Yilin Dong April 1, 2020

Medaboost — An Improved Ensemble Learning Algorithm in Classification with Multiple Annotations

by

Yilin Dong

Joyce Ho
Adviser

Computer Science

Joyce Ho

Adviser

Davide Fossati

Committee Member

Roberto Franzosi

Committee Member

2020

Medaboost — An Improved Ensemble Learning Algorithm in Classification with Multiple Annotations

By

Yilin Dong

Joyce Ho

Adviser

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Arts with Honors

Computer Science

2020

Abstract

Medaboost — An Improved Ensemble Learning Algorithm in Classification with Multiple Annotations

By Yilin Dong

Classification algorithms build models that can classify new observations based on their features. While those algorithms require a training set of samples' features and labels, in reality, many datasets do not meet the requirement. Since having experts to give out manual labels has a high cost, many industries adopted crowdsourcing, which enables a group of people to contribute to the same labeling task. However, multiple annotations from different annotators cannot apply to classification algorithms because they assume that labels are single and consensus. In this paper, we use truth inference methods to estimate single labels given different annotations from multiple annotators. While the Expectation-Maximization method provides the best accuracy, our empirical results suggest that better predictive performance can be achieved by accounting for disagreements. Thus, we propose Medaboost, a new predictive model, that considers the degree of disagreements between annotators to improve predictive performance. Medaboost outperforms AdaBoost on both synthetic dataset and MIMIC-III dataset under different sets of simulated nurses'.

Medaboost — An Improved Ensemble Learning Algorithm in Classification with Multiple Annotations

By

Yilin Dong

Joyce Ho

Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Arts with Honors

Computer Science

2020

Acknowledgements

I would like to thank my advisor, Joyce Ho, for her guidance through each stage of the process. In addition, I am also grateful to my other committee members, Roberto Franzosi and Davide Fossati, who also provided valuable suggestions at my thesis defense. Finally, I also want to thank Mani Sotoodeh who generated experimental datasets from MIMIC-III and provided editorial feedback of the manuscript.

Table of Contents

Chapter 1

Introduction

The supervised classification problem is one of the most common problems in the machine learning field. The idea is to predict a newly observed sample based on its features after learning a discriminative function from a set of training samples where the labels are known. However, many conventional supervised machine learning methods require a single, consensus label per sample. In reality, single and accurate labels are hard to get. Tasks that require human judgment, such as to recognize objects in a graph, to determine the condition of a system, and to categorize the topic in newspapers. Although manual labeling costs a big amount of money, the quality of labeling is still not guaranteed. Moreover, due to the growth of the internet and the cheaper data storage, unlabeled data piled up quickly. As a result, many machine learning studies and also various industries used crowdsourcing to deal with unstructured and unlabeled data. In crowdsourcing, a large group of people from different backgrounds with different levels of expertise can contribute to the same task. Since crowdsourcing lets a group of people cooperate on the same task, adopting it brings the advantage of labeling a great amount of data in reduced cost and shorter time. Since annotators have different levels of knowledge and personal bias, we need to infer the true labels based on those multiple annotations. This task introduces the truth inference problem.

It is important to obtain accurate labels to train the prediction model because inaccurate ones would train a highly biased model and thus make it meaningless to put the model into practice. A typical solution in truth inference problem is to summarize labels based on annotators' accuracy and bias. In this study, we apply and compare several truth inference methods (including Majority Voting, Weighted Majority, and Expectation-Maximization) to estimate the label as accurately as possible.

Adopting the result labels from truth inference methods to train a classification can build a decent model, but the information from the original multiple annotations is lost. In particular, we believe that the conflicts among multiple annotations are also valuable because this information indicates that certain samples have features that lie in the intersection between labels and thus cause difficulties for human labeling. Thus accounting for the uncertainty associated with the estimated labels can potentially help build a more robust predictive model.

In this work, we aim to construct a new classification model that learns from samples' features, their corresponding true labels (obtained from truth inference methods), and annotators' conflicts (disagreements) on labels. Based on this reasoning, we introduce a new approach called Medaboost, which can improve the performance of the Adaptive Boosting (Adaboost) by assigning a set of initial weights that build upon the information in multiple labels. The sample with a higher agreement ratio is assigned more initial weight in Medaboost because we want to focus on the samples that are more likely to be accurate and are not hard to identify.

We evaluate the efficacy of Medaboost with and without the change in initial weights across several settings. We first construct synthetic data to evaluate Medaboost. Then we apply it to real data in the healthcare field to solve the prediction problem of pressure ulcers or pressure injury.

1.1 A Case Study on Pressure Ulcers

Pressure ulcers (PUs) are a major health problem in the United States, with more than 2.5 million people suffering annually[1]. They are localized tissue damage, mainly caused by consistent pressure. Thus, this injury often occurs when patients have limited mobility. PUs can make patients suffer from pain, increase their morbidity, and are also quite costly given their chronic nature. Once PU occurs, patients will go through a painful recovery process as they can have infection, sepsis, and additional surgical cure. A study indicates that they cost $3.3 billion in the U.S. in 2008[2]. However, most of PUs can be avoided in their early stages if the nursing staff take proper prevention measures. Accurate identification of high-risk patients is crucial for effective prevention strategies and reducing the PU incidence.

However, a study showed that nurses are not accurate since they might misclassify the PU risk level for up to 30% of the patients[3]. An inter-rater agreement study by Waugh et al.[4] also showed that nurses have conflicting judgments on the same patient sample even when they were strictly following the same risk measure. They used Cohen's κ[5] to measure the agreement between nurses that corrects for the agreements by chance, and the Cohen's κ was only 0.472 for PU risk assessment on admission, a lot below the recommended value of 0.610[4, 6]. Therefore, nurses are not perfectly accurate, and they often have disagreements on detecting PU risks. Since predictive models are using labels from nurses, the fact that various nurses provide conflicting and inaccurate labels makes it unrealistic to assume single label in the model. Unfortunately, the existing models that predict the PU risk all use single label in both training and test datasets[7, 8, 9]. Thus, we apply our new model that accounts for not only the actual labels inferred from multiple labels but also the disagreements between nurses' labeling.

Chapter 2

Background

In this section, we introduce the truth inference problem, common methods, and provide details about Adaboost, an ensemble prediction model.

2.1 Truth Inference

Many conventional supervised machine learning methods require a single label per sample. To estimate a single label from multiple annotations, truth inference or truth discovery methods have been proposed to address different annotator assumptions. Here we introduce the general truth inference problem and briefly discuss three common truth inference methods: Majority Voting Algorithm (MV), Weighted Majority Algorithm (WM), and Expectation-Maximization Algorithm(EM). Additional truth inference methods are summarized in a survey by Li et al[10]

2.1.1 Problem Formulation

The general problem of truth inference given multiple noisy labels assumes we have N samples and M annotators. For each sample $i \in \{1, \cdots, N\}$, $y_{x_i}^{a_k} \in \{-1, 1\}$ represents the annotation made by the annotator $a_k : k \in \{1, \cdots, M\}$. The true and estimated label of sample x_i is denoted as z_{x_i} and $\hat{z}_{x_i}$ respectively. The truth inference task is to determine the estimated truth $\hat{z}_{x_i} \in \{-1, 1\}$ for each sample x_i so that it is the same as z_{x_i} for as many samples as possible.

2.1.2 Majority Voting Algorithm(MV).

MV is the simplest method to estimate the truth. Each annotation, $y_{x_i}^{a_k}$ serves as a vote. The estimated truth $\hat{z}_{x_i}$ is the value that gets the most votes (see Algorithm 1). In the scenario where there are equal votes (i.e., no clear majority), MV randomly selects one label as the estimated truth.

Though simple, MV outperforms most of the truth inference methods, especially when the accuracy of most of the annotators is high[11]. However, MV treats all annotators equally and ignores the reliability of annotators.

