The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

STARS: Spatial-Temporal Active Re-sampling for Label-Efficient Learning from Noisy Annotations

Dayou Yu[1*], Weishi Shi[2*], Qi Yu[1†]

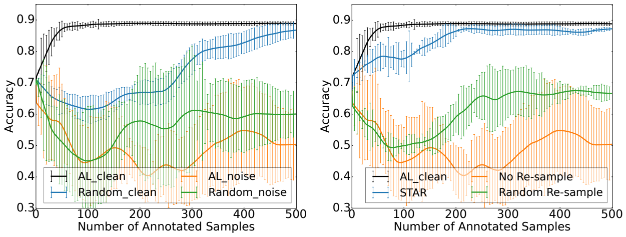
[1] Rochester Institute of Technology
[2]University of North Texas
{dy2507, qi.yu}@rit.edu, weishi.shi@unt.edu

Abstract

Active learning (AL) aims to sample the most informative data instances for labeling, which makes the model fitting data efficient while significantly reducing the annotation cost. However, most existing AL models make a strong assumption that the annotated data instances are always assigned correct labels, which may not hold true in many practical settings. In this paper, we develop a theoretical framework to formally analyze the impact of noisy annotations in AL and show that systematically re-sampling guarantees to reduce the noise rate, which can lead to improved generalization capability. More importantly, the theoretical framework demonstrates the key benefit of conducting active re-sampling on label-efficient learning, which is critical for AL. The theoretical results also suggest essential properties of an active re-sampling function with a fast convergence speed and guaranteed error reduction. This inspires us to design a novel spatial-temporal active re-sampling function by leveraging the important spatial and temporal properties of maximum-margin classifiers. Extensive experiments conducted on both synthetic and real-world data clearly demonstrate the effectiveness of the proposed active re-sampling function.

Introduction

Modern supervised learning techniques, including most deep learning models, require a large volume of labeled data for model training. However, annotating a large number of data samples is both labor-intensive and time-consuming. Active Learning (AL) provides a promising means to reduce the data annotation cost. The key idea is to train an active sampling model to identify most informative samples and only ask their labels from annotators. While traditional AL methods have demonstrated great potential in reducing the data annotation cost (Joshi, Porikli, and Papanikolopoulos 2009; Luo, Schwing, and Urtasun 2013; Yoo and Kweon 2019; Kirsch, Van Amersfoort, and Gal 2019), they are usually vulnerable to the annotation noises introduced through various kinds of human or device errors due to limitation of their knowledge or the constraints from the environment. Thus, their effectiveness may be significantly affected when being deployed in many practical settings.



(a) (b)

Figure 1: (a) impact of noisy annotations, (b) comparison of no re-sampling, random re-sampling, and active re-sampling

Figure 1 (a) demonstrates the major negative impacts of annotation noises to AL. It can be seen that effectiveness of AL becomes completely diminished in the presence of a moderate level of annotation noises at 0.3. We have a few important observations. First, AL under noise, which corresponds to the AL_noise curve in the figure, significantly under-performs an AL model trained via clean data (*i.e.,* AL_clean). In contrast, AL_clean demonstrates a clear advantage over passively learning (*i.e.,* Random_clean), which randomly chooses data instances for annotation and model training. Passive learning eventually still converges to the desirable generalization performance but requires much more annotated instances as compared with AL. Furthermore, AL_noise even performs worse than a passive learning model when being training from noisy annotations (*i.e.,* Random_noise). This is because an AL model tends to choose the most *informative* instances for annotation and these instances can significantly impact the decision surface. When these important instances are wrongly labeled, they could do more harm to the AL model than randomly selected instances. Therefore, an actively trained model could perform even worse under annotation noises. Finally, as expected, models trained from noisy data exhibit a much higher variance, which is undesirable.

Despite significant negative impacts as identified above, how to effectively conduct AL from noisy annotations is still largely unexplored. Traditional methods assume that data instances, once being sampled, can be precisely labeled. However, as discussed earlier, such a strong assumption no longer hold true for many applications. Few existing works propose to perform re-sampling among the annotated data instances and relabel them to alleviate the negative impact of the annotation noises. As a representative work (Lin, Mausam, and

[*]These authors contributed equally.

[†]Corresponding author

Weld 2016), the model needs to determine whether to query a new unlabeled instance from the unlabeled pool or ask the annotator to relabel one from the training set. As a result, each training instance may have a different number of labels as re-sampling and active sampling both precede. The final label of each instance is determined by some aggregation function (*e.g.,* majority vote among all its labels).

However, existing re-sampling models suffer from three fundamental issues. First, it is hard to control the balance between active sampling and re-sampling. The same criterion is typically used for both purposes. However, the training distribution and unlabelled pool distribution could be very different, especially at the early stage of AL. Thus, the same criterion evaluated over these two dataset are not directly comparable. In particular, due to the over-fitting problem caused by lack of data in AL, the same criterion may tend to be overestimated on the training set rather than on the pool set, which forces the model to conduct re-sampling repeatedly and neglects active sampling. Second, for the majority vote to work properly, the same instance needs to be re-sampled multiple times, which may introduce unnecessary annotation costs. Our theoretical results show that while majority vote based re-sampling guarantees to reduce the noise rate, it is far less label-efficient than a well-designed re-sampling mechanism. Finally, the current re-sampling criteria are expensive to evaluate, which typically require retraining the model with each data instance and all its possible labels. This makes re-sampling difficult to scale to real-world (especially multi-class) problems.

