A Universal Unbiased Method for Classification from Aggregate Observations

Zixi Wei[1] Lei Feng[2 3] Bo Han[4 3] Tongliang Liu[5 6 3] Gang Niu[3] Xiaofeng Zhu[7] Hengtao Shen[7]

Abstract

In conventional supervised classification, true labels are required for *individual* instances. However, it could be prohibitive to collect the true labels for individual instances, due to privacy concerns or unaffordable annotation costs. This motivates the study on *classification from aggregate observations* (CFAO), where the supervision is provided to *groups* of instances, instead of individual instances. CFAO is a generalized learning framework that contains various learning problems, such as multiple-instance learning and learning from label proportions. The goal of this paper is to present a novel *universal* method of CFAO, which holds an *unbiased estimator* of the classification risk for *arbitrary losses*—previous research failed to achieve this goal. Practically, our method works by weighing the importance of each label for each instance in the group, which provides purified supervision for the classifier to learn. Theoretically, our proposed method not only guarantees the *risk consistency* due to the unbiased risk estimator but also can be compatible with arbitrary losses. Extensive experiments on various problems of CFAO demonstrate the superiority of our proposed method.

1. Introduction

Classification is one of the most frequently encountered problems in machine learning (Kotsiantis et al., 2006). Over the past ten years, deep learning models have achieved promising performance on various classification tasks, while such a success heavily relies on a large number of high-quality labels for *individual* instances (Jordan & Mitchell, 2015). In many real-world scenarios, it could be prohibitive to collect strong supervision information for individual instances, due to privacy issues, confidentiality concerns, or unaffordable annotation costs. These challenges of individual annotations motivate us to consider the supervision for *groups* of instances, instead of individual instances.

The supervision of groups of instances can be feasible in many realistic applications. For example, to protect the information of individual data points, some summary statistics of groups of data could be disclosed. For the drug activity prediction task (Dietterich et al., 1997), individual annotations are inaccessible, while group annotations are provided. Besides, collecting the supervision information of groups of instances could incur much fewer costs when the annotations costs are unaffordable for individual instances (Zhou, 2018). In addition, making predictions about individual-level behavior based on group-level data also has gained much attention from other fields, such as ecological inference (Schuessler, 1999; Flaxman et al., 2015) and preference learning (Fürnkranz & Hüllermeier, 2003; Chu & Ghahramani, 2005).

The feasibility of group supervision motivates us to investigate the task of *classification from aggregate observations* (CFAO), where we aim to learn a classifier with only supervision on groups of instances. In CFAO, training data are represented by groups of instances and we can only observe the aggregate information of the group. CFAO can be considered as a general learning framework, which contains various learning problems with different types of aggregate information. A well-known problem belonging to CFAO is *multiple-instance learning* (Maron & Lozano-Pérez, 1997; Carbonneau et al., 2018), where the aggregate information is whether the group has at least one positive instance. Another typical problem is *learning from label proportions* (Yu et al., 2013; Scott & Zhang, 2020), where the aggregate information is the proportion of instances from each class in the group. In recent years, CFAO has received increasing attention and some interesting problems of CFAO have been investigated. Bao et al. (2018a) studied *classification from pairwise similarities*, where the aggregate information is whether two instances in the group belong to the same

[1]College of Computer Science, Chongqing University, China [2]School of Computer Science and Engineering, Nanyang Technological University, Singapore [3]RIKEN Center for Advanced Intelligence Project, Japan [4]Department of Computer Science, Hong Kong Baptist University, China [5]Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates [6]Sydney AI Centre, School of Computer Science, The University of Sydney, Australia [7]School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. Correspondence to: Lei Feng <lfengqaq@gmail.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

class (similar) or not (dissimilar). Cui et al. (2020) studied *classification from triplet comparisons*, where the aggregate information is whether one instance is more similar to the other one, compared with the third one.

The goal of this paper is to propose a *universal* approach that can be applied to various types of aggregate observations for CFAO. To the best of our knowledge, there is currently only one universal approach (Zhang et al., 2020) that can be used in various problems of CFAO. This approach is based on maximum likelihood estimation and can gain the theoretical property of restricted classifier consistency under specific conditions. However, this approach has both practical and theoretical limitations. For the practical limitation, it fails to consider differentiating the true label for each instance in the group, which could limit its empirical performance due to the lack of any purified supervision information of individual instances. For the theoretical limitation, it cannot guarantee the *risk consistency*, i.e., the estimator (by aggregate observations) is *biased* to the classification risk (by fully labeled data). Besides, it has a strict restriction on the used loss function due to the log-likelihood, making it not flexible enough when the loss needs to be changed with the dataset in practice.

In this paper, we propose a universal unbiased method for CFAO, which holds an *unbiased estimator of the classification risk*. Although many previous studies have explored unbiased risk estimators (UREs) to solve specific weakly supervised learning problems (Ishida et al., 2018; Cao et al., 2021; Feng et al., 2021), they only focused on a certain learning problem. This limits the usage of existing UREs, and whether there exists a universal URE for various problems of CFAO is still unknown. We for the first time give an affirmative answer to this question. Our proposed universal URE works by weighing the importance of each label for instance in the group, which could guide the classifier to identify the true label for gaining purified supervision of individual instances. Theoretically, the risk consistency of our method can be naturally guaranteed due to its unbiasedness, which also makes our method compatible with arbitrary losses. Extensive experiments on various problems of CFAO demonstrate the superiority of our method.

2. Preliminary Knowledge

In this section, we introduce preliminary knowledge of the supervised classification task and the task of classification from aggregate observations.

2.1. Supervised Classification

Let $\mathcal{X} \in \mathbb{R}^d$ be the feature space with d dimensions and $\mathcal{Y} \in [k]$ be the label space. We denote by $\boldsymbol{x} \in \mathcal{X}$ a feature vector and $y \in \mathcal{Y}$ the ground-truth label of $\boldsymbol{x}$. Each example $(\boldsymbol{x}, y)$ is supposed to be sampled from an underlying distribution with probability density $p(\boldsymbol{x}, y)$. For the k-class classification task, the goal is to train a learning function $f : \mathcal{X} \mapsto \mathbb{R}^k$ that tries to the classification risk defined as

$$R(f) = \mathbb{E}_{p(\boldsymbol{x},y)}\big[\mathcal{L}(\boldsymbol{x}, y; f)\big], \tag{1}$$

where $\mathbb{E}_{p(\boldsymbol{x},y)}[\cdot]$ denotes the expected value over $p(\boldsymbol{x}, y)$ and $\mathcal{L}$ denotes a classification loss (e.g., the softmax cross entropy loss: $\mathcal{L}(\boldsymbol{x}, y; f) = -\log \frac{\exp(f_y(\boldsymbol{x}))}{\sum_{j=1}^{k}\exp(f_j(\boldsymbol{x}))}$), where $f_y(\boldsymbol{x})$ is the y-th element of $f(\boldsymbol{x})$. The predicted label $\hat{y}$ of $\boldsymbol{x}$ is given as $\hat{y} = \operatorname{argmax}_{y\in\mathcal{Y}} f_y(\boldsymbol{x})$. Since the probability density $p(\boldsymbol{x}, y)$ is not accessible, we need to collect identically and independently distributed training examples $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$ and minimize the empirical version of Eq. (1) instead:

$$\widehat{R}(f) = \sum\nolimits_{i=1}^{n}\big[\mathcal{L}(\boldsymbol{x}^{(i)}, y^{(i)}; f)\big], \tag{2}$$

which is referred to as *empirical risk minimization* principle (Vapnik, 1999).

2.2. Classification from Aggregate Observations

For the CFAO task, we aim to learn a classifier with only supervision (i.e., aggregate information) on groups of instances. Concretely, given a group of instances $\boldsymbol{x}_{1:m} = \{\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots, \boldsymbol{x}_m\}$ where m denotes the size of the group, their true labels $y_{1:m} = \{y_1, y_2 \ldots, y_m\}$ are unavailable, and we only know an aggregate label $z \in \mathcal{Z}$ where $\mathcal{Z}$ represents the space of aggregate labels. Each aggregate label can be obtained from $y_{1:m}$ via an aggregate function $g : \mathcal{Y}^m \to \mathcal{Z}$, i.e., $z = g(y_{1:m})$. Our goal is to use observations of $(\boldsymbol{x}_{1:m}, z)$ to train a classifier that can predict the true label as accurately as possible.

In Table 1, we show some typical examples of the CFAO task and their corresponding aggregate functions. All the problems listed in Table 1 can be solved by our proposed universal unbiased risk estimator. In what follows, we introduce two necessary assumptions for mathematically formulating the CFAO task, which were also adopted by the previous study (Zhang et al., 2020).

Assumption 2.1. $p(z|\boldsymbol{x}_{1:m}, y_{1:m}) = p(z|y_{1:m})$.

This assumption indicates that given the true labels $y_{1:m}$, the aggregate label z is independent of the instances $\boldsymbol{x}_{1:m}$. This assumption was commonly used in previous studies (Carbonneau et al., 2018; Cui et al., 2020; Zhang et al., 2020; Bao et al., 2022). It can be implied in the data collection process, e.g., when we first collect $(\boldsymbol{x}, y)$-pairs but only disclose some summary statistics of $y_{1:m}$ for learning due to privacy concerns. It also means that we expect the true label y to carry enough information about $\boldsymbol{x}$, so that we do not need to extract more information from $\boldsymbol{x}_{1:m}$ to obtain z.

Table 1. A brief introduction for typical examples of classification from aggregate observations. For a group of examples $(\boldsymbol{x}_{1:m}, y_{1:m})$, the aggregate label is generated by $z = g(y_{1:m})$.

Classification from $\cdots$	Size m	Aggregate Information	Aggregate Function g
pairwise similarity	$m = 2$	if y_1 and y_2 belong to same class or not	$g(y_1, y_2) = \mathbb{I}[y_1 = y_2]$
triplet comparison	$m = 3$	if $d(y_1, y_2)$ is smaller than $d(y_1, y_3)$	$g(y_{1:3}) = \mathbb{I}[d(y_1, y_2) < d(y_1, y3)]$
multiple instances	$m \geqslant 2$	if at least one positive label exits in $y_{1:m}$ ($k = 2$)	$g(y_{1:m}) = \max(y_{1:m})$
label proportion	$m \geqslant 2$	proportion of data from each class in the group	$g_j(y_{1:m}) = (\sum_{i=1}^m \mathbb{I}[y_i = j])/m$
ordinal rank	$m = 2$	if y_1 is larger than y_2, i.e., $y_1 \geqslant y_2$.	$g(y_1, y_2) = \mathbb{I}[y_1 > y_2]$

Assumption 2.2. $p(y_{1:m}|\boldsymbol{x}_{1:m}) = \prod_{i=1}^m p(y_i|\boldsymbol{x}_i)$.

