Identifying Dependent Annotators in Crowdsourcing

Panagiotis A. Traganitis⋆ and Georgios B. Giannakis†
⋆ Dept. of ECE, Michigan State University, East Lansing, MI 48824, USA
† Dept. of ECE, University of Minnesota, Minneapolis, MN 55455, USA

2022 56th Asilomar Conference on Signals, Systems, and Computers | 978-1-6654-5906-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/IEEECONF56349.2022.10052052

***Abstract*—Crowdsourcing is the learning paradigm that aims to combine noisy labels provided by a crowd of human annotators. To facilitate this label fusion, most contemporary crowdsourcing methods assume conditional independence between different annotators. Nevertheless, in many cases this assumption may not hold. This work investigates the effects of groups of correlated annotators in multiclass crowdsourced classification. To deal with this setup, a novel approach is developed to identify groups of dependent annotators via second-order moments of annotator responses. This in turn, enables appropriate dependence aware aggregation of annotator responses. Preliminary tests on synthetic and real data showcase the potential of the proposed approach.**

***Index Terms*—Crowdsourcing, Weak supervision, Ensemble Learning, Classification**

I. Introduction

Crowdsourcing has recently enjoyed success in numerous learning and data mining tasks, by using crowds of human annotators to accomplish a given task [1]. In particular, *crowdsourced classification* combines classification labels provided by (possibly unreliable) annotators. Most works on crowdsourcing focus on aggregating noisy annotator labels, to obtain results hopefully close to the ground-truth, by assuming conditional independence across annotators.

The simplest label aggregation method for classification is the majority voting rule, which assigns to a datum the label most annotators agree on. While making no strict model assumptions, this method implicitly assumes that all annotators are of roughly equal ability, yielding reduced performance in various scenaria. Sophisticated methods make use of the conditional independence assumption, advocating probabilistic models of annotators, and estimate parameters that characterize their performance. A popular model in this category is the so-called Dawid-Skene model, that uses the expectation-maximization (EM) algorithm to estimate annotator parameters and the unknown labels [2]. As the aforementioned EM algorithm is susceptible to initialization, methods that estimate parameters through the moments of annotator responses, have been advocated to initialize it [3]–[7]. Recent methods have also been proposed to take into account data dependencies [8]–[10], or detect the presence of spammer annotators in a dataset [11], [12].

Nevertheless, in many cases the conditional independence assumption does not hold, thus challenging classical label aggregation methods, as they have to operate under model misspecification. Current approaches tackling this scenario can handle binary crowdsourced classification [13], or, they are performed during the label aggregation stage [14]. A related line of work, aims to identify adversarial annotators from their responses [15]–[17]. The colluding adversaries scenario can be cast as a problem of identifying dependent annotators. In this work, we develop a novel approach to identify groups of dependent annotators from the second-order moments of their responses, prior to the aggregation stage, and in the multiclass classification setting.

Work in this paper was supported by NSF Grants 2103256, 2126052, 2128593, 2212318, 2220292
Emails: traganit@msu.edu, georgios@umn.edu

Notation. Unless otherwise noted, lowercase bold letters, $\boldsymbol{x}$, denote column vectors, uppercase bold letters, $\mathbf{X}$, represent matrices, and calligraphic uppercase letters, $\mathcal{X}$, stand for sets. The (i,j)th entry of matrix $\mathbf{X}$ is denoted by $[\mathbf{X}]_{ij}$; $\text{vec}(\mathbf{X})$ denotes a vector consisting of the stacked columns of $\mathbf{X}$; and $\circ$ denotes the Hadamard (elementwise) product between two vectors or matrices. The Frobenius and nuclear norms of a matrix $\mathbf{X}$ are denoted by $\|\mathbf{X}\|_F$ and $\|\mathbf{X}\|_*$ respectively. $\Pr$ denotes probability, or the probability mass function; $\sim$ denotes "distributed as;" $^\top$ represents transpose; $\text{card}(\mathcal{A})$ denotes the cardinality, i.e. the number of elements, of set $\mathcal{A}$; $\mathbb{E}[\cdot]$ denotes expectation, and $\mathbb{1}(\mathcal{A})$ is the indicator function for the event $\mathcal{A}$, that takes value 1 when $\mathcal{A}$ occurs, and 0 otherwise.