Algorithm 1 Majority Voting Algorithm

```
1: for $i = 1, 2, \ldots N$ do
       $$\hat{z}_{x_i} = \begin{cases} 1 & \text{if } \sum_{k=1}^{M} y_{x_i}^{a_k} > 0 \\ Random\{0,1\} & \text{if } \sum_{k=1}^{M} y_{x_i}^{a_k} = 0 \\ -1 & \text{otherwise} \end{cases}$$
2: end for
```

2.1.3 Weighted Majority Algorithm (WM).

WM was proposed to deal with the reliability of the annotators[12]. It assigns the initial weight 1 to each annotator ($w_{a_k} = 1$), iterates through each sample sequentially, and calculates the label according to the weighted vote. Whenever there is a mismatch between the estimated label and the annotator's label, WM reduces the annotator's weight by multiplying a factor $\beta \in [0, 1)$ so that those annotators with high reliability will have large impacts on the estimated truth. Thus unlike MV, each annotation has a weighted vote as $w_{a_k} \cdot y_{x_i}^{a_k}$ (see Algorithm 2).

Generally, WM is more robust compared to MV with respect to unreliable annotators.

Algorithm 2 Weighted Majority Algorithm

```
$w_{a_k} = 1$
for $i = 1, 2, \ldots N$ do
    $$\hat{z}_{x_i} = \begin{cases} 1 & \text{if } \sum_{k=1}^{M} w_{a_k} \cdot y_{x_i}^{a_k} > 0 \\ -1 & \text{else} \end{cases}$$
    Update weight for k = 1, ... M:
    if $y_{x_i}^{a_k} \neq \hat{z}_{x_i}$ then $w_{a_k} \leftarrow \beta \cdot w_{a_k}$, where $\beta \in [0,1)$
    end if
end for
```

2.1.4 Expectation-Maximization Algorithm(EM).

The EM (Expectation Maximization) truth inference method proposes a more complex model of the reliability of the annotators using a probabilistic graphical model[13]. Under EM, the annotators' reliability is modeled using a confusion matrix π^{a_k}, where each element represents the probability of annotator a_k giving label q given that true label is p.

For a binary classification problem, this confusion matrix is defined as:

$$\pi^{a_k} = \begin{bmatrix} \beta & 1-\beta \\ 1-\alpha & \alpha \end{bmatrix}, \text{ where } \alpha = p(y_{x_i}^{a_k} = 1 \mid z_{x_i} = 1) \text{ and } \beta = p(y_{x_i}^{a_k} = -1 \mid z_{x_i} = -1). \tag{2.1}$$

Note that α, β represent the probability of annotator a_k correctly annotating sample x_i given its true label 1 or -1 respectively. Higher α and β values correspond to a more reliable worker. The estimated ground truth, $\hat{\mathbf{Z}}$, and confusion matrices, $\pi^{\mathbf{A}}$, are iteratively computed using maximum likelihood estimation which optimizes the following:

$$\max_{\hat{\mathbf{Z}},\pi^{\mathbf{A}}} \prod_{i=1}^{N} \sum_{l\in\{-1,+1\}} p(\hat{z}_{x_i} = l) \prod_{a_k\in\mathbf{A}^{x_i}} \pi^{a_k}_{l,y^{a_k}_{x_i}} \tag{2.2}$$

where $\mathbf{A}^{x_i}$ indicates the set of annotators that have annotated sample x_i.

2.2 Adaboost: An Ensemble Classifier

Adaboost is an ensemble method that aimed at improving the performance of base models such as decision trees. It combines multiple weak classifiers to produce a robust classifier[14]. Suppose there are n samples in the training data $(x_1, \hat{z}_{x_1}), \cdots, (x_n, \hat{z}_{x_n})$, where x_i is the set of features for sample i, and $\hat{z}_{x_1}$ is the corresponding label (i.e. the estimated labels from the truth inference method). Then each sample i has a weight $W_t(i)$ associated with the t^{th} iteration. Adaboost assigns equal weights to each sample at first and increases the weights of misclassified samples in each iteration for the classifier in the next iteration. This way, mislabeling those samples will result in more loss, and thus the weak learning algorithm will be trained to avoid mislabelling them. Algorithm 3 shows the pseudo-code for Adaboost.

Algorithm 3 Adaboost algorithm

```
Input: Training data - $(x_1, \hat{z}_{x_1}), \cdots, (x_n, \hat{z}_{x_n})$, learning rate $\eta$, number of iterations $T$, and weak learner $h$
Initialize weight: $W_1(i) = \frac{1}{n}, i \in \{1, \cdots, n\}$
    for $t = 1, 2, \ldots, T$ do
        Find weak learner $h_t$ that minimizes the error $\epsilon_t$:
            $\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq \hat{z}_{x_i}}}^{n} W_t(i)$    (2.3)
        Choose $\alpha_t$:
            $\alpha_t = \eta \cdot ln(\frac{1-\epsilon_t}{\epsilon_t})$, where $\eta \leq 1$ is the learning rate    (2.4)
        Update weight for i = 1, ... n:
            $W_{t+1}(i) = \frac{W_t(i) exp(-\alpha_t \hat{z}_{x_i} h_t(x_i))}{N_t}$    (2.5)
    end for
    Output Classifier:
            $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$    (2.6)
```

Chapter 3

Medaboost: Multiple Experts Disagreement Aware Adaboost

We develop a new framework that builds a robust predictive model when there are multiple annotations (or labels) for each sample. Our framework involves two steps. First, we use an existing truth inference method to estimate the sample label from the multiple annotations. Then, we use the estimated labels and their features to train Medaboost, our proposed new classifier to account for disagreeing annotations. Figure 3.1 shows an overview of our model.

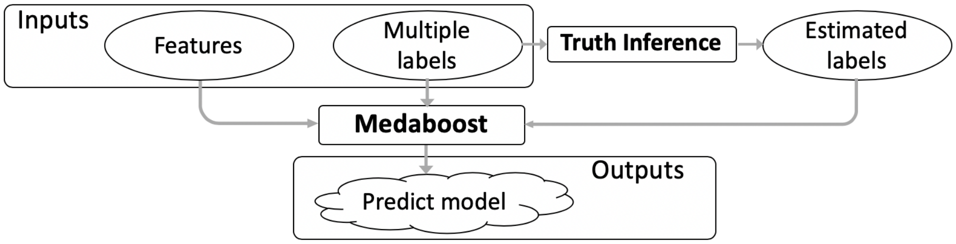


Figure 3.1: Overview of our approach.

3.1 Estimation of Sample Labels

In this study, the raw data is assumed to have only samples' features and their corresponding multiple annotations. A decent prediction model needs to learn from training data that contain single and correct labels. Otherwise, the model would be highly biased to the incorrect data. Therefore, we first need to obtain reliable estimated labels from the original sets of multiple labels. We use truth inference methods that we mentioned earlier to infer the ground truth. After obtaining reliable estimated labels, we adopt these target labels of the training data in our predictive framework. Since it is impossible for truth inference methods to reach 100% accuracy for every dataset, those wrong labels in the estimated labels bring challenges to the prediction model. Standard classifier models will only use the estimated label so they will be negatively impacted by the incorrect labels. Thus, we introduce a robust prediction model that also incorporates the information from multiple annotations to improve the performance of the existing model.

3.2 A Robust Prediction Model

A standard machine learning algorithm, Adaboost, can be vulnerable to overfitting in the presence of noisy labels, since it constantly increases the weights for misclassified samples. The underlying assumption in Adaboost is that all labels are accurate. However, the assumption is wrong in this case. As we explained earlier, the labels to train the model are not accurate because they are adopted from the result of truth inference methods. Even when we use multiple truth inference methods to make the estimation as accurate as possible, bias always exists. We believe that disagreements or conflicts in the multiple annotations may arise from the fact that the samples' features are quite complicated, making the assessment more challenging. Thus we posit that leveraging the uncertainty in samples instead of ignoring it by changing the Adaboost weights will improve the model performance.

Several studies have proposed a re-weighting scheme or changing the initial weights in Adaboost [15, 16, 17]. For instance, Jia st al. showed discarding some small weight samples during the boosting process can increase training speed while keeping prediction performance [15]. Kim et al. built a model based on Adaboost, which assumed that positively labeled samples have a more concentrated feature distribution than those negatively labeled samples[17]. They increased the initial weight for samples with features that are close to the peak of overall distribution[17]. However, their assumption was only proved in the dataset for object detection tasks, so it might not be the case for other datasets. Moreover, they applied different strategies to samples with different labels, so their model will also be worse off when we use inaccurate labels. Hu et al. changed the initial weight in Adaboost to reduce the false-alarm rate (which is the possibility that the model gives out positive labels when the true labels are negative). In their applications, the false-alarm rate raises unnecessary costs. However, we do not need a low false-alarm rate. In fact, in cases like detecting diseases, having a low false-negative rate is also important.

