On Coresets for End-to-end Learning from Crowds

Hang Yang[1], **Zhiwu Li**[1,*], **Witold Pedrycz**[2,3]

[1]Macau Institute of Systems Engineering, Macau University of Science and Technology
[2]Department of Electrical and Computer Engineering, University of Alberta
[3]Systems Research Institute, Polish Academy of Sciences
hangy03@student.must.edu.mo, zwli@must.edu.mo, wpedrycz@ualberta.ca

Abstract

Crowdsourcing is a common approach for training data-hungry models by collecting high-quality labeled data with human labor. With crowdsourcing data, the end-to-end learning paradigm is rising, where the classifier is concatenated with annotator-specific confusion layers and the two parts are co-trained in a parameter-coupled manner. However, learning with the size of a very large set of annotations is a challenge when computation or energy is limited. In this paper, we analyze and refine the coresets for end-to-end learning from crowds under the sensitivity sampling framework. This coreset is a small possible subset of annotations, so one can efficiently optimize the Coupled Cross-Entropy Minimization problem with guaranteed approximation. We first prove the lower bound, which shows no coresets smaller than complete data with confusion layers. Then, with workers' transition matrices $\boldsymbol{A}_r$, we show that with the regularization term $\log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r$, this lower bound can be prevented. Our main result is that under mild assumptions, a smaller coreset exists for the regularized Coupled Cross-Entropy Minimization problem. An upper bound of sensitivity is proposed for designing a sampling algorithm called CrowdCore. The experimental results on synthetic and real-world datasets demonstrate the effectiveness of our analysis.

Introduction

In recent years, deep neural network training on large-scale datasets has achieved remarkable success (LeCun, Bengio, and Hinton 2015; Vaswani et al. 2017; Momeni et al. 2025). For these data-hungry deep models, crowdsourcing serves as a valuable approach for gathering labeled data from human workers (Chu, Ma, and Wang 2021). However, a key challenge arises when dealing with very large datasets of annotations. The size of annotations is usually far larger than the size of instances, e.g., images. Therefore, the computational burden of learning from crowds is worse than that of traditional machine learning, and it is urgent to develop efficient algorithmic techniques for reducing computational complexity.

Coreset (Feldman, Schmidt, and Sohler 2013) is a well-studied technique that can efficiently sketch large datasets without sacrificing performance. For example, existing work shows that empirically, a coreset of size less than 1% of the input is enough to represent the whole dataset with an error less than 0.001 (Tolochinksy, Jubran, and Feldman 2022). For input data with size N, developing coreset means that there will be a smaller subset with size $O(f(N))$ that can play the role of the whole dataset, where $f(N)$ is the sample complexity. Assume that there are R annotators to label N data points. The worst sample complexity will be $O(R \cdot f(N))$. However, this bound is not satisfied since the number of annotations M is much smaller than RN.

Our goal is to find a better bound on the coreset size for crowdsourcing. More detailed, we focus on the end-to-end learning from crowds. "End-to-end" means that the task model and label correction mechanism are co-trained in an end-to-end fashion (Rodrigues and Pereira 2018). The basic paradigm is connecting the confusion layer behind the classifier. Since the parameters of confusion layers $\boldsymbol{A}_r$ and classifier $\boldsymbol{f}$ are coupled optimized with cross-entropy loss, this approach is called Coupled Cross-Entropy Minimization (CCEM). It is proven that under CCEM, the distances between the trained classifier, the confusion layer, and their ground truth are bounded (Ibrahim, Nguyen, and Fu 2023).

Our analysis is under the sensitivity framework, where the key challenge is to find the upper sensitivity function and total sensitivity. Our first observation is that with the original CCEM loss function, there is no coreset smaller than the complete annotations. This lower bound is similar to traditional classification, where a norm regularization of weight is introduced to prevent the lower bound. In CCEM, we find that the regularization term $\log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r$ can help coreset to exist. Geometrically, this term is a surrogate of the volume of $\text{conv}\{\boldsymbol{A}_r\} = \{\boldsymbol{x} \in \mathbb{R}_d | \boldsymbol{x} = \boldsymbol{A}_r\boldsymbol{\theta}, \boldsymbol{\theta} \geq \boldsymbol{0}\}$. Therefore, this term is called volume regularization. This regularization is widely used in improving identifiability in the non-negative matrix factorization tasks (Fu et al. 2019; Ibrahim, Nguyen, and Fu 2023). Nonetheless, its property in coresets has not been studied yet.

Our main result lies in proving that, under mild assumptions, a smaller coreset exists for the regularized CCEM problem. We only add an assumption that limits the eigenvalues of $\boldsymbol{A_r}$. The other assumptions follow the existing works. We first analyze the properties of the new cost function, then bridge the sensitivity to the confusion layer $\boldsymbol{A_r}$. Then, we propose an upper-sensitivity function and derive

*Corresponding author.

the sample probability for each data point. With total sensitivity, the bound of the coreset size is given. Our proof leverages the framework proposed in (Tolochinksy, Jubran, and Feldman 2022) but differs significantly from the study in that work due to different cost functions, assumptions, and data spaces. The contribution can be summarized as follows:

- We provide a rigorous theoretical analysis of coresets for end-to-end learning from crowdsourced data, establishing a lower bound that demonstrates the necessity of using complete data with confusion layers to achieve optimal solutions in the CCEM problem.
- We show that the lower bound identified in our analysis can be avoided with a volume regularization term. We prove, under mild assumptions, the existence of a smaller coreset that can be used to solve the regularized CCEM problem efficiently. This provides an approach to scaling end-to-end learning from crowdsourced data without degrading much accuracy.
- Through empirical evaluation of both synthetic and real-world datasets, the results empirically show that the proposed sampling strategy, CrowdCore, can reduce the number of annotations with small approximation error, and model performance will not significantly decrease.

The originality of this work stems from our analysis and refinement of coresets for end-to-end learning from crowds under the sensitivity sampling framework to optimize the CCEM problem with guaranteed approximation.

Related Works

End-to-end Learning from Crowds. The first work in end-to-end models is CrowdLayer (Rodrigues and Pereira 2018), which applies a learnable crowd layer after the classifier for confusion modeling. After that, TraceReg (Tanno et al. 2019) introduces a regularization term in mapping the classifier output onto the worker-specified output. Besides, CoNAL (Chu, Ma, and Wang 2021) goes further by distinguishing a common confusion from the individual confusion of each worker. UnionNet (Wei et al. 2022) integrates the confusion of all workers into a parametric transition matrix, treating all workers as a unified entity. The CCEM problem is formally analyzed in (Ibrahim and Fu 2021), where the error bounds of learning ground-truth classifier and confusion layers are provided. Besides, the geometry property of the non-negative matrix factorization is introduced to improve the identification (Ibrahim and Fu 2021; Fu et al. 2019).