II. Problem statement and preliminaries

Consider a dataset consisting of N independent and identically distributed (i.i.d.) data $\mathcal{X} := \{\mathbf{x}_n\}_{n=1}^N$ each belonging to one of K possible classes with corresponding labels $\{y_n\}_{n=1}^N$, e.g. $y_n = k$ if $\mathbf{x}_n$ belongs to class k. Class priors are collected in $\boldsymbol{\pi} := [\pi_1, \ldots, \pi_K]^\top = [\Pr(y_n = 1), \ldots, \Pr(y_n = K)]^\top$. M annotators or workers observe $\mathcal{X}$, or subsets of it, and provide noisy estimates of labels, where $g_m(\mathbf{x}_n) \in \{1, \ldots, K\}$ denotes the label assigned to the n-th datum by annotator m. When an annotator does not provide a response for $\mathbf{x}_n$, we encode this by setting $g_m(\mathbf{x}_n) = 0$. Given only the annotator responses $\{g_m(\mathbf{x}_n), m = 1, \ldots, M\}_{n=1}^N$, the *crowdsourced classification* task involves fusing annotator responses to estimate the ground-truth labels, namely $\{\hat{y}_n\}_{n=1}^N$. Note that this is an *unsupervised* problem, as we do not have access to the ground-truth labels $\{y_n\}_{n=1}^N$ or the raw data in $\mathcal{X}$.

A popular probabilistic model for crowdsourced classification, is the so-called Dawid and Skene (DS) model [2]. The DS model posits that given a true label y_n, annotator responses

are independent, that is

$$\Pr\left(g_1(\mathbf{x}_n)=k_1,\ldots,g_M(\mathbf{x}_n)=k_M|y_n=k\right) = \prod_{m=1}^{M}\Pr(g_m(\mathbf{x}_n)=k_m|y_n=k),$$

and consequently an annotator m can be characterized by a $K\times K$ confusion matrix $\mathbf{H}_m$ that has entries $h_{m,k',k} := [\mathbf{H}_m]_{k',k} = \Pr(g_m(\mathbf{x}_n)=k'|y_n=k)$. If annotator confusion matrices $\{\mathbf{H}_m\}_{m=1}^M$ and class priors $\boldsymbol{\pi}$ are known, the label of $\mathbf{x}_n$ can be fused via a maximum a posteriori (MAP) classifier, as

$$\hat{y}_n = \underset{c\in\{1,\ldots,K\}}{\arg\max}\ \log\pi_c + \sum_{m=1}^{M}\log(h_{m,g_m(\mathbf{x}_n),c}), \tag{1}$$

where we used the conditional independence of the annotators. Nevertheless, confusion matrices and priors are not known a priori, and have to be estimated from the available annotator responses, using for instance the methods outlined in [2], [4], [5], [18].

While enabling high quality classification fusion in many crowdsourcing tasks, the DS model does not account for any dependencies that may arise among annotators. The next section will introduce our proposed approach for dealing with annotator dependencies.

III. Identifying groups of annotators

As the plain vanilla DS model does not apply under annotator dependencies, in this work we consider an extended version of it, which has been successfully utilized in [13], [19], [20]. Suppose there exist $J<M$ groups of annotators, and that responses within each annotator group are dependent. In this work, the number of groups J is assumed to be known. Let $\mathcal{M}_j$ denote the set of annotator indices corresponding to group j, and $M_j=\text{card}(\mathcal{M}_j)$ be the cardinality of said group. Under this extended model, dependencies per group j are captured via a latent variable $z_j(\mathbf{x}_n)\in\{1,\ldots,K\}$, conditioned on which the responses of annotators within the group become independent, that is

$$\Pr(\{g_m(\mathbf{x}_n)=k_m\}_{m\in\mathcal{M}_j}|z_j(\mathbf{x}_n)=k) = \prod_{m\in\mathcal{M}_j}\Pr\left(g_m(\mathbf{x}_n)=k_m|z_j(\mathbf{x}_n)=k\right),\ j=1,\ldots,J. \tag{2}$$