Although these methods are not applicable in this case, they proved that changing the initial weight in Adaboost has a significant impact on the formation of a strong classifier because it impacts the sum of weighted classification errors from the very beginning[16]. Therefore, we also adjust the initial weight to prevent Adaboost from overfitting toward the inaccurately labeled samples. Instead of setting all samples to be equal weight (i.e., $W_1(i) = \frac{1}{n}$), we propose to adjust the initial weights to be proportional to the level of observed conflict among annotators.

We believe those conflicting samples indicate that their features are hard to label and are more likely to have wrong estimated labels than consensus samples. Therefore, we choose to decrease their weight.

The initial weight $W_1(i)$ is calculated as:

$$d_{x_i} = \frac{1}{2}(M - |\sum_{k=1}^{M} y_{x_i}^{a_k}|) \tag{3.1}$$

$$W_1(i) = \frac{M - d_{x_i}}{d_{x_i} + M} \cdot \frac{1}{N} \tag{3.2}$$

d_{x_i} is the disagreement number for sample x_i, and N is the normalization factor. For instance: if M = 5, $y_{x_i}^{a_k} = 1$ for k = 1, 2, 3; $y_{x_i}^{a_k} = -1$ for k = 4, 5, then $d_{x_i} = 2$. Higher d leads to lower weight. Since $0 \leq d_{x_i} \leq M - 1$, we know $\frac{1}{(2M-1)N} \leq W_1(i) \leq \frac{1}{N}$. The conflicting samples might have weights close to zero, which means they are heavily discounted in terms of their contribution to the initial classifiers.

The pseudo-code for our entire framework is shown in Algorithm 4. The first part is to estimate the sample labels using an accurate truth inference algorithm. Then, we input the sample features and the estimated labels to Medaboost, which changed the initial weight from Adaboost model.

Algorithm 4 Our framework to deal with multiple annotations

```
Input: Training data - $(x_1, \hat{z}_{x_1}), \cdots, (x_n, \hat{z}_{x_n})$, learning rate $\eta$, number of iterations $T$, and weak learner $h$
Initialize weight for i = 1, ..., n:
    $d_{x_i} = \frac{1}{2}(M - |\sum_{k=1}^{M} y_{x_i}^{a_k}|)$    (3.3)
    $W_1(i) = \frac{M - d_{x_i}}{d_{x_i} + M} \cdot \frac{1}{N}$    (3.4)
for $t = 1, 2, \ldots, T$ do
    Find weak learner $h_t$ that minimizes the error $\epsilon_t$:
        $\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq \hat{z}_{x_i}}}^{n} W_t(i)$    (3.5)
    Choose $\alpha_t$:
        $\alpha_t = \eta \cdot ln(\frac{1 - \epsilon_t}{\epsilon_t})$, where $\eta \leq 1$ is the learning rate    (3.6)
    Update weight for i = 1, ..., n:
        $W_{t+1}(i) = \frac{W_t(i) exp(-\alpha_t \hat{z}_{x_i} h_t(x_i))}{N_t}$    (3.7)
end for
Output Classifier:
    $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$    (3.8)
```

Chapter 4

Experiment Setup

4.1 Data Description

In this study, we conduct experiments on two different datasets. We use a synthetic dataset to simulate the problem setting to explore the impact of several different weighing formulas. We then apply Medaboost on a real-world dataset, the MIMIC-III [18] critical care database. For both datasets, we have sample features and the ground truth label. We also discuss how to create multiple labels to simulate different expert annotations and their reliabilities. The details are explained below.

4.1.1 Synthetic Dataset

This model is aimed at solving the issue of no ground truth but only multiple conflicting annotations are available. However, we need the ground truth to conduct testing metrics for Medaboost's performance evaluation. Therefore, real datasets with both ground truth and annotations from multiple annotators are not easy to find. For the evaluation and investigation purposes, we generate a synthetic dataset.

Features and Labels. In this study, we construct a dataset with 10000 samples. For each sample $i \in \{1,...,10000\}$, it has 10 features ($f_j^{x_i}$, where $j \in \{1,2,...10\}$) and one binary ground truth ($\hat{z}_{x_i} \in \{-1,1\}$). We assign label -1 to 80% of the dataset, and label -1 to 20% of it in order to test whether our model can work well in imbalanced datasets.

To construct the features for the samples, we assume each class sample follows a multivariate Gaussian distribution. We construct two ten-dimensional spheres that are centered at their mean point and follow the standard multivariate normal distribution. The negative class is centered at mean point ($avg_{(-1)}$ =($f_{1,-1},...,f_{10,-1}$) where $f_{j,-1} = 1.5$) with a standard deviation (S.D) of 2.5. The positive class is centered at mean point ($avg_{(1)}$ =($f_{1,1},...,f_{10,1}$) where $f_{j,1} = 5$) with a standard deviation of 2.5. Figure 4.1 shows the normal distribution of the first of the 10 features $f_1^{x_i}$ from the two clusters. The left curve represents the distribution of label -1, and the right one represents that of label 1. We assume that it is easy for annotators to determine the class when the probability density of one class is much higher than the other one. The overlap between the two classes makes it hard to give accurate labels in the intersection.

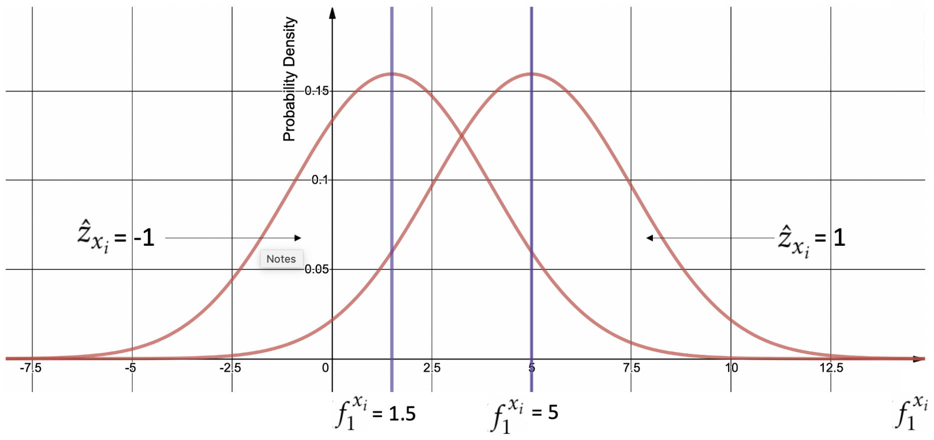


Figure 4.1: A single feature distribution depending on the two classes. The one has a mean of 1.5 is the negative class while the one has a mean of 5 is the positive class.

For each sample, we first determine whether it belongs to the negative class or the positive class (at a ratio of 80:20). Each sample is then drawn from its class-specific multivariate normal distribution. Figure 4.2 shows a 3-d visualization of the first 3 features $(f_1^{x_i}, f_2^{x_i}, f_3^{x_i})$ drawn from our multivariate Gaussian distribution. It is important to note that during the construction of the features, the actual ground truth label is known for every sample.

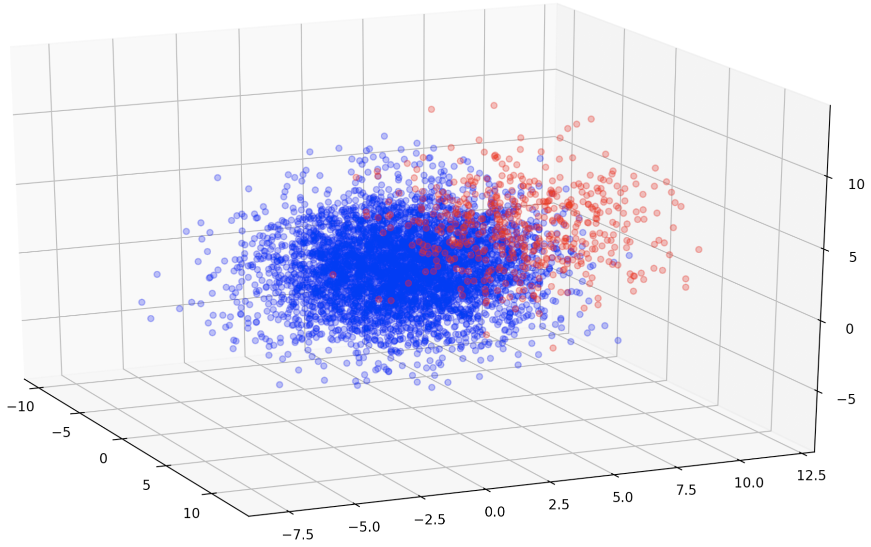


Figure 4.2: A Dataset with Three Features. Blue dots represents negative label, and red dots represents positive label

Multiple Annotations. In this study, we construct the 10 annotations $y_{x_i}^{a_k} \in \{-1, 1\}$ from 10 annotators $a_k, k \in \{1, ..., 10\}$ for each sample i. To simulate annotations from multiple annotators, we assume that the annotators become less accurate when labeling a sample which has a set of features that are located at the intersection of the two clusters. Under this assumption, data belong to the region where the features clearly belong to one of the labels, those annotators often agree with each other. However, in the region of the intersection between the two labels, it is not easy to distinguish between two labels, so the disagreements between annotators increase.

