XXXX:Iterative Mislabel Detection with Early Loss and Influence

Anonymous Author(s)

ABSTRACT

Abstract here.

ACM Reference Format:
Anonymous Author(s). 2023. XXXX:Iterative Mislabel Detection with Early Loss and Influence. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (SIGMOD' 24).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Introduction here.

Problem. Our main research question in this paper is how to *detect mislabeled data in the large training set which will be used for the downstream meachine learning model* and filter it to improve the data quality of training set.

A positive answer to this question is crucial as it can help machine learning model to learn more correctly about the distribution of the training set, so as to obtain better model performance.

Challenges. *Nan* **[Add after we have the technical sections.]**

Our Proposal. Our main idea is to find out the instance in the training set which is labeled incorrectly. More specifically, *Nan* **[Add more details later.]**

Contributions. Our contributions are summarized as follows:

(1)
(2)
(3) *Nan* **[If we can have a good summary of experiments, we can either add here or use a "stitle" section to highlight the empirical findings.]**

2 PRELIMINARY

2.1 Problem Definition

Detection with parameter(gradient?) approximation

The setting of our problems.

Input: A trainging set which contains mislabeled data.

Output: A subset of the initial training set whose parameters are cloest to those of the model trained by removing all the real dirty data in the training set.

Complexity analysis: NP

2.2 Early Loss

The accuracy on "clean" samples is higher than on the "bad" samples, especially in the initial epochs of training. In other words, the training loss on correctly labeled instances is higher than that of the incorrectly labeled instance.

2.3 Parameters Influence

Parameters of the model which is trained by a dataset only contains "clean" samples are more likely to have a larger change when adding one "bad" sample into the dataset, compared with adding a "clean" sample.

3 FRAMEWORK

Nan **[Draw a good figure and clearly explain each step.]**

(1)
(2)
(3)

Nan **[I found that many reviewers will complain that (1) the above figure is too complicated to understand, and (2) the explanation of the figure is not clear. If the reviewers can fully understand and appreciate this framework figure and we have good empirical results, it is more than 50% done.]**

4 INFLUENCE FUNCTION

Theoretical guarantee about the influence function. (add a sample)

5 STOP CONDITION

6 EXPERIMENT

6.1 Experimental Settings

Dataset.We evalute our algorithm on 7 different real-world datasets from a diverse array of domains, whose size varying from the magnitude of 10^3 to 10^6 and the number of class differ from 2 to 100. We first used the dataset form CleanML, which have been performed synthetic mislabel injection with the strategy that flipping 5% of the labels in each class. Then, we also use different kinds of dataset, e.g. image,and dataset with large number of classes to further verify the effectiveness and scalability of our algorithm. The details are listed as follows:

(1) USCensus: This dataset contains 32,561 items about US Census records for adults. Each item has 14 attributes, such as age, education, sex, etc. The classification goal is to predict whether the adult earns more than $50,000.
(2) Marketing: The dataset is about household income,which consists of 8,993 records. A totoal of 14 demo- graphic attributes of each record varying form education to sex. The classification task is to predict if the annual household income is less than $25,000.
(3) EEG: This is a dataset of 14,980 Electroencephalogram recordings with 14 Electroencephalogram attributes e.g., AF3, AF4. The classification task is to predict whether the eye-state is closed or open.
(4) CIFAR-10: The dataset is a computer vision data set for universal object recognition, which contains 50000 32 X 32 RGB color pictures, a total of 10 categories. The task is to predict which kind does the picture belong to. Mislabeled data are artificially introduced by flipping labels of 40% for each type of the dataset randomly. **(need explain?)**
(5) CIFAR-100: This dataset is like CIFAR-10, except that it has a total of 100 classes, and each class contains 500 images. The

classification task is also to predict the category of a given picture.

(6) CovType: This is also a muti-classification dataset, which contains 7 different forest cover type. With a total of 581012 samples, and each sample consist of 54 attributes, such as Elevation, Wilderness Area, Horizontal Distance To Roadway etc.

(7) Mobile Price Prediction: This is a small tabular dataset with only 2000 records. The task is to predict price range of the mobile on the basis of the information about the mobile, specification like Battery power, 3G enabled , wifi ,Bluetooth, Ram etc.**(need remove?)**

(8) *Nan* **[If we can have a good summary of experiments, we can either add here or use a "stitle" section to highlight the empirical findings.]**

Method.We compare our approach against several competing mislabel detection methods. First, we consider 8 baselines:

1. **K-Nearest Neighbor(KNN):** The method counts the number of inconsistencies between the label of a training instance and the labels of its surrounding neighbors. If there is strong evidence of distinction among the labels, the training instance is marked as mislabeled. This kind of method is called local learning method.

2. **Nearest Centroid Neighborhood (NCN):** This method is also belong to local learning method, which assumes that the labels of mislabeled instances tend to disagree with the labels of other instances in their surrounding neighborhood. The difference between KNN and this method is the approach about how to find the nearest neighbor.

3. **Training Set Debugging Using Trusted Items(DUTI):** DUTI utilizes a small set of additional "trusted items"to help detect incorrectly labeled item, which core is to finds the smallest changes for the labels in training set such that the classifier trained on the changed dtaset classify all the trusted items correctly.

4. **Ensemble-based method with consensus filter:** Ensemble-based method assumes that multiple, independent clissifiers often result in conflicting labels about incorrectly labeled training sample. Algorithms that belong to this category vary in terms of how the different classifiers are constructed. Besides, the consensus filter is a strategy which means that a training example could be marked as a mislabel data only if it is misclassified by all the classifiers in the ensemble.

5. **Ensemble-based method with majority vote:** The main idea of this method is tha same as the Ensemble-based method with consensus filter.The only inconsistency between them is that majority vote strategy considers an example to be mislabeled if it disagrees with the majority vote of the classifiers.

6. **Cleanlab:**

7. **Noisy Cross-Validation(NCV):** This method first randomly devide a noisy training set into two halves, then train a neural network for these two half separately. After that, the network which is trained on one half will be applied to another half of the dataset. A sample would be identified as mislabel when its current label is different from its predicted label.

8. **Iterative Noisy Cross-Validation(INCV):** Obviously, this is a iterative method about noisy cross-validation. Apart from selecting mislabel samples, the INCV removes samples that have large categorical cross entropy loss at each iteration.

9. **Self-Ensemble Label Filtering(SELF):** The method

10. **Partition Filter:** This method first partition a noisy dataset into multi-subsets, and then construct a good classifier from each subset. For a given training sample, all classifier will be applied on it, the mislabel sample often have higher probability to have larger misclassified times.

We also consider three variants of our algorithm. The goal is to compare the effectiveness of the cross-validation using both early loss and parameter influence. For all variations, we use the same training paradigms. We consider the following variants:

1. **Without Iteration:**
2. **Without Parameter influence:**
3. **Without Early Loss**

Evaluation Metrics.We mainly focus on the effectiveness of our algorithm and other baselines, so we take f1-score as the most important metric.

6.2 Comparison with Baselines Settings

6.3 Mislabel Ratio Evaluation

6.4 Mislabel Distribution Evaluation

6.5 Model Evaluation

6.6 Stop Condition Evaluation

7 RELATED WORK

8 CONCLUSION

REFERENCES