Report

Noise-robust Classification from Positive Unlabeled Data

Goutam Agarwal

Abstract

Positive and Unlabeled learning (PU learning) aims to train a binary classifier based on only positive and unlabeled examples, where the unlabeled examples could be either positive or negative. This paper proposes a novel PU learning algorithm termed "Loss Decomposition and Centroid Estimation" (LDCE). LDCE converts PU learning into a risk minimization problem in the presence of false negative label noise by regarding all unlabeled examples as negative. By decomposing the hinge loss function and estimating the centroid of negative examples, LDCE effectively mitigates the adverse effects of label noise.

1 Introduction

Traditional supervised machine learning methods assume the availability of negative training data, which is not always the case. Positive and Unlabeled learning (PU learning) addresses this issue by training a binary classifier using positive data and unlabeled data. Existing PU learning methods handle unlabeled data in three main categories, but often suffer from inaccurate identification of negative examples or ineffective handling of label noise. This paper proposes LDCE, which treats unlabeled examples as noisy negatives and formulates PU learning as a noisy label learning problem.

2 Literature Survey

Positive and Unlabeled learning (PU learning) has garnered significant attention in the field of machine learning due to its practical applications in scenarios where only positive examples and unlabeled data are available. Over the years, several approaches have been proposed to address the challenges posed by PU learning. In this section, we review some of the key methodologies and advancements in the field.

2.1 Two-step Strategy

One of the earliest approaches to PU learning is the two-step strategy, where reliable negative examples are first identified from the unlabeled data, followed by training a traditional supervised classifier. Works such as [Liu et al., 2003; 2002; Li and Liu, 2003] fall under this category. However, the accuracy of identifying negative examples can greatly impact the performance of the final model, making this approach sensitive to misclassification errors.

2.2 Cost-sensitive Learning

Our paper also closely related with [Lee and Liu, 2003] Wee Sun Lee and Bing Liu. This research paper addresses the challenge of learning from positive and unlabeled examples using linear functions. It advocate for the utilization of logistic regression on weighted examples, coupled with a performance measure estimate from positive and unlabeled data, to effectively select the regularization parameter via validation set analysis

2.3 Noisy Label Learning

A subset of PU learning methods treats unlabeled examples as negative with label noise, acknowledging that some positive examples may be mislabeled as negative. By framing PU learning as a noisy label learning problem, these methods aim to build robust classifiers resilient to label noise. Biased SVM [Liu et al., 2003] is an example of this approach, where trade-off parameters are used to control the noise rate. However, the performance of such methods heavily relies on the selection of these parameters.

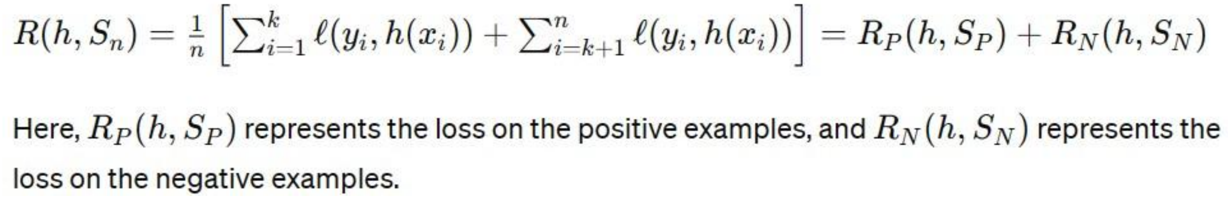
3 Methodology and Approaches

In this section, we present the proposed methodology for addressing the problem of Positive and Unlabeled (PU) learning in the context of sentiment analysis on social media data. Our approach, termed Loss Decomposition and Centroid Estimation (LDCE), aims to improve classification accuracy by explicitly modeling label noise in the negative set and estimating the centroid of noisy negative examples. We provide details on the proposed project, its differentiation from existing work, the machine learning method to be employed, and other required details.

3.1 Proposed Project Details

In this paper, the focus is on positive and unlabeled (PU) learning, where a set of n training examples consists of k positive examples followed by n-k unlabeled examples That might be positive or negative. The objective of the proposed project is to train a binary classifier h on this mixed set such that it accurately assigns labels to unseen test examples.

