RELIABLE AND TRUSTWORTHY: IMPROVING QUALITY OF CROWDSOURCED WORK WITH INTERVAL-VALUED LABELS

by

Makenzie Spurling

A thesis presented to the Department of Computer Science and Engineering and the Graduate School of the University of Central Arkansas in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

Conway, Arkansas
August 2022

ACKNOWLEDGMENTS

For my first acknowledgment, I'd like to thank my thesis advisor Dr. Chenyi Hu. Without him, I would not have chosen to pursue a master's degree or write a thesis at all. His guidance and encouragement throughout the entire process have been an undeniable support. To my faculty advisor Dr. Sinan Kockara, I also extend my thanks for putting up with my long-winded emails and always being there to answer or direct me to answers for my many questions. Dr. Emre Celebi, thank you for all the help you've given me over the years; from help with applying to graduate school to making sure I got registered for classes.

This is also a thank you to all of the faculty members of the University of Central Arkansas's Department of Computer Science and Engineering. For the many years I've spent at UCA, I have had many wonderful professors and advisors who taught me from the first steps of coding to major projects and research works. To my thesis committee members, Mrs. Patricia (Michelle) Talley and Dr. Bernard Chen, thank you for taking the time to review my work and being a part of the process.

It should also be noted that this work was partially supported by the United States National Science Foundation through the grant award NSF/OIA-1946391. More specifically, this work was a part of the DART project of which I have been a member since November 2020. Thank you to the central office whose help with paperwork meant that not only would I get to do research but I would be paid while doing so.

Finally, for my family who has always supported me throughout my academic career: thank you for all of your support and your continued faith in me and my abilities, even when I did not have faith in them. Without all of you standing beside me, I would not have been able to accomplish even as half as much as I have.

VITA

Makenzie Spurling graduated with a Bachelors of Science in Computer Science from the University of Central Arkansas in December 2021. She graduated summa cum laude after three and a half years of hard work. Throughout her years of study she participated in various research projects, publishing one paper in her undergraduate study and two in her graduate study. She also took part in the Department of Computer Science's 4+1 program for an accelerated Master's degree in Computer Science. Through this, she will complete her Master's studies with the thesis option in August of 2022.

ABSTRACT

Using input collected from human crowds through the internet, crowdsourcing has rapidly become an established method for solving applications requiring human input in artificial intelligence and machine learning, such as classification. In classification, the inputs collected from the crowds are called labels and the people giving labels are called crowd-workers. However, labels often suffer from worker uncertainty or inaccuracies due to insufficient knowledge, differences in socio-economic backgrounds, and other outside factors. To solve this issue, interval-valued labels (IVLs) can be used in place of commonly used binary-valued ones to capture uncertainty. Through them, two methods, interval majority voting (IMV) and preferred matching probability (PMP), are adapted to account for uncertainty when making inferences. IVLs also pave the way to quantitatively estimate a worker's reliability in terms of correctness, confidence, stability, and predictability. Reliability can further improve quality of crowdsourcing by applying weights to workers in two learning schemes, weighted interval majority voting (WIMV) and weighted preferred matching probability (WPMP). Experiments on benchmark datasets clearly show that WIMV and WPMP perform better than non-weighted strategies with higher precision, recall, accuracy, and F_1-score. However, workers selected via crowdsourcing come with varying reliabilities and intentions. In addition, there is a threat of people with adverse purposes launching attacks to undermine crowdsourced projects. To combat this, worker reliabilities on gold questions that have known ground truth and regular questions that don't can be compared to detect abnormal behavior and attackers. Through this, anomalies and adversarial attackers have been effectively identified within crowd-workers to improve the quality of crowdsourced work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol	Meaning
G	Set of gold questions
g	Specific gold question
$I(p^+)$	Preferred inference
j	Specific worker
J	Set of workers
$\mathbf{L}_i$	List of IVLs on same instance i by various workers
$\mathbf{L}^j$	List of IVLs by single worker on various observations
$\mathbf{L}^j_G$	List of IVLs from a worker j on G
$\mathbf{L}^j_U$	List of IVLs from a worker j on U
$\mathbf{l}_{gj}$	IVL from worker j on g
$\mathbf{l}_{ij}$	An IVL made for v_i by a worker j
$\underline{l}_{ij}$	j's minimum belief of v_i being an instance of a class
$\bar{l}_{ij}$	j's maximum belief of v_i being an instance of a class
$mid(L_i)$	Midpoint of $\mathbf{L}_i$
$mid(l_{ij})$	Midpoint of $\mathbf{l}_{ij}$
$\mu_{\mathbf{L}_i}$	Mean of $\mathbf{L}_i$
$\mu(\mathbf{L}^j_G)$	Mean of $\mathbf{L}^j_G$
$\mu(\mathbf{L}^j_U)$	Mean of $\mathbf{L}^j_U$
$o(G)$	Ground truth of G
$o(g)$	Ground truth of g
p^+	Probability of v_i being an instance of y
$pdf_{ij}(t)$	Probability density function of a $\mathbf{l}_{ij} \in \mathbf{L}_i$
$rad(L_i)$	Radius of $\mathbf{L}_i$
$rad(l_{ij})$	Radius of $\mathbf{l}_{ij}$
r_j	Worker j's reliability

$\sigma(L_i)$	Standard deviation of $\mathbf{L}_i$
$\sigma(\mathbf{L}_G^j)$	Standard deviation of $\mathbf{L}_G^j$
$\sigma(\mathbf{L}_U^j)$	Standard deviation of $\mathbf{L}_U^j$
U	Set of regular questions
V	Set of observations
$Var(L_i)$	Variance of $\mathbf{L}_i$
v_i	Specific observation
Y	Set of classes
y	Specific class
IMV	Interval majority voting
IVL(s)	Interval-valued label(s)
MV	Majority voting
PMP	Preferred matching probability
WIMV	Weighted interval majority voting
WPMP	Weighted preferred matching probability

CHAPTER I: INTRODUCTION

1.1 Crowdsourcing

This material is extracted and adapted from previously published scientific papers [16], [18], [37], and [38] by the thesis author and others for a more in-depth knowledge and understanding of interval-valued data and its use in crowdsourcing.

Crowdsourcing has very quickly become a popular method for gathering large amounts of information for a project or task. Previously, gathering such large sums of data for a project would require dedicating a lot of time and money, which is unfeasible for the majority outside of big corporations or universities. Relying on the wisdom of the crowd provides a low cost way of accomplishing these previously difficult to do tasks. At its core, crowdsourcing hinges on the voluntary participation of human beings. It is the practice of gathering inputs (labels) from large groups of people (crowd-workers), usually through methods such as the internet or other mediums. To date, there are already large volumes of manually labeled crowdsourced data readily available through crowdsourcing marketplaces such as Amazon Mechanical Turk[1], CrowdFlower[2], and others. This gathered data can be used in various machine learning and artificial intelligence schemes to create inferences for human users.

While crowdsourcing can accomplish much, there are inherent issues with relying on the wisdom of the crowd. One of these is that the participating groups are usually large, relatively open, and rapidly changing. This raises concerns on the quality of the crowd-workers and resulting data. For machine learning algorithms, the quality of data fed into the models directly affects the quality of the output. The same applies in crowdsourcing; the quality of crowdsourcing directly depends on the quality of the collected labels [34] which then subsequently affects the quality of the outcome. Because crowdsourcing is open in nature, crowd-workers usually come from different socio-economic backgrounds with varying cultures or intentions. Factors that affect a worker's ability to give quality data

[1] https://www.mturk.com/

[2] http://crowdflowersites.com/

such as attitude, emotion, and stress level tend to fluctuate throughout the day. Mistakes are more likely when a worker is emotionally unstable, stressed, or tired. Bias in the collected data is also a concern. As reported in [1], Barbosa and Chen demonstrate that biases caused by a worker's socio-economic status can determine the outcome of crowdsourcing. Another issue is that the level of expertise for crowd-workers often varies widely. Workers with a high level of expertise are expected to give better quality labels than those with less. However, a worker with a high level of knowledge and adverse intentions can cause infinitely more harm to crowdsourced tasks. Even experts with no adversarial intentions may disagree on the correct solution when given the exact same data [27]. All of these issues can be traced to a single issue: worker uncertainty. The more uncertainties a worker has, the less reliable the collected labels. When crowdsourcing, managing worker uncertainties is necessary to obtain quality data.

Researchers have almost always been aware that the quality of crowdsourcing depends on the quality of the collected labels more than the model itself [46]. There are many previously proposed solutions that work to minimize the issues with crowdsourcing and data quality. Worker reliability itself is already a highly researched topic with many papers on various methods of managing noise from workers. For one, Bi *et al* studied reasons for noisy labels from factors such as worker dedication, expertise, judgment, and task difficulty in [3]. Qiu *et al* offered methods for selecting workers based on behavior prediction [29]. Tao *et al* reported quality improvements in [39] when utilizing MV-Freq and MV-Beta [35] with worker's reliability. These papers demonstrate how uncertainty management is a major research topic in the scientific community.

1.2 Interval-valued data

In literature, point-valued data and datasets are most often used for uncertainty management. However, point-valued data only captures a single snapshot of an observation at single point in time. In comparison, interval-valued data applies a range of values that are

more suited towards capturing the variations and uncertainties that occur in the real world. Already researchers have developed interval methods for knowledge processing. By using data aggregation strategies from [2], [6], [28], and others, large amounts of point-valued data can be reduced into smaller interval-valued data for more efficient data management and processing.

Intervals also come with their own properties and operations. Researchers have already successfully solved previously hard to solve applications involving uncertainties by applying interval-valued data. Hu and He applied interval least-squares [15] for stock market annual variability forecasting using interval-valued data. It was shown that the average accuracy ratio significantly increased compared to the commonly uses point-valued confidence interval predictions when using the same raw dataset [7], [8], [13]. It has been further verified and validated from the perspective of information theory that the significant improvement comes indeed from interval-valued data [17]. Other successes include but are not limited to [5], [9], [19], [20], [22], [26], [31], and many more.

1.3 Objective of this study

This study applied interval-valued data to crowdsourcing and showed significant quality improvements in crowdsourced data. This was done through various methods such as applying interval-valued data to crowdsourcing labels to capture uncertainty in workers. Then using the uncertainty data, reliability indexes were assigned to workers for weighted crowdsourcing. The established reliabilities were then used to identify significant changes or signs of anomalous behavior that could affect the quality of crowdsourcing. Ultimately, this work demonstrated that using these methods lead to a definite improvement in the quality of inferences made using crowdsourced data.

The remainder of this work is structured as follows. Chapter II provides an overview of the literature on interval-valued labels, their notations, and relevant statistical and probabilistic features. Chapter III tackles uncertainty in workers, interval-valued labels, and

inference making with interval datasets. Chapter IV will further expand on dealing with worker uncertainty through reliability indexes as well as more inference strategies. In Chapter V, worker reliability will be repurposed into identifying and removing anomalous workers and potential attackers. Finally, Chapter VI will draw conclusions about the study based on the findings and suggest future research directions.

CHAPTER II: STATISTICAL AND PROBABILISTIC PROPERTIES OF INTERVAL-VALUED DATASETS

This chapter will be an overview into interval-valued data and its statistical and probabilistic properties. This information, taken from [16], will go over background concepts and notations as well as provide relevant descriptive statistics of interval-valued datasets, probability distributions of interval data and datasets, and a way to derive informational entropy of interval-valued datasets.

2.1 Notations and descriptive statistics

In literature, interval-valued objects use boldfaced letters to separate them from point-valued ones i.e, a is a real where $\mathbf{a}$ is an interval. The endpoints of an interval are the greatest lower bound and least upper bound; written using an underline and overline on the same non-boldfaced letter. Interval $\mathbf{a}$'s greatest lower bound and least upper bound are $\underline{a}$ and $\overline{a}$. Using the bounds, the interval $[\underline{a}, \overline{a}]$ is another way of representing $\mathbf{a}$ and is called an endpoint (or min-max) representation. The midpoint and radius of $\mathbf{a}$ are defined as $mid(\mathbf{a}) = \dfrac{\underline{a} + \overline{a}}{2}$ and $rad(\mathbf{a}) = \dfrac{\overline{a} - \underline{a}}{2}$. This midpoint can also be called the centroid and both terms will be used interchangeably in this text. As the two are point-valued, the terms are written without boldface as $mid(a)$ and $rad(a)$. Like with endpoint representation, $\mathbf{a}$ can be represented using the midpoint and radius as well.