This assumption indicates that the examples in the same group are independent. It is worth noting that all the examples in the training set are independently collected, and we extend such an independent property to the group level.

Combing Assumption 2.1 and Assumption 2.2, we can decompose the joint distribution $p(\boldsymbol{x}_{1:m}, y_{1:m}, z)$ as

$$p(\boldsymbol{x}_{1:m}, y_{1:m}, z) = p(z|y_{1:m}) \prod_{i=1}^m p(y_i|\boldsymbol{x}_i)p(\boldsymbol{x}_i), \quad (3)$$

which would be beneficial for us to derive a universal unbiased risk estimator.

3. The Proposed Method

In this section, we present our universal unbiased method for CFAO. Throughout this paper, we use a slightly different definition of "unbiased", i.e., we say that a method is unbiased if the derived risk of this method on weakly supervised data is equivalent to the ordinary classification risk on fully supervised data (as shown in Eq. (1)). Hence, our key idea is to solve the problem by risk rewriting (Sugiyama et al., 2022), i.e., rewriting the classification risk into an equivalent form that can be estimated from the given aggregate observations. Existing risk rewriting methods can only solve a single problem of CFAO. In this paper, we propose a universal method for CFAO, which is presented as follows.

3.1. Unbiased Risk Estimator

Theorem 3.1. *The classification risk $R(f)$ in Eq. (1) can be equivalently expressed as follows:*

$$R_{\mathrm{agg}}(f) = \mathbb{E}_{p(\boldsymbol{x}_{1:m}, z)}\big[\mathcal{L}_{\mathrm{agg}}(\boldsymbol{x}_{1:m}, z; f)\big]. \quad (4)$$

where $\mathcal{L}_{\mathrm{agg}}(\boldsymbol{x}_{1:m}, z; f)$ (with $z = g(y_{1:m})$ and $y_{1:m}$ is unknown to the learning algorithm) is defined as

$$\mathcal{L}_{\mathrm{agg}}(\boldsymbol{x}_{1:m}, z; f) = \frac{1}{m}\frac{1}{p(z|\boldsymbol{x}_{1:m})} \cdot \quad (5)$$
$$\sum_{i=1}^m \sum_{j=1}^k p(z, y_i = j|\boldsymbol{x}_{1:m}) \cdot \mathcal{L}(\boldsymbol{x}_i, j; f),$$

where $\mathcal{L}(\boldsymbol{x}_i, j; f)$ is an ordinary classification loss function as discussed in Eq. (1).

The proof of Theorem 3.1 is provided in Appendix A.

Theorem 3.1 indicates that we could recover the classification risk by using aggregate observations with a specially defined loss function $\mathcal{L}_{\mathrm{agg}}(\boldsymbol{x}_{1:m}, z; f)$ in Eq. (5). For the introduced loss function $\mathcal{L}_{\mathrm{agg}}$, we do not impose any restrictions on the ordinary classification loss $\mathcal{L}$, hence our method can be compatible with arbitrary losses. We can observe that $\mathcal{L}_{\mathrm{agg}}$ works as an importance-weighting loss, due to the instance-level probability $p(z, y_i = j|\boldsymbol{x}_{1:m})$ and the group-level probability $p(z|\boldsymbol{x}_{1:m})$. Specifically, when $p(z, y_i = j|\boldsymbol{x}_{1:m})$ is small, the probability of the label j being the true label of i-th instance $\boldsymbol{x}_i$ in the group $\boldsymbol{x}_{1:m}$ is small. This means that $(\boldsymbol{x}_i, j)$ is unlikely to be true (i.e., drawn from the fully supervised data distribution $p(\boldsymbol{x}, y)$), therefore we should assign a small weight to $\mathcal{L}(\boldsymbol{x}_i, j; f)$. The group-level probability $p(z|\boldsymbol{x}_{1:m})$ can be regarded as a normalization factor that normalizes the instance-level probability $p(z, y_i = j|\boldsymbol{x}_{1:m})$.

It is worth noting that in Theorem 3.1, we do not impose any restrictions on the group size m, which means that with aggregate observations of arbitrary group sizes, we could still recover the classification risk in Eq. (1). Therefore, for classification from a given set of aggregate observations $\{(\boldsymbol{x}_{1:m}^{(i)}, z^{(i)})\}_{i=1}^n$, we can minimize the following empirical risk (i.e., the empirical version of Eq. (4)):

$$\widehat{R}_{\mathrm{agg}}(f) = \frac{1}{n}\sum_{i=1}^n \mathcal{L}_{\mathrm{agg}}(\boldsymbol{x}_{1:m}^{(i)}, z^{(i)}; f). \quad (6)$$

Here, the key challenge becomes how to calculate $\mathcal{L}_{\mathrm{agg}}$ given an aggregate observation $(\boldsymbol{x}_{1:m}^{(i)}, z^{(i)})$ and a classifier f. According to Eq. (5), we can observe that the remaining issue is to empirically estimate $p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i = j|\boldsymbol{x}_{1:m})$. Once the two probabilities $p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i = j|\boldsymbol{x}_{1:m})$ are estimated by the classifier f, we can substitute the estimated values into $\mathcal{L}_{\mathrm{agg}}$ to obtain the loss value for training the classifier f. It is also noteworthy that different problems of CFAO have different rules on the aggregate function $g(y_{1:m}) = z$. Therefore, $p(z|\boldsymbol{x}_{1:m})$ and

$p(z, y_i = j|\boldsymbol{x}_{1:m})$ can be estimated in different ways in terms of different problems of CFAO.

3.2. Analysis from the EM Perspective

Here, we provide a theoretical justification on how the estimated $\frac{p(z,y_i=j|\boldsymbol{x}_{1:m})}{p(z|\boldsymbol{x}_{1:m})}$ helps train the classifier and differentiates the ground truth label. We show that our method actually maximizes the likelihood $\log(p(z^{(v)}, \boldsymbol{x}_{1:m}^{(v)}; \theta))$ during training when we employ the widely used cross-entropy loss. Let us denote by θ the parameters of the classifier, $p(\cdot; \theta)$ the probability function estimated by the classifier, $S(z) = \{y_{1:m} \in \mathcal{Y}^m \mid g(y_{1:m}) = z\}$ where g denotes the aggregation function, and $\omega_{y_{1:m}}^{(v)}$ the weight corresponding to $y_{1:m}$ for the v-th example in the dataset, where $0 \leqslant \omega_{y_{1:m}}^{(v)} \leqslant 1$ and $\sum_{y_{1:m} \in S(z^{(v)})} \omega_{y_{1:m}}^{(v)} = 1$. Then, we have the following theorem.

Theorem 3.2. *The following inequality holds:*

$$\log(p(z^{(v)} \mid \boldsymbol{x}_{1:m}^{(v)}; \theta)) \geqslant \sum\nolimits_{y_{1:m}^{(v)} \in S(z^{(v)})} \omega_{y_{1:m}}^{(v)} \log(p(y_{1:m}^{(v)}, \boldsymbol{x}_{1:m}^{(v)}; \theta)/\omega_{y_{1:m}}^{(v)}), \tag{7}$$

where the inequality holds with equality when $\omega_{y_{1:m}}^{(v)} = p(y_{1:m}^{(v)} \mid \boldsymbol{x}_{1:m}^{(v)}; \theta)/p(z^{(v)} \mid \boldsymbol{x}_{1:m}^{(v)}; \theta)$ *and Eq. (7) can be considered identical to Eq. (5) during the training.*

The proof of Theorem 3.2 is provided in Appendix B.

Thanks to Theorem 3.2, our method can be considered as an EM algorithm (Moon, 1996) that maximizes a log-likelihood objective. At the E-step, our method assigns $\omega_{y_{1:m}}^{(v)} = \frac{p(y_{1:m}^{(v)}|\boldsymbol{x}_{1:m}^{(v)};\theta)}{p(z^{(v)}|\boldsymbol{x}_{1:m}^{(v)};\theta)}$, to make the inequality holds with equality, i.e., to maximize $\sum_{v=1}^{n} \sum_{y_{1:m}^{(v)} \in S(z^{(v)})} \omega_{y_{1:m}}^{(v)} \log(\frac{p(y_{1:m}^{(v)}, \boldsymbol{x}_{1:m}^{(v)};\theta)}{\omega_{y_{1:m}}^{(v)}})$ with respect to ω when $p(y_{1:m}^{(v)}, \boldsymbol{x}_{1:m}^{(v)}; \theta)$ is fixed. At the M-step, our method aims to maximize Eq. (7) with respect to θ when ω is fixed (i.e., to train the classifier by maximizing the improved lower bound).

4. Realizations of Our Proposed Method

In this section, we describe the realizations of our method for various problems of CFAO, including classification via pairwise similarity (Hsu et al., 2018), classification via triplet comparison (Zhang et al., 2020), multiple-instance learning (Carbonneau et al., 2018), and learning from label proportions (Yu et al., 2013). We also provide the realization of our method for ordinal classification with only ranking or triplet comparison observations, in Appendix C.

It is worth noting that for classification via pairwise similarity or triplet comparison, the learned classifier is *not identifiable*, which means the identifiable mapping from the classifier outputs to semantic classes is lost. This problem is caused by the extremely limited supervision information provided by these two problems. In this case, the classifier consistency cannot be strictly guaranteed (Zhang et al., 2020) while the risk consistency of our proposed method still holds, which also confirms the superiority of our method. Pratically, an identifiable mapping could be obtained if we could obtain a permutation of classes in accordance with the learned classifier (Zhang et al., 2020). A common method to obtain an identifiable mapping is by using a validation dataset to solve the mapping problem. Hsu et al. (2018) proposed a method to obtain an optimal mapping by solving a linear sum assignment problem using the Hungarian algorithm (Kuhn, 1955). In the experiments, we follow (Hsu et al., 2018) to evaluate the performance of classification methods with the obtained optimal mapping.

In what follows, we demonstrate how to estimate $p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i = j|\boldsymbol{x}_{1:m})$ of our proposed $\mathcal{L}_{\text{agg}}$ in Eq. (5), for a certain problem of CFAO with the classifier f. Once the two probabilities $p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i = j|\boldsymbol{x}_{1:m})$ are estimated by the classifier f, we can substitute the estimated values into $\mathcal{L}_{\text{agg}}$ to obtain the loss value for training the classifier f.