Besides the major issues outlined above, a theoretical framework that can be used to analyze the behavior of different re-sampling criteria and quantify the performance improvement (*e.g.,* using error bound) as a result of re-sampling is still missing. In this paper, we propose a novel Spatial-Temporal Active Re-sampling (STARS) model to support label-efficient learning from noisy annotations. The proposed active re-sampling function will reduce the error bound with theoretical guarantees. Figure 1 (b) shows that random re-sampling achieves a better performance than without any re-sampling under annotation noises. However, it converges much slower (also justified by our theoretical results). In contrast, the proposed STARS model converges much faster and to a much better performance, which is very close to an optimal AL model trained on clean data. More importantly, it achieves this performance by using less than 250 annotations, which include labels collected through both active sampling and re-sampling. The promising result shows that this work has the potential to extend the frontiers of AL research by filling out a critical gap in conducting active sampling from noisy annotations. As a result, the proposed research will allow AL models be successfully deployed in many practical settings, where noisy annotations cannot be avoided.

Our main contribution is threefold: (i) we develop the first formal theoretical framework to rigorously justify the negative impact of noisy annotations and establish the value of active re-sampling, (ii) we propose a loss based active re-sampling function with nice theoretical guarantees to reduce the error bound, and (iii) we identify some major limitations of the loss based re-sampling function and develop a novel spatial-temporal active re-sampling function by leveraging the key spatial and temporal properties of maximum-margin classifiers. Extensive experiments on both synthetic and real-world data demonstrate the effectiveness of the proposed active re-sampling model.

Related Work

AL with noisy annotations is primarily investigated under collaborative annotation such as crowd-sourcing (Ipeirotis 2011) and collaborative tagging (Ramezani et al. 2009), where multiple annotators label the data simultaneously (Donmez and Carbonell 2008). In the multi-annotator setting, annotation noise can be reduced through quality control methods (Khattak and Salleb-Aouissi 2011; Chittilappilly, Chen, and Amer-Yahia 2016; Hung et al. 2013). However, in most common scenarios of AL, usually one annotator is available for data labeling or the label is collected from a single device. For example, in the medical domain, the diagnosis of a disease (*i.e.,* a label) is likely to be provided by a single doctor. The difficulty of the task also makes annotation errors inevitable.

Few existing works leverage relabeling strategies to fix the ill-labelled training samples during AL. (Sheng, Provost, and Ipeirotis 2008) proposes to leverage both model and label uncertainty to identify samples for relabelling. However, it is difficult to balance acquiring a new label and relabelling one. (Zhao, Sukthankar, and Sukthankar 2011) proposes to linearly combine the expected loss change and label inconsistency (LI). However, the LI is roughly estimated via local density estimation, which is sensitive to the initialization of AL. (Bouguelia et al. 2018) proposes to use two types of disagreement for relabeling. However, it still requires evaluating all the leave-one-out models thus is very expensive especially in the multi-class case. The sampling score needs to be evaluated for each instance and each class. (Lin, Mausam, and Weld 2016) leverages different types of uncertainty and impact sampling. However, the impact computation is also expensive. Furthermore, majority vote often requires many times of relabeling. Similarly, (Du and Ling 2010) proposes an exploration-exploitation guided AL framework with relabeling. However, the decision making is the same as majority voting. (Zhang, Wang, and Yun 2015) proposes to identify the instances that are the most unreliable. However, the choice of the backward learning process is unclear, and the effectiveness of relabeling is not fully studied.

As pointed out earlier in the paper, existing re-sampling based methods either incur a high computational cost, making them difficult to scale to a large number of classes, or demand a high annotation cost (by relying on majority vote). As a result, they provide insufficient support to an effective AL process. In contrast, the proposed STARS model addresses these limitations by reducing the noise rate with fast convergence and theoretically guaranteed error reduction.

A Theoretical Framework of Active Re-sampling

In this section, we present a formal theoretical framework to clearly establish the benefit of active re-sampling. Through a set of key theoretical results, we demonstrate the negative impact of noisy annotations to supervised learning in general and show that systematic re-sampling in a passive setting provides a viable solution to improve the generalization capability despite of noisy annotations. We then justify that effective active re-sampling could further improve the label complexity, which is essential for label-effective learning in many critical domains.

Problem Setup

Let $\mathcal{D}_T = \{\mathbf{x}, y\}_{n=1}^{N}$ and $\mathcal{D}_U = \{\mathbf{x}\}_{m=1}^{M}$ denote the annotated dataset and unlabelled dataset, For each instance-label pair $\{\mathbf{x}, y\}$, we have $\mathbf{x} \in \mathbb{R}^L$ and $y \in \{-1, 1\}$ ($y \in \{1, 2, ..., K\}$ for multi-class). We denote h as the classification concept from some hypothesis set $\mathcal{H}$, h^* is the optimal concept provided by some learning algorithm (*e.g.,* empirical risk minimization). Traditional pool-based active learning takes sequential steps to update $\mathcal{D}_T$. At each step, the machine samples a data instance from $\mathcal{D}_U$ according to an acquisition function $f(\cdot)$. The sample is then labeled by a noisy annotator $h_\alpha(\cdot)$ and added to $\mathcal{D}_T$. Denote α as the degree of annotation noise: $\alpha = p(h_\alpha(\mathbf{x}) \neq h_0(\mathbf{x}))$, while $h_0(\cdot)$ is the oracle annotator, which is achieved as α approaches to zero: $h_0(\mathbf{x}) = \lim_{\alpha \to 0} h_\alpha(\mathbf{x})$. Let $A(h_\alpha) = \frac{|\{i \in [N]: y_i \neq h_0(\mathbf{x}_i)\}|}{N}$ denote the empirical *data noise* under the assumption that all the data instances are annotated by h_α. Note that $A(h_\alpha)$ is essentially the sample mean of N i.i.d. random variables $\mathbb{1}(\mathbf{x})_{[h_\alpha(\mathbf{x}) \neq h_0(\mathbf{x})]}$, while α is the (unknown) population mean of those variables.

When re-sampling is considered, the model also selects data instances from current training set $\mathcal{D}_T$ according to a re-sampling function $g(\cdot)$. The selected data instance will be relabeled by the same noisy annotator $h_\alpha(\cdot)$. The overall re-sampling process is illustrated by Figure 7 of the Appendix (Yu, Shi, and Yu 2023).

