21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Importance-Weighted Label Prediction for Active Learning with Noisy Annotations

Liyue Zhao[1] Gita Sukthankar[1] Rahul Sukthankar[2]
lyzhao@cs.ucf.edu gitars@eecs.ucf.edu rahuls@cs.cmu.edu
[1]Department of EECS, University of Central Florida
[2]Google Research and Carnegie Mellon University

Abstract

This paper presents a practical method for pool-based active learning that is robust to annotation noise. Our work is inspired by recent approaches to active learning in two different noise-free settings: importance-weighted methods for streams and unbiased pool-based techniques. In our proposed method, we employ an ensemble of classifiers to guide the label requests from a pool of unlabeled training data. We demonstrate, using several standard datasets, that the proposed approach, which employs label prediction in combination with importance-weighting, significantly improves active learning in the presence of annotation noise. Moreover, the ease with which the proposed method can be implemented should make it widely applicable to a broad range of real-world applications.

1. Introduction

Our work is motivated by the recent interest in the use of crowdsourcing [7] as a source of annotations from which to train machine learning systems. However, employing crowdsourcing to label large quantities of data remains challenging for two important reasons: limited annotation budget and label noise. First, although the unit cost of obtaining each annotation is low, the overall cost grows quickly since it is proportional to the number of requested labels, which can number in the millions. This has stimulated the use of approaches, such as active learning [10], that aim to learn high-quality classifiers using only a subset of labeled data. Second, the quality of crowdsourced annotations has been found to be poor [11], with causes ranging from workers who overstate their qualifications, lack of motivation among labelers, haste and deliberate vandalism. Unfortunately, the majority of popular active learning algorithms, while robust to noise in the input features, can be very sensitive to label noise.

The majority of work in active learning makes the "perfect oracle" assumption, meaning that the requested label is guaranteed to be correct. Under such conditions, an effective strategy for active learning is to select samples that would most reduce the set of hypotheses consistent with the observed training labels. Thus, "aggressive" criteria such as least confidence, smallest margin or maximum entropy can enable active learning to obtain high accuracy using a relatively small number of labels. Unfortunately, the potential of label noise causes aggressive active learning methods to fail because even a single incorrect label can suddenly cause the algorithm to incorrectly eliminate the wrong set of hypotheses and thus focus the search on a poor region of the version space. A second (but less well-known) concern is that active learning, by its very nature, creates a biased sampling of the data set since it favors instances that are in confusing regions of the feature space. Unless this bias is accounted for, it leads to sub-optimal learners.

A common strategy for combating annotation noise is to request multiple labels for each selected instance and then to apply majority voting. The simplest approach of requesting the same number of labels for each instance is not usually the most cost-effective since label redundancy increases the overall cost by a multiplicative factor of at least 3. Better results can be obtained by combining active learning with incremental relabeling to solicit additional labels where they are most useful [11, 13, 14]. An alternate strategy, as advocated by agnostic active learning [1], is to make more conservative use of the training data. The basic idea is to omit requesting labels for instances only if all hypotheses under current consideration agree, sample labels from the remaining set and to eliminate only those hypotheses whose lower bound on the expected error is

978-4-9906441-1-6

greater than the minimum upper bound for the expected error over the set of hypotheses.

Our work is inspired by the importance-weighted active learning algorithm (IWAL) [2], which considers each unlabeled element in a stream-based manner to determine whether it should be labeled, and if so, how to weight the data point in a manner that corrects for the effect of biased sampling. As detailed below, we advocate an importance-weighted approach in a pool-based active learning setting. Specifically, this paper makes three contributions: 1) we generate and maintain an unbiased bootstrap ensemble of incrementally trained classifiers that are used to identify and reject potentially inconsistent annotations; 2) rather than requesting labels for unlabeled instances simply based on importance sampling, we identify those instances that are (with high probability) unlikely to require labels; and 3) we present evaluations on standard datasets that show not only the strength of our proposed approach in comparison to established active learning algorithms, but also which specific aspects of our technique lead to the greatest improvements under various noise conditions.