For convenience, we represent $p(y_i = j|\boldsymbol{x}_i)$ by $\eta_j(\boldsymbol{x}_i)$, which denotes the probability of the true label y_i being the j-th class, for the instance $\boldsymbol{x}_i$. Given a classifier f, $\eta_j(\boldsymbol{x}_i)$ can be approximated by applying the softmax function to the classifier output $f(\boldsymbol{x}_i) \in \mathbb{R}^k$, i.e.,

$$\eta_j(\boldsymbol{x}_i) = \frac{\exp(f_j(\boldsymbol{x}_i))}{\sum_{v=1}^{k} \exp(f_v(\boldsymbol{x}_i))}. \tag{8}$$

In what follows, we will demonstrate how $p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i = j|\boldsymbol{x}_{1:m})$ of $\mathcal{L}_{\text{agg}}$ can be derived from $\eta_j(\boldsymbol{x}_i)$ for various problems of CFAO.

4.1. Classification via Pairwise Similarity

In the problem of classification via pairwise similarity, each group has two instances (i.e., $m = 2$), and the aggregate information is whether two instances in the group belong to the same class (similar) or not (dissimilar). In this case, the aggregation function g is defined as $g(y_{1:2}) = \mathbb{I}[y_1 = y_2]$ where $\mathbb{I}$ is the indicator function, which returns 1 if $y_1 = y_2$ and 0 otherwise. This problem was investigated as semi-supervised clustering in early studies (Bilenko et al., 2004; Basu et al., 2004). In recent years, this problem has been studied from the perspective of classification (Hsu et al., 2018; Zhang et al., 2020), based on the maximum likelihood principle (Nishii, 1989).

We also study this problem from the perspective of classification, and our proposed loss function $\mathcal{L}_{\text{agg}}$ can be applied to solve this problem, by using the following proposition.

Proposition 4.1. *For the problem of classification via pairwise similarity ($m=2$), $p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i=j|\boldsymbol{x}_{1:m})$ of $\mathcal{L}_{\text{agg}}$ in Eq. (5) can be empirically estimated by*

$$p(z=1|\boldsymbol{x}_{1:2}) = \sum\nolimits_{j=1}^{k} \eta_j(\boldsymbol{x}_1)\eta_j(\boldsymbol{x}_2),$$
$$p(z=0|\boldsymbol{x}_{1:2}) = 1 - p(z=1|\boldsymbol{x}_{1:2}),$$

and

$$\begin{aligned}
p(z=1, y_1=j|\boldsymbol{x}_{1:2}) &= \eta_j(\boldsymbol{x}_1)\eta_j(\boldsymbol{x}_2),\\
p(z=1, y_2=j|\boldsymbol{x}_{1:2}) &= \eta_j(\boldsymbol{x}_1)\eta_j(\boldsymbol{x}_2),\\
p(z=0, y_1=j|\boldsymbol{x}_{1:2}) &= (1-\eta_j(\boldsymbol{x}_2))\,\eta_j(\boldsymbol{x}_1),\\
p(z=0, y_2=j|\boldsymbol{x}_{1:2}) &= (1-\eta_j(\boldsymbol{x}_1))\,\eta_j(\boldsymbol{x}_2).
\end{aligned}$$

4.2. Classification via Triplet Comparison

In the problem of classification via triplet comparison, each group has three instances (i.e., $m=3$), and the aggregation information is whether one instance is more similar to the other one, compared with the third one. In this case, the aggregate function g is defined as $g(y_{1:3}) = \mathbb{I}\left[d(y_1,y_2) < d(y_1,y_3)\right]$ where $d: \mathcal{Y}\times\mathcal{Y}\mapsto\mathbb{R}$ is a distance measure between classes (a smaller distance means a larger similarity). In our studied classification setting, the distance measure is defined as $d(y,y') = \mathbb{I}[y\neq y']$.

Triplet comparison data has been widely studied in metric learning (Schultz & Joachims, 2003; Kumar & Kummamuru, 2008; Sohn, 2016; Mojsilovic & Ukkonen, 2019). Recently, Cui et al. (2020) showed that we can successfully learn a binary classifier from only triplet comparison data, and Zhang et al. (2020) studied multi-class classification from triplet comparison data. We also study multi-class classification from triplet comparison data, and our proposed loss function $\mathcal{L}_{\text{agg}}$ can be applied to solve this problem, by using the following proposition.

Proposition 4.2. *For the problem of classification via triplet comparison ($m=3$), $p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i=j|\boldsymbol{x}_{1:m})$ of $\mathcal{L}_{\text{agg}}$ in Eq. (5) can be empirically estimated by*

$$p(z=1|\boldsymbol{x}_{1:3}) = \sum\nolimits_{j=1}^{k} \eta_j(\boldsymbol{x}_1)\eta_j(\boldsymbol{x}_2)\big(1-\eta_j(\boldsymbol{x}_3)\big),$$
$$p(z=0|\boldsymbol{x}_{1:3}) = 1 - p(z=1|\boldsymbol{x}_{1:3}),$$

and

$$\begin{aligned}
p(z=1, y_1=j|\boldsymbol{x}_{1:3}) &= \eta_j(\boldsymbol{x}_2)\,(1-\eta_j(\boldsymbol{x}_3))\,\eta_j(\boldsymbol{x}_1),\\
p(z=1, y_2=j|\boldsymbol{x}_{1:3}) &= \eta_j(\boldsymbol{x}_1)\,(1-\eta_j(\boldsymbol{x}_3))\,\eta_j(\boldsymbol{x}_2),\\
p(z=1, y_3=j|\boldsymbol{x}_{1:3}) &= \sum\nolimits_{v\neq j} \eta_v(\boldsymbol{x}_1)\cdot\eta_v(\boldsymbol{x}_2)\eta_j(\boldsymbol{x}_3),\\
p(z=0, y_1=j|\boldsymbol{x}_{1:3}) &= (1-\eta_j(\boldsymbol{x}_2)\,(1-\eta_j(\boldsymbol{x}_3)))\,\eta_j(\boldsymbol{x}_1),\\
p(z=0, y_2=j|\boldsymbol{x}_{1:3}) &= (1-\eta_j(\boldsymbol{x}_1)\,(1-\eta_j(\boldsymbol{x}_3)))\,\eta_j(\boldsymbol{x}_2),\\
p(z=0, y_3=j|\boldsymbol{x}_{1:3}) &= \Big(1-\sum_{v\neq j}\eta_v(\boldsymbol{x}_1)\cdot\eta_v(\boldsymbol{x}_2)\Big)\eta_j(\boldsymbol{x}_3).
\end{aligned}$$

4.3. Learning from Label Proportions

Learning from label proportions (Yu et al., 2014; Quadrianto et al., 2008; Dulac-Arnold et al., 2019; Scott & Zhang, 2020) is also an attractive problem of CFAO. In learning from label proportions, each group has at least two instances (i.e., $m\geqslant 2$), and the aggregate information is the proportion of data from each class in the group. In this problem, the aggregate label space $\mathcal{Z}$ is a k-dimensional vector space (i.e., $\mathcal{Z}=\mathbb{R}^k$). The aggregate function is defined as $g_j(y_{1:m}) = \sum_{i=1}^{m}\mathbb{I}[y_i=j] = z_j$, which corresponds to the number of instances from each class in the group. It is noteworthy that this aggregate function is slightly different from the original aggregate function mentioned in Table 1, where the denominator m is removed for realization convenience, which would not affect the natural property of this problem.

We can also solve this problem using our proposed $\mathcal{L}_{\text{agg}}$ by the following proposition.

Proposition 4.3. *For the problem of learning from label proportions ($m\geqslant 2$), $p(\boldsymbol{z}|\boldsymbol{x}_{1:m})$ and $p(\boldsymbol{z}, y_i=j|\boldsymbol{x}_{1:m})$ of $\mathcal{L}_{\text{agg}}$ in Eq. (5) can be empirically estimated by*

$$p(\boldsymbol{z}|\boldsymbol{x}_{1:m}) = \sum\nolimits_{y_{1:m}\in\delta(\boldsymbol{z})}\prod\nolimits_{i=1}^{m}\eta_{y_i}(\boldsymbol{x}_i),$$

where $\delta(\boldsymbol{z}) = \{y_{1:m}|\boldsymbol{g}(y_{1:m}) = \boldsymbol{z}\}$ and

$$\begin{aligned}
&p(\boldsymbol{z}=\boldsymbol{g}(y_{1:m}), y_i=j|\boldsymbol{x}_{1:m})\\
&\qquad= \sum\nolimits_{y_{1:m\setminus i}\in\delta(\boldsymbol{z},i,j)}\prod\nolimits_{v\neq i}\eta_{y_v}(\boldsymbol{x}_v)\eta_j(\boldsymbol{x}_i),
\end{aligned}$$

where $\delta(\boldsymbol{z},i,j) = \{y_{1:m\setminus i}|\boldsymbol{g}(y_{1:m}) = \boldsymbol{z}, y_i=j\}$.

4.4. Multiple-Instance Learning

Multiple-instance learning (Maron & Lozano-Pérez, 1997; Zhang & Goldman, 2001; Carbonneau et al., 2018; Ilse et al., 2018) is a widely studied weakly supervised learning problem, which also belongs to the task of CFAO. In multiple-instance learning, each group has multiple instances (i.e., $m\geqslant 2$), and the aggregate information is whether the group has at least one positive instance. Since multiple-instance learning focuses on binary classification (i.e., $k=2$), we define the label space $\mathcal{Y}$ as $\{0,1\}$. In this case, the aggregate function is defined as $g(y_{1:m}) = \max(y_{1:m})$, which means that the group that has at least one positive instance is a positive group, otherwise, it is a negative group. Since the output of the binary classifier $f(\boldsymbol{x})\in\mathbb{R}$ is a scalar, we apply the Sigmoid function to approximate $\eta(\boldsymbol{x})$, i.e., $\eta_1(\boldsymbol{x}_i) = \frac{1}{1+\exp(-f(\boldsymbol{x}_i))}$ and $\eta_0(\boldsymbol{x}_i) = 1-\eta_1(\boldsymbol{x}_i)$.

Our proposed loss function $\mathcal{L}_{\text{agg}}$ can also be applied to solve this problem, by the following proposition.

Proposition 4.4. *For the problem of multiple-instance learning ($m\geqslant 2$ and $k=2$), $p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i=j|\boldsymbol{x}_{1:m})$*

Table 2. Statistics of the used benchmark-simulated datasets and the corresponding models. #Sampled groups represents the number of groups sampled for classification via pairwise similarity/triplet comparison/label proportion.