Coresets. Coreset is well-studied in clustering (Feldman, Schmidt, and Sohler 2013), classification (Chen et al. 2022), and regression (Mirzasoleiman, Bilmes, and Leskovec 2020). The most used framework to analyze coresets is provided in (Feldman and Langberg 2011), where the sensitivity of data points is modeled. The work most related to our paper is the coresets for classification. Existing works show that with an additional common regularization term, i.e., norm-2 regularization of classifier parameters, the smaller coreset always exists with sensitivity-based sampling (Tolochinksy, Jubran, and Feldman 2022) or uniform sampling (Alishahi and Phillips 2024; Samadian et al. 2020). There is no direct research on coresets in crowdsourcing, but some work has been done to decrease cost-saving. Probably approximately correct (PAC) is used to study the cost-saving effect, and an upper bound for the minimally sufficient number of crowd labels can be given (Wang and Zhou 2016). Then, the cost complexity, also based on PAC, is proposed to model the trade-off between costs and quality (Fang et al. 2018).

Preliminaries

End-to-end Learning from Crowds

Notation. In this paper, we focus on binary classification with crowdsourcing datasets. Suppose that there are R workers labeling N instances as belonging to $K = 2$ possible classes, and the number of annotations is M. Notation $\boldsymbol{x}_i \in \mathbb{R}^d$ refers to the i-th instance, and $y_{ri} \in \{1, -1\}$ refers to the label from the r-th worker on the i-th instance, where d is the data dimension.

We denote the instance set as $X = \{\boldsymbol{x}_i\}_{i=1}^N$, the annotation set as $Y = \{y_{ri}\}$, where $|Y| = M$, and the unknown instance truth set as $Z = \{z_i\}_{i=1}^N$. Let us represent the classifier as $\boldsymbol{f}$, and the workers' transition matrices as $\{\boldsymbol{A}_r\}_{r=1}^R$, where the columns of $\boldsymbol{A}_r$ are conditional probability distributions. To keep consistent with the existing study (Tolochinksy, Jubran, and Feldman 2022), $\boldsymbol{f}$ is parameterized with a vector $\boldsymbol{w}$ with shape $d \times 1$ as:

$$\boldsymbol{f}(\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\boldsymbol{w}^\top \boldsymbol{x}\right)}. \tag{1}$$

End-to-end Training. In general, the end-to-end training paradigm, classifier $\boldsymbol{f}$ and worker parameters $\{\boldsymbol{A}_r\}_{r=1}^R$ are optimized jointly, called CCEM. Note that the classifier is sequentially combined with a feature extractor and a linear layer. The objective function of CCEM is:

$$\min_{\boldsymbol{f}, \{\boldsymbol{A}_r\}_{r=1}^R} -\frac{1}{M} \sum_{y_{ri} \in Y} \sum_{k=1}^{K} \mathbb{I}(y_{ri} = k) \log[\boldsymbol{A}_r \boldsymbol{f}(\boldsymbol{x}_i)]_k. \tag{2}$$

After proper network initialization, the optimal parameters of crowd layer $\{\boldsymbol{A}_r\}_{r=1}^R$ and classifier $\boldsymbol{f}$ can be estimated with stochastic gradient descent.

Coreset Construction with Sensitivity

To illustrate the concept of sensitivity, we first expand the confusion layer $\boldsymbol{A}_r$ in the binary classification scenario with two variables $a_r, b_r \in [0, 1]$ as $\boldsymbol{A}_r = \begin{bmatrix} a_r & 1 - b_r \\ 1 - a_r & b_r \end{bmatrix}$.

It can be verified that for any $\boldsymbol{\theta}$, if $\mathbf{1}^\top \boldsymbol{\theta} = 1$, $\mathbf{1}^\top \boldsymbol{A}_r \boldsymbol{\theta} = 1$. Then, combine Eqs. (1) and (2), the logistic cost function of a specific data point $(\boldsymbol{x}_i, y_{ri})$ is:

$$\begin{aligned}
\phi(t) &= \log(1 + \mathrm{e}^t), \\
\phi_1(t) &= \log[a_r + (1 - b_r)\mathrm{e}^t], \\
\phi_2(t) &= \log[b_r + (1 - a_r)\mathrm{e}^t], \\
c(\mathbf{x}_i, y_{ri}) &= \begin{cases} \phi(-y_{ri}\mathbf{w}^\top \mathbf{x}_i) - \phi_1(-y_{ri}\mathbf{w}^\top \mathbf{x}_i), & y_{ri} = 1, \\ \phi(-y_{ri}\mathbf{w}^\top \mathbf{x}_i) - \phi_2(-y_{ri}\mathbf{w}^\top \mathbf{x}_i), & y_{ri} = -1. \end{cases}
\end{aligned} \tag{3}$$

Some necessary definitions are given as follows.

Definition 1 (Query space)**.** *A tuple containing:*

- *Input data: complete set of data points $P = (X, Y)$;*
- *Classifier space: $\boldsymbol{f} \in \mathcal{F}$ or equivalently, $\boldsymbol{w} \in \mathbb{R}^d$;*
- *Confusion layer space: matrix space Δ^K, which is simplified to $(a, b) \in ([0, 1], [0, 1])$ when $K = 2$;*
- *Cost function: $c(\boldsymbol{x}_i, y_{ri}; \boldsymbol{f}, \boldsymbol{A}_r)$.*

is called query space for end-to-end learning from crowds.

Note that the confusion layer space is sometimes ignored since the initialization of $\boldsymbol{A}_r$ is determined as an identify matrix. Then, the definition of the coreset is as follows.

Definition 2 (ε-coreset)**.** *Given query space $(P, \mathcal{F}, \Delta^K, c)$, and error parameter $\varepsilon \in (0, 1)$, an ε-coreset for $(P, \mathcal{F}, \Delta^K, c)$ is a tuple (Q, u), where Q is a subset and u is a weight function, such that for every $\boldsymbol{f} \in \mathcal{F}, \boldsymbol{A}_r \in \Delta^K$,*

$$|\sum_{P} c(\boldsymbol{x}_i, y_{ri}) - \sum_{Q} u(\boldsymbol{x}_i, y_{ri}) c(\boldsymbol{x}_i, y_{ri})| \leq \varepsilon \cdot \sum_{P} c(\boldsymbol{x}_i, y_{ri}). \tag{4}$$

It can be seen that there is a relationship between the weight function u and the size M of the complete dataset.