When indicating a single interval, a lowercase boldfaced letter is used. When indicating a collection of real intervals, i.e., an interval-valued dataset, a boldfaced uppercase letter is used. An example interval-valued dataset is $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. Similar to a single interval's endpoints, interval-valued datasets have endpoints which are the sets of the left and right endpoints for all interval data in the dataset. The two end sets, $\underline{X}$ and $\overline{X}$, are written as ordered tuples, $\underline{X} = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n)$ and $\overline{X} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$. While items in the sets are not ordered, $\underline{x}_i \in \underline{X}$ and $\overline{x}_i \in \overline{X}$ will be related to the same interval $\mathbf{x}_i \in \mathbf{X}$. The midpoint and radius of $\mathbf{X}$ are point-valued tuples denoted as $mid(X) = (mid(x_1), mid(x_2), \ldots, mid(X_n))$ and $rad(X) = (rad(x_1), rad(x_2), \ldots, rad(x_n))$, respectively.

Example 1. Given an interval-valued dataset $\mathbf{X}_0 = \{[1,5],[1.5,3.5],[2,3],[2.5,7],[4,6]\}$. The left-endpoint is $\underline{X}_0 = (1, 1.5, 2, 2.5, 4)$ and the right-endpoint is $\overline{X}_0 = (5, 3.5, 3, 7, 6)$. The midpoint of $\mathbf{X}_0$ is $mid(X_0) = \dfrac{\underline{X}_0 + \overline{X}_0}{2} = (3, 2.5, 2.5, 4.75, 5)$, and the radius is $rad(X_0) = \dfrac{\overline{X}_0 - \underline{X}_0}{2} = (2, 1, 0.5, 2.25, 1)$.

The sample dataset $\mathbf{X}_0$ will be used for all future examples in this chapter. Along with the four already given, the mean of $\mathbf{X}$ or the arithmetic average, is another interval statistic. It is written as $\mu_{\mathbf{X}}$ and since $\sum_{i=1}^{n} \mathbf{x}_i = [\sum_{i=1}^{n} \underline{x}_i, \sum_{i=1}^{n} \overline{x}_i]$ in interval arithmetic, it can be expanded into

$$\mu_{\mathbf{X}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i = \left[\frac{\sum_{i=1}^{n} \underline{x}_i}{n}, \frac{\sum_{i=1}^{n} \overline{x}_i}{n}\right] = \left[\mu_{\underline{\mathrm{X}}}, \mu_{\overline{x}}\right]. \tag{1}$$

2.2 Variance and standard deviation

Variance and standard deviation give important information about the distribution and stability of a dataset. The formula for the variance of a point-valued dataset X is

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n} |x_i - \mu|^2, \tag{2}$$

where the term $|x_i - \mu|$ is the distance between $x_i \in X$ and μ, the mean of X. A variance for $\mathbf{X}$ can't be defined using (2) because of the need for a point-valued distance between the two intervals $\mathbf{x}_i$ and $\mu_{\mathbf{X}}$. Based on interval arithmetic [23], the difference between two intervals **a** and **b** is:

$$\mathbf{a} - \mathbf{b} = [\min\{\underline{a} - \underline{b}, \underline{a} - \overline{b}, \overline{a} - \underline{b}, \overline{a} - \overline{b}\}, \max\{\underline{a} - \underline{b}, \underline{a} - \overline{b}, \overline{a} - \underline{b}, \overline{a} - \overline{b}\}] \tag{3}$$

However, the problem with Eq. (3) is that it implies that $|\mathbf{a} - \mathbf{b}| = \max\{|a - b|, \forall a \in \mathbf{a}, \forall b \in \mathbf{b}\}$ i.e., that the absolute value in the equation is the maximum distance between $a \in \mathbf{a}$ and $b \in \mathbf{b}$. Normally, the distance between two nonempty sets is defined as the

minimum distance – not the maximum, which makes this distance unusable. To define variance for an interval-valued dataset, a formula for the distance between two intervals is needed.

Definition 1. Let **a** and **b** be two nonempty intervals. Then, the formula for the distance between **a** and **b** is

$$dist(a,b) = |mid(a) - mid(b)| + |rad(a) - rad(b)| \tag{4}$$

Definition 1 satisfies all mathematical requirements for a distance: the distance is greater than or equal to 0, the distance only equals zero if the two intervals are the same, and for any nonempty intervals **a**, **b**, and **c**, $dist(a,c) \leq dist(a,b) + dist(b,c)$. With a measure for distance in place, the variance of a interval-valued dataset can be calculated. Replacing $x_i - \mu$ in (2) with $dist(\mathbf{x}_i, \mu_{\mathbf{X}})$ as defined above, the point-valued variance of **X** is be written as:

$$Var(X) = \frac{1}{n}\sum_{1}^{n} dist^2(x_i, \mu_X) = \frac{1}{n}\sum_{i=1}^{n} \left[|mid(x_i) - mid(\mu_X)| + |rad(x_i) - rad(\mu_X)|\right]^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}(|mid(x_i) - mid(\mu_X)|)^2 + \frac{1}{n}\sum_{i=1}^{n}(|rad(x_i) - rad(\mu_X)|)^2 +$$
$$+\frac{2}{n}\sum_{1}^{n}(|mid(x_i) - mid(\mu_X)|)(|rad(x_i) - rad(\mu_X)|).$$

The expression above has three terms with all three including $mid(\mu_X)$ and/or $rad(\mu_X)$. Because $\mu_{\mathbf{X}} = \left[\frac{\sum_{i=1}^{n}\underline{x}_i}{n}, \frac{\sum_{i=1}^{n}\overline{x}_i}{n}\right]$, $mid(\mu_{\mathbf{X}}) = \frac{1}{2}\left(\sum_{i=1}^{n}\underline{x}_i/n + \sum_{i=1}^{n}\overline{x}_i/n\right)$ $= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\underline{x}_i + \overline{x}_i}{2}\right) = \frac{1}{n}\sum_{i=1}^{n} mid(x_i) = \mu_{mid(X)}$. Based on this, the first two terms can be condensed into $Var(mid(X))$ and $Var(rad(X))$. The third term relates to the absolute covariance between the midpoint and radius of **X**. Let $\Delta m_i = mid(x_i) - mid(\mu_X)$ and $\Delta r_i = rad(x_i) - rad(\mu_X)$, then the term $\frac{2}{n}\sum_{1}^{n}(|mid(x_i) - \mu_{mid(X)})|)(|rad(x_i) - \mu_{rad(X)})|)$ can be rewritten as $\frac{2}{n}\sum_{1}^{n}|\Delta m_i \Delta r_i|$. The point-valued variance of an interval-valued dataset **X** can now be defined.

Definition 2. Let $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ be an interval-valued dataset, then the point-valued variance of $\mathbf{X}$ is

$$Var(X) = Var(mid(X)) + Var(rad(X)) + \frac{2}{n}\sum_{i=1}^{n} |\Delta m_i \Delta r_i| \tag{5}$$

The variance is point-valued because the midpoints and radii of interval-valued objects are point-valued. The standard deviation of $\mathbf{X}$ is calculated as normal:

$$\sigma(X) = \sqrt{Var(X)} \tag{6}$$

2.3 Probability distributions on interval datasets

An interval-valued dataset $\mathbf{X}$ can be viewed as a sample of an interval-valued population. For each $\mathbf{x}_i \in \mathbf{X}$, there is a related pdf. The pdf of $\mathbf{x}_i \in \mathbf{X}$ is written $pdf_i(x)$ and the pdf for an interval-valued dataset $\mathbf{X}$ is defined below.

Definition 3. A function $f(x)$ is called a probability density function of an interval-valued dataset $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ if and only if $f(x)$ satisfies all of the conditions:

$$\begin{cases} f(x) \geq 0 \; \forall x \in (-\infty, \infty); \\ \int_{-\infty}^{\infty} f(t)dt = 1. \end{cases} \tag{7}$$

The theorem below provides a practical way to calculate a pdf for $\mathbf{X}$.

Theorem 1. Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be an interval-valued dataset; and $pdf_i(x)$ be the pdf of $\mathbf{x}_i$ provided $i \in \{1, 2, \ldots, n\}$. Then,

$$f(x) = \frac{\sum_{i=1}^{n} pdf_i(x)}{n} \tag{8}$$

is a pdf of $\mathbf{X}$.

Proof of Theorem 1 is provided in [16]. Given the $pdf_i(x)$ for each $\mathbf{x}_i \in \mathbf{X}$, the algorithm for calculating the pdf of $\mathbf{X}$ through coding is in pseudo-code below.

Algorithm 1 Finding a pdf for $\mathbf{X}$

Algorithm: Finding a pdf for $\mathbf{X}$
Inputs: $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, $pdf_i(x)$ for every $\mathbf{x}_i \in \mathbf{X}$
Output: $pdf(X)$
`# Initialization`
$c \leftarrow$ concatenating $\underline{x}$ and $\overline{x}$
$c \leftarrow$ sort c
for i from 1 to $2n-1$ **do**
  $segment_i \leftarrow (c_i, c_{i+1}, 0)$
end for
`# Accumulating pdf on each segment`
for $\mathbf{x}_i \in \mathbf{X}$ **do**
  find j and k, such that $c_j = \underline{x}_i$ and $c_k = \overline{x}_i$
  for l from j to k **do**
    $segment_l.pdf \leftarrow segment_l.pdf + pdf_i$
  end for
end for
`# Calculating the pdf`
for i from 0 **to** $2n-1$ **do**
  $segment_i.pdf \leftarrow segment_i.pd/n$
end for
return $segment_i$ for all $i \in \{1, 2, \ldots, 2n-1\}$

Example 2. Find a pdf for the sample dataset $\mathbf{X}_0 = \{[1,5], [1.5,3.5], [2,3], [2.5,7], [4,6]\}$. For simplicity, a uniform distribution is assumed for each pdf_i's, i.e.,

$$pdf_i(x) = \begin{cases} \dfrac{1}{\overline{x}_i - \underline{x}_i} & \text{if } x \in \mathbf{x}_i \\ 0, & \text{otherwise.} \end{cases}$$

When applying Algorithm 1, the pdf is

$$f(X_0) = \frac{\sum_{i=1}^{5} pdf_i(x)}{5} = \begin{cases} 0.05 & \text{if } x \in [1, 1.5] \\ 0.15 & \text{if } x \in (1.5, 2] \\ 0.35 & \text{if } x \in (2, 2.5] \\ 0.39 & \text{if } x \in (2.5, 3] \\ 0.19 & \text{if } x \in (3, 3.5] \\ 0.09 & \text{if } x \in (3.5, 4] \\ 0.19 & \text{if } x \in (4, 5] \\ 0.14 & \text{if } x \in (5, 6] \\ 0.044 & \text{if } x \in (6, 7] \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The pdf is a stair function because of the uniform distribution for each $\mathbf{x}_i$. □

There are additional issues when finding a pdf for an interval-valued dataset using Algorithm 1. Algorithm 1 assumes that $pdf_i(x) = 0, \forall x \notin \mathbf{x}_i$. If that's not the case, then the $2n$ numbers in $\underline{X}$ and $\overline{X}$ will divide $\mathbb{R}$ into $2n + 1$ sub-intervals. These sub-intervals include $(-\infty, \min(\underline{X})), (\max(\overline{X}), \infty)$. Therefore, the accumulation loop in Algorithm 1 will run through all of the $2n + 1$ sub-intervals, and then normalize them by dividing n. An implicit assumption of Theorem 1 is of all $\mathbf{x}_i \in \mathbf{X}$ being equally weighted. That is not necessary and if needed, a positive weight w_i can be placed on each of the pdf_i's as shown in the Corollary 1.

Corollary 1. Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be an interval-valued dataset and pdf_i be the pdf of $\mathbf{x}_i \in X$, then the function

$$f(x) = \frac{\sum_{i=1}^{n} w_i \text{ pdf}_i(x)}{\sum_{i=1}^{n} w_i} \quad \text{where } \forall i \; w_i > 0 \tag{10}$$

is a pdf of $\mathbf{X}$.

2.4 Information entropy of interval-valued datasets

The amount of information within an interval-valued dataset is also a point of interest. Information entropy was introduced by Shannon in his paper "A Mathematical Theory of Communication" [32]. From that paper, the measure of information entropy associated to a possible data value is:

$$H(x) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{11}$$

where $p(x_i)$ is the probability of $x_i \in X$. Since an interval-valued dataset $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ divides the real axis into $2n + 1$ sub-intervals, $\mathcal{P}$ can be used to denoted the partition with $\mathbf{x}^{(j)}$ specifying the j-th element. This gives the equation to calculate the partition as $\mathcal{P} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(2n+1)}\right)$. Example 2 shows that Algorithm 1 can be applied to each $\mathbf{x}^{(j)} \in \mathcal{P}$ to find the pdf_j. With the probability of $\mathbf{x}^{(j)} = \int_{\mathbf{x}^{(j)}} pdf_j(t)dt$ available, (11) can be applied to calculate the entropy of an interval-valued dataset $\mathbf{X}$. A summarization of the steps for finding the entropy of $\mathbf{X}$ is provided in the algorithm below.