Fixing Noisy Annotations through Re-sampling

Since the data noise may be introduced to different types of data instances, an intuitive way to overcome the noise is to increase the training set size by annotating more data to enhance the chance of learning from more clean data. However, the asymptotic analysis given in the following corollary shows that increasing the training size could even enlarge the gap with the true error rate.

Lemma 1. *Consider a multi-class problem with K classes and assume all classes are equiprobable. As $|\mathcal{D}_T| \to \infty$, the gap between the error of a noisy classifier h^* and its true error rate on the clean test data converges to $\left(1 - \frac{K}{K-1}\epsilon\right)\alpha$, where $\epsilon = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_T}[h^*(\mathbf{x}) \neq y]$.*

Proof of the Lemma is provide in the Appendix. Lemma 1 implies that the gap actually increases since ϵ is expected to decrease when the training set $\mathcal{D}_T$ becomes larger. In contrast, reducing α can more effectively close the gap. However, directly changing α is challenging in practice as the annotation behavior of humans is subtle and outside of the control of the AL model. In fact, to improve the annotation quality usually involves years of training and practicing, which can be very expensive, especially in knowledge-rich domains. Next, we show that re-sampling provides an indirect way to manipulate the data noise, which can improve the generalization capability despite of noisy annotations from humans.

The most straightforward way to perform re-sampling is to uniformly sample from $\mathcal{D}_T$ and then conduct a majority vote. We show from the following theorem that even using such a simple re-sampling strategy, we can systematically reduce the data noise. To make the theoretical analysis more intuitive, we focus our discussion on the binary problem.

Theorem 1. *Consider a noisy data instance $\mathbf{x}$ that has been repeatedly labeled for $R > 2$ with the same annotation noise $\alpha < 0.5$ and the final label is determined through a majority vote. The probability of $\mathbf{x}$'s final label remaining uncorrected is guaranteed to be lower than α.*

Proof sketch. Let α_R^{maj} denote the probability of $\mathbf{x}$ still being uncorrected after conducting majority vote on R repeated annotations. This corresponds to the correct label has been annotated for no more than $\lfloor \frac{R}{2} \rfloor$ times. Since the probability to assign the correct label is α, we have

$$\alpha_R^{\text{maj}} = F\left(\lfloor \frac{R}{2} \rfloor; R, 1 - \alpha\right) = \sum_{n=0}^{\lfloor \frac{R}{2} \rfloor} \binom{R}{n} (1-\alpha)^n \alpha^{(R-n)}$$

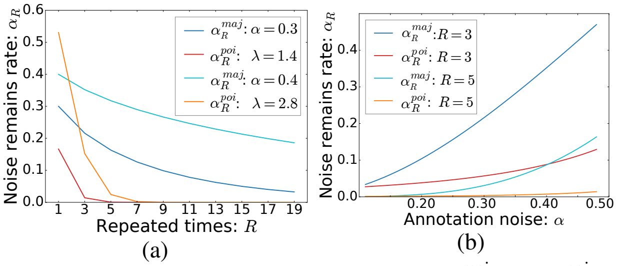
where F is the cumulative distribution function of binomial distribution $\text{Bin}(\lfloor \frac{R}{2} \rfloor; R, 1 - \alpha)$. To avoid ties in majority vote, we assume that N_1 takes odd values. It is straightforward to show that $\alpha_R^{\text{maj}} < \alpha$ for $R = 3$. We then use induction to show that $\alpha_{R+2}^{\text{maj}} < \alpha_R^{\text{maj}}$, which will complete the proof. A detailed proof is provided in the Appendix. □

From Passive Re-sampling to Active Re-sampling

While the majority vote based re-sampling can help to improve the generalization capability of a model, it incurs a high annotation cost, which does not provide a label efficient learning scheme suitable for AL. Ideally, once a noisy data instance has been correctly relabeled, it should not be re-sampled or sampled with a smaller chance. Thus, an optimal active re-sampling function g^* should select training instance according to its likelihood of being wrongly labeled. Given such g^*, we can show that the noise rate will decrease much faster than the passive re-sampling based on majority vote. We first prove that using an optimal re-sampling function, the noise rate is guaranteed to decrease. We then show it can converge to a clean training set in a much faster rate. -1mm

Theorem 2. *Given a re-sampling function g^* that does not revisit a corrected labeled instance $\mathbf{x}$, then (i) the noise level of the re-sampled dataset can be modeled by a Poisson distribution: Pois(λ) with $\lambda = \frac{1}{1-\alpha}$; (ii) the probability of $\mathbf{x}$'s final label remaining uncorrected α_R^{poi} is guaranteed to be lower than α after being relabeled $R > 2$ times.*

Proof sketch. The proof of part (i) directly follows that g^* no longer samples a data instance once its label is correct and given the noise rate at α, it takes on average $\frac{1}{1-\alpha}$ steps to correct a label. Proof of part (ii) can be done by using the analytical form of the Poisson distribution and through induction. Details are provided in the Appendix. □



(a)

(b)

Figure 2: Noise rate decreasing speed: α_R^{maj} Vs. α_R^{poi}; (a) $\alpha = 0.3, 0.4$; (b) $R = 3, 5$.

Numerical Analysis. Based on the theoretical results in Theorems 1 and 2, we conduct important numerical analysis to compare the noise rate decreasing speed between passive and active re-sampling. In Figure 2 (a), we first fix $\alpha = 0.3$, $\lambda = \frac{1}{1-\alpha} = 1.4$ and show the uncorrected noise rate along with the re-sampling frequency R. It is clear that active re-sampling with an optimal re-sampling function converges to a clean dataset (*i.e.,* zero noise rate) significantly faster than passive re-sampling.