Proposition 1. *Assume that (Q, u) is a ε-coreset for $(P, \mathcal{F}, \Delta^K, c)$, where cost c is always positive. The following inequality holds:*

$$|\sum_{Q} u(\boldsymbol{x}_i, y_{ri}) - M| \leq \varepsilon M. \tag{5}$$

Proof. Let $\boldsymbol{w} = \boldsymbol{0}$, and $\boldsymbol{A}_r = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. We have that for any data point, $c(\boldsymbol{x}_i, y_{ri}) = \phi(0) = \log 2$. By Definition 2, $|\sum_P \log 2 - \sum_Q u(\boldsymbol{x}_i, y_{ri}) \log 2| \leq \varepsilon \sum_P \log 2$. Since $\sum_P 1 = M$, we have $|\sum_Q u(\boldsymbol{x}_i, y_{ri}) - M| \leq \varepsilon M$. □

Definition 3 (Sensitivity)**.** *Given query space $(P, \mathcal{F}, \Delta^K, c)$, the sensitivity of a data point $(\boldsymbol{x}_i, y_{ri})$ is defined as the supreme value of fraction between the cost of this point and the cost of all points for all possible classifiers, i.e.,*

$$\text{sensitivity}(\boldsymbol{x}_i, y_{ri}) \triangleq \sup_{\boldsymbol{f} \in \mathcal{F}, \boldsymbol{A}_r \in \Delta^K} \frac{c(\boldsymbol{x}_i, y_{ri})}{\sum_P c(\boldsymbol{x}_i, y_{ri})}, \tag{6}$$

where the denominator $\sum_P c(\boldsymbol{x}_i, y_{ri})$ is assumed to be positive. Since it is almost impossible to get the closed form of sensitivity, a function $s(\boldsymbol{x}_i, y_{ri})$ is called an upper sensitivity function if for all $\boldsymbol{x}_i, y_{ri}$, $s(\boldsymbol{x}_i, y_{ri}) \geq \text{sensitivity}(\boldsymbol{x}_i, y_{ri})$.

The total sensitivity for $(P, \mathcal{F}, \Delta^K, c)$ is

$$S \triangleq \sum_{P} s(\boldsymbol{x}_i, y_{ri}). \tag{7}$$

With an upper bound for the sensitivity and total sensitivity, a sensitivity-based sample from P with a size of M' is a set of M', independent and identically distributed (i.i.d.), draws from the complete set of data points, where the sample probability is $\text{Prob}(\cdot) = \frac{s(\cdot)}{S}$. Then, the coreset size is bounded according to the following theorem.

Theorem 1 ((Feldman and Langberg 2011))**.** *Given query space $(P, \mathcal{F}, \Delta^K, c)$, let s be the upper sensitivity function, and S be the upper total sensitivity. Let D be the VC-dimension of the loss function, and $\varepsilon, \delta \in (0, 1)$ be the error and probability control parameters. Then, Q is a random sample of size M' that*

$$M' \geq \frac{10S}{\varepsilon^2}(D \log S + \log(\frac{1}{\delta}))), \tag{8}$$

where the sample probability of point $i \in [1, M]$ selected each time is s_i/S. Let the associated weight u_i for each point $i \in [1, M]$ be $\frac{S}{s_i\,|Q|}$. We have that (Q, u) is an ε-coreset with probability at least $1 - \delta$.

Lower Bound

Given a query space, the first question is whether the coreset exists. Studies such as those in (Tolochinksy, Jubran, and Feldman 2022; Samadian et al. 2020) show that for pure logistic regression, where the cost is cross-entropy between prediction and ground truth, the ε-coreset does not exist. The key lemma used is as follows.

Lemma 1 ((Tolochinksy, Jubran, and Feldman 2022))**.** *If every data point $(\boldsymbol{x}_i, y_{ri})$ in P has $\text{sensitivity}(\boldsymbol{x}_i, y_{ri}) = 1$, then the only ε-coreset for P is P itself, i.e., no smaller coreset.*

To determine the lower bound of corsets for crowdsourcing learning, we explore three cases that control the size of instances and annotators.

Case (a): Multiple Instances, Single Annotator. Assume that there are N instances $\{\boldsymbol{x}_i\}_{i=1}^N$ and one annotator whose confusion layer $\boldsymbol{A}_0$ has diagonal elements a_0 and b_0. In this case, there are only single annotator, i.e., $M = N$. This situation will degrade to logistic regression when a_0 and b_0 are both fixed to 1. For points $\boldsymbol{x}_i$ scattered on a circle which does not pass through the zero point:

$$\begin{aligned}
\text{sensitivity}(\boldsymbol{x}_i, y_{0i}) &= \sup_{\boldsymbol{w} \in \mathbb{R}_d, a_0, b_0 \in [0,1]} \frac{c(\boldsymbol{x}_i, y_{0i})}{\sum_P c(\boldsymbol{x}_i, y_{0i})} \\
(\text{let } a_0 = b_0 = 1) &\geq \sup_{\boldsymbol{w} \in \mathbb{R}_d} \frac{c(\boldsymbol{x}_i, y_{0i}; a_0 = 1, b_0 = 1)}{\sum_P c(\boldsymbol{x}_i, y_{0i}; a_0 = 1, b_0 = 1)} \\
(\text{Eq. 3}) &= \sup_{\boldsymbol{w} \in \mathbb{R}_d} \frac{\phi(-y_{0i} \boldsymbol{w}^\top \boldsymbol{x}_i)}{\sum_P \phi(-y_{0i} \boldsymbol{w}^\top \boldsymbol{x}_i)} \\
(*) &= 1.
\end{aligned} \tag{9}$$

We briefly describe the intuition of $(*)$. A geometric trick is used in the logistic cost function, where the data point $\boldsymbol{x}_i$ is augmented with y_{ri} to become $y_{ri}\boldsymbol{x}_i$. With this trick, all points in Fig. 1(a) can be seen as positive samples. For each $y_{ri}\boldsymbol{x}_i$, there is a hyper-plane that can perfectly separate this point from others, i.e., for $y_{ri}\boldsymbol{x}_i$, one has $y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i > 0$ holds, while for any other points $y_{rj}\boldsymbol{x}_j$, $y_{rj}\boldsymbol{w}^\top \boldsymbol{x}_j < 0$. Let $\|\boldsymbol{w}\|_2 \to \infty$. We have that $\phi(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i) \to \infty$ for $y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i < 0$, and $\phi(y_{rj}\boldsymbol{w}^\top \boldsymbol{x}_j) \to 0$ for