In order to construct the 10 annotators' labeling based on our assumed situation, we first calculate the Euclidean distance ($d_{x_i,avg_{\{-1,1\}}}$) between the sample's mean point ($avg_{(-1)}$ or $avg_{(1)}$) and the sample x_i's feature point ($f_1^{x_i},...,f_{10}^{x_i}$) by using the formula:

$$d_{x_i,avg_{\{-1,1\}}} = \sqrt{\sum_{j=1}^{10}(f_{j,\{-1,1\}} - f_j^{x_i})^2} \tag{4.1}$$

The annotation standard for all annotators is defined using the Euclidean distance and the clusters' standard deviations. $S.D_{-1}$ is the standard deviation of class -1, and $S.D_1$ represents that for class 1. First, we calculate the value of $d_{x_i,avg_{-1}} - d_{x_i,avg_1}$, which is the distance between class -1's mean point and the sample point minus distance between class 1's mean point and the sample point. If the sample is closer to one of the class mean, then it is more likely that this sample belongs to that class (or label). Then, we divide the possible values of this difference into 8 groups, the division is based on the two clusters' standard deviations (see Equation (4.2)). We calculate the true percentage of $z_{x_i} = -1$ and $z_{x_i} = 1$ in each of the 8 conditions. Equation(4.2) shows the percentage of $z_{x_i} = -1$. We set the truth to be our annotation standard.

$$P(z_{x_i} = -1) = \begin{cases} 1, & \text{for } d_{x_i,avg_{-1}} - d_{x_i,avg_1} \leq -(S.D_{-1} + S.D_1) \\ 1, & \text{for } -(S.D_{-1} + S.D_1) < d_{x_i,avg_{-1}} - d_{x_i,avg_1} \leq -\frac{1}{2}(S.D_{-1} + S.D_1) \\ 0.99, & \text{for } -\frac{1}{2}(S.D_{-1} + S.D_1) < d_{x_i,avg_{-1}} - d_{x_i,avg_1} \leq -\frac{1}{4}(S.D_{-1} + S.D_1) \\ 0.91, & \text{for } -\frac{1}{4}(S.D_{-1} + S.D_1) < d_{x_i,avg_{-1}} - d_{x_i,avg_1} \leq 0 \\ 0.56, & \text{for } 0 < d_{x_i,avg_{-1}} - d_{x_i,avg_1} \leq \frac{1}{4}(S.D_{-1} + S.D_1) \\ 0.22, & \text{for } \frac{1}{4}(S.D_{-1} + S.D_1) < d_{x_i,avg_{-1}} - d_{x_i,avg_1} \leq \frac{1}{2}(S.D_{-1} + S.D_1) \\ 0, & \text{for } \frac{1}{2}(S.D_{-1} + S.D_1) < d_{x_i,avg_{-1}} - d_{x_i,avg_1} \leq (S.D_{-1} + S.D_1) \\ 0, & \text{for } (S.D_{-1} + S.D_1) < d_{x_i,avg_{-1}} - d_{x_i,avg_1} \end{cases} \tag{4.2}$$

To simulate annotators' actions and adjust their accuracy, we set two parameters to make them deviate from the ground truth. We use a set of 10 deviation factors dev^{a_k} to differentiate the 10 annotators. For instance, the actual possibility that an annotator a_k labels sample x_i as -1 ($P(y_{x_i}^{a_k} = -1)$) would be the annotation standard that we calculated from the ground truth ($P(z_{x_i} = -1)$) plus the deviation factor dev^{a_k}. The purpose of having this factor is to make each annotator's action differ from each other. When $|dev^{a_k}|$ is high, the annotator deviates more from the ground truth and thus becomes more inaccurate. To control this factor, we introduce our first parameter $R \in (0,1)$ to limit the range of the deviation factor. i.e. $dev^{a_k} \in [-R,R]$. dev^{a_k} is a random number within the range.

Moreover, our assumption in this study is that when a sample gets closer to the overlapping area of two labels, it is more difficult for annotators to label accurately. Therefore, we introduce a set of degree factor $frac \in (0,1)$ to control the degree of deviation. When the sample is far from the overlapping area, which means $P(z_{x_i} = -1)$ and $P(z_{x_i} = 1)$ differs a lot, $frac$ should be small, and vice versa. Therefore, each condition in Equation (4.2) has one $frac$. In the Equation (4.3) below shows the possibility that an annotator a_k labels sample x_i as -1 ($P(y_{x_i}^{a_k} = -1)$).

$$\begin{gathered}\text{if } P(z_{x_i} = -1) < -frac \cdot dev^{a_k}, \text{ then } dev^{a_k} = -dev^{a_k} \\ P(y_{x_i}^{a_k} = -1) = P(z_{x_i} = -1) + frac \cdot dev^{a_k} \\ \text{where } P(y_{x_i}^{a_k} = -1) \geq 0, P(y_{x_i}^{a_k} = 1) \geq 0, dev^{a_k} \in (-R,R).\end{gathered} \tag{4.3}$$

The chance that this annotator labels 1 to the same sample is calculated as $P(y_{x_i}^{a_k} = 1) = 1 - P(y_{x_i}^{a_k} = -1)$.

10 sets of different $frac$ and R are selected, and the mean reliability and the S.D of reliability is calculated for each annotator.

Table 4.1 shows the ten datasets that we generated. The mean reliability of annotators ranges from 74% to 91%.

Table 4.1: Ten datasets with different parameters

No.	**frac**	**R for** dev^{a_k}	**Annotators' mean (S.D) reliability**
1	[0.1, 0.3, 0.8, 0.9, 0.9, 0.8, 0.3, 0.1]	0.8	82.2% (2.2%)
2	[0.2, 0.3, 0.6, 0.9, 0.9, 0.6, 0.3, 0.2]	0.7	85.1% (2.6%)
3	[0.2, 0.3, 0.4, 1, 1, 0.4, 0.3, 0.2]	0.6	88.1% (2.8%)
4	[0.3, 0.4, 0.6, 0.9, 0.9, 0.6, 0.4, 0.3]	0.8	77.6% (6.4%)
5	[0.1, 0.3, 0.4, 0.7, 0.7, 0.4, 0.3, 0.1]	0.7	91.4% (3.6%)
6	[0.1, 0.3, 0.4, 0.7, 0.7, 0.4, 0.3, 0.1]	0.0	89.2% (4.6%)
7	[0.3, 0.4, 0.6, 0.9, 0.9, 0.6, 0.4, 0.3]	0.8	77.2% (6.2%)
8	[0.1, 0.3, 0.4, 0.7, 0.7, 0.4, 0.3, 0.1]	0.65	86.2% (5.6%)
9	[0.2, 0.3, 0.6, 1, 1, 0.6, 0.3, 0.2]	0.6	86.8% (3.4%)
10	[0.3, 0.4, 0.8, 0.9, 0.9, 0.8, 0.4, 0.3]	0.9	74.8% (6.7%)

4.1.2 MIMIC-III Critical Care Database

We use the MIMIC-III [18] critical care database, a freely-available dataset, to allow for easy replication of our experiments. This dataset is one of the most commonly used benchmarks for analytic studies as it contains various information including demographics, lab results, diagnosis, and notes. MIMIC-III contains information about 38,597 patients admitted to intensive care units (ICU) at a large tertiary care hospital, between 2001-2012.