Algorithm 2 Finding entropy of an interval-valued dataset $\mathbf{X}$

Algorithm: Finding entropy of an interval-valued dataset $\mathbf{X}$
Inputs: $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, $pdf_i(x)$ for every $\mathbf{x}_i \in \mathbf{X}$
Output: Entropy(X)
`# Find the partition for the real axis`
$c \leftarrow$ concatenating $\underline{x}$ and $\overline{x}$
$c \leftarrow$ sort c
c forms a $2n + 1$ partition $\mathcal{P}$ of $(-\infty, \infty)$
`# Find the probability for each` $\mathbf{x}^{(j)} \in \mathcal{P}$
for j from 1 **to** $2n + 1$ **do**
  Find pdf_j on $\mathbf{x}^{(j)}$ with Algorithm 1
  Calculate $p_j = \int_{\mathbf{x}^{(j)}} pdf_j(x)dx$
end for
`# Calculate the entropy`
Entropy(X) $\leftarrow 0$
for j from 1 **to** $2n + 1$ **do**
  $Entropy(X) \leftarrow Entropy(X) - p_j \log p_j$
end for
return $Entropy(X)$

The example below finds the entropy of the sample dataset $\mathbf{X}_0$ while continuing to assume uniform distribution as in Example 2.

Example 3. Applying (9), the probability of each interval $\mathbf{x}^{(j)}$ is calculated as

$$p(x) = \int_{\mathbf{x}^{(j)}} pdf(t)dt = \begin{cases} 0.025, & \mathbf{x}^{(1)} = [1, 1.5] \\ 0.075, & \mathbf{x}^{(2)} = [1.5, 2] \\ 0.175, & \mathbf{x}^{(3)} = [2, 2.5] \\ 0.197, & \mathbf{x}^{(4)} = [2.5, 3] \\ 0.098, & \mathbf{x}^{(5)} = [3, 3.5] \\ 0.048, & \mathbf{x}^{(6)} = [3.5, 4] \\ 0.194, & \mathbf{x}^{(7)} = [4, 5] \\ 0.144, & \mathbf{x}^{(8)} = [5, 6] \\ 0.044, & \mathbf{x}^{(9)} = [6, 7] \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The entropy of $\mathbf{X}_0$ is $Entropy(X_0) = -\sum_i p_i \log p_i = 2.019$. □

CHAPTER III: CROWDSOURCING WITH INTERVAL-VALUED LABELS

This chapter's material is sampled from [18] as more background into crowdsourcing and interval-valued labels (IVLs). While crowdsourcing can be applied to much broader topics, in this context it is used for classification. More specifically, a binary classification model is used, which has a set of observations V and a set of classes Y. The objective is to determine whether an observation $v_i \in V$ is instance of a class $y \in Y$ using a worker's labels. In the past, Y has been assumed to have a cardinality of one ($|Y| = 1$) to simplify the problem without losing generality. This is because if $|Y| = m > 1$, then for the multiple classes, m, determining whether v_i is an instance of each of the classes would get checked repeatedly. When assuming $|Y| = 1$, the problem simplifies to answering if v_i is an instance of a class y? For binary classification, the answer can be yes (1) or no (0) and is defined as the ground truth. The list of answers (labels) collected from crowd-workers on the same observation v_i is L_i. A machine learning strategy can then be applied to L_i to make an inference on the observation, with the goal of matching the ground truth.

Traditionally, binary-valued labeling is the go-to labeling strategy when using binary classification. However, binary-valued labels are inherently flawed since there is an implicit assumption that a worker has total belief in their choice. In reality, this is not always the case. Workers may often face uncertainty when selecting between 0 and 1 definitively in practice. These uncertainties are inevitable when crowdsourcing, which is why properly managing them is essential to the quality of crowdsourced work. To include uncertainty when learning, 'soft' labels [36] were suggested in 1995. However, due to the lack of modeling techniques and available software tools, the methods weren't broadly applied in the past [33]. It has not been until recently that researchers have published works with applications of soft labeling when crowdsourcing. Sheng *et al* was able to apply soft labeling to study majority voting and pairing with multiple noisy labeling [35] while Zhang *et al* reported an improvement of label quality when using TSK Fuzzy Classifier [47].

3.1 Updated notations for IVLs

Interval-valued labels are an alternative to binary-valued ones. Unlike with binary-valued labels, using IVLs allows workers to specify their uncertainty on an observation v_i. An IVL made for a $v_i \in V$ by a worker $j \in J$ is written as $\mathbf{l}_{ij} = [\underline{l}_{ij}, \bar{l}_{ij}] \subseteq [0, 1]$. If j believes that v_i is 60-70% likely to be an instance of y, then the resulting label for v_i would be the interval [0.6, 0.7]. This not only ensures that the label contains the worker's uncertainty, but that there is no worry of having to recreate it afterwards. The worker j's minimum and maximum belief in v_i being an instance of any given class is $\underline{l}_{ij}$ and $\bar{l}_{ij}$. The midpoint (centroid) of $\mathbf{l}_{ij}$ is $mid(l_{ij}) = \frac{\underline{l}_{ij}+\bar{l}_{ij}}{2}$ and can be any number between and including 0 and 1 since $\mathbf{l}_{ij} \subseteq [0, 1]$. When $\mathrm{mid}(l_{ij}) > 0.5$, a worker j leans towards accepting v_i into the class. This is a positive IVL. Conversely, if $\mathrm{mid}(l_{ij}) < 0.5$, then the worker leans towards rejecting v_i being from the class and is a negative IVL. Otherwise, the IVL is neither positive nor negative and implies a tie. This is summarized as:

$$\begin{cases} 0 & \text{if } mid(l_{ij}) < 0.5, \\ tie & \text{if } mid(l_{ij}) = 0.5, \\ 1 & \text{if } mid(l_{ij}) > 0.5. \end{cases} \tag{13}$$

The radius of $\mathbf{l}_{ij}$, $rad(l_{ij}) = \frac{\bar{l}_{ij}-\underline{l}_{ij}}{2}$, reflects the maximum variation from the centroid.

Since $\mathbf{l}_{ij}$ specifies the belief of v_i being an instance of y, its difference from one, i.e. $1 - \mathbf{l}_{ij} = [1 - \bar{l}_{ij}, 1 - \underline{l}_{ij}]$, reflects the belief of v_i not being an instance of y. For example, $\mathbf{l}_{ij} = [0.6, 0.7]$ also means that j believes that v_i is not an instance of y within the range $[0.3, 0.4]$. Notice that a real is actually a narrow interval that's greatest lower and least upper bound is the exact same. These are still IVLs that will have a midpoint of the real and a radius of 0. For instance, the midpoint and radius of $\mathbf{l}_{ij} = [0.8, 0.8]$ are 0.8 and 0. Of course, an IVL can also be 0 or 1, as per the binary classification.

Previously, the list of labels sourced from the crowd was defined as L_i. Applying notation for intervals, $\mathbf{L}_i$ will be used to denote the list of IVLs on the same instance i by various workers $j \in J$. Similarly, $\mathbf{L}^j$ is a list of IVLs made by a single worker j on different observations. A list of IVLs can also be defined as an interval-valued dataset or $\mathbf{L}_i = [\mathbf{l}_{i1}, \mathbf{l}_{i2}, \ldots, \mathbf{l}_{in}]$. Hence, many of the notations described in Chapter II can be rewritten for IVLs. The midpoint and radius of $\mathbf{L}_i$ are $\mathrm{mid}(L_i)$ and $\mathrm{rad}(L_i)$. Based on Eq. (1) the the mean of $\mathbf{L}_i$ is the interval

$$\mu_{\mathbf{L}_i} = \frac{1}{n}\sum_{i=1}^{n} l_{ij} = \left[\frac{\sum_{i=1}^{n} \underline{l}_{ij}}{n}, \frac{\sum_{i=1}^{n} \bar{l}_{ij}}{n}\right], \tag{14}$$

with it's bounds consisting of the mean of all upper and lower bounds of the intervals within $\mathbf{L}_i$. The variance of $\mathbf{L}_i$ derived from Eq. (5) is another real value denoted as

$$Var(L_i) = Var(\mathrm{mid}(L_i)) + Var(\mathrm{rad}(L_i)) + \frac{2}{n}\sum_{i=1}^{n} |\Delta m_i \Delta r_i| \tag{15}$$

where $\Delta m_i = \mathrm{mid}(l_i) - \mu(\mathrm{mid}(L_i))$ and $\Delta r_i = \mathrm{rad}(l_i) - \mu(\mathrm{rad}(L_i))$. The standard deviation of $\mathbf{L}_i$ would then be

$$\sigma(L_i) = \sqrt{Var(L_i)}. \tag{16}$$

Rewriting (8) for IVLs, Eq. (17) provides a pdf for $\mathbf{L}_i$:

$$f(t) = \frac{\sum_{j=1}^{n} pdf_{ij}(t)}{n}, \tag{17}$$

where $pdf_{ij}(t)$ is a pdf of a random variable $\mathbf{l}_i \in \mathbf{L}_i$ and $pdf_{ij}(t) = 0$ if $t \notin \mathbf{l}_{ij}$.

3.2 Preprocessing IVLs

Preprocessing is a common and often necessary practice for data mining or machine learning. It consists of two stages: cleaning and normalization. To clean $\mathbf{L}_i$, the dataset is searched for out of range labels. An out of range label is one that is empty or not within the range $[0, 1]$. These labels are often discarded, but some cases can be fixed. If an empty label is due to the lower and upper bounds being reversed, swapping the order fixes the issue. Otherwise, a truncation or linear transformation can be applied with proper justifications. An example would be fixing the upper bound of an interval to a max, such as 1 in binary classification, if the upper bound is greater than the max. In addition to empty labels, Eq. (13) suggests that an $\mathbf{l}_{ij}$ with a midpoint of 0.5 would not provide any useful information. Let ϵ be a preset small positive threshold. When searching the dataset, any $\mathbf{l}_{ij}$s whose midpoints are within an ϵ distance from 0.5 are removed since they contain little useful information. This also has the added bonus of reducing noise in the collected labels.

For normalization, the radius of each IVL can be adjusted using a δ-normalization for all $\mathbf{l}_{ij} \in \mathbf{L}_i$ as follows. After choosing a threshold $\delta > 0$, each $\mathbf{l}_{ij}$ is replaced with the interval $[mid(\mathbf{l}_{ij}) - \delta, mid(\mathbf{l}_{ij}) + \delta]$. This can cause an issue as a δ-normalization may cause an out-of-range error. When this happens, the IVL can either be truncated and/or asymmetrically normalized in order to clean it. In practice, the value of δ may be selected experimentally as a proportion of the radius of $\mu(\mathbf{L}_i)$. An optional centroid normalization may be performed on IVLs containing 0.5. Instead of removing those labels, they can adjusted using symmetric cancellation centered at 0.5. After adjusting, the remaining interval can be shifted by aligning its nearest endpoint to 0.5.

3.3 Making inferences on L

When making inferences using $\mathbf{L}_i$, any strategy has to be modified to take in interval-valued labels. Take Majority Voting (MV), a simple strategy that has been applied to crowd-

sourcing in the past. For regular MV, the list of binary-valued labels L_i consists of 0s and 1s only. Similarly, $\mathbf{L}_i$ will consist of only positive or negative IVLs after preprocessing. Eq. (13) provides the preference of j to accept or reject v_i being from y. Let c^- and c^+ be the counts of negative and positive IVLs in $\mathbf{L}_i$. Then, an inference for $\mathbf{L}_i$ using MV is

$$y = \begin{cases} 0 & \text{if } c^+ < c^-, \\ 1 & \text{if } c^+ > c^-, \\ tie & \text{otherwise} \end{cases} \tag{18}$$

Eq. (18) applies centroids only and produces the same inference as MV that uses binary-valued labels. However, the goal of inference making is to make inferences that match the ground truth. When the votes are close in MV, the inference may mismatch the ground truth. By taking $\mathbf{L}_i$ and an inference, the probability of the inference matching the ground truth can be found.