The hidden variables $\{z_j(\mathbf{x}_n)\}_{j=1}^L$ are also assumed conditionally independent given the ground-truth label y_n of the datum $\mathbf{x}_n$, that is

$$\Pr\left(z_1(\mathbf{x}_n)=k_1,\ldots,z_J(\mathbf{x}_n)=k_J|y_n=k\right) = \prod_{j=1}^{J}\Pr\left(z_j(\mathbf{x}_n)=k_j|y_n=k\right). \tag{3}$$

Similar to the plain vanilla DS model, we can define priors for hidden variables $\boldsymbol{\theta}_j := [\theta_{j,1},\ldots,\theta_{j,K}]^\top = [\Pr(z_j(\mathbf{x}_n)=1),\ldots,\Pr(z_j(\mathbf{x}_n)=K)]^\top$ for $j=1,\ldots,J$, and confusion matrices for each annotator; thus, if annotator m belongs to group j, the corresponding confusion matrix has entries $[\mathbf{H}_m]_{k',k}=\Pr(g_m(\mathbf{x}_n)=k'|z_j(\mathbf{x}_n)=k)$, while the confusion matrix $\boldsymbol{\Psi}_j$ per hidden variable j, has entries $\psi_{j,k',k} := [\boldsymbol{\Psi}_j]_{k',k} = \Pr\left(z_j(\mathbf{x}_n)=k'|y_n=k\right)$. In this setup, label fusion can be accomplished in two stages [13], [19]: At the first stage, annotator confusion matrices $\{\hat{\mathbf{H}}_m\}_{m\in\mathcal{M}_j}$ and priors for latent variables $\{\hat{\theta}_{j,c}\}_{c=1}^K$ are estimated per group j, via standard crowdsourcing algorithms [2], [5]. Then the responses per group are aggregated and estimates of the hidden variables $\{\hat{z}_j(\mathbf{x}_n)\}_{n=1,j=1}^{N,J}$ are obtained, as

$$\hat{z}_j(\mathbf{x}_n) = \underset{c\in\{1,\ldots,K\}}{\arg\max}\ \log\hat{\theta}_{j,c} + \sum_{m=1}^{M}\log(\hat{h}_{m,g_m(\mathbf{x}_n),c}). \tag{4}$$

At the second stage, latent variable confusion matrices $\{\hat{\boldsymbol{\Psi}}_j\}_{j=1}^J$ and class priors $\hat{\boldsymbol{\pi}}$ are estimated using again standard crowdsourcing algorithms on the latent variables. The final estimates of the labels $\{\hat{y}_n\}_{n=1}^N$, are obtained by aggregating hidden variables,

$$\hat{y}_n = \underset{c\in\{1,\ldots,K\}}{\arg\max}\ \log\hat{\pi}_c + \sum_{j=1}^{J}\log(\hat{\psi}_{j,\hat{z}_j(\mathbf{x}_n),c}). \tag{5}$$

The two step label fusion procedure is listed in Alg. 1. However, for this two-stage procedure to be successful, knowledge of the groupings of annotators is critical.

Algorithm 1 Two-stage label fusion

1: **Input:** Annotator responses $\{g_m(\mathbf{x}_n)\}_{n=1,m=1}^{N,M}$, annotator groups $\{\mathcal{M}_j\}_{j=1}^J$.
2: **Output:** Estimates of data labels $\{\hat{y}_n\}_{n=1}^N$
3: **for** $j=1,\ldots,J$ **do**
4: Estimate $\boldsymbol{\theta}_j, \{\mathbf{H}_m\}_{m\in\mathcal{M}_j}$ using e.g. [2], [5], [6].
5: Estimate hidden labels $\{\hat{z}_j(\mathbf{x}_n)\}_{n=1}^N$ via (4).
6: **end for**
7: Estimate $\boldsymbol{\pi}, \{\boldsymbol{\Psi}_j\}_{j=1}^J$ using e.g. [2], [5], [6].
8: Estimate data labels $\{\hat{y}_n\}_{n=1}^N$ via (5).