2. Related Work

Given the limited space, our survey of related work focuses solely on the subset that relates directly to our proposed method. Active learning iteratively seeks to select the most informative instance based on current knowledge; see [10] for a recent survey. In stream-based active learning, the algorithm is presented with a sequence of unlabeled instances and the problem can be formulated as choosing whether to request a label for the current instance. By contrast, in a pool-based setting, the active learner is presented with a set of unlabeled data from which instances are selected. As discussed in the Introduction, early work in active learning was dominated by "aggressive" strategies that are not designed with label noise in mind; representative examples of this work include [4, 8, 12].

Our work is strongly influenced by A^2 [1], and IWAL [2], which are recent approaches to active learning that consider imperfect oracles. UPAL [6], while designed for the noise-free scenario, also seeks unbiased sampling of unlabeled data. This bias issue is related to that of "dataset shift", where training and test distributions diverge [3]. Failing to take it into account can degrade classifier accuracy, even when label noise is not an issue.

Algorithm 1 Proposed active learning algorithm

Require:
Input:
$\mathcal{X}$: set of (initially unlabeled) instances;
p_{min}: lower bound for importance sampling;
θ: label prediction confidence threshold;
T: period over which ensemble retraining occurs;
B: labeling budget.
Output:
h^*: The optimal classifier.

$\mathcal{T} \leftarrow$ randomly drawn subset of $\mathcal{X}$ (e.g., 10%);
Request (noisy) labels for $\mathcal{T}$ from oracle;
Train ensemble of probabilistic SVMs $\mathcal{H} = \{h_1, \ldots, h_K\}$ using bootstrap sampling on $\mathcal{T}$;
while $B > 0$ **do**
  Compute $\pi(\mathbf{x})\ \forall \mathbf{x} \in \mathcal{X}$ using Equation 1;
  Select a sample $\mathbf{x}_i$ from $\mathcal{X} \setminus \mathcal{T}$ using importance sampling with normalized CDF from π;
  if $s(\mathbf{x}_i, \mathcal{H}) < \theta$ **then**
    $y_i \leftarrow$ predicted label for $\mathbf{x}_i$ using $\mathcal{H}$;
  else
    Request label for $\mathbf{x}_i$ from oracle;
    $B \leftarrow B - \text{cost}(\mathbf{x}_i)$;
    After every T queries to oracle:
    (Optional:) Retrain ensemble $\mathcal{H}$ on bootstrap of $\mathcal{T}$;
  end if
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{x}_i\}$ with weight $1/\pi(\mathbf{x}_i)$;
end while
Return SVM h^* trained on weighted set of labeled data.

3. Method

We formulate the problem as follows. Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denote the (initially unlabeled) pool of instances. We model the oracle as a process that, given an instance x_i, returns an incorrect label with i.i.d. probability η and the correct label otherwise. We assume that the noise rate $\eta \in (0, 0.5)$ is unknown. Following standard approaches in active learning, we initially request noisy labels for an initial set $\mathcal{L}$ of randomly selected instances, where $|L|$ is typically 10% of $|\mathcal{X}|$. Using these, we train an ensemble of K support vector machines (SVMs) on bootstrap samples [5] of $|\mathcal{L}|$ elements, drawn (with replacement) from the labeled set $\mathcal{L}$. This set of K SVMs, denoted as $\mathcal{H} = \{h_1, \ldots, h_K\}$, correspond to the bootstrap estimators employed in the bootstrap variant of IWAL [2]. In our implementation, we assume without loss of generality that SVM outputs are interpreted probabilistically, e.g., using [9].

Our proposed method now diverges from IWAL in several respects, as detailed in Algorithm 1. Rather than simulating a stream by drawing unlabeled samples from the unlabeled pool $\mathcal{X} \setminus \mathcal{L}$, we sample directly from the pool in a more computationally efficient manner us-

ing importance sampling. Specifically, we calculate the probability $\pi(\mathbf{x})$ of sampling an unlabeled element $\mathbf{x}$ by running it through the K classifiers in $\mathcal{H}$ using:

$$\pi(\mathbf{x}_i) = p_{\min} + (1 - p_{\min}) \left(h_{\max}(\mathbf{x}_i) - h_{\min}(\mathbf{x}_i) \right), \quad (1)$$