Dataset	#Train	#Val	#Test	#Classes	#Features	#Sampled groups	Model
MNIST	45,000	15,000	10,000	10	784	120,000/120,000/30,000	5-layer LeNet
Kuzushiji-MNIST	45,000	15,000	10,000	10	784	120,000/120,000/30,000	5-layer LeNet
Fashion-MNIST	45,000	15,000	10,000	10	784	120,000/120,000/30,000	5-layer LeNet
CIFAR-10	37,500	12,500	10,000	10	3,072	120,000/120,000/30,000	22-layer DenseNet
SVHN	54,942	18,315	26,032	10	3,072	120,000/120,000/30,000	22-layer DenseNet
msplice	1,905	635	635	3	240	6,350/6,350/1,587	3-layer MLP (d-300-k)
optdigits	3,372	1,124	1,124	10	62	11,240/11,240/2,810	3-layer MLP (d-300-k)
pendigits	6,594	2,199	2,199	10	16	21,984/21,984/5,496	3-layer MLP (d-300-k)
usps	5,578	1,860	1,860	10	256	18,596/18,596/4,649	3-layer MLP (d-300-k)
vehicle	507	169	170	4	18	1,692/1,692/423	3-layer MLP (d-300-k)

of $\mathcal{L}_{\text{agg}}$ in Eq. (5) can be empirically estimated by

$$p(z=0|\boldsymbol{x}_{1:m}) = \prod_{i=1}^{m} \eta_0(\boldsymbol{x}_i),$$
$$p(z=1|\boldsymbol{x}_{1:m}) = 1 - p(z=0|\boldsymbol{x}_{1:m}),$$

and

$$\begin{aligned} p(z=1, y_i=1|\boldsymbol{x}_{1:m}) &= \eta_1(\boldsymbol{x}_i), \\ p(z=1, y_i=0|\boldsymbol{x}_{1:m}) &= (1-\prod_{j\neq i}\eta_0(\boldsymbol{x}_j))\eta_0(\boldsymbol{x}_i), \\ p(z=0, y_i=0|\boldsymbol{x}_{1:m}) &= \prod_{j\neq i}\eta_0(\boldsymbol{x}_j)\eta_0(\boldsymbol{x}_i), \\ p(z=0, y_i=1|\boldsymbol{x}_{1:m}) &= 0. \end{aligned}$$

5. Experiments

In this section, we conduct extensive experiments to empirically demonstrate the effectiveness of our proposed method in various problems of CFAO. For classification via pairwise similarity/triplet comparison/label proportion, we use five popular large-scale benchmark datasets including MNIST (LeCun et al., 1998), Kuzushiji-MNIST (Clanuwat et al., 2018), Fashion-MNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011), and CIFAR-10 (Krizhevsky et al., 2009) and five regular-scale datasets from the UCI Machine Learning Repository (Dua & Graff, 2017) including usps, pendidigts, optdigits, msplice, and vehicle. For multiple-instance learning, we use five common benchmark datasets in this area (Dietterich et al., 1997; Andrews et al., 2002), inclduing Musk1, Musk2, Elephant, Fox, and Tiger. Since our proposed method can be compatible with arbitrary models and losses, we use various base models, including 5-layer LeNet (LeCun et al., 1998), 22-layer DenseNet (Huang et al., 2017), and 3-layer (d-300-k) Multilayer Perceptron on the above datasets. For the classification loss $\mathcal{L}$ in our proposed $\mathcal{L}_{\text{agg}}$, we simply adopt the widely used softmax cross entropy loss for multi-class classification and adopt the logistic loss for binary classification. Detailed descriptions of the used datasets and the corresponding models are provided in Table 2. The details of our algorithmic procedure, hyperparameter settings, and the characteristics of datasets for multiple-instance learning are provided in Appendix D.

For classification via pairwise similarity/triplet comparison/label proportion, the aggregate observations are randomly generated from the training set with replacement, according to Assumption 2.2. Since the learned classifier for classification via pairwise similarity/triplet comparison is not identifiable, we follow Hsu et al. (2018); Zhang et al. (2020) to evaluate the performance by modified accuracy that allows any permutation of classes, and the optimal permutation is obtained by solving a linear sum assignment problem using the Hungarian algorithm (Kuhn, 1955).

We run five trials on each dataset and record the mean accuracy and standard deviation (mean $\pm$ std). The best performance among all the methods is highlighted in boldface. We also conduct paired t-test at 5% significance level, and use $\bullet/\circ$ to denote whether our proposed universal unbiased method (UUM) is significantly better/worse than a compared method.

5.1. Classification via Pairwise Similarity

Experimental setup. For classification via pairwise similarity, the size of the generated training set is 120,000 for large-scale benchmark datasets and is twice the size of the original training set for regular-scale UCI datasets. We compare our proposed method with three methods of classification via pairwise similarity, including the universal method based on log-likelihood (Zhang et al., 2020) and two representation/metric learning methods including the Siamese network (Koch et al., 2015) and the contrastive loss (Hadsell et al., 2006). Since the output of the two representation/metric learning methods is a vector representation, we use the K-means clustering algorithm (Bock, 2007) on vector representations to obtain unidentifiable class predictions and evaluate the performance by modified accuracy with the optimal permutation of classes (Kuhn, 1955).

Table 3. Test accuracy (mean ± std) of each method for classification via pairwise similarity on large-scale benchmark datasets.

METHOD	MNIST	KUZUSHIJI	FASHION	CIFAR10	SVHN
LOG-LIKELIHOOD	98.87 ± 0.06%●	93.40% ± 0.64%	88.75 ± 0.31%●	71.45 ± 0.47%●	91.06 ± 0.24%●
SIAMESE	98.63 ± 0.15%●	89.25 ± 1.87%●	82.20 ± 1.45%●	58.16 ± 1.92%●	87.42 ± 4.06%●
CONTRASTIVE	98.86 ± 0.14%●	93.63 ± 0.53%●	89.17 ± 0.31%	23.69 ± 0.65%●	91.30 ± 0.33%●
UUM (OURS)	**98.99 ± 0.04%**	**94.04 ± 0.50%**	**89.19 ± 0.37%**	**72.52 ± 0.68%**	**92.41 ± 0.46%**

Table 4. Test accuracy (mean ± std) of each method for classification via pairwise similarity on regular-scale UCI datasets.

METHOD	MSPLICE	OPTDIGITS	PENDIGITS	USPS	VEHICLE
LOG-LIKELIHOOD	94.93 ± 0.43%	98.10 ± 0.49%	88.18 ± 14.14%	96.87 ± 0.17%●	**79.41 ± 4.23%**
SIAMESE	94.62 ± 0.42%	90.05 ± 3.50%●	77.13 ± 1.95%●	81.96 ± 6.16%●	45.76 ± 5.29%
CONTRASTIVE	93.36 ± 0.36%●	95.78 ± 0.83%●	90.50 ± 1.14%●	93.52 ± 0.46%●	60.35 ± 2.31%
UUM (OURS)	**94.99 ± 0.71%**	**98.31 ± 0.37%**	**96.95 ± 4.14%**	**97.02 ± 0.13%**	78.71 ± 4.20%

Table 5. Test accuracy (mean ± std) of each method for classification via triplet comparison on large-scale benchmark datasets.

METHOD	MNIST	KUZUSHIJI	FASHION	CIFAR10	SVHN
LOG-LIKELIHOOD	98.92 ± 0.11%	93.46 ± 0.10%●	88.76 ± 0.28%	70.57 ± 0.59%●	90.61 ± 0.60%●
TRIPLET	97.40 ± 0.10%●	85.65 ± 0.98%●	82.14 ± 2.68%●	47.09 ± 3.80%●	84.10 ± 2.15%●
(2+1) TUPLE	96.85 ± 0.33%●	80.26 ± 1.46%●	74.19 ± 1.12%●	45.53 ± 1.58%●	74.02 ± 1.28%●
UUM (OURS)	**99.06 ± 0.09%**	**93.88 ± 0.29%**	**89.18 ± 0.47%**	**72.41 ± 0.98%**	**92.44 ± 0.29%**

Table 6. Test accuracy (mean ± std) of each method for classification via triplet comparison on regular-scale UCI datasets.

DATASET	MSPLICE	OPTDIGITS	PENDIGITS	USPS	VEHICLE
LOG-LIKELIHOOD	95.09 ± 0.90%	**98.15 ± 0.21%**	64.30 ± 35.63%●	96.42 ± 0.34%	75.53 ± 3.28%
TRIPLET	92.69 ± 1.47%●	91.81 ± 1.31%●	65.66 ± 17.16%	87.04 ± 3.46%●	65.88 ± 5.37%●
(2+1)TUPLE	91.43 ± 1.32%●	81.64 ± 6.29%●	64.36 ± 0.12%	74.72 ± 0.96%●	63.88 ± 3.50%●
UUM (OURS)	**95.37 ± 1.10%**	98.13 ± 0.41%	**66.48 ± 36.24%**	**96.49 ± 0.59%**	**76.71 ± 2.06%**

Experimental results. Table 3 and Table 4 report the test accuracy (mean ± std) of each method for classification via pairwise similarity on large-scale benchmark datasets and UCI datasets, respectively. As can be observed from Table 3, UUM achieves the best performance among all the methods on all the large-scale benchmark datasets and significantly outperforms the compared methods in most cases. Table 4 shows that UUM achieves the best performance on 4 out of 5 UCI datasets. From the two tables, we can see that UUM significantly outperforms other compared methods when a complex model (i.e., DenseNet) is used on large-scale datasets. This phenomenon indicates that UUM could obtain a more precise estimation of the importance of each label for each instance in the group, with a powerful model on larger-scale datasets.

5.2. Classification via Triplet Comparison

Experimental setup. For classification via triplet comparison, the size of the generated training set is 120,000 for large-scale benchmark datasets and twice the size of the original training set for UCI datasets. We compare our proposed UUM with three methods of classification via triplet comparison, including the universal method based on log-likelihood (Zhang et al., 2020) and two representation/metric learning methods including the triplet loss (Schroff et al., 2015) and the (2+1) tuple loss (Sohn, 2016). We also use the K-means clustering algorithm (Bock, 2007) on vector representations obtained by the triplet loss and the (2+1) tuple loss to get unidentifiable class predictions and evaluate the performance by modified accuracy with the optimal permutation of classes (Kuhn, 1955).