The ensuing subsections outline how the second-order moments of annotator responses can be utilized to infer annotator groups. For brevity's sake we describe the case for $J=2$ groups, although the proposed approach can be readily scaled for $J>2$. Further annotators are assumed to be aligned, that is the first M_1 belong to the first group and the rest to the second.

A. Annotator co-occurrence

Consider the $K\times K$ co-occurrence matrix $\mathbf{R}_{m,m'}$ between two annotators $m,m'\neq m$, that has entries $[\mathbf{R}_{m,m'}]_{k,k'} = \Pr\left(g_m(\mathbf{x}_n)=k, g_{m'}(\mathbf{x}_n)=k'\right)$. If annotators m,m' belong to the same group, e.g. $m,m'\in\mathcal{M}_1$, then it can be shown that their co-occurrence matrix is given by [19]

$$\begin{aligned}\mathbf{R}_{m,m'} &= \mathbf{H}_m\text{diag}(\boldsymbol{\Psi}_1\boldsymbol{\pi})\mathbf{H}_{m'}^\top \\ &= \mathbf{H}_m\text{diag}^{1/2}(\boldsymbol{\Psi}_1\boldsymbol{\pi})\text{diag}^{1/2}(\boldsymbol{\Psi}_1\boldsymbol{\pi})\mathbf{H}_{m'}^\top = \mathbf{U}_{1,m}\mathbf{U}_{1,m}^\top,\end{aligned} \tag{6}$$

where we have used the fact that the entries of $\mathbf{\Psi}_1\boldsymbol{\pi}$ are non-negative, and defined $\mathbf{U}_{j,m} := \mathbf{H}_m \text{diag}^{1/2}(\mathbf{\Psi}_j\boldsymbol{\pi})$. Accordingly, the co-occurrence matrix between annotators m, m'' that belong to different groups, e.g. $m \in \mathcal{M}_1, m'' \in \mathcal{M}_2$, is

$$\begin{aligned}\mathbf{R}_{m,m''} &= \mathbf{H}_m\mathbf{\Psi}_1\text{diag}(\boldsymbol{\pi})\mathbf{\Psi}_2\mathbf{H}_{m''}^\top \\ &= \mathbf{H}_m\mathbf{\Psi}_1\text{diag}^{1/2}(\boldsymbol{\pi})\text{diag}^{1/2}(\boldsymbol{\pi})\mathbf{\Psi}_2^\top\mathbf{H}_{m''}^\top \\ &= \mathbf{V}_{1,m}\mathbf{V}_{2,m''}^\top,\end{aligned} \tag{7}$$

with $\mathbf{V}_{j,m} := \mathbf{H}_m\mathbf{\Psi}_j\text{diag}^{1/2}(\boldsymbol{\pi})$. Eq.'s (6) and (7) indicate that co-occurrence matrices between two annotators admit different factorizations when annotators belong to the same group, and when they belong to different ones. Nevertheless, simple algebraic manipulations show that,

$$\begin{aligned}\mathbf{V}_{j,m} &= \mathbf{U}_{j,m}\text{diag}^{-1/2}(\mathbf{\Psi}_\ell\boldsymbol{\pi})\mathbf{\Psi}_j\text{diag}^{1/2}(\boldsymbol{\pi}) \\ &= \mathbf{U}_{j,m}\mathbf{C}_j,\end{aligned} \tag{8}$$

where $\mathbf{C}_j := \text{diag}^{-1/2}(\mathbf{\Psi}_j\boldsymbol{\pi})\mathbf{\Psi}_j\text{diag}^{1/2}(\boldsymbol{\pi})$. Thus, in both cases, the decompositions of the co-occurrence matrix are linearly related. Finally, the co-occurrence of an annotator with itself is given by $\mathbf{R}_{m,m} = \text{diag}(\mathbf{H}_m\mathbf{\Psi}_j\boldsymbol{\pi})$ [19], which cannot be decomposed in a form similar to (6) or (7).