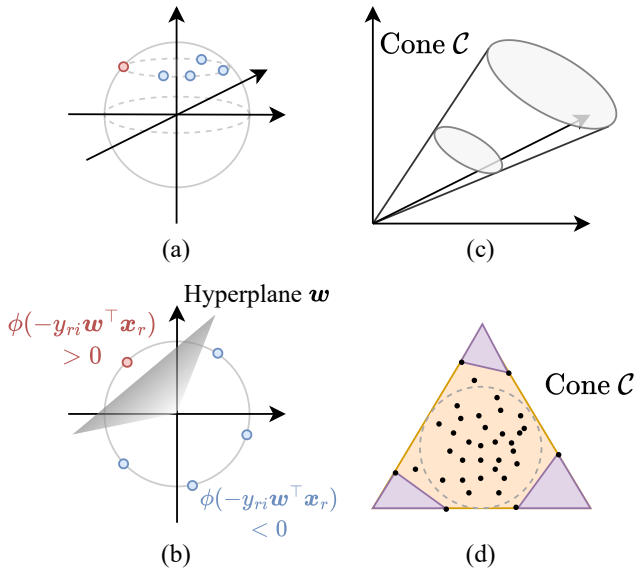

Figure 1: (a) Data points in $\mathbb{R}^3$ scattered on a circle, (b) Cross-sectional view of (a), the hyperplane $\boldsymbol{w}$ exists for every data point; (c) A second-order cone $\mathcal{C}$ in $\mathbb{R}^3$, and (d) Cross-sectional view of (c), where the dots denote the rows of $\boldsymbol{A}_r$ and the circle denotes the cone $\mathcal{C}$.

$y_{rj}\boldsymbol{w}^\top \boldsymbol{x}_j > 0$. Therefore, $\sup_{\boldsymbol{w}\in\mathbb{R}_d} \frac{\phi(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i)}{\sum_P \phi(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i)} = \lim_{\|\boldsymbol{w}\|_2\to\infty} \frac{\phi(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i)}{\sum_P \phi(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i)} = 1$.

The above analysis holds for every point $(\boldsymbol{x}_i, y_{ri})$. Therefore, the sensitivity of every point is 1. According to Lemma 1, no smaller coreset exists in this case.

Solution (a): Add Norm Regularization $\|\boldsymbol{w}\|_2$. To prevent this lower bound, the norm-2 regularization term $\|\boldsymbol{w}\|_2$ is added to the cost function. In this way, when $\boldsymbol{w} \to \infty$, the cost of all points will approach ∞, and thus the sensitivity of any data point will not be 1. With this regularization term and some assumptions, the coreset size is bounded by different sampling methods.

Theorem 2 ((Alishahi and Phillips 2024)). *The query space with a single annotator is $(P, \mathcal{F}, (a_0, b_0), c)$, where the confusion layer can be fixed without loss of generality, i.e., $a_0 = b_0 = 1$, and cost function is $c(y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i) = \phi(y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i) + \|\boldsymbol{w}\|_2$. Assume that for all $i, \|\boldsymbol{x}_i\|_2 \le 1$, and any s-sensitivity sample with size M' by specific sampling strategy guarantees an ε-coreset with a probability of at least $1-\delta$, with sensitivity-based sampling, $M' = O(\frac{d^2 \log M}{\varepsilon^2})$ holds.*

Case (b): Single Instance, Multiple Annotators. Assume that there are M annotators with confusion layer $\{\boldsymbol{A}_r\}_{r=1}^M$ and only one instance $\boldsymbol{x}_0$. Similarly to case (a), we analyze the sensitivity of data point $(\boldsymbol{x}_0, y_{r0})$. We find that for the cost function in Eq. (3), the sensitivity of every point is 1, which implies that there is no nontrivial coreset. This observation is formally stated as follows.

Concept	Case (a)	Case (b)
Perspective	Instance	Annotator
Problematic	$c \to \infty$	$c \to 0$
Reg. Term	$\|\boldsymbol{w}\|_2$	$\log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r$
Existing Usage	Avoid overfitting	Improve identifiability
Geometry	Length	Volume

Table 1: Correspond concepts in classification and crowdsourcing.

Theorem 3 (No coreset for one-instance crowdsourcing)**.** *Given a query space $(P, \mathcal{F}, \Delta^K, c)$, where the instance space contains only one item. If the cost is a CCEM loss function, i.e., Eq. (3), and let the error parameter be $\varepsilon \in (0,1)$, then there is only an ε-coreset Q such that $Q = P$.*

Proof. It can be checked that:

- If $a_r = 1$ and $b_r = 1$, then $c(\boldsymbol{x}_0, y_{r0}) = \phi(-y_{r0}\boldsymbol{w}^\top \boldsymbol{x}_0)$,
- If $y_{r0} = 1$, $a_r = 1$ and $b_r = 0$, then $c(\boldsymbol{x}_0, y_{r0}) = 0$,
- If $y_{r0} = -1$, $a_r = 0$ and $b_r = 1$, then $c(\boldsymbol{x}_0, y_{r0}) = 0$.

Therefore, for any data point, we have:

$$\begin{aligned}\text{sensitivity}(\boldsymbol{x}_0, y_{r0}) &= \sup_{\boldsymbol{w}\in\mathbb{R}_d, a_r, b_r\in[0,1]} \frac{c(\boldsymbol{x}_0, y_{r0})}{\sum_P c(\boldsymbol{x}_0, y_{r0})} \\ &\ge \sup_{\boldsymbol{w}\in\mathbb{R}_d} \frac{c(\boldsymbol{x}_i, y_{ri}; a_r = 1, b_r = 1)}{c(\boldsymbol{x}_i, y_{ri}; a_r = 1, b_r = 1) + \sum_{P'} 0} \\ &= 1,\end{aligned} \tag{10}$$

where $P' = P - \{(\boldsymbol{x}_0, y_{r0})\}$.

With Lemma 1, one concludes that there is no smaller coreset for P. □

Solution (b): Add Volume Regularization $\log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r$. What we need to prevent this lower bound is an additional item that dominates the cost when the cross-entropy loss approaches zero. We found that the volume regularization satisfied this requirement.