Experimental results. Table 5 and Table 6 report the test accuracy (mean ± std) of each method for classification via triplet comparison on large-scale benchmark datasets and UCI datasets, respectively. The experiments for classification via triplet comparison show similar results compared with the experiments for classification via pairwise similarity. Hence the superiority of our proposed method is also demonstrated. The experimental results for classification via triplet comparison also demonstrate that the performance of

Table 7. Test accuracy (mean $\pm$ std) of each method for learning from label proportions on large-scale benchmark datasets.

DATASET	MNIST	KUZUSHIJI	FASHION	CIFAR10	SVHN
LOG-LIKELIHOOD	$\mathbf{98.99 \pm 0.06\%}$	$94.06\% \pm 0.27\%\bullet$	$89.72 \pm 0.23\%$	$70.56 \pm 0.39\%$	$90.61 \pm 0.43\%$
ROT	$98.95 \pm 0.08\%$	$94.32\% \pm 0.22\%$	$89.40 \pm 0.31\%\bullet$	$70.44 \pm 0.48\%\bullet$	$\mathbf{92.09 \pm 0.17\%}$
UUM (OURS)	$98.96 \pm 0.10\%$	$\mathbf{94.41 \pm 0.16\%}$	$\mathbf{89.87 \pm 0.20\%}$	$\mathbf{70.77 \pm 0.66\%}$	$91.76 \pm 0.18\%$

Table 8. Test accuracy (mean $\pm$ std) of each method for learning from label proportions on regular-scale UCI datasets.

DATASET	MSPLICE	OPTDIGITS	PENDIGITS	USPS	VEHICLE
LOG-LIKELIHOOD	$95.12 \pm 0.84\%\bullet$	$98.35 \pm 0.38\%$	$99.35 \pm 0.13\%$	$96.89 \pm 0.31\%\bullet$	$79.39 \pm 1.44\%$
ROT	$95.21 \pm 0.79\%\bullet$	$98.40 \pm 0.27\%$	$99.35 \pm 0.17\%$	$97.18 \pm 0.32\%$	$\mathbf{80.14 \pm 0.23\%}$
UUM (OURS)	$\mathbf{95.62 \pm 0.64\%}$	$\mathbf{98.43 \pm 0.38\%}$	$\mathbf{99.38 \pm 0.17\%}$	$\mathbf{97.24 \pm 0.41\%}$	$79.41 \pm 1.93\%$

Table 9. Test accuracy (mean $\pm$ std) of each method for multiple-instance learning on common benchmark datasets.

DATASET	ELEPHANT	FOX	TIGER	MUSK1	MUSK2
LOG-LIKELIHOOD	$98.50 \pm 1.27\%$	$94.30 \pm 1.96\%$	$98.70 \pm 0.67\%$	$99.35 \pm 1.46\%\bullet$	$80.20 \pm 12.70\%\bullet$
MINIMAX-FEATURE	$90.90 \pm 2.46\%\bullet$	$81.00 \pm 2.65\%\bullet$	$91.70 \pm 1.72\%\bullet$	$98.59 \pm 1.46\%\bullet$	$98.43 \pm 1.29\%\bullet$
UUM (OURS)	$\mathbf{99.00 \pm 1.06\%}$	$\mathbf{94.90 \pm 1.98\%}$	$\mathbf{99.00 \pm 0.61\%}$	$\mathbf{100.00 \pm 0.00\%}$	$\mathbf{99.61 \pm 0.37\%}$

UUM would be more remarkable when large-scale datasets and more complex models are used.

5.3. Learning from Label Proportions

Experimental setup. For learning from label proportions, the size of the generated training set is 30,000 for benchmark datasets and half of the size of the original training set for UCI datasets. The group size m is set to 6. We compare our proposed method with two methods of learning from label proportions, including the log-likelihood method (Zhang et al., 2020) and ROT (Dulac-Arnold et al., 2019).

Experimental results. Table 7 and Table 8 report the test accuracy of each method for learning from label proportions on large-scale datasets and UCI datasets, respectively. We can find that UUM achieves the best performance for learning from label proportions in most cases. Hence the effectiveness of our proposed method is also demonstrated in the problem of learning from label proportions.

5.4. Multiple-Instance Learning

Experimental setup. For multiple-instance learning, we collect 5 widely used benchmark datasets, including Elephant, Fox, Tiger, Musk1, and Musk2. We randomly split the given datasets into training, validation, and test sets by 60%, 20% and 20% for each trial. Since these datasets only contain aggregate labels, we evaluate the performance by group-level accuracy. We compare our UUM with two methods, including the log-likelihood method (Zhang et al., 2020) and the minimax-feature method (Gärtner et al., 2002). We use the linear model as the base model to realize our method and the compared methods, for fair comparison.

Experimental results. Table 9 reports the test accuracy of each method for multiple-instance learning on common benchmark datasets. It can be seen that our proposed method achieves the best performance on all datasets and significantly outperforms the minimax-feature method in all cases. Therefore, the effectiveness of our method in multiple-instance learning is also validated.

6. Conclusion

In this paper, we investigated an interesting learning task called classification from aggregate observations, where we aim to learn a classifier with supervision on groups of instances, instead of supervision on individual instances. This task is quite general and contains a variety of learning problems such as multiple-instance learning and learning from label proportions. To handle this task, we proposed a novel universal method that holds an unbiased estimator of the classification risk for arbitrary losses. Our method has both practical and theoretical advantages. Practically, our method works by importance weighting for each instance and each label in the group, which provides purified supervision for the classifier to learn. Theoretically, our provided unbiased risk estimator not only guarantees the risk consistency of our method but also can be compatible with arbitrary losses. Comprehensive experimental results validated the effectiveness of our proposed method in various problems of classification from aggregate observations. In future work, we plan to extend our proposed universal unbiased method to the regression setting with aggregate observations.

Acknowledgements

This research is supported, in part, by the Joint NTU-WeBank Research Centre on Fintech (Award No: NWJ-2021-005), Nanyang Technological University, Singapore. Lei Feng is also supported by the National Natural Science Foundation of China (Grant No. 62106028), Chongqing Overseas Chinese Entrepreneurship and Innovation Support Program, CAAI-Huawei MindSpore Open Fund, and Chongqing Artificial Intelligence Innovation Center. Bo Han is supported by NSFC Young Scientists Fund No. 62006202, and Guangdong Basic and Applied Basic Research Foundation No. 2022A1515011652.

References

Andrews, S., Tsochantaridis, I., and Hofmann, T. Support vector machines for multiple-instance learning. In *NeurIPS*, pp. 577–584, 2002.

Bao, H., Niu, G., and Sugiyama, M. Classification from pairwise similarity and unlabeled data. In *ICML*, pp. 452–461, 2018a.

Bao, H., Sakai, T., Sato, I., and Sugiyama, M. Convex formulation of multiple instance learning from positive and unlabeled bags. *Neural Networks*, 105:132–141, 2018b.

Bao, H., Shimada, T., Xu, L., Sato, I., and Sugiyama, M. Pairwise supervision can provably elicit a decision boundary. In *AISTATS*, 2022.

Basu, S., Bilenko, M., and Mooney, R. J. A probabilistic framework for semi-supervised clustering. In *KDD*, pp. 59–68, 2004.

Bilenko, M., Basu, S., and Mooney, R. J. Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, pp. 11, 2004.

Bock, H.-H. Clustering methods: a history of k-means algorithms. *Selected Contributions in Data Analysis and Classification*, pp. 161–172, 2007.

Cao, Y., Feng, L., Xu, Y., An, B., Niu, G., and Sugiyama, M. Learning from similarity-confidence data. In *ICML*, pp. 1272–1282, 2021.

Carbonneau, M.-A., Cheplygina, V., Granger, E., and Gagnon, G. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

Chu, W. and Ghahramani, Z. Preference learning with gaussian processes. In *ICML*, pp. 137–144, 2005.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Cui, Z., Charoenphakdee, N., Sato, I., and Sugiyama, M. Classification from triplet comparison data. *Neural Computation*, 32(3):659–681, 2020.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Dulac-Arnold, G., Zeghidour, N., Cuturi, M., Beyer, L., and Vert, J.-P. Deep multi-class learning from label proportions. *arXiv preprint arXiv:1905.12909*, 2019.

Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., and Sugiyama, M. Provably consistent partial-label learning. In *NeurIPS*, volume 33, pp. 10948–10960, 2020.

Feng, L., Shu, S., Lu, N., Han, B., Xu, M., Niu, G., An, B., and Sugiyama, M. Pointwise binary classification with pairwise confidence comparisons. In *ICML*, pp. 3252–3262, 2021.

Flaxman, S. R., Wang, Y.-X., and Smola, A. J. Who supported obama in 2012? ecological inference through distribution regression. In *KDD*, pp. 289–298, 2015.

Fürnkranz, J. and Hüllermeier, E. Pairwise preference learning and ranking. In *ECMLP-KDD*, pp. 145–156, 2003.

Gärtner, T., Flach, P. A., Kowalczyk, A., and Smola, A. J. Multi-instance kernels. In *ICML*, pp. 7, 2002.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pp. 1735–1742, 2006.

Hsu, Y.-C., Lv, Z., Schlosser, J., Odom, P., and Kira, Z. Multi-class classification without multi-class labels. In *ICLR*, 2018.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *CVPR*, pp. 4700–4708, 2017.

Ilse, M., Tomczak, J., and Welling, M. Attention-based deep multiple instance learning. In *ICML*, pp. 2127–2136. PMLR, 2018.

Ishida, T., Niu, G., and Sugiyama, M. Binary classification from positive-confidence data. In *NeurIPS*, 2018.

Jordan, M. I. and Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Koch, G., Zemel, R., Salakhutdinov, R., et al. Siamese neural networks for one-shot image recognition. In *ICML*, 2015.

Kotsiantis, S. B., Zaharakis, I. D., and Pintelas, P. E. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2): 83–97, 1955.

Kumar, N. and Kummamuru, K. Semisupervised clustering with metric learning using relative comparisons. *IEEE Transactions on Knowledge and Data Engineering*, 20 (4):496–503, 2008.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Maron, O. and Lozano-Pérez, T. A framework for multiple-instance learning. In *NeurIPS*, volume 10, 1997.

Mojsilovic, S. and Ukkonen, A. Relative distance comparisons with confidence judgements. In *ICDM*, pp. 459–467. SIAM, 2019.

Moon, T. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996. doi: 10.1109/79.543975.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nishii, R. Maximum likelihood principle and model selection when the true model is unspecified. In *Multivariate Statistics and Probability*, pp. 392–403. Elsevier, 1989.

Quadrianto, N., Smola, A. J., Caetano, T. S., and Le, Q. V. Estimating labels from label proportions. In *ICML*, pp. 776–783, 2008.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, June 2015.