First, select a known distribution such as uniform for each $\mathbf{l}_{ij} \in \mathbf{L}_i$. Then, find a *pdf* of $\mathbf{L}_i$ using Eq. (17). If any $l \in \mathbf{l} = (0.5, 1]$, that suggests that v_i is an instance of y. This makes the probability of v_i being an instance of y, denoted as p^+, the definite integral:

$$p^+ = \int_{0.5}^{1} f(t)dt \tag{19}$$

Binary classification means an inference is either 1 or 0 and Eq. (8) assures $\int_0^1 f(t)dt = 1$. Because p^+ is the probability of the inference valued 1 with $\mathbf{L}_i$ provided, the probability of the inference valued 0 is $\int_0^{0.5} f(t)dt = 1 - p^+$.

Instead of checking the probability of matching for an arbitrary inference, p^+ can be applied to make a preferred inference $I(p^+)$ from $\mathbf{L}_i$ as the following:

$$I(p^+) = \begin{cases} 0 & \text{if } p^+ < 0.5, \\ tie & \text{if } p^+ = 0.5, \\ 1 & \text{if } p^+ > 0.5. \end{cases} \tag{20}$$

When there is not a tie, Eq. (20) ensures that the inference from $\mathbf{L}_i$ has a greater than 50% probability of matching the ground truth. If there is a tie, then another IVL is needed to break it. Algorithm 3, termed preferred matching probability (PMP), summarizes how to derive a preferred inference from $\mathbf{L}_i$. The algorithm returns the inference from $\mathbf{L}_i$ with a preferred matching probability $\max\{p^+, 1-p^+\} > 50\%$ unless $p^+ = 0.5$. The higher the probability of matching is, the less amount of uncertainty $\mathbf{L}_i$ contains. The overall uncertainty in $\mathbf{L}_i$ can then be measured with

$$\zeta = 1 - \max\{p^+, 1-p^+\}, \tag{21}$$

where ζ is the uncertainty index of $\mathbf{L}_i$. Obviously, $0 \leq \zeta \leq 0.5$. This means that other than pdf_{ij} selection, both values of matching probability and ζ depend on $\mathbf{L}_i$ only. The quality of the output depends on the quality of the input.

Algorithm 3 Deriving a preferred inference from $\mathbf{L}_i$

Algorithm: Deriving a preferred inference from $\mathbf{L}_i$
Inputs: $\mathbf{L}_i = \{\mathbf{l}_{i1}, \mathbf{l}_{i2}, \ldots, \mathbf{l}_{in}\}$
Output: 0 or 1
$\mathbf{L}_i \leftarrow$ pre-processing $\mathbf{L}_i$
for j from 1 to $|L_i|$ **do**
  $pdf_{ij} \leftarrow$ select a pdf for $\mathbf{l}_{ij}$
end for
$f(t) \leftarrow$ apply Algorithm 1 with $\mathbf{L}_i$ and all pdf_{ij} as inputs
$p^+ \leftarrow \int_{0.5}^{1} f(t)dt$
$I(p^+) \leftarrow$ Eq. (20) with input p^+
return $I(p^+)$

CHAPTER IV: CROWD-WORKERS RELIABILITY

IVLs are designed to capture worker uncertainty, but this does not automatically assure that there are no low-quality workers. Workers often have varying levels of expertise and should not be given equal sway in inference making. To combat this, four worker reliability indexes, correctness, confidence, stability, and predictability, are derived using IVLs. These indexes can be used to weight a worker's labels based on performance rather than assuming all workers are equal. Computational experiments will then be run to compare quality differences between weighted and unweighted inference strategies.

4.1 Crowd worker reliabilities

The first measure of worker reliability is correctness which is the number of accurately labeled observations by a worker. A common approach to estimate a worker's correctness is using a set of gold questions. The ground truth of gold questions is known but is made deliberately unknown to workers when labeling. A list of gold questions is $G = [g_1, g_2, \ldots, g_k]$, with the ground truth being a binary string denoted as $o(G) \in \{0,1\}^k$. The IVL from a worker j on $g \in G$ is $\mathbf{l}_{gj} = [\underline{l}_{gj}, \bar{l}_{gj}]$. Taking the known ground truth of $g \in G$, $o(g)$, the center-correctness of $\mathbf{l}_{gj}$ is:

$$\text{center\_correctness}(l_{gj}) = \begin{cases} 1 - \text{mid}(l_{gj}) & \text{if } o(g) = 0, \\ \text{mid}(l_{gj}) & \text{if } o(g) = 1. \end{cases} \tag{22}$$

The center-correctness relies on both midpoint and the ground truth. For simplification, the ground truth is assumed to be equal to 1 for all $g \in G$. By doing so, $\mathbf{l}_{gj}$ can be replaced with its difference from 1, i.e. $1 - \mathbf{l}_{gj} = [1 - \bar{l}_{gj}, 1 - \underline{l}_{gj}]$ when $o(g) = 0$ with no issues because it doesn't change the center-correctness.

Take an IVL $\mathbf{l}_{gj} = [0.5, 0.7]$ with a $o(g)$ of 0. The midpoint of the IVL is 0.6 with a center-correctness of $1 - 0.6 = 0.4$. Taking that same IVL but converting $o(g)$ to 1 and replacing $\mathbf{l}_{gj}$ with $[1 - 0.7, 1 - 0.5] = [0.3, 0.5]$, gives a midpoint of 0.4. Eq. (22)

says that this is the same center-correctness regardless of which ground truth is used. It is from now on assumed that $o(g) = 1$ for all gold questions when $\mathbf{l}_{gj}$ is replaced with $1 - \mathbf{l}_{gj}$. Assuming this makes the $\mathrm{mid}(l_{gj})$ of an IVL on a gold question g by j the center-correctness. The values of $\underline{l}_{gj}$ and $\overline{l}_{gj}$ also refers to the label's min- and max-correctness, like the center-correctness.

Assume that $\mathbf{L}_G^j = [\mathbf{l}_{g_1 j}, \mathbf{l}_{g_2 j}, \ldots, \mathbf{l}_{g_k j}]$ is a list of IVLs from a specific worker j on G. Eq. (1) gives the mean of $\mathbf{L}_G^j$ as $\mu(\mathbf{L}_G^j) = [\mu(\underline{L}_G^j), \mu(\overline{L}_G^j)]$. The mean provides an estimate of a worker's overall correctness as the average min-, max-, and center-correctness: $\mu(\underline{L}_G^j), \mu(\overline{L}_G^j)$, and $\mathrm{mid}(\mu(L_G^j))$. Similarly, a worker j's average min-, max-, and center-correctness provides the means of the worst, best, and average correctness of j. The standard deviations of $\underline{L}_G^j, \overline{L}_G^j$, and $\mathrm{mid}(L_G^j)$ also provide information on the stability of j's min-, max-, and center-correctness. Along with G, there is also a list of regular questions U that have an unknown ground truth. $\mathbf{L}_U^j$ represents a worker j's IVLs on U. Where gold questions are for testing workers, regular questions are what crowdsourcing wants to answer.

Information about a worker's confidence can be found within an IVL. The midpoint of a particular IVL, $\mathrm{mid}(l)$, represents the degree of a worker's belief towards 0 or 1. A worker's confidence in their belief is reflected in the distance between the midpoint and 0.5, i.e. $|\mathrm{mid}(l) - 0.5|$. A worker can also have no confidence picking 0 or 1. This happens when $\mathrm{mid}(l) = 0.5$ which is why 0.5 is subtracted from the midpoint. Additionally, the radius of $\mathbf{l}$ specifies the maximum possible variation from the centroid. When $\mathrm{rad}(l) = 0$, the label is point-valued. A point-valued $\mathbf{l}$ means that the worker is confident in the labels value. Otherwise, the label contains the worker's uncertainty over a range. The maximum possible value of $\mathrm{rad}(l)$ is 0.5, so the difference between 0.5 and the radius of $\mathbf{l}$, i.e. $0.5 - \mathrm{rad}(l)$, is the measure of a worker's confidence on the centroid. Ultimately, the confidence of a worker on a single $\mathbf{l}$ is a combination of a worker's belief in their answer with the maximum

variation of the centroid. Worker confidence can then be defined through

$$\text{conf}(l) = |\text{mid}(l) - 0.5| + 0.5 - \text{rad}(l). \tag{23}$$

Since both $|\text{mid}(l) - 0.5|$ and $0.5 - \text{rad}(l)$ will result in values between 0 and 0.5, the confidence of $\mathbf{l}$ will be between 0 and 1. This is unlike binary-valued labels where confidence is always 100%. For a binary-valued label $l = 0$ (or 1), $|\text{mid}(l) - 0.5| = 0.5$ and $\text{rad}(l) = 0$. The confidence of a binary-valued label is not distinguishable, where IVLs can be differentiated using their confidence values. For instance, the confidence values of $[0.8, 0.9]$ and $[0.5, 0.7]$ are $0.8 = 0.35 + 0.45$ and $0.5 = 0.1 + 0.4$, respectively.

The mean of $\mathbf{L}_G^j$ also reflects a worker's overall confidence as $|\text{mid}(\mu(L_G^j)) - 0.5| + 0.5 - \text{rad}(\mu(L_G^j))$. This statement brings up two important caveats about j's confidence. The first is that a worker's overall confidence is not the same as the mean of conf($\mathbf{l}_{g_ij}$). The second is that the confidence of an IVL does not have any ties to the ground truth so calculating confidence is independent from the actual ground truth of the label. Estimating a worker's confidence using Eq. (23) only uses the mean of the IVL. Following this logic, the overall confidence of a worker on a set of regular questions U is calculable. Then by comparing the worker's confidences of $\text{conf}(L_G^j)$ and $\text{conf}(L_U^j)$, a worker's consistency can be statistically checked. If G is a good sample of U, then $\text{conf}(L_G^j)$ and $\text{conf}(L_U^j)$ should be statistically consistent.

The stability of a worker's labels comes from the standard deviation. As mentioned, a worker' j's min-, max-, center-correctness and confidence are point-valued. Since they are point-valued, the standard deviations of those measures can be calculated as normal. Using Eq. (16) $\sigma(L^j)$ can be calculated from $\mathbf{L}^j$ to estimate a worker's overall stability. Like the overall confidence of a worker, the standard deviation of $\mathbf{L}^j$ does not rely on the ground truth. This means that $\sigma(L_G^j)$ and $\sigma(L_U^j)$ can be compared for any significant differences in stability.

The predictability of a worker is also the entropy of $\mathbf{L}^j$ as applied according to information theory. Applying Algorithm 2, the entropy of $\mathbf{L}^j$ defined as $H(L^j)$ can be calculated. $H(L^j)$ then quantitatively measures a worker's predictability. According to the minimum entropy principle, a worker is more predictable if their entropy is less than that of other workers. Like the reliability measures of confidence and stability, the calculating entropy is independent of the ground truth. For a specific worker, the values of $H(L_G^j)$ and $H(L_U^j)$ can be calculated and statistically compared for differences as well.

4.2 Selection strategies

Worker selection is one of the most basic tools for quality improvement. The purpose of selection is to choose workers for crowdsourcing who pass certain criteria to assure expert labels. Previously, studies in worker selection only applied correctness based on binary-valued labels on gold questions [21], [29], etc. This changes with the introduction of interval-valued labels. A worker now has the four estimated reliability measures available as criteria in worker selection. The different reliability measures may differentiate j from other workers. To use the reliabilities for worker selection the relationship between the reliabilities should be investigated. Take a worker's correctness and confidence. The estimated center-correctness from $\mu(\mathbf{L}_G^j)$ of j is $\mathrm{mid}(\mu(L_G^j))$, which is between 0 and 1. The radius of $\mu(\mathbf{L}_G^j)$, $\mathrm{rad}(\mu(L_G^j))$, is between 0 and $\min\{\mathrm{mid}(\mu(L_G^j)), 1 - \mathrm{mid}(\mu(L_G^j))\}$. This ensures that $\mu(\mathbf{L}_G^j) \subseteq [0, 1]$. Based on Eq. (23), the confidence range for a given center-correctness is $|\mathrm{mid}(\mu(L_G^j)) - 0.5| + 0.5 - \min\{\mathrm{mid}(\mu(L_G^j)), 1 - \mathrm{mid}(\mu(L_G^j))\}$.

Fig. 4.1 illustrates the range of worker confidence versus correctness. The figure suggests that having a high correctness correlates with having a high level of confidence. Taking a look at the correctness range from 90% to 100% shows that a worker's level of confidence would be at least 80%. However, the same does not hold true when confidence is high. When a worker's confidence is at 80% or above; then according to the figure, the worker's correctness could be less than 20% or more than 80%. If using confidence as

the criteria for worker selection, then getting workers who have a correctness above 80% is incredible, but ultimately unsustainable since these workers are offset by those whose correctness is below 20%. However, since this work uses a binary classification model the labels from a worker with a correctness less than 20% can still be used. This is done by replacing the workers label $\mathbf{l}_{ij}$ with $1 - \mathbf{l}_{ij}$. As previously stated, replacing a label with its difference from one does not affect the center-correctness. Using this, it is expected that the average center-correctness of a worker would be above 80%.