In our scenario, the risk of classifier h consists of two components. The first component is the loss on the clean positive examples, and the second component is the loss on the corrupted negative examples, defined as:

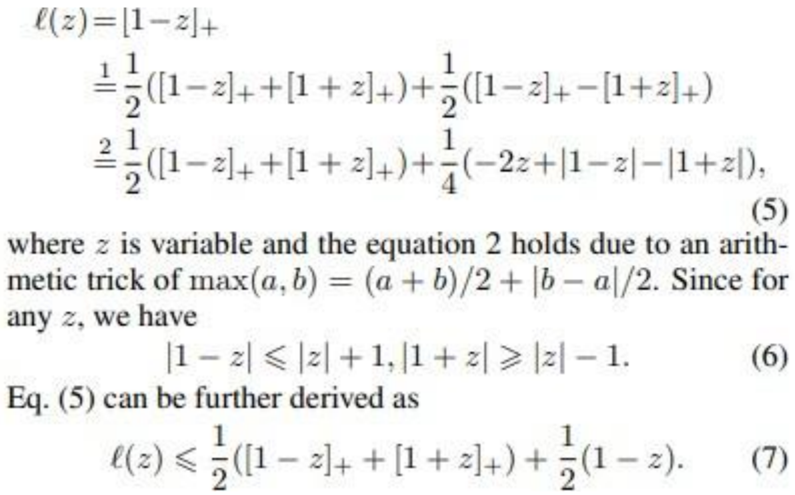


$$R(h, S_n) = \frac{1}{n}\left[\sum_{i=1}^{k} \ell(y_i, h(x_i)) + \sum_{i=k+1}^{n} \ell(y_i, h(x_i))\right] = R_P(h, S_P) + R_N(h, S_N)$$

Here, $R_P(h, S_P)$ represents the loss on the positive examples, and $R_N(h, S_N)$ represents the loss on the negative examples.

In this The loss on positive part can be easily computed as ground truth labels are known to us, the problem lies in computing loss on corrupted negative part, thus in the section below we will define how negative loss is handled.

Here we are deriving the bound on negative loss by further decomposing into two parts as one part composed of even function and other part composed of odd function from which further only the odd part is affected by the noise.



$$\begin{aligned}\ell(z) &= [1-z]_+\\ &\overset{1}{=} \frac{1}{2}([1-z]_+ + [1+z]_+) + \frac{1}{2}([1-z]_+ - [1+z]_+)\\ &\overset{2}{=} \frac{1}{2}([1-z]_+ + [1+z]_+) + \frac{1}{4}(-2z + |1-z| - |1+z|),\end{aligned} \tag{5}$$

where z is variable and the equation 2 holds due to an arithmetic trick of $\max(a,b) = (a+b)/2 + |b-a|/2$. Since for any z, we have

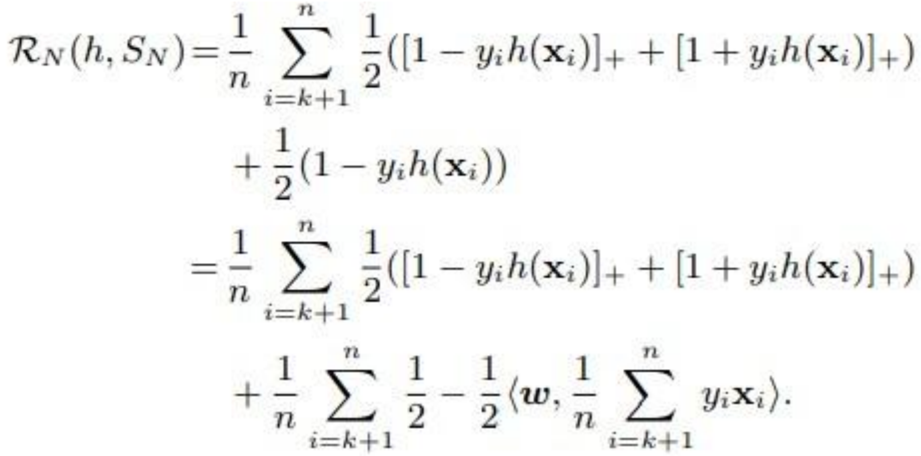
$$|1-z| \leqslant |z| + 1, |1+z| \geqslant |z| - 1. \tag{6}$$

Eq. (5) can be further derived as

$$\ell(z) \leqslant \frac{1}{2}([1-z]_+ + [1+z]_+) + \frac{1}{2}(1-z). \tag{7}$$

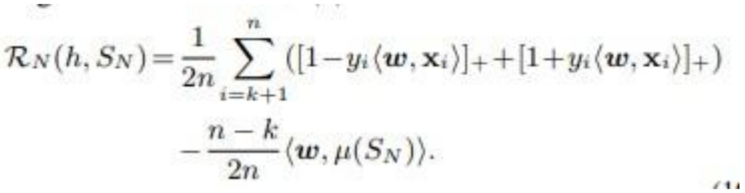
Here equation 7 is the bound obtained on the negative loss which consist of two parts first being even function in z and other is odd function so only odd parts is affected and now we will proceed our calculation on odd part only