Remind that $\boldsymbol{A}_r = \begin{bmatrix} a_r & 1-b_r \\ 1-a_r & b_r \end{bmatrix}$. We have $\log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r = \log(a_r + b_r - 1)^2$. Either $a_r \to 1$ and $b_r \to 0$ or $a_r \to 0$ and $b_r \to 1$, $\log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r \to \infty$ holds.

Coincidentally, this regularization term is used to improve the identifiability of crowdsourcing (Ibrahim, Nguyen, and Fu 2023), where the volume of $\boldsymbol{A}_r$ cone is maximized. We summarize the corresponding concepts in coresets for logistic regression and crowdsourcing in Table 1.

Case (c): Multiple Instances, Multiple Annotators. In this case, it is intuitive that to sample the smallest coreset, the norm and volume regularization terms should both be used. The cost function is:

$$\begin{aligned} c(\boldsymbol{x}_i, y_{ri}) = \|\boldsymbol{w}\|_2 - \log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r + \\ \begin{cases} \phi(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i) - \phi_1(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i), & y_{ri} = 1, \\ \phi(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i) - \phi_2(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i), & y_{ri} = -1. \end{cases} \end{aligned} \tag{11}$$

In the next section, we show that a better coreset does exist for any input.

Coresets with Volume Regularization

Main Results

We make the following assumptions.

Assumption 1 (Bounded $\|\boldsymbol{x}_i\|_2$, (Alishahi and Phillips 2024)). *For all $\boldsymbol{x}_i, \|\boldsymbol{x}_i\|_2 < 1$, i.e., Prob$(\|\boldsymbol{x}_i\|_2 \geq 1) = 0$. This assumption is easy to satisfy with normalization.*

Assumption 2 (Bounded $\log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r$). *Let $\xi \in (0,1)$. The minimum eigenvalue of $\boldsymbol{A}_r$ is greater than or equal to ξ, which implies that $|a_r + b_r - 1| \geq \xi$, and $\log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r \geq 2\log\xi$. Note that $\log\det \boldsymbol{A}_r^\top \boldsymbol{A}_r \leq 0$.*

Remark 1. *Assumption 2 is natural when there is no malicious annotator whose annotations are independent of instances. If the minimum eigenvalue of $\boldsymbol{A}_r$ is 0, or $a_r + b_r = 1$, it can be checked that the output probability is always $[a_r, b_r]^\top$. Without loss of generality, in our analysis, $a_r + b_r > 1$. If $a_r + b_r < 1$, with multiplying by an additional permutation matrix, $a_r + b_r > 1$ (Ibrahim, Nguyen, and Fu 2023).*

With the above assumption, and also, with the cost function Eq. (11), we first define auxiliary functions:

$$s_A(y_{ri}, a_r, b_r) = \begin{cases} \frac{2\log|a_r+b_r-1|+\log(b_r-a_r+1)-\log 2}{2\log|a_r+b_r-1|+\log(1-b_r)}, & y_{ri}=1, \\ \frac{2\log|a_r+b_r-1|+\log(a_r-b_r+1)-\log 2}{2\log|a_r+b_r-1|+\log(1-a_r)}, & y_{ri}=-1. \end{cases} \tag{12}$$

$$b_A(y_{ri}, a_r, b_r) = s_A(y_{ri}, a_r, b_r) \cdot (2 - \frac{1}{\log|a_r + b_r - 1|}). \tag{13}$$

$$b_x(\boldsymbol{x}_i) = 2 + \|\boldsymbol{x}_i\|_2^2. \tag{14}$$

Then, the sensitivity is bound as follows.

Theorem 4 (Sensitivity). *Given input data $P = (X, Y) = \{(\boldsymbol{x}_i, y_{ri})\}$ such that $|P| = M$, let the data points ascending sorted by $b_A(y_{ri}, a_r, b_r) \cdot b_x(\boldsymbol{x}_i)$. Then the sensitivity of every $(\boldsymbol{x}_i, y_{ri})$ in position m is bounded by $s(\boldsymbol{x}_i, y_{ri}) = O(\frac{L(b_A(y_{ri},a_r,b_r)\cdot b_x(\boldsymbol{x}_i)+1)}{m})$, where L is a sufficiently large constant, and the total sensitivity is*

$$t = \sum_P s(\boldsymbol{x}_i, y_{ri}) = O(\log M + \sum_P \frac{L(b_A b_x + 1)}{m}).$$

Compared with the existing bound of sensitivity which only considers the index position m and the norm of the data point $\|\boldsymbol{x}_i\|_2$, our bound extensively considers the factor of the annotator b_A. With this upper sensitivity bound, the sampling strategy is given in Algorithm 1, and the bound of coreset size is given in the following theorem.

Theorem 5 (Coreset). *Given input data $P = (X, Y) = \{(\boldsymbol{x}_i, y_{ri})\}$ such that $|P| = M$, let $\varepsilon, \delta \in (0,1)$, for every $\boldsymbol{x}_i \in \mathbb{R}^d, y_{ri} \in \{1, -1\}$, the cost function is given as Eq. (11). Let (Q, u) be the output of Algorithm 1 while the input is query space $(P, \mathcal{F}, \Delta^K, c)$. Then, with probability at least $1 - \delta$, (Q, u) is an ε-coreset for $(P, \mathcal{F}, \Delta^K, c)$. And the size of Q is bounded by $|Q| = O(\frac{\log M}{\varepsilon^2}(d\log\log M + \log\frac{1}{\delta}))$.*

Algorithm 1: CrowdCore Sample Strategy.

Input: Query space: $(P, \mathcal{F}, \Delta^K, c)$, and coreset size M'.
Output: Coreset (Q, u), where Q is a subset of P and u is a weight function.

1: Sort the data points $(\boldsymbol{x}_i, y_{ri})$ in input P ascending by $b_A(y_{ri}, a_r, b_r) \cdot b_x(\boldsymbol{x}_i)$, and record index position m.
2: Compute the upper sensitivity for data points in position m such that $s(\boldsymbol{x}_i, y_{ri}) = \frac{L(b_A(y_{ri}, a_r, b_r) \cdot b_x(\boldsymbol{x}_i) + 1)}{m}$, where L is a sufficiently large constant.
3: Compute total sensitivity $S = \sum_P s(\boldsymbol{x}_i, y_{ri})$.
4: Sample M' data points from P to Q with probability $\frac{s(\boldsymbol{x}_i, y_{ri})}{S}$, and set weight $u(\boldsymbol{x}_i, y_{ri}) = \frac{S}{M' s(\boldsymbol{x}_i, y_{ri})}$.
5: **return** Coreset (Q, u).