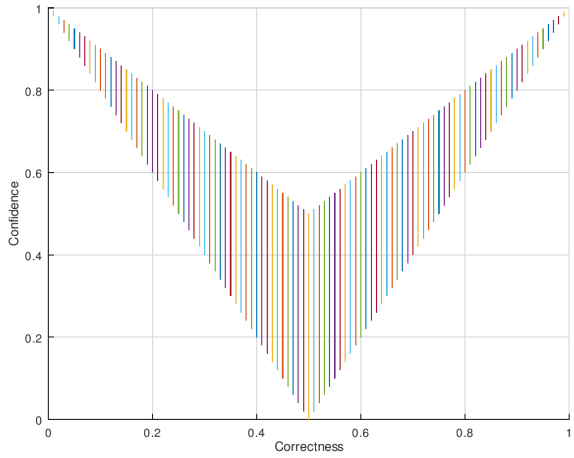

Figure 4.1: Range of confidence vs. correctness

The figure also shows that as the confidence level approaches 0, the center-correctness will converge on 0.5. Labels with low confidences like this would likely result in a tie. Because of this, replacing labels with the difference with 1 is not recommended with lower confidences if the correctness is below 50%. The label does not give enough information to justify such an act. Based on the above discussion, a set of guidelines for worker selection with IVLs is thus:

- Workers with a high level of confidence are preferred above a given threshold. A confidence threshold of above 80% guarantees the average correctness would be above 80% assuming a difference from 1 replacement when $\mathrm{mid}(\mu(L_G^j)) \leq 0.2$.

- When $\mu(\mathbf{L}_G^j)$ has mediocre confidence (between 40-60%), then worker correctness varies within a broad range of values. If the confidence level is 40%, correctness can fall anywhere between 30-70%. If selecting a worker with mediocre confidence is necessary, selecting the one with the highest correctness out of the group is preferred.

- A worker with a low confidence level is not helpful in labeling. If a worker's confidence is below 20%, then the average of the worker's IVLs would be at most 10% away from being a tie (50%). Low confidence workers should be removed through worker selection and reliability thresholds.

A worker's confidence also provides an additional criteria, other than worker correctness, that can be used for worker selection.

4.3 Weighted and unweighted inference making

The main objective of crowdsourcing is to obtain an inference through crowd-collected labels that matches the actual ground truth. Previously Eq. (18), a modification to Majority Voting, was used to obtain an inference from an interval-valued dataset. The equation counts the number of positive and negative IVLs to derive an inference. A more straightforward modification of the equation is to use the centroid $\mathrm{mid}(l_{ij})$ directly instead of counting each IVL. That is

$$W_i^+ = \sum_{\mathbf{l}_{ij} \in \mathbf{L}_i \wedge \mathrm{mid}(l_{ij}) > 0.5} \mathrm{mid}(l_{ij}), \text{ and} \tag{24}$$

$$W_i^- = \sum_{\mathbf{l}_{ij} \in \mathbf{L}_i \wedge \mathrm{mid}(l_{ij}) < 0.5} 1 - \mathrm{mid}(l_{ij}). \tag{25}$$

This gives the equation for interval majority voting (IMV) as

$$y_i = \begin{cases} 1 & \text{if } W_i^+ > W_i^-, \\ 0 & \text{if } W_i^+ < W_i^-, \\ tie & \text{otherwise.} \end{cases} \tag{26}$$

However, not all workers should have the same weight in inference making. Labels are not all created equal and some should be weighted more than others [41]. Instead of only using the centroid for inference making, j's reliability r_j can be used as the weight of the worker's label $\mathbf{l}_{ij}$. When doing so the algorithm for weighted interval majority voting (WIMV) becomes

$$W_i^+ = \sum_{\mathbf{l}_{ij} \in \mathbf{L}_i \wedge \text{mid}(l_{ij}) > 0.5} r_j \times \text{mid}(l_{ij}), \text{ and} \tag{27}$$

$$W_i^- = \sum_{\mathbf{l}_{ij} \in \mathbf{L}_i \wedge \text{mid}(l_{ij}) < 0.5} r_j \times (1 - \text{mid}(l_{ij})). \tag{28}$$

By using Eqs. (27) and (28) instead of (24) and (25) to make an inference with Eq. (26), WIMV can be used to make an inference from $\mathbf{L}_i$. The strength of the inference with WIMV (or IMV) can then be found using

$$\hat{p} = \max \left\{ \frac{W_i^+}{W_i^+ + W_i^-}, \frac{W_i^-}{W_i^+ + W_i^-} \right\}. \tag{29}$$

In the discussion above, any $\mathbf{l}_{ij} \in \mathbf{L}_i$ with $\text{mid}(l_{ij}) = 0.5$ are ignored, as those labels are not helpful towards inference making.

Majority voting was not the only inference making scheme previously discussed. The other strategy, preferred matching probability, applies a pdf of $\mathbf{L}_i$ defined in Eq. (17) to derive an inference. The probability of the overall preference of 1 on the observation is given by Eq. (19) since any value $t > 0.5$ implies a preference towards 1. Eq. (20) then results in an inference with PMP. Similar to Majority Voting, PMP can be improved by

applying reliability to the algorithm. The arithmetic average used in PMP gives all pdf_{ij} equal weight when making inferences. Similar to WIMV, the equation can be modified by multiplying j's reliability r_j with pdf_{ij} to act as a weight of worker reliability when calculating f_i. This gives the equation

$$f_i(t) = \frac{\sum_{j=1}^{m} r_j \times pdf_{ij}(t)}{\sum_{j=1}^{m} r_j}. \tag{30}$$

From there it is straightforward to verify that the $f_i(t)$ in Eq. (30) is a pdf of $\mathbf{L}_i$ too. Applying it with Eq. (19), p^+ can be evaluated to then make an inference using Eq. (20). This is called weighted preferred matching probability (WPMP). Like with PMP, the probability of matching is also $\hat{p} = \max\{P^+, 1 - P^+\}$ for WPMP.

There is one more question to answer about inference making: what reliability value should be chosen as the weight r_j? The main objective of learning is making an inference that is the same as the ground truth. Among the four reliability measures, only the correctness is directly associated with ground truth. Since the other three reliability indexes are not connected to the ground truth, a worker who has high confidence, stability, and predictability has a high possibility of incorrectly deducing the ground truth. This means that the most optimal reliability to use as the weight r_j would be a worker j's correctness.

4.4 Software design and implementation

To test the quality of the proposed methods for applying worker reliability, a pool of workers with various levels of reliability is generated. Fig. 4.2 shows an example pool with workers clustered according to correctness and confidence. Workers are generated using random seeds set between 0 and 1 with a uniform distribution. These random seed are the basis of the worker's IVLs. This leads to a lot of variation on where workers fall within Fig. 4.2. The variation may lead to problems with defining a set number of workers since depending on the generated seeds, there may be some seeds that don't appear as often if

the threshold is set at a very high percentage such as 90% or higher. An example of this is that starting out there could be 20 seeds that are in the more than 90% range, but after generating IVLs there are only 10 workers that fall within that range. In general, the number of selected workers should be the minimum number of workers within the chosen range of reliability. Taking this into account, a group of at least ten workers who pass a chosen reliability threshold (in this case worker confidence) are randomly selected for inference making. These workers are saved off to minimize variables when comparing inferences.

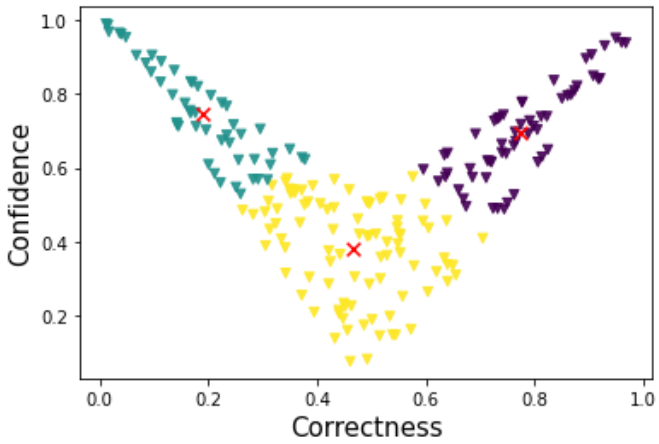

Figure 4.2: A pool of workers with different reliability

These selected workers then provide random IVLs on a benchmark CEKA [45] dataset, $Income94$. The $Income94$ dataset is a binary dataset with a total of 600 observations that have fifteen attributes. CEKA datasets provide the true class of the observations, with the $Income94$ observations being evenly split between 0 and 1. Using the collected IVLs, the proposed strategies of MV, IMV, PMP, WIMV, and WPMP are applied to make inferences. Since the dataset provides the ground truth, these inferences can be compared against the known ground truth to measure their accuracy. For each of the inferences obtained through the strategies, the confusion matrices [42] are recorded for the purpose of quality comparison between the methods. Table 4.1 records the results of a single run on the $Income94$ dataset with 180 observations that used a worker confidence threshold of 60% and no label correction, i.e. without the $1 - \mathbf{l}_{ij}$ replacement of $\mathbf{l}_{ij}$ when j's correct-

ness is very low. The worker confidence is used as the threshold measure since worker correctness is already being used as the weight for the strategies, although it could be used in tandem with confidence for even finer selection. The table indicates that both WIMV and WPMP produce much better results in terms of reduced false positives and negatives, while IMV and PMP are shown to miss a majority of ground truth. In addition, MV was only able to classify 148 items with the remaining 32 items being marked as a tie.

Strategy	TP	FP	TN	FN
MV	24	48	37	39
IMV	34	56	44	46
PMP	25	53	39	48
WIMV	74	3	97	6
WPMP	75	1	99	5

Table 4.1: Confusion matrices for strategies without correction

As a comparison, Table 4.2 shows the results of another test on the same dataset using the same confidence threshold of 60% and the same workers but utilizing label correction via replacing $\mathbf{l}_{ij}$ with $1 - \mathbf{l}_{ij}$ when j's confidence is greater than 90%. In this table, all five strategies produce better results than that reported in Table 4.1. For MV, IMV, and PMP, the number of false positive and negatives has shown a definite decrease although they are still not as good as WIMV and WPMP. These results are from correcting labels for workers with high confidence and low correctness. However, with or without worker correction the proposed weighted strategies of WIMV and WPMP are still the closest to being able to accurately match the ground truth.

Strategy	TP	FP	TN	FN
MV	45	30	55	18
IMV	60	30	70	20
PMP	49	35	62	23
WIMV	79	0	100	1
WPMP	78	2	98	2

Table 4.2: Confusion matrices for strategies with correction

Tables 4.1 and 4.2 indicate that both WIMV and WPMP can produce much better results with random workers whose confidence level is at least 60%. It is impractical to verify the findings with various confidence thresholds through counting numbers on confusion matrices. Instead, applying the metrics of recall, precision, accuracy, and F_1-score through the confusion matrices can be used to quantitatively measure the performance.

4.5 Computational results

The experiment design was implemented in Python 3 and the resulting metrics are shown graphically. These experiments continue to use the $Income94$ dataset. Fig. 4.3 shows the changes of recall, precision, accuracy, and F_1-score in the inference strategies as the confidence threshold increases by 1%. At each confidence threshold, ten workers are randomly selected from the pool for each run. The workers give IVLs on gold questions and the performance metrics are calculated based on the resulting confusion matrices. These scores are then averaged over forty runs. There is no specific reason for averaging over forty runs, the number only needs to be big enough to minimize any outliers due to the random factor. Label correction is also used on labels for workers with a correctness lower than 20% and high confidence.