First step is to assume that linear classifier is h=<w,xi> and by introducing this our equation 7 will be constructed as



$$\begin{aligned}\mathcal{R}_N(h, S_N) &= \frac{1}{n}\sum_{i=k+1}^{n} \frac{1}{2}([1 - y_i h(\mathbf{x}_i)]_+ + [1 + y_i h(\mathbf{x}_i)]_+)\\ &\quad + \frac{1}{2}(1 - y_i h(\mathbf{x}_i))\\ &= \frac{1}{n}\sum_{i=k+1}^{n} \frac{1}{2}([1 - y_i h(\mathbf{x}_i)]_+ + [1 + y_i h(\mathbf{x}_i)]_+)\\ &\quad + \frac{1}{n}\sum_{i=k+1}^{n}\frac{1}{2} - \frac{1}{2}\langle \boldsymbol{w}, \frac{1}{n}\sum_{i=k+1}^{n} y_i \mathbf{x}_i\rangle.\end{aligned}$$

In the above equation h(xi) is the predicted value and yi is true label. So first part can be computed easily but in 2nd part of odd function the term <w,1/n(sum(i=k+1 to n)yixi> can not be determined easily. To handle this we had introduced a term centroid explicitly for positive and negative part as positive centroid and negative centroid like $\mu(S_N) = \frac{1}{n-k}\sum_{i=k+1}^{n} y_i \mathbf{x}_i$and $\mu(\widetilde{S_N}) = \frac{1}{n-k}\sum_{i=k+1}^{n} \widetilde{y_i} \mathbf{x}_i$ After introducing this term in above equation , the equation Reduces as below:



$$\begin{aligned}\mathcal{R}_N(h, S_N) &= \frac{1}{2n}\sum_{i=k+1}^{n} ([1 - y_i\langle \boldsymbol{w}, \mathbf{x}_i\rangle]_+ + [1 + y_i\langle \boldsymbol{w}, \mathbf{x}_i\rangle]_+)\\ &\quad - \frac{n-k}{2n}\langle \boldsymbol{w}, \mu(S_N)\rangle.\end{aligned}$$

Now the overall estimation of loss lies in computation of centroid of positive part But it is also difficult to analyze it , hence for that sake we had taken support of the theorem mentioned in proposed paper which give us unbiased estimator to estimate centroid of positive part with the help of negative centroid. Result of theorem is E[u(Sn_bar)]=(1-2pn)u(Sn)
Here in above equation we introduced l-2 regularized norm and also a constraint on u(sn_bar) like $(\mu - \hat{\mu}(\widetilde{S_N}))^\top \hat{\Sigma}(\hat{\mu}(\widetilde{S_N}))(\mu - \hat{\mu}(\widetilde{S_N})) \leq \beta,$ here in the middle covariance matrix is included which is estimated simply by

$$\hat{\Sigma}[\mu(\widetilde{S_N})] = \sum_{i=k+1}^{n} \frac{\mathbf{x}_i^\top \mathbf{x}_i}{n-k} - \frac{1}{n-k} \sum_{i=k+1}^{n} \frac{\mathbf{x}_i^\top \widetilde{y}_i}{n-k} \sum_{i=k+1}^{n} \frac{\mathbf{x}_i \widetilde{y}_i}{n-k}.$$

Then the term u is calculated by using lagrange variable method as

$$\mu = \hat{\mu}(\widetilde{S_N}) + (\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1} \boldsymbol{w} \sqrt{\beta / (\boldsymbol{w}^\top (\hat{\Sigma}(\hat{\mu}(\widetilde{S_N})))^{-1} \boldsymbol{w})}.$$

Here all the terms except w are known to us where beta can be estimated by cross validation and w is computed using gradient descent

3.2 Differentiation from Existing Work

Unlike existing PU learning methods that often rely on two-step strategies, cost-sensitive learning, or noisy label learning approaches, LDCE introduces a novel framework that explicitly accounts for false negative label noise in the negative set. By decomposing the classification loss and estimating the centroid of noisy negative examples, LDCE offers a more robust and accurate solution in the performed experiments . This approach is fundamentally different from traditional methodologies and represents a significant advancement in the field of PU learning.