Schuessler, A. A. Ecological inference. *Proceedings of the National Academy of Sciences*, 96(19):10578–10581, 1999.

Schultz, M. and Joachims, T. Learning a distance metric from relative comparisons. In *NeurIPS*, volume 16, 2003.

Scott, C. and Zhang, J. Learning from label proportions: A mutual contamination framework. In *NeurIPS*, pp. 22256–22267, 2020.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, volume 29, 2016.

Sugiyama, M., Bao, H., Ishida, T., Lu, N., and Sakai, T. *Machine learning from weak supervision: An empirical risk minimization approach.* MIT Press, 2022.

Vapnik, V. N. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yu, F., Liu, D., Kumar, S., Tony, J., and Chang, S.-F. \proptosvm for learning with label proportions. In *ICML*, pp. 504–512. PMLR, 2013.

Yu, F. X., Choromanski, K., Kumar, S., Jebara, T., and Chang, S.-F. On learning from label proportions. *arXiv preprint arXiv:1402.5902*, 2014.

Zhang, Q. and Goldman, S. Em-dd: An improved multiple-instance learning technique. In *NeurIPS*, volume 14, 2001.

Zhang, Y., Charoenphakdee, N., Wu, Z., and Sugiyama, M. Learning from aggregate observations. In *NeurIPS*, volume 33, pp. 7993–8005, 2020.

Zhou, Z.-H. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

A. Proof of Theorem 3.1

The classification risk $R(f) = \mathbb{E}_{p(\boldsymbol{x},y)}[\mathcal{L}(\boldsymbol{x},y;f)]$ can be expressed as

$$R(f) = \int_{\mathcal{X}} \sum\nolimits_{y=1}^{k} p(\boldsymbol{x},y)\mathcal{L}(\boldsymbol{x},y;f)\mathrm{d}\boldsymbol{x}.$$

When we take m examples into one group, we can represent $R(f)$ as follows:

$$\begin{aligned}
R(f) &= \int_{\mathcal{X}} \sum_{y_{1:m}} p(\boldsymbol{x}_{1:m}, y_{1:m}) \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\boldsymbol{x}_i, y_i; f)\mathrm{d}\boldsymbol{x}_{1:m} \\
&= \int_{\mathcal{X}} p(\boldsymbol{x}_{1:m}) \sum_{y_{1:m}} p(y_{1:m}|\boldsymbol{x}_{1:m}) \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\boldsymbol{x}_i, y_i; f)\mathrm{d}\boldsymbol{x}_{1:m} \\
&= \mathbb{E}_{p(\boldsymbol{x}_{1:m})}\Big[\sum_{y_{1:m}} p(y_{1:m}|\boldsymbol{x}_{1:m}) \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\boldsymbol{x}_i, y_i; f)\Big] \\
&= \mathbb{E}_{p(\boldsymbol{x}_{1:m})}\Big[\sum\nolimits_{y_{1:m}} p(z|\boldsymbol{x}_{1:m}) \frac{p(y_{1:m}|\boldsymbol{x}_{1:m})}{p(z|\boldsymbol{x}_{1:m})} \cdot \frac{1}{m}\sum\nolimits_{i=1}^{m} \mathcal{L}(\boldsymbol{x}_i, y_i; f)\Big] \\
&= \mathbb{E}_{p(\boldsymbol{x}_{1:m})}\Big[\sum_{z\in\mathcal{Z}} p(z|\boldsymbol{x}_{1:m}) \sum_{y_{1:m}\in S(z)} \frac{p(y_{1:m}|\boldsymbol{x}_{1:m})}{p(z|\boldsymbol{x}_{1:m})} \cdot \frac{1}{m}\sum\nolimits_{i=1}^{m} \mathcal{L}(\boldsymbol{x}_i, y_i; f)\Big],
\end{aligned}$$

where $S(z) = \{y_{1:m} \in \mathcal{Y}^m | g(y_{1:m}) = z\}$ contains all the possibilities of $y_{1:m}$ that satisfy the condition $g(y_{1:m}) = z$. Then, we have

$$\begin{aligned}
R(f) &= \mathbb{E}_{p(\boldsymbol{x}_{1:m})}\Big[\sum_{z\in\mathcal{Z}} p(z|\boldsymbol{x}_{1:m}) \sum_{y_{1:m}\in S(z)} \frac{p(y_{1:m}|\boldsymbol{x}_{1:m})}{p(z|\boldsymbol{x}_{1:m})} \cdot \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\boldsymbol{x}_i, y_i; f)\Big] \\
&= \mathbb{E}_{p(\boldsymbol{x}_{1:m},z)}\Big[\sum_{y_{1:m}\in S(z)} \frac{p(y_{1:m}|\boldsymbol{x}_{1:m})}{p(z|\boldsymbol{x}_{1:m})} \cdot \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\boldsymbol{x}_i, y_i; f)\Big] && (9) \\
&= \mathbb{E}_{p(\boldsymbol{x}_{1:m},z)}\Big[\frac{1}{m}\frac{1}{p(z|\boldsymbol{x}_{1:m})}\sum_{i=1}^{m}\sum_{j=1}^{k} \sum_{y_{1:m\setminus i}\in S(z,j)} p(y_{1:m\setminus i}|\boldsymbol{x}_{1:m\setminus i}) \cdot p(y_i = j|\boldsymbol{x}_i) \cdot \mathcal{L}(f(\boldsymbol{x}_i), j)\Big], && (10)
\end{aligned}$$

where we have switched the two summations in the last equality above, and $y_{1:m\setminus i} = \{y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_m\}$ and $S(z,j) = \{y_{1:m\setminus i} \in \mathcal{Y}^{m-1} \mid g(y_1 \cdots y_{i-1}, j, y_{i+1} \cdots, y_m) = z\}$ contains all the possibilities of $y_{1:m\setminus i}$ on the condition of $y_i = j$. It is worth noting that

$$\begin{aligned}
&\sum_{y_{1:m\setminus i}\in S(z,j)} p(y_{1:m\setminus i}|\boldsymbol{x}_{1:m\setminus i}) \\
&= \frac{\sum_{y_{1:m\setminus i}\in S(z,j)} p(y_{1:m\setminus i}|\boldsymbol{x}_{1:m\setminus i}) p(y_i = j|\boldsymbol{x}_i)}{p(y_i = j|\boldsymbol{x}_i)} \\
&= \frac{p(g(y_{1:m}) = z, y_i = j|\boldsymbol{x}_{1:m})}{p(y_i = j|\boldsymbol{x}_i)} && (11) \\
&= \frac{p(z, y_i = j|\boldsymbol{x}_{1:m})}{p(y_i = j|\boldsymbol{x}_i)} && (12)
\end{aligned}$$

By substituting Eq. (11) into Eq. (10), we obtain

$$\begin{aligned}
R(f) &= \mathbb{E}_{p(\boldsymbol{x}_{1:m},z)}\Big[\frac{1}{m}\frac{1}{p(z|\boldsymbol{x}_{1:m})}\sum_{i=1}^{m}\sum_{j=1}^{k} \sum_{y_{1:m\setminus i}\in S(z,j)} p(y_{1:m\setminus i}|\boldsymbol{x}_{1:m\setminus i}) \cdot p(y_i = j|\boldsymbol{x}_i) \cdot \mathcal{L}(f(\boldsymbol{x}_i), j)\Big] \\
&= \mathbb{E}_{p(\boldsymbol{x}_{1:m},z)}\Big[\frac{1}{m}\frac{1}{p(z|\boldsymbol{x}_{1:m})} \cdot \sum_{i=1}^{m}\sum_{j=1}^{k} p(z, y_i = j|\boldsymbol{x}_{1:m}) \cdot \mathcal{L}(f(\boldsymbol{x}_i), j)\Big] \\
&= R_{\text{agg}}(f),
\end{aligned}$$

where completes the proof of Theorem 3.1.

B. Proof of Theorem 3.2

The log likelihood $\sum_{v=1}^{n}\log(p(z^{(v)},x^{(v)}_{1:m};\theta))$ could be transformed into:

$$\begin{aligned}
&\sum_{v=1}^{n}\log(p(z^{(v)},x^{(v)}_{1:m};\theta)) && (13)\\
&=\sum_{v=1}^{n}\log(\sum_{y^{(v)}_{1:m}\in S(z^{(v)})} p(z^{(v)},y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)) && (14)\\
&=\sum_{v=1}^{n}\log(\sum_{y^{(v)}_{1:m}\in S(z^{(v)})} p(z^{(v)}\mid y^{(v)}_{1:m},x^{(v)}_{1:m})p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)) && (15)\\
&=\sum_{v=1}^{n}\log(\sum_{y^{(v)}_{1:m}\in S(z^{(v)})} p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)) && (16)\\
&=\sum_{v=1}^{n}\log(\sum_{y^{(v)}_{1:m}\in S(z^{(v)})} \omega^{(v)}_{y_{1:m}}\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}}) && (17)\\
&\geqslant\sum_{v=1}^{n}\sum_{y^{(v)}_{1:m}\in S(z^{(v)})} \omega^{(v)}_{y_{1:m}}\log(\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}}), && (18)
\end{aligned}$$

where the second equality is due to $p(z|y_{1:m})=p(z|y_{1:m},x_{1:m})=1$ when $g(y_{1:m})=z$. The last inequality relies on Jensen's inequality and the properties of the weight $\omega^{(v)}_{y_{1:m}}$ (i.e., $0\leqslant\omega^{(v)}_{y_{1:m}}\leqslant 1$ and $\sum_{y_{1:m}\in S(z^{(v)})}\omega^{(v)}_{y_{1:m}}=1$). The inequality holds with equality when $\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}}$ is a constant. i.e., $\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}}=C$ where C is a constant.