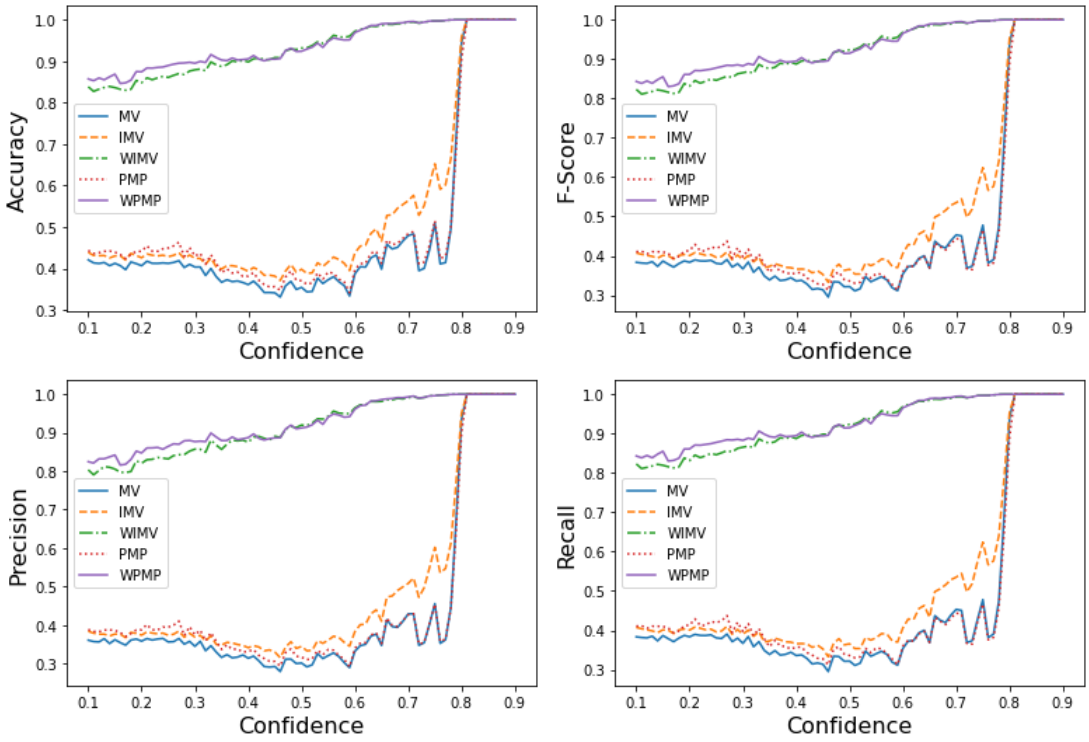

Figure 4.3: Performance measures vs. confidence threshold on Income94

It is evident that all four of metric figures have similar results. When the confidence threshold is lower than 60%, the qualitative measures for the inference made with the non-weighted strategies MV, IMV, and PMP are below 50%. As the confidence threshold increases beyond 60%, the measures start showing an increasing trend. At 80% confidence, there is a sharp spike of increase for the non-weighted strategies until the values finally converge at 1. Unlike the unweighted strategies, the measures for the weighted strategies WIMV and PMP remain above 0.8 even at lower confidences. WPMP is shown to be ahead of WIMV when the confidence is low, falls a small amount below it when confidence reaches 50%, and then falls in line with WIMV until 80% confidence where both converge at 1. PMP follows a similar pattern to WPMP, but instead of overtaking IMV, the values for IMV remain higher than PMP until the two converge at around 80%. From 50% and onward, PMP and MV have very similar results. From these figures, it is clear that the proposed weighted strategies produce significantly better values than the non-weighted strategies on the $Income94$ dataset.

However, the findings above are only on a single dataset. In order to verify that the findings are sound, the experiment is run again using the same workers but a different binary dataset from CEKA named Car. The Car dataset contains 1,594 observations that have seven attribute. 1,210 of those observations have a ground truth of 0 and the remaining 384 have a ground truth of 1. Fig. 4.4 reports the resulting performance measures. The four figures are very similar to those found using the $Income94$ dataset. The only noticeable differences between the two sets is that the precision and F_1-score start lower than the others in the Car dataset, around 0.1 versus the 0.4 in recall and accuracy, when the confidence threshold is low. This difference is likely due to Car being an imbalanced dataset. The accuracy and recall figures look similar to their counterparts in the $Income94$ dataset. While the precision and F_1-score start lower, the two measures still follow the same pattern of rise and fall as the $Income94$ figures.

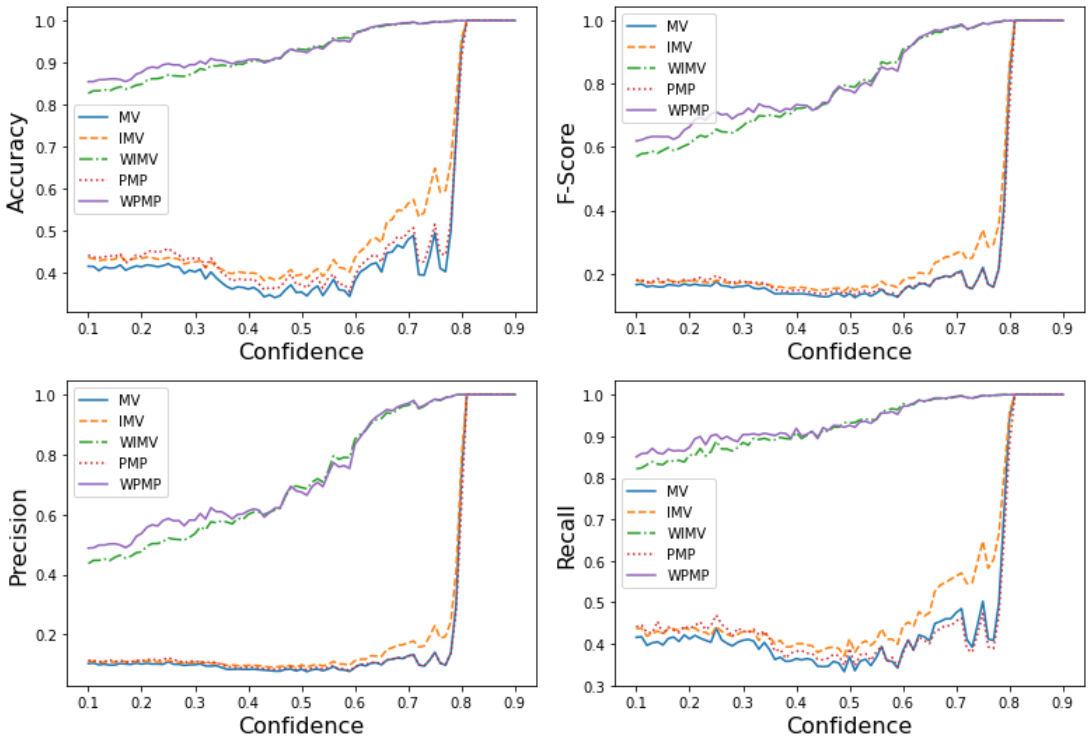

Figure 4.4: Performance measures vs. confidence threshold on Car

Regardless of the differences, all eight figures demonstrate a significant improvement on inferencing when using the weighted strategies instead of the unweighted. This is made especially apparent when the confidence threshold is low. In addition, making confidence the measure for worker selection has shown to be an effective way to get worker labels matching the ground truth when crowdsourcing. Another interesting result is that when the confidence threshold is high enough, all inference strategies lead to the ground truth after utilizing IVLs from naive attackers through label correction. The significant quality improvements on the inferences comes from the worker reliability weighted strategies proposed in this work.

CHAPTER V: ANOMALY DETECTION

The previous chapter assumed that every work will perform according to their reliability without considering anomalies. Given the varying intentions of workers and outside factors that can have an impact on the decision making process, this assumption is not assured. Quality assurance in real world applications requires anomaly detection and attacker identification being built-in parts of the process. This will include anomaly detection strategies using worker reliability, attacker identification and removal, and anomaly detection over time intervals. Computational experiments for examining the effects of rejecting anomalous workers or attacker on inference making will be run to compare quality.

5.1 Anomaly detection

Abnormal behavior in workers is unavoidable. To detect anomalies, detecting changes in reliability is paramount. This is done by comparing a worker's reliability indicators on a set of gold questions G and regular questions U. One problem with this is that calculating the correctness for U is impossible since it requires knowing the ground truth of the questions. However, the three remaining reliabilities can still be used since they are calculated independently of the ground truth. When comparing the reliabilities, if G well samples U, then the overall confidence, stability, and predictability derived from $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ should be statistically consistent. If there are significant changes in the reliabilities, this indicates a possibly anomaly.

In order to statistically compare a worker's confidence on $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$, the means $\mu(\mathbf{L}_G^j)$ and $\mu(\mathbf{L}_U^j)$ should be tested for significant differences. A standard method for examining if two samples have the same mean or not is the t-test [24]. Given that the t-statistic of two point-valued samples S_1 and S_2 is

$$t = \frac{|\mu(S_1) - \mu(S_2)|}{\sqrt{\frac{\sigma^2(S_1)}{|S_1|} + \frac{\sigma^2(S_2)}{|S_2|}}}, \tag{31}$$

whether the means of S_1 and S_2 are statistically consistent can be derived using a chosen statistical significance. Since $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ are interval-valued, the already proposed Eqs. (14), (15), and (16) calculate the mean, variances, and standard deviations of the lists. Taking these measures and using Eq. (4) for distance between intervals, the t-statistic for $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ is

$$t = \frac{\text{dist}\left(\mu(\mathbf{L}_G^j), \mu(\mathbf{L}_U^j)\right)}{\sqrt{\frac{\sigma^2(L_G^j)}{|L_G^j|} + \frac{\sigma^2(L_U^j)}{|L_U^j|}}}. \tag{32}$$

The degree of freedom in the previous equation is $|L_G^j| - 1$ because $|L_G^j|$ is less than $|L_U^j|$ usually. The number of gold questions available is normally much smaller than the number of regular questions for obvious reasons. On thing to note is that when calculating $\mu(\mathbf{L}_G^j)$ in Eq. (32), the label should not be replaced with the difference from 1 when the ground truth is 0 or $\mu(\mathbf{L}_G^j)$ will skew to the right of 0.5.

Another method for detecting anomalies is using a worker's stability. Remember that stability is derived from the standard deviation of the list of IVLs. To compare $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ using stability, the standard deviations $\sigma(\mathbf{L}_G^j)$ and $\sigma(\mathbf{L}_U^j)$ need to be compared for significant differences. Eq. (2) also allows for calculating the variances $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$. Once the values have been calculated, the F-test [25] can then be applied to compare the standard deviations of the list of IVLs on the gold and regular questions. If the two variances are not significantly different, then their ratio will be close to 1. It should be mentioned that if a worker's $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ are determined statistically different that the worker is not automatically an malicious attacker. The t- and F-tests determine only if a worker has behaved anomalously and may need evaluation into why their behavior has changed. Also note that the tests only use the worker's stability and confidence – not predictability. This is because predictability between workers does not vary much and the research behind an interval method for comparing the statistical consistency is still lacking. As stated before, the quality of a worker is not constant, outside factors can cause fluctuations in the quality of labels. While crowdsourcing does not provide access to the possible outside factors that explain

these fluctuations, by using worker's labels and statistically checking for differences the quality of a worker can be monitored and addressed, if needed.

5.2 Identifying possible attackers

As with any type of open group structure, there is a possibility of attackers with destructive intentions. A challenging task in crowdsourcing is identifying and excluding those who are very knowledgeable but have adversary purposes [4], [29]. In this work, attackers are categorized into two groups: naive or sophisticated. Naive attackers who are informed will strive to classify all observations opposite the actual ground truth. In the previous chapter, these are the workers with high confidence and low correctness. These attackers are neutralized through the replacement of their labels with the difference from 1, allowing their labels to still be used in inference making. Where naive attackers are easy to combat, sophisticated attackers are more complicated as they are more likely to identify gold questions and answer them with a high level of accuracy and then deliberately label the regular questions incorrectly.

In order to accurately identify both kinds of attackers, workers associated with a high or low correctness should be targeted. While a naive attackers IVLs can still be used, $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ will first need to be checked to determine if they are statistically consistent with each other. By doing so, this guarantees that the worker will be consistent in their label making, even though they label opposite the ground truth, and that the IVLs can be replaced with the difference from 1. If the two are shown to be statistically inconsistent, the IVLs are discarded. Conversely, identifying a possible sophisticated attacker requires checking if $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ are significantly different statistically. This shows that the worker had actively changed their method of labeling when going from gold questions to regular questions. In previous works such as [29] and others, very sophisticated attackers have been shown to purposefully get a worse correctness score when answering gold questions to avoid suspicion. When a worker tries to lower their correctness using IVLs, the confi-

dence level will change. Therefore, instead of only running tests on workers with high or low correctness, t- and F-tests should be performed on every $j \in J$ to identify possible attackers. Any worker that fails both tests has a high likelihood of being an attacker and should be marked for evaluation.