We further loosen the requirement of using an optimal re-sampling function by doubling λ as 2.8. This essentially allows each noisy label to be re-sampled close to 3 times (vs. 1.4 times as in the optimal setting). In this case, active re-sampling still reduces the noise rate much faster than the passive one. Further increasing α to 0.4 makes the convergence much slower, which is as expected. In Figure 2 (b), we fix the re-sampling frequency $R = 3, 5$ and investigate the impact of α. It confirms that active re-sampling is more effective in reducing the noise rate given the same R and its advantage becomes more obvious with the increase of α.

Loss-Based Active Re-sampling. Both the theoretical result and numerical analysis demonstrate the clear advantage of conducting active re-sampling for label-efficient learning from noisy annotations. In addition, the numerical analysis confirms that even with a much larger λ, which allows a data instance to be revisited multiple times even after its label has been corrected, active re-sampling can still reduce the noise rate with a highly desirable speed. This further confirms the potential of using a well-designed re-sampling function to support AL from noisy annotations.

As stated above, a suitable re-sampling function should sample according to the likelihood of a training instance-label pair having the wrong label. Since the true labeling function h_0 is unknown, we can choose to use the model h^* trained on the noisy data as its proxy, which leads to a loss-based re-sampling function: $g^{\text{LOSS}}(\mathbf{x}) := p[y \neq h^*(\mathbf{x})]$.

To see why $g^{\text{LOSS}}(\mathbf{x})$ is more likely to assign a higher sampling score to a noisy instance, recall that h^* is obtained by minimizing some loss over the noisy training data:

$$\begin{aligned} h^* &= \arg\min_{h\in\mathcal{H}} \mathbb{E}_{\mathbf{x}}[\mathcal{L}_h(\mathbf{x}, y)] = \arg\min_{h\in\mathcal{H}} \mathbb{E}_{\mathbf{x}}[||h(\mathbf{x}) - y||^P] \\ &= \arg\min_{h\in\mathcal{H}} (1-\alpha) \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_T^{\text{cle}}} [||h(\mathbf{x}) - h_0(\mathbf{x})||^P] + \qquad (1) \\ &\quad \alpha \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_T^{\text{noi}}} [||h(\mathbf{x}) - (1 - h_0(\mathbf{x}))||^P] \qquad (2) \end{aligned}$$

where we have partitioned the training set $\mathcal{D}_T = \mathcal{D}_T^{\text{cle}} \cup \mathcal{D}_T^{\text{noi}}$ into a clean and noisy sets. As a result, the loss is also partitioned into two parts. Since $\alpha < 0.5$ is assumed, h^* would make more effort on approaching the true label function h_0 in the clean population $\mathcal{D}_T^{\text{cle}}$. To minimize the overall loss, (3) is more likely to be true than (4).

$$p(y \neq h^*(\mathbf{x})|\mathbf{x} \sim \mathcal{D}_T^{\text{cle}}) > p(y \neq h^*(\mathbf{x})|\mathbf{x} \sim \mathcal{D}_T^{\text{noi}}) \qquad (3)$$

$$p(y \neq h^*(\mathbf{x})|\mathbf{x} \sim \mathcal{D}_T^{\text{cle}}) < p(y \neq h^*(\mathbf{x})|\mathbf{x} \sim \mathcal{D}_T^{\text{noi}}) \qquad (4)$$

The inequality (3) implies that the model loss provides a valid re-sampling criterion that has the ability to pay more attention to the noisy data. To this end, we propose

$$g^{\text{LOSS}}(\mathbf{x}) := p[y \neq h^*(\mathbf{x})] \propto \mathcal{L}_{h^*}(\mathbf{x}, y) \qquad (5)$$

We further justify the effectiveness of loss based active re-sampling by restricting our focus to a specific type of models, Maximal-Margin Classifiers (MMCs). Representative MMCs, such as support vector machines (SVMs), have been commonly used for AL due to their sparse nature and good generalization capability. There are two additional key reasons of using an SVM as a base learning model for active re-sampling: (i) analysis of error bound could be conducted through the number of support vectors (see the theorem below), and (ii) the important spatial (or geometric) properties in the dual space could lead to more effective re-sampling mechanisms (detailed in the next section).

Theorem 3. *Relabeling a data instance with a high (hinge) loss is guaranteed to reduce the error bound for an SVM classifier $h^{SVM}(\mathbf{x})$.*

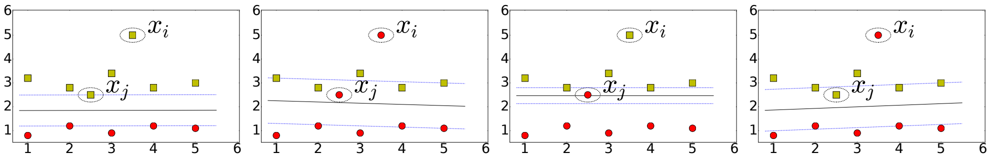
Proof sketch. The proof follows the leave-one-out (LOO) error analysis of SVMs: the average LOO error for N samples is an unbiased estimate of the average generalization error for samples of size $N - 1$ (Mohri, Rostamizadeh, and Talwalkar 2018). It also leverages the important property of support vectors with a large loss. Details are in the Appendix. □

Spatial-Temporal Active Re-sampling

A loss based function provides a valid active re-sampling mechanism with guaranteed performance improvement when being used with MMCs. However, since re-sampling will be conducted along with AL, which is essentially a dynamic process, it is critical to consider the temporal re-sampling order of the training instances. As we show below, simply re-sampling by following the order of the model loss on each training instance may lead to a slow convergence.