where $p_{\min}$ is a constant (typically set to 0.1) and $h_j(.)$ is the probabilistic output of an SVM classifier from $\mathcal{H}$, with $h_{\min}(.)$ and $h_{\max}(.)$ denoting $\min_{h \in \mathcal{H}} h(\mathbf{x}_i)$ and $\max_{h \in \mathcal{H}} h(\mathbf{x}_i)$, respectively. In other words, $\pi(\mathbf{x})$ is highest when there is significant disagreement between any two classifiers in the bootstrap ensemble and lowest when all of them agree. We sample from the data according to π simply by generating a cumulative probability distribution, where each instance $\mathbf{x}_i$ in the unlabeled pool $\mathcal{X} \setminus \mathcal{L}$ corresponds to an interval in (0,1] proportional to $\pi(\mathbf{x}_i)$. Then, a random number drawn in (0,1] maps to the selected instance. We emphasize that this sampling process induces a *biased* sampling that favors those instances with high $\pi(.)$, i.e., those that most merit labeling. We make two important observations. First, this bias leads the distribution of data in $\mathcal{L}$ to draft away from that in the original data set $\mathcal{X}$. Unless the bias is rectified, training on $\mathcal{L}$ may not result in classifiers that are well suited to solving the original problem. Second, it is important to sample using $\pi(.)$ rather than pursuing an aggressive strategy such as ranking unlabeled instances according to $\pi(.)$ and requesting labels only for the top few, since that would exacerbate the sampling bias.

Based on these observations, after obtaining a label for instance $\mathbf{x}_i$, we add it to the labeled set $\mathcal{L}$ with a weight of $1/\pi(\mathbf{x}_i)$, which unbiases the distribution of labeled data. Intuitively, this can be viewed as finely sampling the uncertain data points and adding them with small weights, while coarsely sampling the more certain data points and adding them with larger weights.

Unlike IWAL, we do not necessarily request a label for every point $\mathbf{x}_i$ drawn from the unlabeled pool. For some instances, including some for which $\pi(\mathbf{x}_i)$ is high due to disagreement among a pair of classifiers, it is possible that the overwhelming majority of the remaining classifiers is still in agreement. For such instances, particularly when η is low, we can short-circuit the labeling process and add $\mathbf{x}_i$ to $\mathcal{L}$ with the majority predicted label, without consulting the oracle. This form of label prediction, which can be viewed as a variant of self-training, is employed only in cases where the ensemble is confident, formalized as $s(\mathbf{x}_i, \mathcal{H}) > \theta$, where

$$s(\mathbf{x}_i, \mathcal{H}) = \frac{1}{|\mathcal{H}|} \left| \sum_{h_k \in \mathcal{H}} h_k(\mathbf{x}_i) \right|. \quad (2)$$

In practice, we employ a fixed threshold $\theta = 0.75$ for the experiments reported in this paper. One would expect label prediction to occur more frequently and with greater accuracy when η is low; this is confirmed in our experiments below.

Additionally, after requesting T=50 labels from the oracle, we can retrain our ensemble classifier set $\mathcal{H}$ from the current training set $\mathcal{T}$ using a fresh round of bootstrap sampling. We elect not to retrain at every iteration simply for computational reasons. Retraining the ensemble set is another major point of difference between our proposed method and bootstrap IWAL, which uses the same set throughout active learning. As our ensemble becomes a more representative sample of $\mathcal{X}$, we can dynamically adjust $p_{\min}$ according to a schedule to reflect our increased confidence. However, in the experiments below, we employ a fixed $p_{\min}$ = 0.1 to make it easier for others to duplicate our results.

The final classifier is obtained by training a classifier using all of the labeled data (from both oracle and predictions) in $\mathcal{L}$. In practice, this would be done once at the termination of active learning. However, in our experiments shown below, to show the incremental progress of the different algorithms, we train classifiers throughout the active learning process on the weighted set of labeled data to date and show performance on a held-out test set as a function of requested labels.

4. Experimental Results

We have conducted a comprehensive series of experiments using several standard datasets. Fig. 1 presents results on three datasets: *mushroom* and *spambase* (both from the UCI[1] Machine Learning Repository, and *MNIST*[2] digits.