5.3 Dynamically monitor behavior through time-intervals

Malicious intent is not the only reason for abnormal worker behavior. Factors such as experience, stress, environment, and even random mistakes can lead to bad quality labels every so often. Due to this, the behavior of a crowd-worker j can vary from moment to moment. The solution to this is to monitor a workers behavior dynamically rather than statically. By presenting new gold questions to j periodically, $\mathbf{L}_G^j$ effectively becomes a time series. With the assumption that behavior before the time series does not have influence on the current behavior and after, a time window T can be used to contain the IVLs under consideration. A worker j's IVLs on gold and regular questions within the time window T can then be denoted using $\mathbf{L}_{G_T}^j$ and $\mathbf{L}_{U_T}^j$, respectively. To detect possible anomalies, t- and F-tests can be performed on $\mathbf{L}_{G_T}^j$ and $\mathbf{L}_{U_T}^j$ within the time window T. For convenience, the assumption that $|L_{G_T}^j| = k$ holds true. When the next $\mathbf{l}_{g_{k+1}j}$ becomes available the window is moved one step forward. Doing so, the list of gold question IVLs within the time window $T+1$ is $\mathbf{L}_{G_{T+1}}^j = \{\mathbf{l}_{g_2 j}, \mathbf{l}_{g_3 j}, \ldots, \mathbf{l}_{g_{k+1} j}\}$. The IVLs of regular questions within the time-window $T+1$ is $\mathbf{L}_{U_{T+1}}^j$. As the time window T increases, by performing t- and F-tests on $\mathbf{L}_{G_{T+1}}^j$ and $\mathbf{L}_{U_{T+1}}^j$, possible anomalies can be detected and identified.

5.4 Software design and implementation

For statically detecting anomalies and attackers, a set of a hundred crowd-workers J with various reliabilities is generated. Next, three binary CEKA datasets [45] $Income94$, $Sick$, and $Vote$ are chosen because they include the ground truth. The $Sick$ dataset contains

3,772 observations with thirty attributes. For it, 3,541 observations have a ground truth of 0 and 231 have a ground truth of 1. For the $Vote$ dataset, there are 435 observations with seventeen attributes. 267 of the observations have a ground truth of 0 and the remaining 168 have a ground truth of 1. A portion (one third) of each dataset forms the set of gold questions G. That results in 200 gold questions from $Income94$, 1,257 from $Sick$, and 145 from $Vote$. The remaining will form the set of regular questions U.

Next, all workers $j \in J$ provide IVLs on G and U. This gives a $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ for each generated worker. The IVLs in $\mathbf{L}_G^j$ will follow the workers reliability indexes with slight randomization so that the reliabilities are not completely stagnant. For the majority of workers, their labels in $\mathbf{L}_U^j$ will also follow the worker's reliability. For workers designated as attackers, their IVLs will be modified to simulate an attacker's IVLs. One experiment only detects anomalies in behavior that may or may not be caused by adverse intentions, and the other specifically targets attackers. How a worker's IVLs are altered may change based on the experiment being run.

While checking a worker's behavior through a single time period is helpful, in reality worker behavior is not static. The reliabilities of real-world workers would change whenever they participate in crowdsourcing, even if that change is small. For the dynamic detection experiments, the J defined in the previous paragraph can still be used. The three CEKA datasets are still split into portions, but are treated as pools of questions. Instead of having a workers give IVLs on all questions at once, the workers are given a smaller number of starting questions from G and U with one being added as the time window moves forward. This would continue until the total number of IVLs reaches a predetermined max or until all questions have been used. Intervals will be added to $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ as new questions from G and U are introduced. The tests will be run to compare $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ for various time intervals after a specific number of questions has been labeled by the workers. Workers acting as attackers are tracked throughout the process and give bad IVLs as the questions are introduced. The effectiveness of the anomaly detection and attacker

identification via the test statistics will be observed through both the static and dynamic experiments. The experiment results and impacts of the attackers on the inferences are reported in the following.

5.5 Computational results

The first experiment tests whether the proposed statistical tests can effectively detect anomalies within crowd-workers. For this, a fifth of the worker pool is chosen to behave abnormally, regardless of reliability. The designated anomalous workers provide IVLs on U that are set away from their actual reliability indexes. This is done by modifying the IVLs through random variation on the lower and upper bounds. Then for all workers, t- and F-tests' are performed to determine if the confidence and stability of $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ differ statistically or not using a confidence level of at least 95%. If only one of the two reliabilities differ, then the worker is possibly anomalous. It is when both reliabilities differ that the worker is most likely anomalous and should be marked for examination. The results of the test on the $Income94$ dataset is given in Fig. 5.1, which visually highlights the detected anomalous workers out of the entire worker pool through a correctness-confidence graph. Anomalous workers are depicted as dots with x's through them in the figure. Out of the twenty workers selected from the pool to act anomalous, sixteen were successfully identified. Additionally, the experiment was run forty times and the best, worst, and average percentage for identified anomalies versus actual anomalies was recorded. The collected percentages were 100%, 65%, and 84.1%. These show that the method can detect anomalous workers effectively the majority of the time. However, anomalies do not automatically mean a worker is an attacker. A worker with previously bad reliability giving better labels would be marked as anomalous based on these methods. How can attackers be identified and neutralized when inference making?

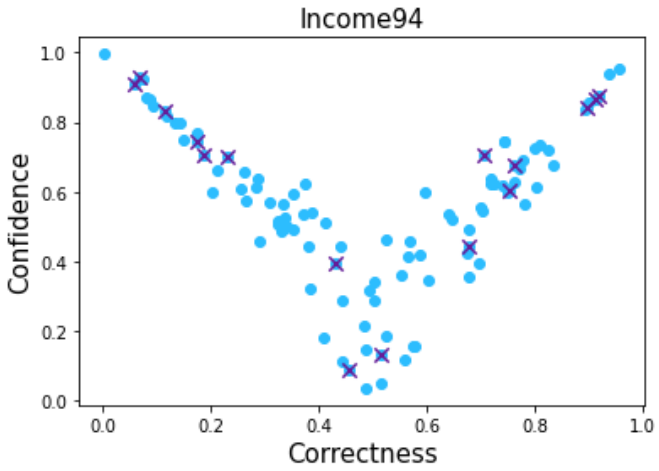

Figure 5.1: Statistically inconsistent workers in Income94 dataset

Previously when obtaining inferences, only naive attackers were managed during the process with sophisticated attackers not being addressed at all. This leaves the question of how to identify attackers before the inference making process? As previously stated, to identify knowledgeable attackers the focus should be placed on workers with a high level of confidence who have either a high or low correctness. A low level of confidence usually signals that a worker has an insufficient level of knowledge. Labels from such workers are normally discarded. Thus, to identify possible attackers, only workers with a high level of confidence need to be checked for consistency. Regardless of whether a worker is a naive or sophisticated attacker, the IVLs in $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ would not be consistent statistically. For the experiment, only workers with a greater than 60% confidence threshold give IVLs. Twenty out of the hundred workers are picked to simulate attackers. After performing the statistical tests for each $j \in J$, the null hypothesis that $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ are the same statistically at a given level of significance can be accepted or rejected. Fig. 5.2 highlights workers who fail the tests and demonstrates the overall effectiveness of the proposed approach for identifying potential attackers. Taking a closer look at the graph, it is noticeable that only fourteen of the twenty designated attackers were found. This is expected. Just because the means and standard deviation of $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ are consistent does not mathematically ensure

that a worker will behave normally, although the inconsistency is what suggests the worker is likely an attacker.

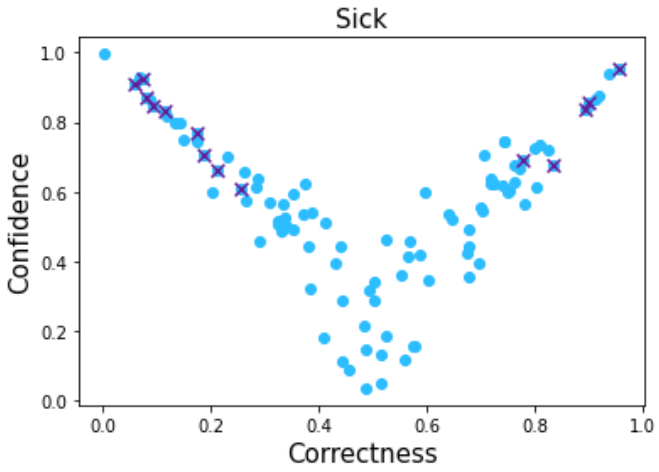

Figure 5.2: Statistically inconsistent workers in Sick dataset

The previous experiments detected anomalies statically through a large list of IVLs for workers on gold and regular questions. For dynamically monitoring workers, the workers are not given access to the entirety of the two question pools at one time. Instead, new IVLs are provided to workers to act as different time intervals. After a certain number of IVLs has been provided to the workers, the anomaly detection tests are run on $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$. Like with the previous experiments, twenty workers are singled out to act as attackers. Fig 5.3 shows the results of anomaly detection at 5, 10, 50, 100, 150, and 200 IVLs. It can be seen that for the lower numbers of IVLs not all anomalies are detected by the method. However, as the number of IVLs increases, the anomalies start to be singled out more. This is because unlike with static experiments, this system remembers anomalies from previous time intervals. Of course like with static detection this system can be used for detecting adversarial attackers by implementing a reliability threshold for above 60% confidence. The difference between the two is that by actively monitoring workers, the method can catch anomalies while crowdsourcing rather than after. Anomalies can be caught in the act and removed before they have given a large number of IVLs on regular questions. This method is preemptive rather than reactive.

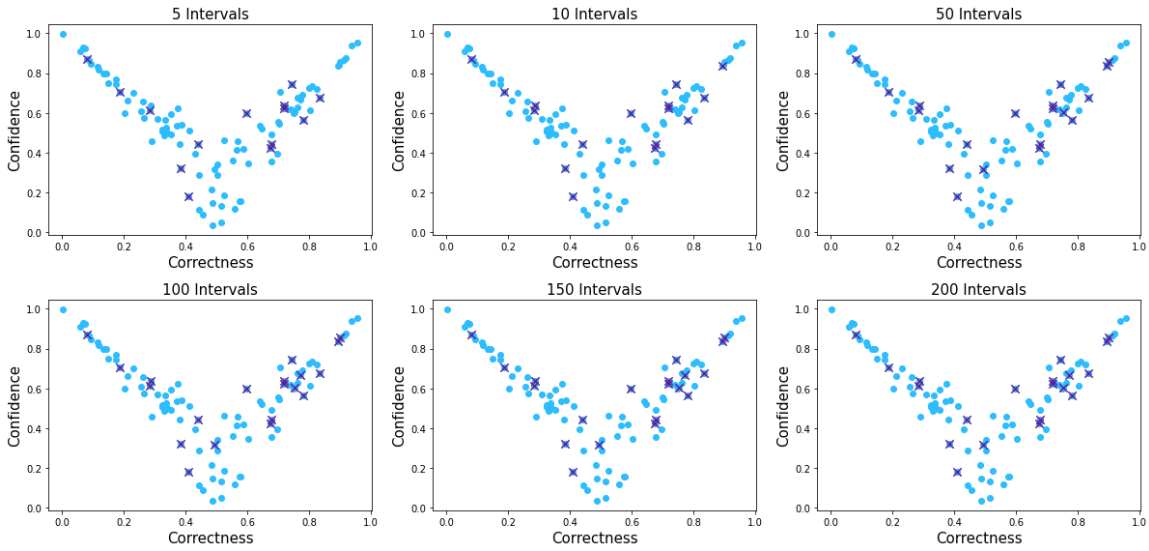

Figure 5.3: Anomalies detected through different time intervals

The main purpose of identifying attackers is to be able to exclude them and improve the quality of crowdsourced work. What is the use of identifying attackers if they are not definitively dealt with? To view the impact of removing attackers on inferencing, the weighted strategy WIMV was employed to make new inference on the $Vote$ dataset with and without removing attackers. WIMV was used instead of WPMP since both weighted strategies provide similar results and WIMV is simpler to implement. IVLs are collected from fifteen randomly selected workers for each $u \in U$. An inference is then made on each $\mathbf{L}_U^j$ using WIMV. Confusion matrices are then calculated since the test datasets include the ground truth. The process is run twice, with one being attacker-included and the other attacker-excluded. The tests are run twenty times due to the random factor and the average of those runs is used for the confusion matrices. Using the confusion matrices, quantitative measures of recall, precision, accuracy, and F_1-score are calculated. Fig. 5.4 then compares F_1-scores of attacker-included and -excluded runs on the $Vote$ test dataset. The horizontal axis represents the number of attackers present in the entire worker pool. At a lower number of attackers, the figure shows for the attacker-included test the non-attacker workers are able to overpower the attackers bad labels and still achieve a good result. However, with the increasing number of attackers, the chance that a selected worker is an attacker goes up

and correspondingly the F_1-score drastically falls. Conversely, when excluding identified attackers, the F_1-score remains perfect or close to it even when facing a large number of attackers. Even though the attacker-excluded does not identify all attackers, identifying and removing the majority allows good workers to overpower the bad labels.