Proofs of Main Results

Define function $g(t) = g(-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i) = c - \|\boldsymbol{w}\|_2 = \phi(t) - \phi_{ri}(t) - \log(a_r + b_r - 1)^2$, where c is the cost function defined on Eq. (11), and $\phi_{ri}(t)$ depends on y_{ri}. The properties of the function g are as follows.

- Given y_{ri}, the function g is increasing with $-y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i$,
- The minimum is $\min g = \min(\log\frac{1}{a_r}, \log\frac{1}{b_r}) - \log(a_r + b_r - 1)^2$ when $y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i \to -\infty$.
- The maximum is $\max g = \max(\log\frac{1}{1-a_r}, \log\frac{1}{1-b_r}) - \log(a_r + b_r - 1)^2$ when $y_{ri}\boldsymbol{w}^\top \boldsymbol{x}_i \to \infty$.
- $g(0) \geq \log 2 - \max(\log(a_r - b_r + 1), \log(b_r - a_r + 1)) - \log(a_r + b_r - 1)^2 \geq 0$.

Using these properties, the key is to find the zero point of $g(-t) = t^2$. Since the function g is always positive and increasing, it is clear that there is only one zero point t_0. However, it is hard to give a closed form for this root. We give a bound of t_0 related to (a_r, b_r).

Proposition 2. *For $g(t) = \phi(t) - \phi_{ri}(t) - \log(a_r + b_r - 1)^2$, let $\hat{t}_0 = \sqrt{-\log(a_r + b_r - 1)^2} \in [0, \sqrt{-2\log\xi}]$, it holds $g(-\hat{t}_0) > \hat{t}_0^2$.*

Proof. By the symmetry of a and b, let $\phi_{ri} = \phi_1$. The proof for $\phi_{ri} = \phi_2$ is similar. Let $\hat{t}_0 = \sqrt{-\log(a_r + b_r - 1)^2}$. We have

$$\begin{aligned} g(-\hat{t}_0) =& \phi(-\hat{t}_0) - \phi_1(-\hat{t}_0) - \log(a + b - 1)^2 \\ =& \phi(-\hat{t}_0) - \phi_1(-\hat{t}_0) + \hat{t}_0^2, \end{aligned} \tag{15}$$

therefore, $g(-\hat{t}_0) - \hat{t}_0^2 = \phi(-\hat{t}_0) - \phi_1(-\hat{t}_0) \geq \min(\log\frac{1}{a_r}, \log\frac{1}{b_r}) > 0$. This bound t_0 is tight since when $a_r \to 1, b_r \to 0$, $g(-\hat{t}_0) = \hat{t}_0^2$. □

Lemma 2. *For $g(t) = \phi(t) - \phi_{ri}(t) - \log(a_r + b_r - 1)^2$, let $c > 0$. Then, for every $t > 0$,*

$$\frac{g(ct) + t^2}{g(-ct) + t^2} \leq \frac{\max g}{g(0)}(2 - \frac{1}{\log|a_r + b_r - 1|})(2 + c^2). \tag{16}$$

	MNIST				MiniBooNE			
	UniSamp	SenSamp	GradSamp	CrowdCore	UniSamp	SenSamp	GradSamp	CrowdCore
1%	1.422±0.986	0.647±0.439	2.408±1.467	**0.471±0.212**	1.091±0.425	1.180±0.699	1.233±0.388	**0.804±0.130**
Time	8.63±0.62	15.36±0.16	200.22±9.58	16.49±0.70	0.29±0.01	4.94±0.09	4.99±0.10	4.98±0.03
3%	0.789±0.418	0.599±0.360	1.181±0.677	**0.467±0.191**	0.636±0.268	0.785±0.315	1.039±0.207	**0.471±0.050**
Time	23.98±1.78	31.92±1.89	300.10±9.39	32.85±1.49	0.48±0.02	4.93±0.04	5.13±0.12	5.07±0.03
5%	0.784±0.510	0.422±0.257	0.856±0.654	**0.394±0.175**	0.451±0.179	0.668±0.269	1.062±0.484	**0.345±0.090**
Time	39.26±0.67	47.51±2.12	590.95±10.36	47.15±2.25	0.65±0.02	5.04±0.03	5.21±0.02	5.22±0.02
7%	0.568±0.336	**0.312±0.188**	0.890±0.729	0.344±0.169	0.424±0.164	0.534±0.189	0.601±0.223	**0.343±0.129**
Time	58.33±3.12	58.18±2.29	790.95±10.36	62.72±2.55	0.84±0.02	5.15±0.03	5.33±0.06	5.32±0.04
9%	0.362±0.229	0.317±0.089	0.793±0.634	**0.256±0.208**	0.313±0.172	0.478±0.244	0.628±0.153	**0.235±0.107**
Time	63.78±3.58	77.79±4.97	900.28±10.63	73.52±1.98	1.04±0.02	5.25±0.04	5.41±0.04	5.40±0.02

Table 2: Approximation Error and Running Time of Synthetic Annotations.

Proof. See the appendix. □

Then, we give the proof of our main results. For a data point $(\boldsymbol{x}_i, y_{ri})$, according to Lemma 2,

$$\frac{g(\|\boldsymbol{x}_i\|_2 t) + t^2}{g(-\|\boldsymbol{x}_i\|_2 t) + t^2} \leq \frac{\max g}{g(0)} \left(2 - \frac{1}{\log|a_r + b_r - 1|}\right)(2 + \|\boldsymbol{x}_i\|_2^2). \quad (17)$$

We divide the bound in Eq. (17) into two parts:

$$b_A(y_{ri}, a_r, b_r) = \frac{\max g}{g(0)}(2 - \frac{1}{\log|a_r + b_r - 1|}), \quad (18)$$

$$b_x(\boldsymbol{x}_i) = 2 + \|\boldsymbol{x}_i\|_2^2. \quad (19)$$

According to Lemma 4.2 in (Tolochinksy, Jubran, and Feldman 2022), for $(r', i') \neq (r, i)$, it holds

$$c(\boldsymbol{x}'_i, y_{r'i'}) \leq \max_P c(\boldsymbol{x}_i, y_{ri}) \leq L(b_A b_x + 1) c(\boldsymbol{x}_i, y_{ri}), \quad (20)$$

where L is a sufficiently large constant.