Cohort Selection

Hospital stays were chosen as our unit of prediction, to resemble the real-world situation of nurses' assessment of charts. We first discarded non-sensible hospital stays such as records containing negative length of stay ($\sim 50K$ unique stays had reasonable stays). Since the prevalence of PU is extremely low in the younger population, hospital stays of individuals aged 20 years or less were removed. In addition, only hospital stays between 2-120 days were considered, since lab events were not likely to be measured for shorter stays. Longer stays (more than 120 days) also indicated medically complex patients. The exclusion criteria reduced the number of unique hospitals stays to $\sim 26K$.

Establishing PU. To establish sufficiently reliable ground truth labels (i.e. presence of PU or its absence), we consider two sources of information, the ICD-9 diagnosis codes and notes for each hospital stay. We used the following ICD-9 codes to determine the potential presence of PU: [707, 707.1, 707.2, 707.3, 707.4, 707.5, 707.6, 707.7, 707.9, 707.11, 707.21, 707.22, 707.23, 707.24, 707.25]. We also searched for the following keywords in the notes: [Pressure Ulcer Prevention, Skin Surveillance, Decubitus Ulcers, Impaired Tissue Integrity, Impaired Skin Integrity, Bed Sores, Pressure Ulcer, Pressure sore].

If both sources indicated the presence of PU, this constituted as a positive sample. Similarly, any stay that did not have any indication in both sources was labeled as a negative sample. A stay that indicated PU either in only the notes or in the ICD-9 codes were not used due to the potential ambiguity.

Establishing PU Case/Control Samples. The ratio of positive samples (PU) to negative samples (no PU) in the 26k hospital stays is significantly imbalanced (3.5%) and can negatively impact the predictive model. Thus, we designed a case-control study. Case-control study is very common in medical research and has also been suggested for use in Artificial Intelligence for healthcare[19]. Each positive PU stay is matched with 4 negative PU stays based on similarity in terms of age, gender, the total length of stay, and the ICU length of stay. The number of potential negative matches was chosen as 4 to introduce enough diversity in matched stays' characteristics. Negative samples do not need to be unique for each positive sample. In fact, some negative samples are matched with multiple positive ones. A total of 856 stays were identified as positive samples for PU and an additional 2733 stays were selected as negative samples for PU for our study. Thus, the final cohort contains 3589 hospital samples with a 31.3% prevalence rate of PU.

Feature Construction. For our prediction task, multiple layers of features are constructed from various tables in MIMIC-III to be merged together. The layers include demographics, counts of times each ICD-9 diagnosis class appears, and lab measurements during each hospital stay. The demographics of the patient for each hospital stay were extracted from MIMIC-III Admissions and Patients tables, we partly used [20] for these preprocessing steps. To establish diagnosis classes, we used the standard chapters of ICD-9 diagnosis codes [21] and summed the number of diagnosis codes falling into each chapter for building diagnosis features. For lab events features, we used the MIMIC-Extract pipeline [22] and aggregated the lab results for each day. We also used only the first 2 days of hospital stay average lab measurements. Missing lab measurements were imputed using its mean across available stays. All lab measurement features were normalized using the min-max normalization. A summary of the features in each layer, used in our prediction task, is given in Table 4.2. A total of 89 features were used for each hospital stay.

Table 4.2: Layers of Features used in Prediction of PU

Feature Layer	**Features Available in that Layer** ***(Number of Features in the layer)***
Demographics	{Age, Gender, ICU Type, Admission Type, Insurance Type, Religion, Ethnicity, Marital Status, Length of Stay in ICU, Total Length of Stay} (**10 Features**)
ICD-9 Diagnosis Charts' Counts	{ Blood, Circulatory, Congenital, Digestive, Endocrine, External, Genitourinary, Ill defined, Infectious, Injury, Mental, Muscular, Neoplasms, Nervous, Pregnancy, Prenatal, Respiratory, Skin conditions except PU} (***18 Features***)
Lab Measurements	{Albumin Bilirubin, Calcium, Cardiac Index, Chloride, Cholesterol, Co2, Creatinine, Diastolic Blood Pressure, Eosinophils, Fibrinogen, Fraction Inspired Oxygen, Glucose, Heart Rate, Height, Hematocrit, Hemoglobin, Lactate Dehydrogenase, Lactic Acid, Lymphocytes, Magnesium, Mean Blood Pressure, Monocytes, Neutrophils, Oxygen Saturation, Ph, Phosphate, Phosphorous, Platelets, Potassium, Red Blood Cell Count, Respiratory Rate, Sodium, Venous Pvo2, Weight, White Blood Cell Count, etc.} (***61 Features***)

Simulation of Nurse Annotation. To simulate a group of nurses labeling stays for PU, we consider two parameters: (1) mean reliability of nurses, or how likely on average the group of nurses are to annotate a true case of PU as positive; and (2) variability in nurses' degrees of expertise to reflect the seniority or training received. A previous study showed that the inter-rater agreement between nurses for PU is around 70%[4]. We chose the two parameters' ranges based on this evidence to generate annotations. Moreover, to capture the true conditions of the PU annotation process, we conducted our experiments with 3 and 5 nurses, since it is infeasible to employ more nurses for annotations.

As is typical in crowdsourcing literature, we adopt the Beta distribution for the generation of each nurse's reliability [23]. The reliabilities of a given number of nurses are drawn as samples of a Beta distribution with its mean set to the average reliability and its standard deviation (nurses' conflict level) either set to high or low. Intuitively, conflicts between a group of less reliable nurses are more than that of more reliable nurses. Thus, we use a lower variance when the Beta distribution mean increases. In other words, a group of nurses with higher average reliability will have less conflict. Once each nurse's reliability is established, its annotations are generated by randomly choosing (100-reliability) percentages of positive cases of PU

(+1) and flipping their labels into negative (-1).

We conduct experiments varying the number of nurse annotators ($M \in \{3, 5\}$); the group's average reliability (mean $\in \{0.5, 0.57, 0.64, 0.7\}$); and the standard deviation of the nurses' conflict level (low and high). For each distribution setting (mean and standard deviation), we generate 2 different annotation datasets. An example of one draw for 5 annotators ($M = 5$) is shown in Table 4.3. In total, there are $2 \times 4 \times 2 \times 2 = 32$ datasets to run.

Table 4.3: Example of a Nurses' Annotation Simulation with Different Parameters for Beta Distribution.

Group No.	**Reliability mean**	**Reliability standard deviation**	**Example of 5 nurses reliabilities**
1	50%	Low (5.6%)	[52.5%, 51.7%, 35.7% , 50.5%, 53.1%]
2	50%	High (11%)	[56.6%, 39.8%, 50.4%, 56.5%, 37.6%]
3	57%	Low (5.5%)	[58.9%, 56.9%, 45.1%, 58.7% , 55.7%]
4	57%	High (9.7%)	[61.5%, 61.4%, 40.5%, 59.2%, 52.4%]
5	64%	Low (4.7%)	[62.5%, 59.8%, 63.5%, 67.2%, 68.1%]
6	64%	High (9.4%)	[62.6%, 75.6%, 76.4%, 56.1% , 70.2%]
7	70%	Low (4.1%)	[76.7%, 73.4%, 67.5%, 69.4%, 73.7%]
8	70%	High (8.2%)	[68.7%, 75.1%, 79.4%, 56.2%, 83.8%]

4.2 Evaluation

4.2.1 Training, Validation, and Test Data

We divide our data (10000 in synthetic dataset; 3589 stays in MIMIC-III) into 64% training, 16% validation, and 20% test set. We construct 10 different sets of train-validation-test splits in order to get more accurate evaluation results. After the split, we impute the missing values with column mean for each set individually.

4.2.2 Truth Inference Methods

We used MV, WM, and EM to perform truth inference (as discussed in Section 2.1). For the WM method, we adopted the typical re-weighting multiplier $\beta = 0.5$ [24]. For the EM method, we used an existing implementation [23] and set the number of iterations to 20. To estimate the labels for the model training, we evaluate the accuracy of the three truth inference methods' estimated labels compared to the ground truth. In this study, we only run truth inference methods on the training data and validation data.