In this case, we have

$$\begin{aligned}
\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{C}&=\omega^{(v)}_{y_{1:m}} && (19)\\
\sum_{y^{(v)}_{1:m}\in S(z^{(v)})}\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{C}&=\sum_{y^{(v)}_{1:m}\in S(z^{(v)})}\omega^{(v)}_{y_{1:m}} && (20)\\
\sum_{y^{(v)}_{1:m}\in S(z^{(v)})}\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{C}&=1 && (21)\\
\sum_{y^{(v)}_{1:m}\in S(z^{(v)})}p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)&=C && (22)\\
\sum_{y^{(v)}_{1:m}\in S(z^{(v)})}p(y^{(v)}_{1:m},x^{(v)}_{1:m},z^{(v)};\theta)&=C && (23)\\
p(x^{(v)}_{1:m},z^{(v)};\theta)&=C, && (24)
\end{aligned}$$

where the last derivation is due to the fact that we exhausted all the possible $y^{(v)}_{1:m}$ in $S(z^{(v)})$. In this way, we have

$$\omega^{(v)}_{y_{1:m}}=\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{p(x^{(v)}_{1:m},z^{(v)};\theta)}=\frac{p(y^{(v)}_{1:m}\mid x^{(v)}_{1:m};\theta)}{p(z^{(v)}\mid x^{(v)}_{1:m};\theta)}. \tag{25}$$

Therefore, the E-step of our method is to set $\omega^{(v)}_{y_{1:m}} = \frac{p(y^{(v)}_{1:m}|x^{(v)}_{1:m};\theta)}{p(z^{(v)}|x^{(v)}_{1:m};\theta)}$, to make the inequality holds with equality, i.e., to maximize $\sum_{v=1}^{n}\sum_{y^{(v)}_{1:m}\in S(z^{(v)})}\omega^{(v)}_{y_{1:m}}\log(\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}})$ with respect to ω when $p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)$ is fixed.

On the other hand, the M-step of our method is to maximize $\sum_{v=1}^{n}\sum_{y^{(v)}_{1:m}\in S(z^{(v)})}\omega^{(v)}_{y_{1:m}}\log(\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}})$ (i.e., Eq. (18)) with respect to θ when ω is fixed.

For the M-step, ω is fixed, thus we have

$$\log(\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}}) \tag{26}$$

$$= \log(\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{p(x_{1:m};\theta)}) + (\log(p(x_{1:m};\theta)) - \log(\omega^{(v)}_{y_{1:m}})) \tag{27}$$

$$= \log(p(y_{1:m}|x_{1:m});\theta) + \log(\frac{p(x_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}}) \tag{28}$$

$$= \sum_{i=1}^{m}\log(p(y_i|x_i;\theta)) + \log(\frac{p(x_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}}) \tag{29}$$

$$= \sum_{i=1}^{m}\log(p(y_i|x_i;\theta)) + \log(\frac{p(x_{1:m})}{\omega^{(v)}_{y_{1:m}}}), \tag{30}$$

where the last term $\log(\frac{p(x_{1:m})}{\omega^{(v)}_{y_{1:m}}})$ is a constant when ω is fixed. When the cross-entropy loss is applied, $\mathcal{L}(x_i,y_i;f) = -\log(p(y_i|x_i;\theta))$. Therefore, maximizing $\log(p(y_{1:m}|x_{1:m}))$ is equivalent to minimizing $\mathcal{L}(x_i,y_i;f)$.

It is noteworthy that the objective function we analyze above (i.e., $\sum_{y^{(v)}_{1:m}\in S(z^{(v)})}\omega^{(v)}_{y_{1:m}}\log(\frac{p(y^{(v)}_{1:m},x^{(v)}_{1:m};\theta)}{\omega^{(v)}_{y_{1:m}}})$) can be considered identical to the Eq. (9), i.e., $\sum_{y_{1:m}\in S(z^{(v)})}\frac{p(y_{1:m}|x_{1:m})}{p(z|x_{1:m})}\cdot\frac{1}{m}\mathcal{L}(x_i,y_i;f)$ except for the differences of the two constant terms $\frac{1}{m}$ and $\log(\frac{p(x_{1:m})}{\omega^{(v)}_{y_{1:m}}})$ when training the classifier, which is also identical to the final objective function used in our paper.

In summary, the E-step of our method improves the lower bound (i.e., Eq. (18)) of the likelihood $\log(p(z^{(v)},x^{(v)}_{1:m};\theta))$ and the M-step of our method trains the classifier by maximizing the improved lower bound, which indicates that our method actually maximizes the likelihood $\log(p(z^{(v)},x^{(v)}_{1:m};\theta))$.

C. Additional CFAO Problems

C.1. Rank observation

In the next two aggregate observation learning tasks, we focus on ordinal regression. Ordinal regression has a similar label space $\mathcal{Y} = \{1, 2\cdots k\}$ compared with muti-class classification. But there exists an order between different labels in label space, i.e., $1 < 2 < 3\cdots < k$. For notation convenience, we use $\eta_j(\boldsymbol{x}_i)$ to denote the probability $p(y_i \leqslant j|\boldsymbol{x}_i)$ of the true label y_i less or equal to j. Specifically, $\eta_0(\boldsymbol{x}_i) = 0$ and $\eta_k(\boldsymbol{x}_i) = 1$, and $p(y_i = j|\boldsymbol{x}_i)$ can be calculated by $\eta_j(\boldsymbol{x}_i) - \eta_{j-1}(\boldsymbol{x}_i)$. In the rank observation task, there are two instances in one bag (i.e., $m = 2$), and the aggregate function is defined as $g(y_{1:2}) = \mathbb{I}[y_1 < y_2]$.

This task is also a CFAO task in an ordinal classification setting, which can be solved by the following proposition.

Proposition C.1. *For the problem of classification via ordinal ranks ($m = 2$), $p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i = j|\boldsymbol{x}_{1:m})$ of $\mathcal{L}_{\text{agg}}$ in Eq. (5) can be empirically estimated by*

$$p(z = 1|\boldsymbol{x}_{1:2}) = \sum_{j=1}^{k}\eta_{j-1}(\boldsymbol{x}_1)(\eta_j(\boldsymbol{x}_2) - \eta_{j-1}(\boldsymbol{x}_2)),$$

$$p(z = 0|\boldsymbol{x}_{1:2}) = 1 - p(z = 1|\boldsymbol{x}_{1:2}),$$

and

$$
\begin{aligned}
p(z=1, y_1=j|\boldsymbol{x}_{1:2}) &= (1-\eta_j(\boldsymbol{x}_2))\,(\eta_j(\boldsymbol{x}_1)-\eta_{j-1}(\boldsymbol{x}_1)),\\
p(z=1, y_2=j|\boldsymbol{x}_{1:2}) &= \eta_{j-1}(\boldsymbol{x}_1)(\eta_j(\boldsymbol{x}_2)-\eta_{j-1}(\boldsymbol{x}_2)),\\
p(z=0, y_1=j|\boldsymbol{x}_{1:2}) &= \eta_j(\boldsymbol{x}_2)(\eta_j(\boldsymbol{x}_1)-\eta_{j-1}(\boldsymbol{x}_1)),\\
p(z=0, y_2=j|\boldsymbol{x}_{1:2}) &= (1-\eta_{j-1}(\boldsymbol{x}_1))\,(\eta_j(\boldsymbol{x}_2)-\eta_{j-1}(\boldsymbol{x}_2)).
\end{aligned}
$$

C.2. Ordinal triplet observation

Ordinal triplet observation task is similar to triplet observation task, each bag has three instances (i.e. $m=3$) and the aggregate function is defined as $g(y_{1:3})=\mathbb{I}\left[d(y_1,y_2)<d(y_1,y_3)\right]$. In ordinal regression, we define $d(y_1,y_2)=|y_1-y_2|$.

Proposition C.2. *For the problem of classification via ordinal triplet comparison ($m=3$), $p(z|\boldsymbol{x}_{1:m})$ and $p(z,y_i=j|\boldsymbol{x}_{1:m})$ of $\mathcal{L}_{\text{agg}}$ in Eq. (5) can be empirically estimated by*

$$
\begin{aligned}
p(z=1|\boldsymbol{x}_{1:3}) &= \sum_{j=1}^{k}\sum_{v=1}^{k}(\eta_v(\boldsymbol{x}_3)-\eta_{v-1}(\boldsymbol{x}_3))p(|y_2-j|<|v-j||\boldsymbol{x}_2)(\eta_j(\boldsymbol{x}_1)-\eta_{j-1}(\boldsymbol{x}_1))\\
p(z=v|\boldsymbol{x}_{1:3}) &= 1-p(z=1|\boldsymbol{x}_{1:3})
\end{aligned}
$$

and

$$
\begin{aligned}
p(z=1,y_1=j|\boldsymbol{x}_{1:3}) &= \sum_{v=1}^{k}(\eta_v(\boldsymbol{x}_3)-\eta_{v-1}(\boldsymbol{x}_3))p(|y_2-j|<|v-j||\boldsymbol{x}_2)(\eta_j(\boldsymbol{x}_1)-\eta_{j-1}(\boldsymbol{x}_1))\\
p(z=1,y_2=j|\boldsymbol{x}_{1:3}) &= \sum_{v=1}^{k}(\eta_v(\boldsymbol{x}_1)-\eta_{v-1}(\boldsymbol{x}_1))p(|y_3-v|>|j-v||\boldsymbol{x}_3)(\eta_j(\boldsymbol{x}_2)-\eta_{j-1}(\boldsymbol{x}_2))\\
p(z=1,y_3=j|\boldsymbol{x}_{1:3}) &= \sum_{v=1}^{k}(\eta_v(\boldsymbol{x}_1)-\eta_{v-1}(\boldsymbol{x}_1))p(|y_2-v|<|j-v||\boldsymbol{x}_2)(\eta_j(\boldsymbol{x}_3)-\eta_{j-1}(\boldsymbol{x}_3))\\
p(z=0,y_1=j|\boldsymbol{x}_{1:3}) &= \sum_{v=1}^{k}(\eta_v(\boldsymbol{x}_3)-\eta_{v-1}(\boldsymbol{x}_3))(1-p(|y_2-j|<|v-j||\boldsymbol{x}_2))(\eta_j(\boldsymbol{x}_1)-\eta_{j-1}(\boldsymbol{x}_1))\\
p(z=0,y_2=j|\boldsymbol{x}_{1:3}) &= \sum_{v=1}^{k}(\eta_v(\boldsymbol{x}_1)-\eta_{v-1}(\boldsymbol{x}_1))(1-p(|y_3-v|>|j-v||\boldsymbol{x}_3))(\eta_j(\boldsymbol{x}_2)-\eta_{j-1}(\boldsymbol{x}_2))\\
p(z=0,y_3=j|\boldsymbol{x}_{1:3}) &= \sum_{v=1}^{k}(\eta_v(\boldsymbol{x}_1)-\eta_{v-1}(\boldsymbol{x}_1))(1-p(|y_2-v|<|j-v||\boldsymbol{x}_2))(\eta_j(\boldsymbol{x}_3)-\eta_{j-1}(\boldsymbol{x}_3))
\end{aligned}
$$