3.3 Machine Learning Method

The LDCE algorithm employs a supervised learning approach based on Support Vector Machines (SVM) with a modified loss function. Specifically, LDCE extends the standard SVM formulation to incorporate loss decomposition techniques, allowing for the separate modeling of noise in the negative class. Additionally, LDCE estimates the centroid of noisy negative examples to improve the decision boundary between positive and negative instances. By optimizing the modified loss function using gradient-based methods, LDCE learns a discriminative classifier that is robust to label noise

4 Dataset Detail

The experiments are conducted on synthetic datasets, UCI benchmark datasets, and real-world datasets including handwritten digit recognition datasets. The datasets are preprocessed and divided into positive, negative, and unlabeled sets as required for PU learning experiments. **Synthetic Dataset:**

The dataset comprises 1000 examples in total, featuring two classes: positive and negative. Each class corresponds to a Gaussian distribution, resulting in a two-dimensional dataset. There are 500 examples for each class, making it balanced

To match the dataset with required condition of the paper we had taken noise rate to be 0.2 that is considered k=400 which are truly positive and rest are considered as unlabeled with noise included in it.

Vote dataset:

The "vote" dataset from the UCI Machine Learning Repository is a collection of votes from the U.S. House of Representatives on 16 key issues. Each instance in the dataset represents a single congressman's vote on each issue, with attributes indicating their political party (Democrat or Republican) and their vote (yes, no, or unknown). This dataset contains 435 instances, with missing values denoted by '?' symbols.

As a part of preprocessing in the features including yes or no and some instances of ? encoded them in numerical values as 1 for yes -1 for no and 0(neutral) for ? . similarly in the labels we identified democrats and republicans as positive and negative class.

To satisfy the criteria as per proposed paper we have assumed noise rate firstly to be 0.2 so out of 267 positive instances 253 are recognised as true positive rest are considered as unlabeled and negative

5 Experiment

LDCE is compared with state-of-the-art PU learning methods such as Weighted SVM, Unbiased PU, and Non-Negative PU. The comparison is conducted on synthetic, Vote benchmark data, and real-world datasets using cross-validation. Here for the synthetic dataset we can examine that our proposed LDCE algorithm give us an hyperplane that perfectly separates the positive and negative points which is not in case of existing models

To perform the training on the above dataset we had trained our model with the help of two algorithms specified below

Algortihms 1 is basically dividing the corrupted dataset into g number of groups and then it will calculates the mean corresponding to each of the group then g number of lists will be created to capture the deviations of mean of each group from all other groups , then the median for each of these list will be stored in a different list and then the group corresponding to minimum median will be taken into consideration for further calculation of centroid

Here first algorithm computes u(Sn_bar) negative centroid estimator with which we can compute covariance matrix that is used in algorithm 2

Algorithm 2 uses gradient descent and algorithm 1 recursively to determine converged w . As part of training we had feeded our data(modified from original to meet criteria of proposed paper) to algorithm 2 to calculate parameters like covariance matrix(captures the strength of relationship between features of dataset).

Also the training examples (normally the whole modified dataset in all cases) are used to calculate negative centroid estimator by feeding the dataset to algorithm 1

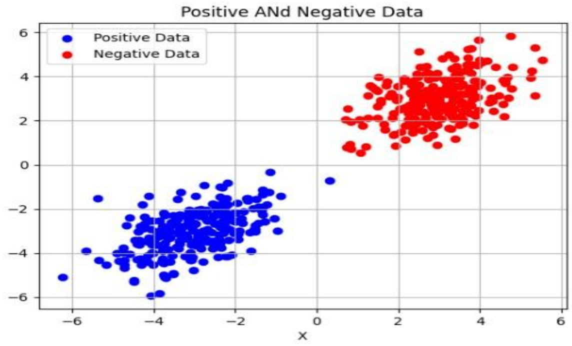
The optimisation settings that are embedded in above mentioned 2nd algorithms are gradient descent and lagrange variable method .

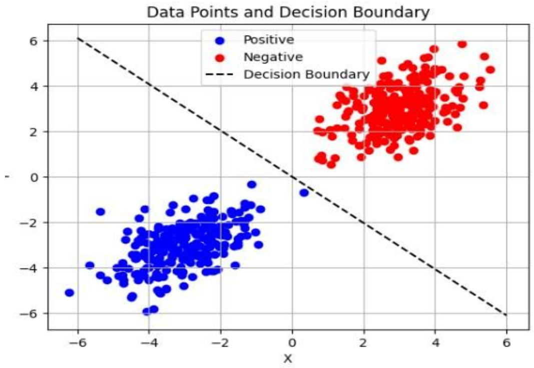
Gradient descent method is performed on the whole loss defined as one for positive part and another for negative part , by considering it as the objective function need to be minimized and lagrange method is used for calculating u(That was include as a variable in the constraints of objective function after defining the bounds over it)

6 Result

LDCE consistently outperforms existing PU learning methods across all experiments. Synthetic dataset experiments demonstrate LDCE's ability to accurately separate positive and negative examples even in the presence of label noise. Benchmark dataset experiments show significant performance improvements by LDCE compared to other methods. Real-world dataset experiments, particularly in handwritten digit recognition, highlight LDCE's effectiveness in practical applications.