Determining the Temporal Re-sampling Order

We first show that the temporal order of re-sampling plays an important role in AL and the loss based re-sampling might lead to slow convergence or even trapped in a local optimum. As shown in Figure 3 (a), the clean data have two separate classes, while its noisy version has two incorrectly labeled instances as shown in Figure 3 (b). They represent two types of data instances that have non-zero losses with $\mathbf{x}_i$ being an outlier and $\mathbf{x}_j$ being on-the-margin. Using the model loss, $\mathbf{x}_i$ will be re-sampled before $\mathbf{x}_j$ due to a larger loss. However, this may deviate from the ultimate goal of re-sampling, which



(a) (b) (c) (d)

Figure 3: Importance of temporal re-sampling order (a) clean data, (b) noisy data, (c) loss based re-sampling, (d) an alternative (and better) temporal order for re-sampling

is to clean the noises for the purpose of recovering the optimal decision boundary with clean data shown in Figure 3 (a). As the model first corrects the outlier $\mathbf{x}_i$, the decision boundary will be misplaced temporally as shown in Figure 3 (c). Such distortion affects the accuracy of loss evaluation thus is harmful for the next re-sampling step as the on-the-margin sample might no longer stand out from other trivial (clean) instances with the drastic change of the decision boundary. An alternative (and better) temporal order of re-sampling is to first correct the on-the-margin instance $\mathbf{x}_j$ to push the decision boundary towards the right direction. While the decision boundary has been effectively shifted, the margin size remains almost the same as opposed to a significant shrink of the margin size in Figure 3 (c). This nice property benefits from the special design of the SVM loss function that uses a l_1-loss to penalize large errors, which makes the margin less sensitive to outliers. In the next re-sampling step, it further corrects the outlier $\mathbf{x}_i$ to make a final adjustment to the decision boundary. Figure 3 (d) shows that this best recovers the desired decision boundary.

Following the above analysis, we categorize all training instances into three types: `Type-I`: on-margin noise, `Type-II`: outlier noise, and `Type-III`: clean data. `Type-I` is most useful for active re-sampling as revealing the true labels of them has a direct impact on the current decision boundary. `Type-II` is also useful for re-sampling as revealing their true labels helps describe the data distribution more precisely. However, correcting their labels could lead to model oscillation especially in the early stage of AL. Ideally, this type of data should be re-sampled after `Type-I`. `Type-III` provides no value for active re-sampling and should attract least attention from the sampling function.

Design of the Active Re-sampling Function

In order to properly sample instances from the `Type-I` group, it is essential to quantify the instance distance to the current decision surface. To this end, we propose to use the magnitude of the SVM decision function as re-sampling criterion. An SVM classifier can be formulated as the linear combination of basis functions $h(\mathbf{x}) = \mathbf{w}^\top\phi(\mathbf{x})$, where $\mathbf{w}$ can be learned as follows:

$$\arg\min_{\mathbf{w}} \sum \text{Error}[y_n h(\mathbf{x}_n)] + \lambda||\mathbf{w}||^2 \tag{6}$$

with hinge loss $\text{Error}[y_n h(\mathbf{x}_n)] = [1 - y_n h(\mathbf{x}_n)]_+$.

Since the magnitude of SVM decision function (DEC) is proportion to the perpendicular distance of a data instance to the current decision surface, we define

$$g^{\text{DEC}}(\mathbf{x}) = |\mathbf{w}^\top\phi(\mathbf{x})| \propto \frac{|y\mathbf{w}^\top\phi(\mathbf{x})|}{||\mathbf{w}||} \tag{7}$$

where the last term quantifies perpendicular distance. We have used the fact that $y \in \{-1, 1\}$ and $||\mathbf{w}||$ is a constant for every input $\mathbf{x}$. As a result, g^{DEC} favors data instances located close to the current decision surface, which gives preference to sample `Type-I` instances before others.

Solely relying on g^{DEC} for the entire AL process may overly penalize `Type-II` instances, which makes them re-sampled even after some `Type-III` instances. To address this issue, we propose to leverage label inconsistency (LIC) to encourage sampling `Type-II` instances:

$$g^{\text{LIC}}(\mathbf{x}) = \sum_{(\mathbf{x}_i, y_i)\in\mathcal{D}_T} ||k(\mathbf{x}, \mathbf{x}_i)\mathbf{w}^\top\phi(\mathbf{x}_i) - y|| \tag{8}$$

where $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$ is a kernel function. $g^{\text{LIC}}(\mathbf{x})$ measures how neighborhood predictions of $\mathbf{x}$ disagree with its label, y. A large score means that the close neighbors of $\mathbf{x}$ (evaluated using the kernel function) are predicted to have different labels than y, which implies $\mathbf{x}$ is more likely to be a `Type-II` instance (*i.e.,* outlier).

A Joint Active Re-sampling Function. Both g^{DEC} and g^{LIC} are defined to leverage the key spatial information from the global (*i.e.,* distance to the decision surface) and local (*i.e.,* difference from neighbor instances) perspectives, to give preference to `Type-II` and `Type-I` instances, respectively. To achieve a desired temporal re-sampling order as described earlier, we propose to dynamically balance these two spatial re-sampling criteria to first re-sample on-margin instances and then shift to outliers to stabilize the overall AL performance and avoid slow convergence:

$$g^{\text{STARS}}(\mathbf{x}) = (1-\tau)g^{\text{DEC}}(\mathbf{x}) + \tau\left[g^{\text{LIC}}(\mathbf{x})\right]^{(-1)} \tag{9}$$

where 'STARS' standards for Spatial Temporal Active Re-sampling and $0 \leq \tau \leq 1$ is initialized as 0 and will continue to increase during the entire AL process. Note that the dependencies of the re-sampling function on $\mathbf{w}$ can be removed through the dual representation. In particular, substituting $\mathbf{w}$ with the optimal dual solution leads to

$$\begin{aligned} g^{\text{STARS}}(\mathbf{x}) = (1-\tau)|\mathrm{k}^\top(\mathbf{x})(\mathbf{a}\odot\mathbf{y})| + \\ \tau||\mathrm{k}(\mathbf{x})\odot\mathbf{h} - \mathbf{y}\mathbb{1}_N||^{(-1)} \end{aligned} \tag{10}$$

where $\mathrm{k}(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x})..., k(\mathbf{x}_N, \mathbf{x})]^\top$,$\mathbf{y} = (y_1, ...y_N)^\top$, $\mathbf{h} = [h(\mathbf{x}_1), ...h(\mathbf{x}_N)]^\top$, and $\mathbf{a} = (a_1, , ...a_N)^\top$ are Lagrange multipliers with $a_n > 0$ indicating $\mathbf{x}_n$ is a support vector. The first term in (10) leverages the feature relationship while the second term focuses on the label relationship. Together, these two terms capture the full spatial information from the reproducing kernel Hilbert space induced by kernel $k(\cdot, \cdot)$.