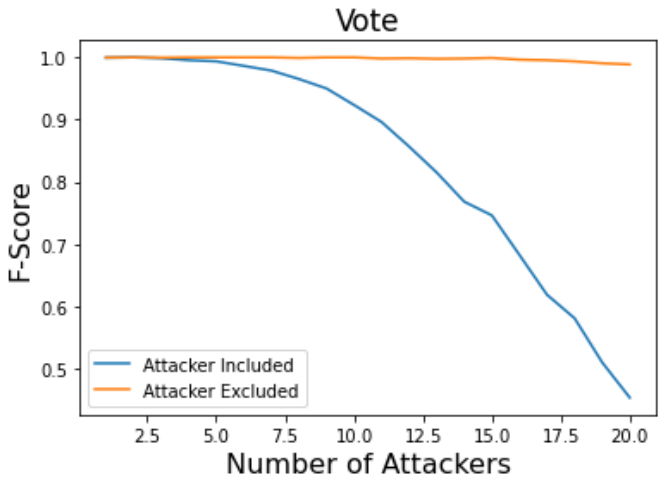

Figure 5.4: F_1-score with increasing numbers of attackers

CHAPTER VI: CONCLUSIONS AND FUTURE WORKS

Interval-valued labels have proven to be a reliable and trustworthy method for improving the quality of crowdsourced work. IVLs contain more information than regular labels and come with their own sets of statistical and probabilistic properties. By using them, the reliability of a particular worker can be estimated in terms of correctness, confidence, stability, and predictability. With this, worker selection can be employed for choosing quality crowd-workers through reliability thresholds. Along with worker selection, two weighted algorithms WIMV and WPMP, were used to make weighted inferences on IVLs. When applying worker selection to these algorithms, the results prove that the two approaches produce significantly better quality inferences when comparing against other strategies.

The success of applying worker reliability to crowdsourcing leads to applying worker reliability in detecting anomalous behavior in crowd-workers. It is not always assured that a worker will perform according to their reliability indexes. For this, strategies for anomaly detection and attacker identification have been outlined in this work. Of the four reliabilities only the confidence, stability, and entropy are independent of the ground truth and can be compared through gold and regular questions. By comparing the resulting reliabilities for statistical consistency, abnormal behavior from a worker can be recognized. Assuming that the gold questions well sample the regular, any disparities within the means and/or standard deviations of a workers list of IVLs on gold and regular questions, $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$, suggest a worker acts anomalously. When applying the detection strategies, all strategies have been effective in identifying malicious attackers from regular crowd-workers as shown in Fig. 5.2. Fig. 5.4 further shows how important including and excluding attackers can be on the quality of crowdsourcing. With an increasing numbers of attackers, the quality measure of F_1-score is shown falling dramatically in the attacker-included runs. When removing attackers, the quality from the WIMV algorithm returns near perfect results.

There are two items to take note of. One is that the detection approach does not mathematically guarantee that all anomalous workers will be detected. The other is that

a worker's means and standard deviations of $\mathbf{L}_G^j$ and $\mathbf{L}_U^j$ being consistent does not assure that a worker is not an attacker. This shows that further research into worker reliability and anomaly detection is still required. A couple of topics for exploration include the utilization of an approximated spectrum or applying a confidence interval towards anomaly detection. More research into dynamic anomaly detection with time intervals is also a possibility, especially as this work does not take into account outlier when shifting the time intervals. Moving away from anomaly detection, there is also an avenue of research in applying neural networks towards interval-valued labeling. IVLs could also be applied to text summarization through the numerical rankings associated with text comments for measuring a reviewers reliability. Regardless of the future directions interval-valued labeling could be taken, this work shows that interval-valued labels, worker reliability, and anomaly detection are critical for quality improvements in crowdsourcing and an important topic within the scientific community.

REFERENCES

[1] Barbosa, N., and Chen, M.: Rehumanized crowdsourcing: a labeling framework addressing bias and ethics in machine learning. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1-12, (2019)

[2] Bentkowska, U.: New types of aggregation functions for interval-valued fuzzy setting and preservation of pos-B and nec-B-transitivity in decision making problems. Information Sciences 424(C), pp. 385-399, (2018)

[3] Bi, W., Wang, L., Kwok, J., and Tu, Z.: Learning to predict from crowdsourced data. UAI'14: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, pp. 82-91 (2014)

[4] Checco, A., Bates, J., and Demartini, G.: Adversarial attacks on crowdsourcing quality control. J. of Artificial Intelligence Research 67, pp. 375-408, (2020)

[5] Dai, J., Wang, W., Mi, J.: Uncertainty measurement for interval-valued information systems. Information Sciences, pp. 63-78, (2013)

[6] Grabisch, M., Marichal, J., Mesiar, R., Pap, E.: Aggregation functions. Cambridge University Press, New York, (2009)

[7] He, L., and Hu, C.: Midpoint method and accuracy of variability forecasting. J. Empirical Economics 38, pp. 705-715, (2009)

[8] He, L., Hu C.: Impacts of interval computing on stock market forecasting. J. of Computational Economics 33(3), pp. 263-276, (2009)

[9] Hu, C. and *et al*: Knowledge processing with interval and soft computing. Springer-Verlag, London (2008)

[10] Hu, C., Frolov, A., Kearfott, R., Yang, Q.: A general iterative sparse linear solver and its parallelization for interval Newton methods. Reliable Computing 1, pp. 251–263, (1995)

[11] Hu, C., Cardenas, A., Hoogendoorn, S. et al. An interval polynomial interpolation problem and its Lagrange solution. Reliable Computing 4, pp. 27-38, (1998)

[12] Hu, C.: Using interval function approximation to estimate uncertainty. In: Huynh VN., Nakamori Y., Ono H., Lawry J., Kreinovich V., Nguyen H.T. (eds) Interval / Probabilistic Uncertainty and Non-Classical Logics. Advances in Soft Computing, vol 46. Springer, Berlin, Heidelberg (2008)

[13] Hu, C. and He, L.: An application of interval methods to stock market forecasting. J. Reliable Computing 13, pp. 423-434, (2007)

[14] Hu, C.: A note on probabilistic confidence of the stock market ILS interval forecasts. J. Risk Finance 11 (4), pp. 410-415, (2010)

[15] Hu, C.: Interval function and its linear least-squares approximation. ACM SNC '11: Proceedings of the 2011 International Workshop on Symbolic-Numeric Computation, pp. 16-23, (2012)

[16] Hu, C., and Hu, ZH.: On statistics, probability, and entropy of interval-valued datasets. In: Lesot MJ. et al. (eds) Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU 2020. Communications in Computer and Information Science 1239, pp. 422-435, (2020)

[17] Hu, C., and Hu, ZH.: A computational study on the entropy of interval-valued datasets from the stock market. In: Lesot MJ. et al. (eds) Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU 2020. Communications in Computer and Information Science 1239, pp. 407-421, (2020)

[18] Hu, C. Sheng, SV., Wu, N., and Wu, X.: Managing uncertainties in crowdsourcing with interval-valued labeling. In: Rayz J., Raskin V., Dick S., Kreinovich V. (eds) Explainable AI and Other Applications of Fuzzy Techniques. NAFIPS 2021. Lecture Notes in Networks and Systems 258, pp. 166-178, (2021)

[19] Huynh, V., and *et al*: On decision making under interval uncertainty: A new justification of Hurwicz optimism-pessimism approach and its use in group decision making. 39th Int. Sym. on Multiple-Valued Logic, pp. 214-220, (2009)

[20] Korvin, A., Hu, C., and Chen, P.: Generating and applying rules for interval valued fuzzy observations. Lecture Notes in Computer Science 3177, pp. 279-284, (2004)

[21] Li, H., Liu, Q.: Cheaper and Better: Selecting Good Workers for Crowdsourcing. https://arxiv.org/abs/1502.00725 (2015)

[22] Marupally, P., Paruchuri, V., Hu, C.: Bandwidth variability prediction with rolling interval least squares (RILS). In: Proceedings of the 50th ACM SE Conference, Tuscaloosa, AL, USA, March 29-31, 2012, pp. 209-213, (2012)

[23] Moore, R. E.: Methods and applications of interval analysis. SIAM Studies in Applied Mathematics, Philadelphia, (1979)

[24] NIST: Do two processes have the same mean? https://www.itl.nist.gov/div898/handbook/prc/section3/prc31.htm

[25] NIST: F-test, www.itl.nist.gov/div898/handbook/eda/section3/eda359.htm

[26] Nordin, B., Hu, C., Chen, B., and Sheng, V.S.: Interval-valued centroids in K-means algorithms. In: Proceedings of the 11th IEEE Int. Conf. on Machine Learning and Applications (ICMLA), pp. 478-481, (2012)

[27] Parer, J., and Hamilton, E.: Comparison of 5 experts and computer analysis in rule-based fetal heart rate interpretation. Am J. Obstetrics Gynecology 203(5), pp. 451.E1-451.E7, (2010)

[28] Pkala, B.: Uncertainty data in interval-valued fuzzy set theory: Properties, Algorithms and Applications (1st ed.) Springer, (2018)

[29] Qiu, L., *et al*: CrowdSelect: Increasing accuracy of crowdsourcing tasks through behavior prediction and user selection. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 539-548, (2016)

[30] Rajpal, S., Goel, K., and Mausam, M.: POMDP-based control of workflows for crowdsourcing. Proceedings of the 32nd International Conference on Machine Learning, pp. 52-85, (2015)

[31] Rhodes, C., Lemon, J., and Hu, C.: An interval-radial algorithm for hierarchical clustering analysis. 14th IEEE Int. Conference on Machine Learning and Applications (ICMLA), pp. 849-856, (2015)

[32] Shannon, C. -E.:, A mathematical theory of communication. The Bell System Technical Journal 27, pp. 379-423, (1948)

[33] Sheng, V.S., Provost, F., and Ipeirotis, P.: Get another label? Improving data quality and data mining using multiple, noisy labelers. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 614-622, (2008)

[34] Sheng, V.S., and Zhang, J: Machine learning with crowdsourcing: A brief summary of the past research and future directions. Proceedings of the AAAI Conference on Artificial Intelligence 33(1), pp. 9837-9843, (2019)

[35] Sheng, VS., Zhang, J., Bin, G., Wu, X.: Majority voting and pairing with multiple noisy labeling. IEEE Transactions on Knowledge and Data Engineering 31(7), pp. 1355-1368 (2019)

[36] Smyth, P.: Learning with probabilistic supervision. In Computational Learning Theory and Natural Learning Systems Vol. III, MIT Press, (1995)

[37] Spurling, M., Hu, C., Sheng, VS., Zhang, H.: Estimating crowd-worker's reliability with interval-valued labels to improve the quality of crowdsourced work. 2021 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 01-08, (2021)

[38] Spurling, M., Hu, C., Zhan, H., Sheng, V.S.: Anomaly Detection in Crowdsourced Work with Interval-Valued Labels. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU 2022. Communications in Computer and Information Science, vol 1601. Springer, Cham, (2022)

[39] Tao, F., Jiang, L., Li, C.: Label similarity-based weighted soft majority voting and pairing for crowdsourcing. Knowledge and Information Systems 62, pp. 2521-2538, (2020)

[40] Wang, G., Wang, T., Zheng, H., and Zhao, B.: Man vs. machine: practical adversarial detection of malicious crowdsourcing workers. Proc. of the 23rd USENIX Security Symposium, pp. 239-254, (2014)

[41] Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J.: Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. Advances in Neural Information Processing Systems 22, pp. 2035-2043, (2008)

[42] Wikipedia, Confusion matrix en.wikipedia.org/wiki/Confusion_matrix

[43] Wikipedia, Information entropy, https://en.wikipedia.org/wiki/Entropy_(information_theory)

[44] Wolfram Mathworld, Binary normal distribution http://mathworld.wolfram.com/BivariateNormalDistribution.html

[45] Zhang, J., Sheng, VS., Nicholson, B., and Wu, X.: CEKA: A Tool for Mining the Wisdom of Crowds. Journal of Machine Learning Research 16, pp. 2853-2858, (2015)

[46] Zhang, J., Wu, X., and Sheng, V.S.: Learning from crowdsourced labeled data: a survey. Artif. Intell. Rev 46, pp. 543-576, (2016)

[47] Zhang, X., Pan, X., and Wang, S.: Label quality improvement in crowdsourcing with ensemble TSK fuzzy classifier. 2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 290-296, (2019)