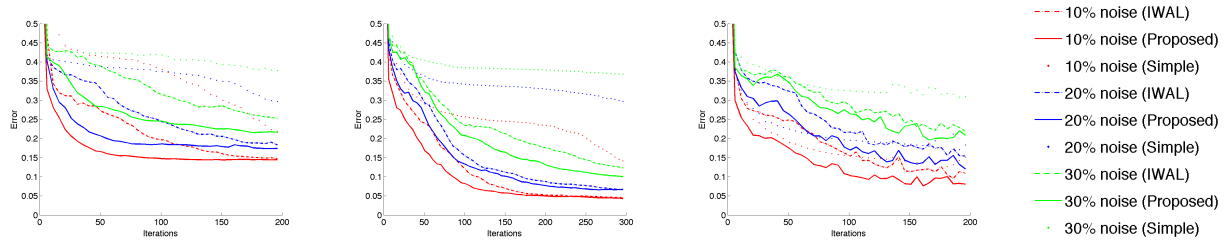
To enable repeatability and a controlled evaluation testbed, we corrupt the data using different levels of i.i.d. annotation noise; results shown here employ $\eta = \{0.1, 0.2, 0.3\}$, with knowledge of η withheld from the algorithm.

Fig. 1 summarizes our results. We compare four algorithms: (1) Tong & Koller [12], denoted as "aggressive"; (2) IWAL [2]; (3) Proposed method. Each curve is the average computed over 50 independent runs. To ensure a fair comparison, aggressive, IWAL and proposed were all initialized with exactly the same set of randomly-selected data ($\mathcal{T}$) in a given run. The graphs plot accuracy on the test set for a classifier trained only using the selected data to date, against the number of requested labels. We discuss the results on each dataset in greater detail below.

Fig. 1(a) shows classification results on the *spambase* dataset, which consists of 4601 instances with 56

[1] http://archive.ics.uci.edu/ml/

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



(a) spambase (b) mushroom (c) MNIST

Figure 1. Classification error on three standard datasets. Proposed method (solid line) outperforms both Tong & Koller (aggressive) and importance-weighted active learning (IWAL).

attributes, and 7% internal misclassification error. We use PCA to reduce the input dimensions from 56 to 20. Fig. 1(b) presents results on *mushroom*, which contains 8124 instances with 23 attributes. Finally, Fig. 1(c) shows results on *MNIST*, where we use the digits '3' and '5' for testing. The dataset has 1902 instances with 784 attributes. We use PCA to reduce dimensionality to 20. In all cases, we seed the active learning with a pool of 200 randomly-selected instances that are labeled with the same noise rate, from which K=41 bootstrap samples, each of $|\mathcal{T}|$=40 are selected.

We see that in all cases, the proposed method (solid line) outperforms both aggressive and IWAL algorithms at the given noise levels. We observe that aggressive active learning typically converges to a higher error rate, confirming our belief that such techniques are noise-seeking and can focus on only one portion of the dataset. As expected, we see that the asymptotic behavior of our method is similar to that of IWAL, but the label prediction enables the proposed method to perform better with fewer requested labels.

5. Conclusion

We present an approach to pool-based active learning motivated by IWAL that is designed to perform under noisy conditions. Asymptotically, our algorithm performs as well as IWAL but makes more efficient use of the available data through label prediction. The ease with which our method can be implemented makes it practical, particularly for applications in crowdsourcing. In future work, we are particularly interested in extending our work to combat non-iid sources of label noise, which can be common in crowdsourced oracles.

Acknowledgments

This research is supported in part by DARPA N10AP20027.

References

[1] M. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *ICML*, 2006.

[2] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *ICML*, 2009.

[3] J. Q. Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009.

[4] C. Dagli, S. Rajaram, and T. Huang. Utilizing information theoretic diversity for svm active learning. In *ICPR*, 2006.

[5] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. CRC Press, 1994.

[6] R. Ganti and A. Gray. UPAL: Unbiased pool based active learning. In *AISTATS*, 2011.

[7] J. Howe. The rise of crowdsourcing. *Wired*, 2006.

[8] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *SIGIR*, 1994.

[9] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, 1999.

[10] B. Settles. Active learning literature survey. Technical report, University of Wisconsin, Madison, 2010.

[11] V. Sheng, F. Provost, and P. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *KDD*, 2008.

[12] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *JMLR*, 2, 2001.

[13] L. Zhao, G. Sukthankar, and R. Sukthankar. Incremental relabeling for active learning with noisy crowdsourced annotations. In *IEEE SocialCom*, 2011.

[14] Y. Zheng, S. Scott, and K. Deng. Active learning from multiple noisy labelers with varied costs. In *ICDM*, 2010.