Corollary 1. *Relabeling a data instance with a low g^{STARS} score is guaranteed to reduce the error bound for an SVM classifier $h^{SVM}(\mathbf{x})$.*

Reducing High Re-sampling Variance. At the early stage of AL, the model prediction on specific data instances tends to have a large variance as AL is supposed to change the model drastically with the newly labeled instances and the training is unstable due to limited training data. Such a high variance will transfer to the proposed re-sampling function. As a result, a score $[g^{\text{STARS}}(\mathbf{x})]^{(t)}$ evaluated at the t-th iteration alone may not truly reflect the desirable re-sampling need due to the impact of the high variance. In our experiments, we observe that the $g^{\text{STARS}}(\mathbf{x})$ of a clean data instance is both high and stable during the learning process while among the instances with a low $g^{\text{STARS}}(\mathbf{x})$, only true noisy data would have steady low scores for a consecutive number of learning iterations. Similar observations has also been reported in curriculum learning based models, where the truly difficult data instances tend to have consistently large losses during the model training process. To address the high re-sampling variance, we propose to further incorporate a memorization unit that leverages the historical re-sampling scores of a data instance. Specifically, we expect the impact of the least recently evaluated $g^{\text{STARS}}(\mathbf{x})$ quickly fade out, which can be achieved by computing the exponential moving average:

$$[g^{\text{STARS}}(\mathbf{x})]^{(t)} = (1-\gamma)[g^{\text{STARS}}(\mathbf{x})]^{(t)} + \gamma[g^{\text{STARS}}(\mathbf{x})]^{(t-1)}$$

where γ is the decay factor, which decreases along with AL as the model becomes more stable.

Experiments

Synthetic Data Experiments

In the synthetic experiments, we compare STARS with loss based re-sampling (LOSS) and two of its individual components: DEC and LIC. Figures 4 (a)-(d) show the snapshots of the decision boundary (solid line) predicted by each model after 500 annotations (400 sampling+100 re-sampling). We use an uncertainty based sampling strategy, BvSB (Joshi, Porikli, and Papanikolopoulos 2009) for AL to sample new data instances. Margins are plotted (dashed lines) and data instances are plotted in red and blue squares. The instances selected for relabeling are colored in orange while the ones being relabeled correctly are rounded by blue circles. Figure 4 (g) shows the optimal decision boundary achieved by AL on clean data. As we compare each decision boundary, we observe that although STARS does not correct as many noisy samples as DEC and LOSS do, its converged decision boundary is closest to the optimal one. This confirms the importance of the re-sampling order: correcting the noise in the 'critical' data is more important than correcting more noisy data in random. Specifically, in Figure 4 (c) LOSS puts too much attention to correct the outlier noises (`Type-II`). Although a large portion of the outlier noises are successfully corrected, they contribute marginal help in terms of forming the true decision boundary. Figure 4 (b) shows that LIC exhibits the same 'exploration' behavior as LOSS does and ends up with the similar distorted decision boundary. Meanwhile, DEC in Figure 4 (a) demonstrates a clear 'exploitation' behavior as expected. We can observe that DEC does approach to the true decision boundary closer than LOSS due to the focus on correcting the noises near the decision boundary. However, the converged decision boundary poorly depicts the data distribution away from the decision boundary as DEC lacks proper exploration.

We provide a more detailed view on how STARS adjusts the re-sampling order according to different stages of AL to achieve the 'exploit-then-explore' behavior in the Appendix. Overall, the re-sampling strategies all perform better than the random re-sampling or no re-sampling. This can also be reflected by Figure 1 (b). Last, we show the AL curve that captures the model performance in entire learning process in Figure 4 (h), which further confirms that STARS maintains the advantage in model fitting for most parts of the learning.

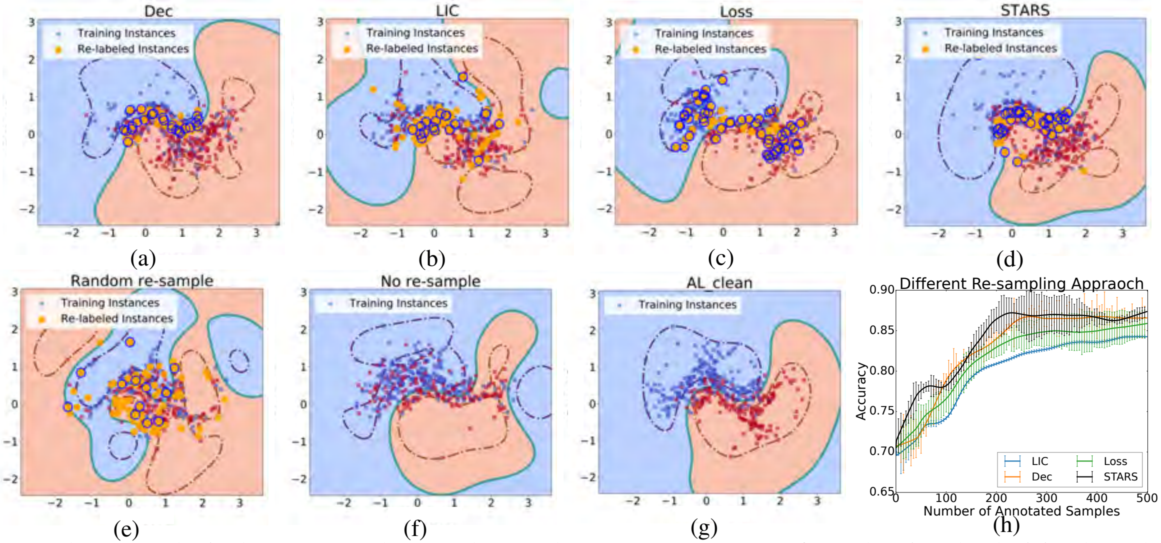
Real Data Experiments

We select 5 real-world datasets (Dua and Graff 2017; Shi and Yu 2018) from different domains: medical, bioinformatics, image recognition, and automatic systems. The chosen datasets have features varying from 8 to 1,554 and classes ranging from 10 to 50. The annotation noise α is set to 0.3. For STARS, we linearly increase τ from 0.2 to 0.7 and fix γ to 0.2. Additional details are in the Appendix.