4.2.3 Prediction Models

We use the decision tree as the weak learner for AdaBoost and Medaboost since the decision tree has been widely applied in healthcare especially for heterogeneous data of various types [25], as in our dataset.

Because the data has more negative labels, we balance the two labels when we train the basic decision tree by adjusting the weight for each class $C \in \{-1, 1\}$ inversely proportional to the number of classes' occurrences as: $W_C = \frac{n}{2 \cdot (\text{occurrence of class C})}$.

The prediction models (i.e., AdaBoost and Medaboost) are first trained on the training set. The validation set is then used to optimize two hyper-parameters: numbers of iterations T and learning rate η. A higher number of iterations means more weak classifiers are combined in the final prediction combines. A higher learning rate means sample weights at each iteration increase or decrease as shown in Equations (3.6) and (3.7). The model is then trained on the entire training and validation data (80% of the data) using the best pair of T and η. The prediction of the classifier is then evaluated on our test set.

4.2.4 Evaluation Metrics

Since our data is imbalanced toward negative class, we cannot use accuracy because labelling all samples as negative gives a high accuracy but cannot make a good model. Therefore, we evaluate the models using the Area Under the Receiver Operating Characteristic Curve (AUC). AUC is generally insensitive to class imbalance and is the de facto measure of discrimination in literature.

For the truth inference part, we use accuracy to evaluate their performance because it provides the most intuitive result. It is calculated as:

$$\frac{\text{the number of samples that are correctly labelled}}{\text{the total number of samples}} \tag{4.4}$$

A result with high AUC and high accuracy indicates good performance.

Chapter 5

Result

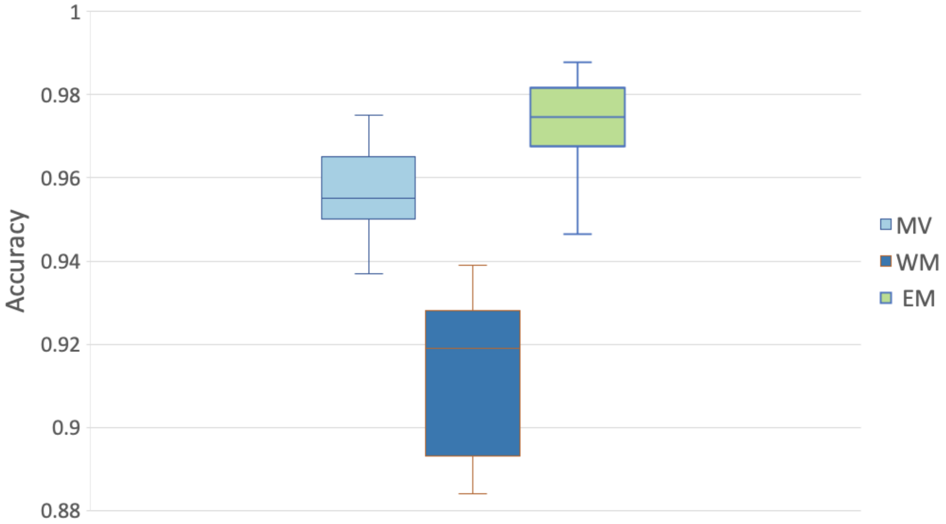
5.1 Performance of truth inference methods

5.1.1 Synthetic Dataset

We use accuracy as the metric to evaluate the performance of the three truth inference methods: MV, WM, and EM. Since we do not need to use the test set (20% of the samples), we only calculate the accuracy of the three methods using 8000 samples.

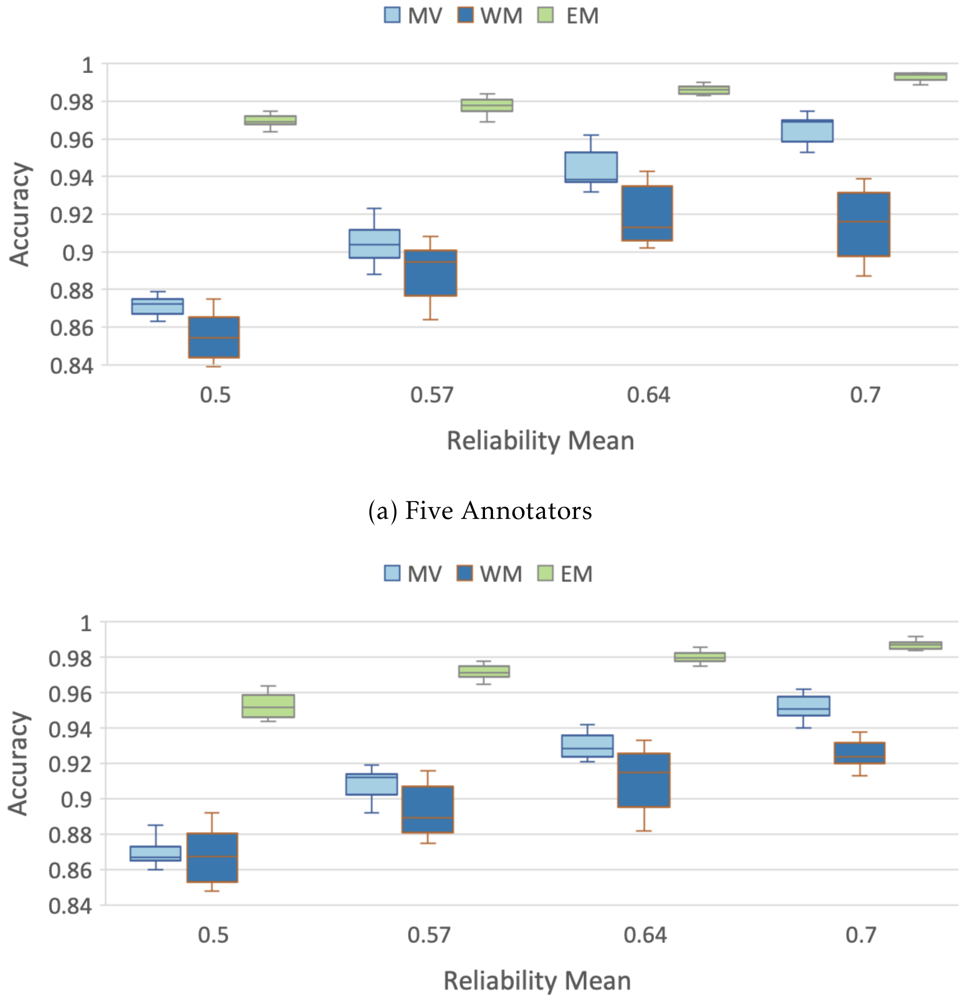
For each of the 10 datasets, we calculate the accuracy for 10 different train-validation-test splits. In the end, we use box plot to visualize the 10*10 = 100 accuracy results for each of the three methods. Figure 5.1 shows the result. From the figure, we can see that three methods all achieve high accuracy, with medians greater than 92%. In comparison, EM performs the best among the three methods, with a median of 97.5%. MV is the second-best truth inference method, with a median of 95.5%. WM performs the worst, with a median being 92%.

Figure 5.1: Accuracy of MV, WM, and EM Using Synthetic Datasets



5.1.2 MIMIC-III Critical Care Database

Similar to Synthetic dataset, we evaluate the performance of MV, WM, and EM on just the training and validation data. Given that there are 10 different train-validation-test splits, we present the average accuracy based on the average reliability of the nurses. Figure 5.2 shows the average accuracy of MV, WM, and EM over the mean of groups' sample reliability for annotations from 3 nurses and 5 nurses. We observe from the figure that EM is always higher than the other two methods. For example, in Figure 5.2a, even for the lowest mean reliability of 0.50, the accuracy of EM is as high as 95.75%, while that of MV and WM are 83.07% and 82.58% respectively. We also observe that when the reliability of the annotators increases, the average generally increase, with some minor fluctuation of WM. Comparison of the two figures also suggests that it is always better to have more annotators as the accuracy with 5 annotators (Figure 5.2a) is slightly higher than 3 annotators (Figure 5.2b). Based on the results in Figure 5.2, we use EM as the truth inference algorithm for the predictive models.