Then, sort points in P with $b_A b_x$. For data point $(\boldsymbol{x}_i, y_{ri})$ in the m position, we have

$$\sum_{\text{Top } m} c(\boldsymbol{x}'_i, y_{r'i'}) \geq \sum_{\text{Top } m} \frac{1}{L(b_A b_x + 1)} c(\boldsymbol{x}_i, y_{ri}) \geq \frac{m \cdot c(\boldsymbol{x}_i, y_{ri})}{L(b_A b_x + 1)}. \quad (21)$$

Consider the cost on all data points in P, it holds

$$\sum_P c(\boldsymbol{x}_i, y_{ri}) \geq \sum_{\text{Top } m} c(\boldsymbol{x}'_i, y_{r'i'}) \geq \frac{m \cdot c(\boldsymbol{x}_i, y_{ri})}{L(b_A b_x + 1)}. \quad (22)$$

Therefore, the upper sensitivity function is derived, i.e.,

$$\text{sensitivity}(\boldsymbol{x}_i, y_{ri}) = \sup_{\boldsymbol{f} \in \mathcal{F}, \boldsymbol{A}_r \in \Delta^K} \frac{c(\boldsymbol{x}_i, y_{ri})}{\sum_P c(\boldsymbol{x}_i, y_{ri})} \leq \frac{L(b_A b_x + 1)}{m} = s(\boldsymbol{x}_i, y_{ri}), \quad (23)$$

where m is the index after sorting.

Finally, summarize the upper sensitivity function over all data points to get the total sensitivity:

$$S = \sum_P s(\boldsymbol{x}_i, y_{ri}) = O(\log M + \sum_P \frac{L(b_A b_x + 1)}{m}). \quad (24)$$

Theorems 4 and 5 are directly derived from Theorem 1 and Eq. (24).

Experiments

Experimental Setup

To solve these three research questions, we conduct experiments on both synthetic datasets and real-world datasets.

- RQ1: Though there is a theoretical guarantee of the approximation error of cost, how does the coreset sampling algorithm perform with the large-scale dataset?
- RQ2: How does the model perform on unseen data with different coreset sampling algorithms?
- RQ3: How does the annotators' confusion matrix affect the approximation quality and generalization performance of the constructed coreset?

Four comparable sampling algorithms are used. **UniSamp**: Sample annotations with the same probability (Alishahi and Phillips 2024). **SenSamp**: Only consider the norm of inputs $\|\boldsymbol{x}_i\|_2$ in the sensitivity (Tolochinksy, Jubran, and Feldman 2022). **GradSamp**: Set the sample probability to the norm of the gradient of parameters. **CrowdCore**: See Algorithm 1.

Extending to Multiple Classes

A one-vs-rest strategy is taken to handle multi-class classification. We specify the label class as positive and the rest of the classes as negative.

For $K > 2$ classes, the complete confusion matrix $\boldsymbol{A}$ is a $K \times K$ matrix. For a sample with label y, the one-vs-rest confusion matrix is $\begin{bmatrix} a & 1-b \\ 1-a & b \end{bmatrix}$, where $a = \boldsymbol{A}_{y,y}$, and $b = 1 - \frac{1}{K-1} \sum_{k \neq y} \boldsymbol{A}_{k,y}$.

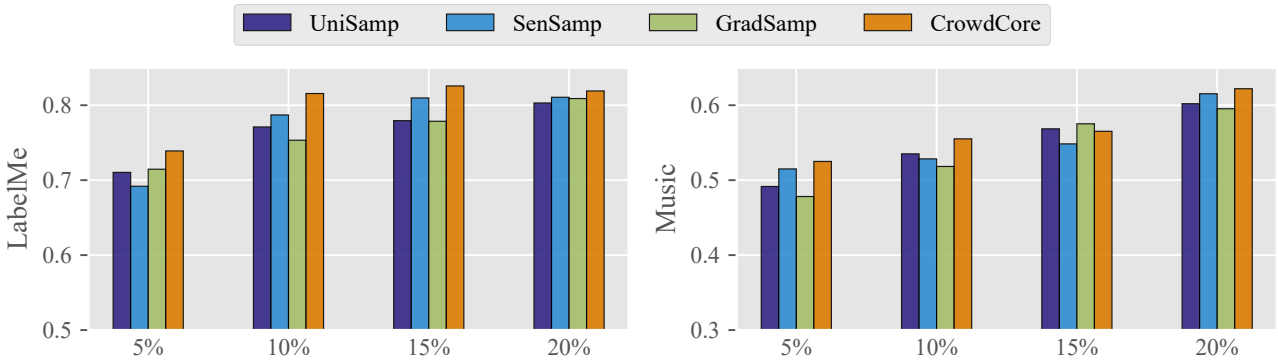

Figure 2: Performance of Classifiers Trained on Real-World Annotations.

Performance on Synthetic Annotations (RQ1)

Settings. We take MNIST (Deng 2012) and MiniBooNE (Aguilar-Arevalo et al. 2009) datasets. The number and dimension of MiniBooNE data points are 70,000 and 784, respectively. The number and dimension of MNIST data points are 130,064 and 50, respectively. To generate noisy annotations, $R = 5$ annotators are simulated. The ground-truth of the confusion layer is generated by $\boldsymbol{A}_r = \text{Normalization}(\boldsymbol{I}+0.2\boldsymbol{M}_{\text{rand}})$, where $\boldsymbol{M}_{\text{rand}}$ is sampled from uniform distribution in range $[0, 1]$.

Results. We run sampling algorithms and report the mean and standard error of the mean absolute approximation error (1e-3) and the running time (in seconds), as shown in Table 2. As the sample size increases, the approximation error for all sampling methods generally decreases. CrowdCore and SenSamp perform better than the other two, implying that sensitivity-based sampling is helpful for preserving dataset information. When the coreset size is pretty small, the CrowdCore performs significantly better than others.

Performance on Human Annotations (RQ2)

Settings. LabelMe (Rodrigues and Pereira 2018) is an open-source dataset containing 59 annotators, 1,000 instances, and 8 classes. The backbone is the VGG-16 network. Music (Rodrigues, Pereira, and Ribeiro 2014) is a music genre classification dataset. The backbone is a 3-layer MLP classifier. Normalization is used to control the feature norm before the last network layer. The training epochs for LabelMe and Music are set to 25 and 120, respectively.

Results. We run sampling algorithms and report the classification accuracy on test data, as shown in Figure 2. CrowdCore consistently maintains the highest performance, demonstrating its effectiveness in finding coresets, whereas GradSamp shows a significant decline in model performance, suggesting that the coreset sizes for this strategy should be larger. The performance gap between CrowdCore and other methods increases at smaller coresets, indicating that it is better at finding more relevant data.