For the calculation of $p(|y_2-v|<|j-v||\boldsymbol{x}_2)$ and $p(|y_3-v|>|j-v||\boldsymbol{x}_3)$ in above equation:

$$
\begin{aligned}
p(|y_2-v|<|j-v||\boldsymbol{x}_2) &= \max(\eta_{|j-v|+v-1}(\boldsymbol{x}_2)-\eta_{v-|j-v|}(\boldsymbol{x}_2),0)\\
p(|y_3-v|>|j-v||\boldsymbol{x}_3) &= 1-\left(\eta_{v+|j-v|}(\boldsymbol{x}_3)-\eta_{v-|j-v|-1|}(\boldsymbol{x}_3)\right)
\end{aligned}
$$

The function of $\max$ *in the first equation is to make the equality holds when $|j-v|=0$.*

D. Experiments Details

D.1. Training Algorithm

The pseudo-code of our proposed algorithm is presented in Algorithm 1. The training algorithm work with a similar process with RC (Feng et al., 2020) which treats $\eta(\boldsymbol{x})$ as weights and updates $\eta(\boldsymbol{x})$ during the training process. The main training

Algorithm 1 RC Algorithm

```
Input: Model $f$, epoch $T_{\text{max}}$, iteration $I_{\text{max}}$, size of label space $k$, whether to use log-likelihood to initialize flag$_{\text{init}}$, log-likelihood initialize epoch $T_{\text{init}}$, whether to use confidence matrix flag$_{\text{mat}}$, aggregate observation training set $\mathcal{D} = \{(\boldsymbol{x}_{1:m}^{(i)}, z^{(i)})\}_{i=1}^{n}$;
 1: if flag$_{\text{mat}}$=TRUE then
 2:   Initialize confidence tensor $C \in \mathbb{R}^{n\times m\times k}$,$C_{ijv} = \frac{1}{k}, \forall 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m, 1 \leqslant v \leqslant k$, we use $C_{ijv}$ to store $\eta_v(\boldsymbol{x}_j^{(i)})$;
 3: end if
 4: for $t = 1, 2, \ldots, T_{\text{max}}$ do
 5:   Shuffle $\mathcal{D} = \{(\boldsymbol{x}_{1:m}^{(i)}, z^{(i)})\}_{i=1}^{n}$;
 6:   for $j = 1, 2, \ldots I_{\text{max}}$ do
 7:     Fetch mini-batch $\mathcal{D}_j$ from $\mathcal{D}_j$;
 8:     if flag$_{\text{init}}$ and $t \leqslant T_{\text{init}}$ then
 9:       Update model $f$ by log-likelihood method (Zhang et al., 2020);
10:     else
11:       if flag$_{\text{mat}}$=True then
12:         Fetch $\eta(\boldsymbol{x}_v^{(i)}), 1 \leqslant v \leqslant m$ from $C$;
13:       else
14:         Calculate $\eta(\boldsymbol{x}_v^{(i)}), 1 \leqslant v \leqslant m$ by model $f$ and detach the gradient;
15:       end if
16:       Update model $f$ by $\hat{R}(f)_{\text{agg}}$ in Eq. (6);
17:     end if
18:     if flag$_{\text{mat}}$ =TRUE then
19:       Update $C$ by model $f$;
20:     end if
21:   end for
22: end for
Output: $f$.
```

process is provided between line 11 to line 19 in Algorithm 1. It contains 2 steps: 1) obtaining approximated $\eta(\boldsymbol{x})$. 2) using the approximated $\eta(\boldsymbol{x})$ to calculate $\hat{R}_{\text{agg}}(f)$ and update model f. Since different CFAO tasks need different training strategies, we provide two strategies to obtain approximated $\eta(\boldsymbol{x})$. One way is to approximate $\eta(\boldsymbol{x})$ using the current model outputs. The other way is to approximate $\eta(\boldsymbol{x})$ by the model outputs from last epoch. We implement this by storing the outputs of model in a matrix and fetching $\eta(\boldsymbol{x})$ from the matrix during training. These two strategies corresponding to whether to use a radical way to update weights during training. We use flag$_{\text{mat}}$ to denote the strategy to update weights during training. We use a matrix to store $\eta(\boldsymbol{x})$ when flag$_{\text{mat}}$ is set to TRUE, otherwise we obtain $\eta(\boldsymbol{x})$ by current model outputs. The matrix is initialized uniformly, which means we initialize all elements in the matrix to $\frac{1}{k}$.

$p(z|\boldsymbol{x}_{1:m})$ and $p(z, y_i = j|\boldsymbol{x}_{1:m})$ play an important rule in UUM to provide purified supervision for the classifier to learn. Especially, the value of these two probability functions depends on two components during the training process, i.e. $\eta(\boldsymbol{x})$ approximated by the model outputs and z given by the aggregate observation dataset. During the warm-up phase, $p(z \mid \boldsymbol{x}_{1:m})$ and $p(z, y_i = j \mid \boldsymbol{x}_{1:m})$ mainly depend on z since the model learned limited information in warm-up phase.

However, in some CFAO problems, z provides little information to weights during the warm-up phase, e.g. pairwise similarity and triplet comparison. If we set $\eta_j(\boldsymbol{x}) = \frac{1}{k}$ for all j, $\frac{p(z,y_i=j|\boldsymbol{x}_{1:m})}{p(z|\boldsymbol{x}_{1:m})}$ would equal to $\frac{1}{k}$ for all i and j no matter z takes 0 or 1, which means aggregate label would provides little information to help UUM approximating weights precisely.

Table 10. Test performance of UUM on flag$_{\text{init}}$=FALSE for pairwise similarity

Dataset	MNIST	KUZUSHIJI	FASHION	CIFAR10	SVHN
log-likelihood	$98.87 \pm 0.06\%$	$93.40\% \pm 0.64\%$	$88.75 \pm 0.31\%$	$71.45 \pm 0.47\%$●	$91.06 \pm 0.24\%$●
UUM(ours)	$\mathbf{99.01 \pm 0.13}\%$	$\mathbf{93.66 \pm 0.47}\%$	$\mathbf{89.00 \pm 0.49}\%$	$\mathbf{73.45 \pm 0.40}\%$	$\mathbf{92.37 \pm 0.23}\%$

In order to obtain precise weights, we could use another method which dose not rely on weights to warm up model. We use log-likelihood as the warm-up method in our algorithm. The warm-up phase is shown in line 9 in Algorithm 1. We use $\text{flag}_{\text{init}}$ and T_{init} to denote whether to use log-likelihood to warm up model and the number of warm-up epoch respectively.

We also conduct experiments on UUM on $\text{flag}_{\text{init}} = \text{FALSE}$ for pairwise similarity on benchmark datasets. We trained model for 100 epochs. For MINIST, Kuzushiji-MNIST and Fashion, the falg_{mat} is set to FALSE in first 30 epoch and set to TRUE in last 70 epochs. For CIFAR-10 and SVHN, the flag_{mat} is set to FALSE in first 50 epoch and set to TRUE in last 50 epochs. The experimetal results are provided on Table 10.

D.2. Training Details

We used Adam (Kingma & Ba, 2015) optimizer with 0 weight decay to train the model. The learning rates were 1e-3, 1e-3 and 2e-1 for benchmark datasets, UCI datasets and MIL datasets respectively. The batch size is 128 for benchmark datasets and UCI datasets. We search the batch size from (128, 256,512,1024,2048,4096) for MIL datasets. The model is trained for 100, 200, and 3500 epochs for benchmark datasets, UCI datasets and MIL datasets respectively. We evaluate test performance on the model obtained the best performance on validation sets.

The $\text{flag}_{\text{init}}$ in Algorithm 1 is set TRUE for pairwise similarity and triplet comparison and set FALSE for MIL and LLP. flag_{mat} is set TRUE for classification via pairwise similarity/triplet comparison/learning from label proportions, and set FALSE for multiple-instance learning. The value of T_{init} in pairwise similarity and triplet comparison is set to 20 and 100 for benchmark datasets and UCI datasets respectively.

D.3. Benchmark Datasets for Multiple-Instance Learning

We use five commonly used benchmark datasets in MIL studies (Dietterich et al., 1997; Andrews et al., 2002), including Musk1, Musk2, Elephant, Fox, and Tiger. For these datasets, Musk1 has 47 positive bags and 45 negative bags. Musk2 consists of 39 positive bags and 63 negative bags. The other three datasets contain 100 positive bags and 100 negative bags. It is worth noting that these datasets are too small to evaluate the task of MIL from similar and dissimilar bags, we follow Bao et al. (2018b) to augment them for increasing the number of bags. Specifically, bags chosen randomly from the original datasets were duplicated and then Gaussian noise with mean zero and variance 0.01 was added to each dimension. In this way, we increased the number of samples in the Musk datasets (Musk1 and Musk2) 10 times and the Corel datasets (Elephant, Fox, and Tiger) 5 times. Table 11 reports the characteristics of these datasets[1] after preprocessing.

Table 11. The characteristics of the used benchmark datasets for multiple-instance learning.

Dataset	# Features	# Positive bags	# Negative bags	# Avg. Pos. Ins. per bag	# Avg. Neg. Ins. per bag
Musk1	166	475	445	2.2±2.5	2.9±7.0
Musk2	166	413	607	8.9±22.7	49.9±169.7
Elephat	230	504	496	3.9±4.2	3.2±3.6
Fox	230	498	502	3.2±3.6	3.4±3.8
Tiger	230	506	494	2.8±3.1	3.4±3.9

E. Additional Experiments for Variant Size of Training Set

Since the size of training sets may vary from a large range depending on different tasks in practice, we conducted additional experiments by reducing the sample size for training. The experimental results are shown in Table 12 and Table 13. As shown in Table 12 and Table 13, when reducing the sample size for training, our proposed method consistently outperforms other compared methods.

[1] http://www.cs.columbia.edu/~andrews/mil/datasets.html

Table 12. Experiments on classification via pairwise similarity with different aggregate observation sample size formed by CIFAR10

Method	10000	30000	60000	90000	120000
log-likelihood	33.30%	56.15%	66.30%	70.97%	71.45%
siamese	31.74%	41.63%	48.01%	54.27%	58.16%
contrastive	20.23%	20.35%	20.66%	21.79%	23.69%
UUM	**34.14%**	**59.02%**	**69.17%**	**72.64%**	**72.52%**

Table 13. Experiments on classification via triplet comparison with different aggregate observation sample size formed by CIFAR10

Method	10000	30000	60000	90000	120000
log-likelihood	31.74%	53.15%	66.66%	68.67%	70.57%
triplet	26.48%	33.81%	7.21%	39.85%	47.09%
(2+1)tuple	24.64%	29.74%	36.65%	40.55%	45.53%
UUM	**33.03%**	**56.44%**	**63.38%**	**72.97%**	**72.09%**