(a) Five Annotators

(b) Three Annotators

Figure 5.2: Accuracy of MV, WM, and EM over Different Mean Reliability of Annotators

5.2 Robustness and advantage of Medaboost.

5.2.1 Synthetic Dataset

We calculate AUC for both Medaboost and baseline (Adaboost) to evaluate their performance. For each of the ten synthetic datasets, we take 10 different train-validation-test splits to calculate the AUC using the test set. We then use box plot to visualize the 10 AUC in different splits for each dataset. Figure 5.3 shows AUC of Medaboost and Adaboost, which is grouped by different mean accuracy of 10 experts.

From the figure, we observe that the median of Medaboost is always higher than that of our baseline. On average, the median AUC of Medaboost is 97.0%, and that of our baseline is 96.4%, 0.6% higher.

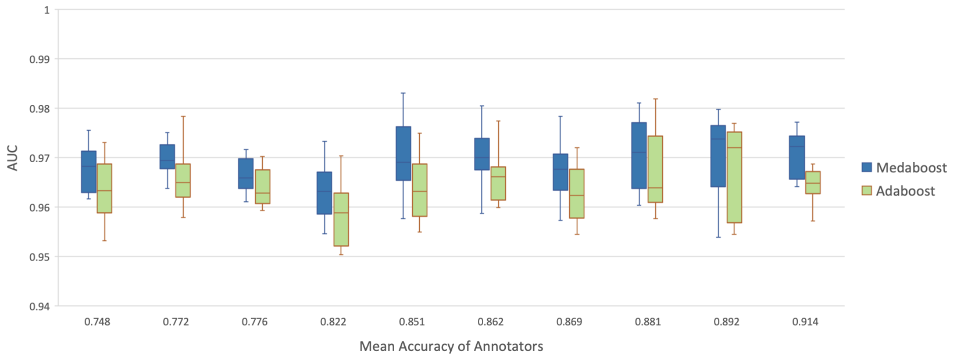


Figure 5.3: AUC of Medaboost and Adaboost under different mean accuracies of experts

To further analyze the advantage of Medaboost over the baseline, we also conduct a one-sided paired t-test. We run the test on all AUC values in 10 splits and 10 datasets ($10 \times 10 = 100$ AUC values). Table 5.1 shows the result of the t-test. Our null hypothesis is that the AUC of Medaboost is strictly smaller then that of the baseline. Take the first t-test as an example. On average, Medaboost's AUC is 0.969% (with S.D 0.006) higher than baseline's. As $p < 0.05$, we can reject our null hypothesis. It means that using Medaboost can have a statistically significant increase than using baseline.

Table 5.1: Performance of Medaboost and baseline on datasets with different M and S.D. The table presents the one-sided paired t-test for the Medaboost's AUC and Adaboost's AUC.

	Paired Difference of Medaboost-Adaboost		**p-value**
	Mean Difference	S.D Difference	**(one-sided)**
Synthetic Dataset	0.004	0.003	0.000

5.2.2 MIMIC-III Critical Care Database

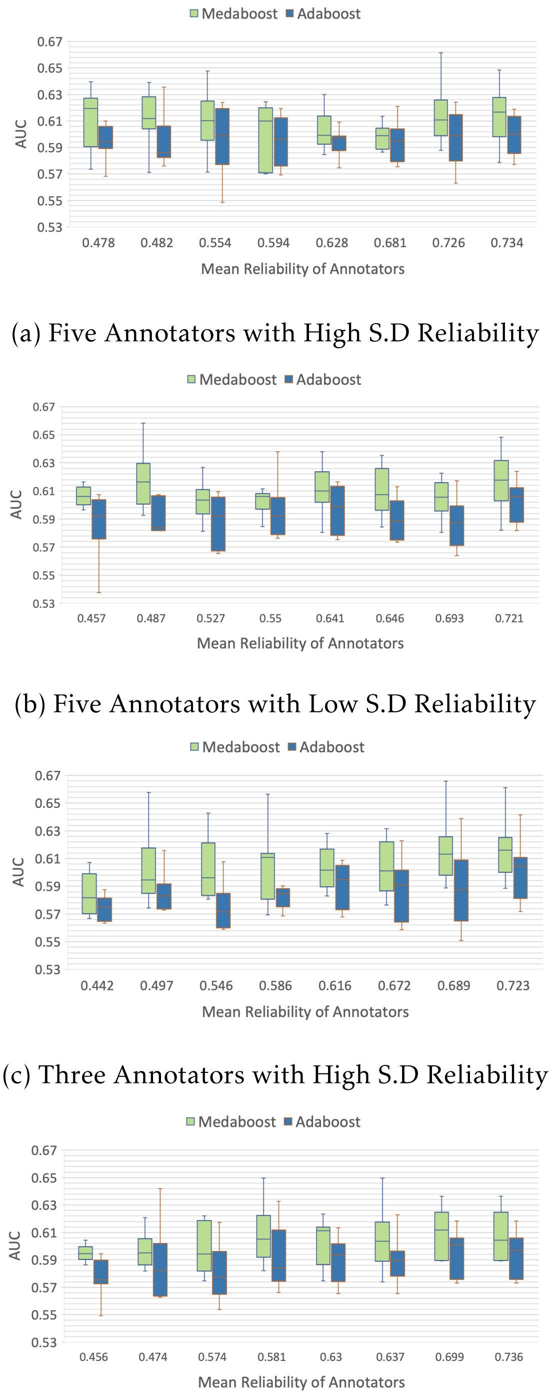
Similar to the synthetic dataset experiment, we conduct a one-sided paired t-test for 4 different groups by fixing the number of annotators (i.e., 3 or 5) and the standard deviation level (i.e., high or low), and each group has 8 datasets with different mean reliability. For each dataset, we calculate the AUC values for 10 different splits. Our null hypothesis is that the AUC of Medaboost is strictly smaller than that of baseline. Thus, we run a paired t-test for 80 paired differences of AUC values between Medaboost and Adaboost for each of the 4 groups. Table 5.2 shows the result of the t-test. As an example, consider the first t-test with 5 nurses and a high standard deviation level. On average, Medaboost's AUC is 1.3% (with S.D. of 0.016) higher than Adaboost. This suggests that Medaboost improves the predictive performance compared to the baseline. The table shows that all p-values are smaller than 0.05, so Medaboost is better than Adaboost across all 4 groups.

Sensitivity to different nurses' mean reliability. To analyze the predictive models' sensitivity to different nurses' mean reliability, we compare the test set's AUC value on datasets with different means, while keeping

Table 5.2: Performance of Medaboost and baseline on datasets with different M and S.D. The table presents the one-sided paired t-test for the Medaboost's AUC and Adaboost's AUC.

	Paired Difference of Medaboost-Adaboost		**p-value**
Dataset	Mean Difference	S.D Difference	**(one-sided)**
5 nurses high S.D	0.013	0.016	0.000
5 nurses low S.D	0.018	0.013	0.000
3 nurses high S.D	0.017	0.013	0.000
3 nurses low S.D	0.017	0.013	0.000

the number of annotators and the standard deviation the same. Figure 5.4a shows the boxplots of the AUC across the 10 different splits for both Medaboost and Adaboost (baseline) with 5 annotators and a high standard deviation in the nurses' reliabilities. From the figure, we observe that the median AUC of Medaboost (colored green) is higher than that of the Adaboost (colored blue). On average, the median AUC of Medaboost is 0.611, 0.014 higher than that of the baseline (0.597), with the highest difference at 0.025. We also observe that there are no major differences in predictive performance even as the mean value gets higher when there is a high standard deviation in the nurses' reliabilities.



(a) Five Annotators with High S.D Reliability

(b) Five Annotators with Low S.D Reliability

(c) Three Annotators with High S.D Reliability

(d) Three Annotators with Low S.D Reliability

Figure 5.4: Predictive performance with varying number of annotators and different reliability parameters.

Sensitivity to different number of nurses. To analyze the predictive models' sensitivity to different number of nurses, we compare the AUC on datasets with M = 3 and M = 5, and keep standard deviation the same (either high or low). Figure 5.4 shows the AUC performance of both Medaboost and Adaboost (baseline). Each sub-figure summarizes the 8 datasets with the same number of annotators and standard deviation, while each boxplot summarizes the results from 10 train-validation-test splits.