Case Study (RQ3)

We conduct a case study in LabelMe to analyze how annotator identifiability $|a_r + b_r - 1|$ influence coreset selection.

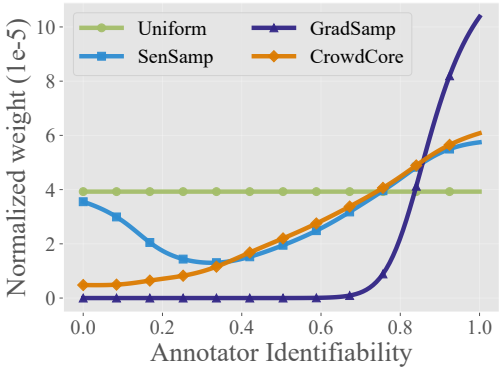

Figure 3: Weight across Sampled Annotations.

We take annotators in the dataset with different confusion properties, including: 1) high-quality with $a_r, b_r \to 1$, 2) systematic-bias with $a_r, b_r \to 0$, and 3) non-identifiable with $a_r + b_r \to 1$. For the former two, identifiability $|a_r + b_r - 1| \to 1$ and for the last one, identifiability $|a_r + b_r - 1| \to 0$. Figure 3 shows the expected value of the normalized weights as a function of $|a_r + b_r - 1|$. We find that CrowdCore shows a clear preference for data labeled by high-quality and systematic-biased annotators, while actively downweights samples from non-identifiable annotators. This result shows that CrowdCore not only selects informative instances but also implicitly performs data denoising, enhancing robustness with different annotators.

Conclusion

In this paper, we focus on the coresets for end-to-end learning from crowdsourced data. By analyzing the CCEM problem with data sensitivity, we provide: (1) It is impossible to find coresets smaller than the complete dataset with confusion layers under the CCEM loss. (2) With the volume regularization term, coresets exist for any input data. (3) After proposing the upper-sensitivity function, we derive a novel sample algorithm called CrowdCore. Experimental validation on synthetic and real datasets confirms that our proposed method can effectively sample better coresets compared to some existing sampling strategies.

Acknowledgments

This work is supported by the Science and Technology Development Fund (FDCT) under Grant number 0029/2023/RIA1, and the AI Super Computing Platform of Macau University of Science and Technology.

References

Aguilar-Arevalo, A.; Anderson, C.; Bartoszek, L.; Bazarko, A.; Brice, S. J.; Brown, B.; Bugel, L.; Cao, J.; Coney, L.; Conrad, J.; et al. 2009. The miniboone detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 599(1): 28–46.

Alishahi, M.; and Phillips, J. M. 2024. No dimensional sampling coresets for classification. In *Proceedings of the 41st International Conference on Machine Learning*, 1008–1049. Vienna, Austria.

Chen, J.; Yang, Q.; Huang, R.; and Ding, H. 2022. Coresets for relational data and the applications. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 434–448. New Orleans, LA, United States.

Chu, Z.; Ma, J.; and Wang, H. 2021. Learning from crowds by modeling common confusions. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 5832–5840. Virtual Conference.

Deng, L. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Fang, Y.; Sun, H.; Chen, P.; and Huai, J. 2018. On the cost complexity of crowdsourcing. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 1531–1537. Stockholm, Sweden.

Feldman, D.; and Langberg, M. 2011. A unified framework for approximating and clustering data. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, 569–578. San Jose, CA, United States.

Feldman, D.; Schmidt, M.; and Sohler, C. 2013. Turning big data into tiny data: constant-size coresets for k-means, PCA and projective clustering. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1434–1453. New Orleans, LA, United States.

Fu, X.; Huang, K.; Sidiropoulos, N. D.; and Ma, W.-K. 2019. Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Processing Magazine*, 36(2): 59–80.

Ibrahim, S.; and Fu, X. 2021. Crowdsourcing via annotator co-occurrence imputation and provable symmetric nonnegative matrix factorization. In *Proceedings of the 38th International Conference on Machine Learning*, 4544–4554. Virtual Conference.

Ibrahim, S.; Nguyen, T.; and Fu, X. 2023. Deep learning from crowdsourced labels: Coupled Cross-Entropy Minimization, identifiability, and regularization. In *Proceedings of the 11th International Conference on Learning Representations*, 1–39. Kigali, Rwanda.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature*, 521(7553): 436–444.

Mirzasoleiman, B.; Bilmes, J.; and Leskovec, J. 2020. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, 6950–6960. Virtual Conference.

Momeni, A.; Rahmani, B.; Scellier, B.; Wright, L. G.; McMahon, P. L.; Wanjura, C. C.; Li, Y.; Skalli, A.; Berloff, N. G.; Onodera, T.; Oguz, I.; Morichetti, F.; del Hougne, P.; Gallo, M. L.; Sebastian, A.; Mirhoseini, A.; Zhang, C.; Markovic, D.; Brunner, D.; Moser, C.; Gigan, S.; Marquardt, F.; Ozcan, A.; Grollier, J.; Liu, A. J.; Psaltis, D.; Alù, A.; and Fleury, R. 2025. Training of physical neural networks. *Nature*, 645(8079): 53–61.

Rodrigues, F.; Pereira, F.; and Ribeiro, B. 2014. Gaussian process classification and active learning with multiple annotators. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, 433–441. Beijing, China.

Rodrigues, F.; and Pereira, F. C. 2018. Deep learning from crowds. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, 1611–1618. New Orleans, LA, United States.

Samadian, A.; Pruhs, K.; Moseley, B.; Im, S.; and Curtin, R. 2020. Unconditional coresets for regularized loss minimization. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 482–492. Palermo, Italy.

Tanno, R.; Saeedi, A.; Sankaranarayanan, S.; Alexander, D. C.; and Silberman, N. 2019. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11236–11245. Long Beach, CA, United States.

Tolochinksy, E.; Jubran, I.; and Feldman, D. 2022. Generic coreset for scalable learning of monotonic kernels: Logistic regression, sigmoid and more. In *Proceedings of the 39th International Conference on Machine Learning*, 21520–21547. Baltimore, MD, United States.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Łukasz Kaiser; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010. Long Beach, CA, United States.

Wang, L.; and Zhou, Z.-H. 2016. Cost-saving effect of crowdsourcing learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2111–2117. New York, NY, United States.

Wei, H.; Xie, R.; Feng, L.; Han, B.; and An, B. 2022. Deep learning from multiple noisy annotators as a union. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12): 10552–10562.