Below are the significant plots obtained after performing training on synthetic dataset with the help of LDCE algorithm





After performing the experiment on Vote dataset (Modified after inclusion of noise in unlabeled) and then considering the dataset with Actual dataset as Testing Sample , Then Testing accuracy came out to be like below in our prescribed code:

Accuracy for test data of vote came out to be 0.86

7 Future Work

Currently we have performed our experiments on a few of the datasets mentioned in the proposed paper , So as part of the future work we will try to perform our experiment on all the datasets listed in our paper.

Also in algorithm 2 we had used gradient descent to find the absolute converged w. We had observed that in a few datasets including a significantly large amount of features The algorithm does not converge fastly , thus we will try to explore several possibilities that will handle this issue such as including Newton method instead of gradient descent.

Then with the modified version of the algorithm all the experiments will be performed again and respective results will be compared .

7 Work Done After Mid Term Review:

As part of our ongoing research, we were tasked with enhancing the efficiency of our LDCE (Local Data Cleaning and Enhancement) algorithm by incorporating stochastic gradient descent (SGD) in place of the conventional gradient descent optimization method. This modification aimed to expedite convergence time, particularly on large datasets, without compromising model performance.

Firstly, we defined stochastic gradient descent as a variant of gradient descent wherein the model is trained on random subsets of the dataset rather than the entire dataset at each iteration. This stochastic sampling approach introduces randomness, enabling faster convergence while still approximating the gradient of the objective function.

To implement stochastic gradient descent, we crafted a function that selects a specified number of random samples (in our case, 50) at each iteration and computes the gradient of the objective function based solely on this subset. This function utilizes random sampling to ensure diverse data representation across iterations, thereby enhancing model generalization.

Subsequently, we integrated this newly defined stochastic gradient descent function into our LDCE algorithm, replacing the previous gradient descent method. The only alteration required was to call the stochastic gradient descent function instead of the conventional gradient descent function within the LDCE algorithm.

For experimentation, we evaluated the modified LDCE algorithm on two distinct datasets. Firstly, we revisited the handwritten digit recognition dataset previously utilized in our research. Notably, the F1 score obtained on the testing dataset improved to 0.93, a significant enhancement from the previous implementation. Moreover, the convergence time drastically reduced from 45 minutes to a mere one minute, underscoring the efficacy of stochastic gradient descent.

Additionally, we applied the modified LDCE algorithm to the balance scale dataset, comprising 625 instances and five attributes. Preprocessing involved designating the "R" class as positive and all other classes as negative. Following dataset corruption per LDCE requirements, we calculated the weight vector using the newly integrated stochastic gradient descent approach. The resultant F1 score for this dataset stood at 0.53, indicative of the algorithm's capability to handle diverse datasets and maintain competitive performance.

8 Conclusion

This paper introduces a new learning method called "Loss Decomposition and Centroid Estimation" (LDCE) for handling partially labeled data. LDCE addresses the challenge of learning from positive and unlabeled (PU) examples by treating the unlabeled instances as negative with potential labeling errors. By doing this, LDCE transforms PU learning into a noisy label learning problem and employs loss decomposition to explicitly model these noisy labels. The key idea is to estimate the centroid of labeled examples accurately to mitigate the impact of noise. Experimental results on both artificial and real-world datasets demonstrate that LDCE outperforms existing PU learning methods.

All the results obtained with the code of our algorithm align with the results mentioned in the respective proposed paper such as perfectly separating the data points using hyperplane and aligning the accuracies of UCI benchmark datasets with accuracies mentioned in the paper. Also we had utilized stochastic gradient descent instead of gradient descent and apply the modified algorithm to larger dataset to check its efficiency.

References

1. Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In International Conference on Machine Learning, 2003. [icml-03.pdf (uic.edu)] Accessed on: 31th March, 2024.
2. Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In Proceedings of The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008. [posonly.pdf (ucsd.edu)]Accessed on: 31th March, 2024.
3. Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In Advances in Neural Information Processing Systems, 2013. [Learning with noisy labels | Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1 (acm.org)]Accessed on: 31th March, 2024.