Figures 5.4a and 5.4b illustrates the AUC when $M = 5$, while Figures 5.4c and 5.4d summarizes the performance when $M = 3$. A comparison of high standard deviation of reliability (Figures 5.4a and 5.4c) illustrate the importance of having more nurses annotations especially in the presence of high standard deviation. For 5 annotators, the median AUC values range between 0.600 and 0.619 while for 3 annotators the range is between 0.582 and 0.616. The same trend is found even with low standard deviation (Figures 5.4b and 5.4d) as the AUC ranges between 0.605 and 0.622 for $M = 5$ and 0.593 to 0.615 for $M = 3$.

Sensitivity to nurses' conflict level (or standard deviation of reliability) To determine the robustness to the standard deviation of the nurses' reliabilities, we compare the predictive models' AUC between the two levels of standard deviation and keep M and range of mean the same. Figure 5.4a and 5.4c show the AUC when the S.D of nurses' reliability is high, while Figure 5.4b and 5.4d represent when S.D of nurses' reliability is low. By comparing AUC with 5 annotators, we can see that the degree of S.D reliability does not affect the performance of Medaboost. For high S.D, the mean of median AUC values is 0.611, and it is the same as that for low S.D(0.611). For AUC with 3 annotators, the mean of median AUC values is 0.603 for high S.D, 0.001 higher than that for low S.D (0.602). Therefore, no obvious trend is found for the degree of S.D reliability.

Chapter 6

Conclusions

In this study, we investigated the performance of several truth inference methods to predict true labels based on multiple annotations. Our results suggest that EM is the most accurate method and is a reasonable choice to estimate the true label. We also propose Medaboost, a new predictive model that adjusts the initial weight of the Adaboost algorithm based on annotators' disagreements on each sample.

In our model, the more conflicts on one sample, the less weight that sample gets to reflect the greater uncertainty about the sample's label. From our experimental studies on both synthetic dataset and MIMIC-III dataset, we illustrated the benefit of Medaboost as it outperforms Adaboost, an ensemble prediction model. Additionally, the performance of Medaboost is generally robust to the number of annotators, the mean reliability of the annotators, and the conflict levels across the annotators. We note that this is a pilot study on a relatively small case-control study with well-established ground truth labels for the patients. Thus more experiments are necessary to provide evidence of Medaboost's viability, but the early results are promising.

There are several limitations in our study. First, in the MIMIC-III the number of annotators is limited to reflect practical considerations for PU annotations. While Medaboost can be applied to other applications, additional experiments are necessary to explore a large range of annotators. Similarly, we only generate synthetic data under a fixed ground truth distribution. More experiments need to be done using more datasets. Second, we only use simulated nurses' labels in our dataset to evaluate three truth inference methods and Medaboost. In reality, the studies should be conducted with real patient assessments from a panel of nurses. Third, the discrimination power of our predictive model when the annotators' mean accuracy is low (such as in the MIMIC-III dataset) is still low with an AUC of 0.61. Additional features can be explored such as mining unstructured text to improve the predictive performance of PU identification. Fourth, our baseline does not reflect whether using the truth inference methods is helpful in this study because we use the results from truth inference methods to train Adaboost. In further studies, we will also set an Adaboost model without truth inference methods as one of our baselines. We will deal with the multiple annotations by duplicate that sample for each of its annotations. For instance, if each sample has 3 annotations, we create three samples with the same feature set and one of the three annotations per sample. In this way, we can also evaluate the necessity of including the truth inference step in Medaboost. Finally, Medaboost only considers information from multiple annotations, i.e. the disagreements in them and the estimated labels generated from them. The use of features is the same as the normal prediction models. In future studies, the distributions of samples' features and their relationship with disagreements in multiple annotations can be investigated.

Bibliography

[1] Bauer K, Rock K, Nazzal M, Jones O, Qu W. Pressure Ulcers in the United States' Inpatient Population From 2008 to 2012: Results of a Retrospective Nationwide Study. Ostomy Wound Manage. 2016;62(11):30–38.

[2] Van Den Bos J, Rustagi K, Gray T, Halford M, Ziemkiewicz E, Shreve J. The $17.1 billion problem: the annual cost of measurable medical errors. Health Affairs. 2011;30(4):596–603.

[3] Schoonhoven L, Grobbee DE, Bousema MT, Buskens E, prePURSE study group. Predicting pressure ulcers: cases missed using a new clinical prediction rule. Journal of advanced nursing. 2005;49(1):16–22.

[4] Waugh SM, Bergquist-Beringer S. Inter-rater agreement of pressure ulcer risk and prevention measures in the National Database of Nurs-

ing Quality Indicators®(NDNQI). Research in nursing & health. 2016;39(3):164–174.

[5] Cohen J. A coefficient of agreement for nominal scales. Educational and psychological measurement. 1960;20(1):37–46.

[6] Polit DF, Beck CT. Nursing research: Generating and assessing evidence for nursing practice. Lippincott Williams & Wilkins; 2008.

[7] Sook CI, Eunja C. Predictive Bayesian Network Model Using Electronic Patient Records for Prevention of Hospital-Acquired Pressure Ulcers. Journal of Korean Academy of Nursing. 2011;41(3).

[8] Cho I, Park I, Kim E, Lee E, Bates DW. Using EHR data to predict hospital-acquired pressure ulcers: a prospective study of a Bayesian Network model. International journal of medical informatics. 2013;82(11):1059–1067.

[9] Kaewprag P, Newton C, Vermillion B, Hyun S, Huang K, Machiraju R. Predictive models for pressure ulcers from intensive care unit electronic health records using Bayesian networks. BMC medical informatics and decision making. 2017;17(2):65.

[10] Li Y, Gao J, Meng C, Li Q, Su L, Zhao B, et al. A survey on truth discovery. ACM Sigkdd Explorations Newsletter. 2016;17(2):1–16.

[11] Li Y, IP Rubinstein B, Cohn T. Truth Inference at Scale: A Bayesian Model for Adjudicating Highly Redundant Crowd Annotations. In: The World Wide Web Conference; 2019. p. 1028–1038.

[12] Littlestone N, Warmuth MK, et al. The weighted majority algorithm. University of California, Santa Cruz, Computer Research Laboratory; 1989.

[13] Dawid AP, Skene AM. Maximum likelihood estimation of observer error-rates using the EM algorithm. Applied statistics. 1979;p. 20–28.

[14] Freund Y, Schapire RE. A desicion-theoretic generalization of on-line learning and an application to boosting. In: European conference on computational learning theory. Springer; 1995. p. 23–37.

[15] Jia Hx, Zhang Y. Fast adaboost training algorithm by dynamic weight trimming. Chinese Journal of Computing. 2009;32:336–341.

[16] Hu W, Hu W, Maybank S. Adaboost-based algorithm for network intrusion detection. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2008;38(2):577–583.

[17] Kim K, Choi HI. Adjusting initial weights for Adaboost learning. International Conference on Computer Applications and Information Processing Technology (CAIPT). 2017;p. 1–5.

[18] Johnson AE, Pollard TJ, Shen L, Li-wei HL, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Scientific data. 2016;3:160035.

[19] Gyöngyösi M, Ploner M, Porenta G, Sperker W, Wexberg P, Strehblow C, et al. Case-based distance measurements for the selection of controls in case-matched studies - application in coronary interventions. Artif Intell Medicine. 2002;26(3):237–253.

[20] Cummings D. Predicting hospital length-of-stay at time of admission; 2018. [Online; accessed 8-March-2020]. https://towardsdatascience.com/predicting-hospital-length-of-stay-at-time-of-admission-55dfdfe69598.

[21] Marathon Studios I. ICD-9-CM Chapters; 2020. [Online; accessed 8-March-2020]. https://icd.codes/icd9cm.

[22] Wang S, McDermott MBA, Chauhan G, Hughes MC, Naumann T, Ghassemi M. MIMIC-Extract: A Data Extraction, Preprocessing, and Representation Pipeline for MIMIC-III. arXivorg. 2019 Jul;.

[23] Zheng Y, Li G, Li Y, Shan C, Cheng R. Truth inference in crowdsourcing: is the problem solved? Proceedings of the VLDB Endowment. 2017;10(5):541–552.

[24] Kolter JZ, Maloof MA. Dynamic weighted majority: An ensemble method for drifting concepts. Journal of Machine Learning Research. 2007;8(Dec):2755–2790.

[25] Podgorelec V, Kokol P, Stiglic B, Rozman I. Decision trees: an overview and their use in medicine. Journal of medical systems. 2002;26(5):445–463.