-paced active learning: Query the right thing at the right time,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5117–5124.

[46] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, 2014.

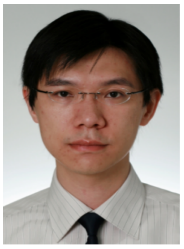
[47] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, “Learning with noisy labels,” in *Neural Information Processing Systems*, 2013, pp. 1196–1204.

[48] M.-F. Balcan, A. Broder, and T. Zhang, “Margin based active learning,” in *Computational Learning Theory*. Springer, 2007, pp. 35–50.

[49] S. Dasgupta, A. T. Kalai, and C. Monteleoni, “Analysis of perceptron-based active learning,” in *Computational Learning Theory*. Springer, 2005, pp. 249–263.

[50] Y. Hu, D. Zhang, Z. Jin, D. Cai, and X. He, “Active learning via neighborhood reconstruction,” in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. Citeseer, 2013, pp. 1415–1421.

Xiaofeng Cao is currently working toward the Ph.D. degree at Australian Artificial Intelligence Institute, University of Technology Sydney. His research interests include PAC learning theory, agnostic learning algorithm, and generalization analysis.



Ivor W. Tsang is an ARC Future Fellow and Professor of Artificial Intelligence, at University of Technology Sydney (UTS). He is also the Research Director of the Australian Artificial Intelligence Institute. His research focuses on transfer learning, learning with noisy supervision and big data analytics for data with extremely high dimensions. In 2019, his JMLR paper titled ”Towards ultrahigh dimensional feature selection for big data” received the International Consortium of Chinese Mathematicians Best Paper Award. In 2020, Prof Tsang was recognized as the AI 2000 AAAI/IJCAI Most Influential Scholar in Australia for his outstanding contributions to the field of AAAI/IJCAI between 2009 and 2019. His works on transfer learning granted him the Best Student Paper Award at CVPR 2010 and the 2014 IEEE TMM Prize Paper Award. In addition, he had received the prestigious IEEE TNN Outstanding 2004 Paper Award in 2007. He serves as a Senior Area Chair/Area Chair for NeurIPS, ICML, AISTATS, AAAI and IJCAI. He serves as the Editorial Board for the JMLR and MLJ, as well as an Associate Editor for the IEEE Trans. on Big Data and the IEEE Trans. on Emerging Topics in Computational Intelligence.