In addition to two proposed re-sampling methods and using random re-sampling as a baseline, we also compare with the most related re-sampling work that is applicable to our setting, impactEXP (Lin, Mausam, and Weld 2015). The model is originally designed for binary problems and the extension to multiple classes can be computationally prohibitive as the number of classes increases. Thus, we can only run this baseline on datasets with smaller classes (*i.e.,* Yeast and Auto-drive) with a meaningful sampling time for AL. Figures 5 (a)-(e) show that STARS consistently outperforms other baselines on all datasets. Among all the models, impactEXP improves slowly mainly because it adopts majority vote, which has been proved to be less efficient than the proposed re-sampling strategies equipped by LOSS and STARS. LOSS also shows a competing performance on three datasets with a relatively large feature space. The increase of the feature size makes it more challenging to accurately capture the spatial properties of the feature space but LOSS is more robust to this change. As part of the ablation study, Figures 5 (f)-(j) compare STARS with its two components, LIC and DEC. Again, STARS shows a clear advantage on all datasets, which justifies the effectiveness of spatial-temporal based re-sampling. Some other re-sampling methods are restricted to binary problems and are expensive to generalize to multiple classes. We compare with them in binary settings and the results are consistent with the multi-class results. Additional results and ablation studies are in the Appendix.

Extension to Other Models

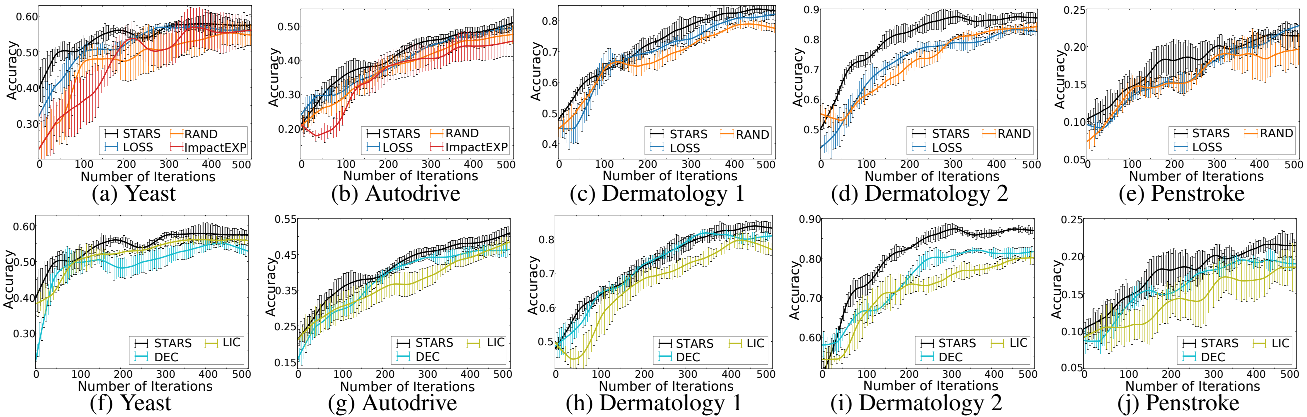
While SVM has been used to demonstrate the proposed theoretical framework on active re-sampling, it is important to note that many fundamental components of the framework are generally applicable to other classification models. To demonstrate the potential extension, we apply STARS to



(a) (b) (c) (d)

(e) (f) (g) (h)

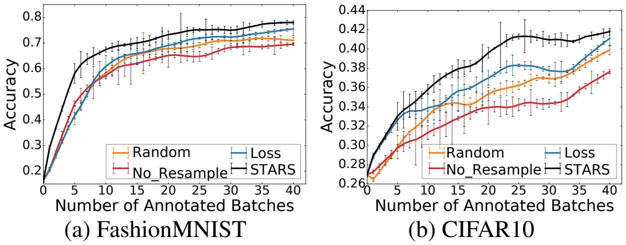
Figure 4: Results on synthetic data: (a)-(e) The snapshots at the convergence stage of AL showing the decision boundary of the model and the distribution of the re-labeled samples picked by different re-sampling strategies over the entire AL process; (f)-(g) decision boundary of no-re-sampling and AL from clean data); (h) AL performance comparison.



(a) Yeast (b) Autodrive (c) Dermatology 1 (d) Dermatology 2 (e) Penstroke

(f) Yeast (g) Autodrive (h) Dermatology 1 (i) Dermatology 2 (j) Penstroke

Figure 5: Real-world AL Results: (a)-(e) Comparison with other baselines; (f)-(j) ablation study.



(a) FashionMNIST (b) CIFAR10

Figure 6: Re-sampling results using a DNN

deep neural networks (DNN) through the general deep kernel learning (DKL) framework (Wilson, *et al.* 2016). Under DKL, $\phi(\mathbf{x})$ in (6) is achieved through a non-linear mapping given by a deep architecture and $h(\mathbf{x})$ can be interpreted as the logit, which is obtained through a linear mapping from $\phi(\mathbf{x})$ through $\mathbf{w}$. The logit shares a similar property as the magnitude of the SVM decision function: a small absolute logit implies $\mathbf{x}$ is close to the decision boundary. Since in active deep learning, the samples are usually labeled in a batch, we also adjust the re-sampling strategy to use small batches. The detailed configuration and additional results can be found the Appendix. Figure 6 shows that the loss-based re-sampling strategy performs very close to the random re-sampling strategy, which means that the magnitude of the loss is less indicative of the potentially mislabeled data points for deep learning models. Meanwhile, STARS shows a clear advantage over both strategies, which justifies the effectiveness of the proposed temporal re-sampling order. Furthermore, the label consistency criterion also becomes more effective due to an improved data representation $\phi(\mathbf{x})$ learned by a DNN.

Conclusion

We focus on AL from noisy annotations by developing a formal theoretical framework to prove the negative impact of annotation noises and suggest effective ways to conduct active re-sampling with performance and convergence guarantees. A novel spatial-temporal active re-sampling (STARS) model is designed accordingly and tested on both synthetic and real-world data under noisy settings. One future direction is to develop the theoretical guarantee on the reduction of the error bound for deep learning models. Meanwhile, it is also interesting to extend our model to a non-uniform annotation environment, where the annotation error changes along with the annotator's domain knowledge, skill, and other factors.

Acknowledgements

This research was supported in part by an NSF IIS award IIS-1814450 and an ONR award N00014-18-1-2875. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

References

Bouguelia, M.-R.; Nowaczyk, S.; Santosh, K.; and Verikas, A. 2018. Agreeing to disagree: Active learning with noisy labels without crowdsourcing. *International Journal of Machine Learning and Cybernetics*, 9(8): 1307–1319.

Chittilappilly, A. I.; Chen, L.; and Amer-Yahia, S. 2016. A survey of general-purpose crowdsourcing techniques. *IEEE Transactions on Knowledge and Data Engineering*, 28(9): 2246–2266.

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

Du, J.; and Ling, C. X. 2010. Active learning with human-like noisy oracle. In *2010 IEEE International Conference on Data Mining*, 797–802. IEEE.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. Institution: University of California, Irvine, School of Information and Computer Sciences.

Hung, N. Q. V.; Tam, N. T.; Tran, L. N.; and Aberer, K. 2013. An evaluation of aggregation techniques in crowdsourcing. In *International Conference on Web Information Systems Engineering*, 1–15. Springer.

Ipeirotis, P. 2011. Crowdsourcing using mechanical turk: quality management and scalability. In *Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011*, 1.

Joshi, A. J.; Porikli, F.; and Papanikolopoulos, N. 2009. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2372–2379. IEEE.

Khattak, F. K.; and Salleb-Aouissi, A. 2011. Quality control of crowd labeling through expert evaluation. In *Proceedings of the NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*, volume 2, 5.

Khetan, A.; Lipton, Z. C.; and Anandkumar, A. 2017. Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*.

Kirsch, A.; Van Amersfoort, J.; and Gal, Y. 2019. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32.

Lin, C. H.; Mausam, M.; and Weld, D. S. 2015. Reactive learning: Actively trading off larger noisier training sets against smaller cleaner ones. In *Proceedings of the 32nd International Conference on Machine Learning, Lille, France (ICML)*.

Lin, C. H.; Mausam, M.; and Weld, D. S. 2016. Re-active learning: Active learning with relabeling. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Luo, W.; Schwing, A.; and Urtasun, R. 2013. Latent structured active learning. *Advances in Neural Information Processing Systems*, 26.

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2018. *Foundations of machine learning*. MIT press.

Mozafari, B.; Sarkar, P.; Franklin, M.; Jordan, M.; and Madden, S. 2014. Scaling up crowd-sourcing to very large datasets: a case for active learning. *Proceedings of the VLDB Endowment*, 8(2): 125–136.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Ramezani, M.; Sandvig, J. J.; Schimoler, T.; Gemmell, J.; Mobasher, B.; and Burke, R. 2009. Evaluating the impact of attacks in collaborative tagging environments. In *2009 International Conference on Computational Science and Engineering*, volume 4, 136–143. IEEE.

Sheng, V. S.; Provost, F.; and Ipeirotis, P. G. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 614–622.

Shi, W.; and Yu, Q. 2018. An Efficient Many-Class Active Learning Framework for Knowledge-Rich Domains. In *2018 IEEE International Conference on Data Mining (ICDM)*.

Whitehill, J.; Wu, T.-f.; Bergsma, J.; Movellan, J.; and Ruvolo, P. 2009. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In Bengio, Y.; Schuurmans, D.; Lafferty, J.; Williams, C.; and Culotta, A., eds., *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

Yoo, D.; and Kweon, I. S. 2019. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 93–102.

Younesian, T.; Zhao, Z.; Ghiassi, A.; Birke, R.; and Chen, L. Y. 2021. QActor: Active Learning on Noisy Labels. In *Asian Conference on Machine Learning*, 548–563. PMLR.

Yu, D.; Shi, W. S.; and Yu, Q. 2023. Appendix: STARS: Spatial-Temporal Active Re-Sampling for Label-Efficient Learning from Noisy Annotations. https://github.com/ritmininglab/STARS.git.

Zhang, X.-Y.; Wang, S.; and Yun, X. 2015. Bidirectional active learning: A two-way exploration into unlabeled and labeled data set. *IEEE Transactions on Neural Networks and Learning Systems*, 26(12): 3034–3044.

Zhao, L.; Sukthankar, G.; and Sukthankar, R. 2011. Incremental relabeling for active learning with noisy crowdsourced annotations. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, 728–733. IEEE.

Zhao, Z.; Birke, R.; Han, R.; Robu, B.; Bouchenak, S.; Mokhtar, S. B.; and Chen, L. Y. 2021. Enhancing robustness of on-line learning models on highly noisy data. *IEEE Transactions on Dependable and Secure Computing*, 18(5): 2177–2192.

Zhao, Z.; Cerf, S.; Birke, R.; Robu, B.; Bouchenak, S.; Mokhtar, S. B.; and Chen, L. Y. 2019. Robust anomaly detection on unreliable data. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 630–637. IEEE.