B. Inferring annotator groups

Let $\bar{\mathbf{R}}$ be an $MK \times MK$ block matrix, whose (m, m')-th block is the $K \times K$ matrix $\mathbf{R}_{m,m'}$, i.e. [7]

$$\bar{\mathbf{R}} = \begin{bmatrix} \mathbf{R}_{1,1} & \mathbf{R}_{1,2} & \dots & \mathbf{R}_{1,M} \\ \vdots & & \ddots & \vdots \\ \mathbf{R}_{M,1} & & \dots & \mathbf{R}_{M,M} \end{bmatrix}. \tag{9}$$

Upon defining the $M_1K \times K$ matrices $\bar{\mathbf{U}}_1 := [\mathbf{U}_{1,1}^\top, \dots, \mathbf{U}_{1,M_1}^\top]^\top$, $\bar{\mathbf{V}}_1 := [\mathbf{V}_{1,1}^\top, \dots, \mathbf{V}_{1,M_1}^\top]^\top$, and the $M_2K \times K$ matrices $\bar{\mathbf{U}}_2, \bar{\mathbf{V}}_2$ accordingly, and using (8), $\bar{\mathbf{R}}$ can be written as

$$\bar{\mathbf{R}} = \begin{bmatrix} \bar{\mathbf{U}}_1 & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{U}}_2 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{U}}_1^\top & \mathbf{C}_1^\top\bar{\mathbf{V}}_2^\top \\ \mathbf{C}_2^\top\bar{\mathbf{V}}_1^\top & \bar{\mathbf{U}}_2^\top \end{bmatrix} + \bar{\mathbf{D}} = \mathbf{L} + \bar{\mathbf{D}} \tag{10}$$

where $\mathbf{L} := \begin{bmatrix} \bar{\mathbf{U}}_1 & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{U}}_2 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{U}}_1^\top & \mathbf{C}_1^\top\bar{\mathbf{V}}_2^\top \\ \mathbf{C}_2^\top\bar{\mathbf{V}}_1^\top & \bar{\mathbf{U}}_2^\top \end{bmatrix}$ and $\bar{\mathbf{D}}$ is a block diagonal matrix, with the m-th diagonal block being $\mathbf{R}_{m,m} - \mathbf{U}_{j,m}\mathbf{U}_{j,m}$, for $m \in \mathcal{M}_j$. As $J < K$ the block matrix $\hat{\mathbf{R}}$, exhibits a low-rank plus block-diagonal structure. Also (10) indicates that the low rank matrix $\mathbf{L}$ exhibits structure that is dependent on the groupings of annotators. The form of $\bar{\mathbf{R}}$ suggests a two-step procedure for estimating annotator groups. First, we need to remove the influence of $\bar{\mathbf{D}}$ from $\bar{\mathbf{R}}$. This can be done via robust PCA [21], as $\bar{\mathbf{D}}$ does not adhere to the low rank structure of $\mathbf{L}$. Nevertheless, we have access only to the sample version of (10), $\hat{\bar{\mathbf{R}}}$, which can be readily formed by estimating sample co-occurrence matrices as

$$[\hat{\mathbf{R}}_{m,m'}]_{k',k} = \frac{\sum_{n=1}^N \mathbb{1}(g_m(\mathbf{x}_n) = k, g_{m'}(\mathbf{x}_n) = k')}{N_{m,m'}}, \tag{11}$$

with $N_{m,m'}$ being the number of data that annotators m, m' have both provided responses for. If two annotators do not have any overlap, that is $N_{m,m'} = 0$, we set $\hat{\mathbf{R}}_{m,m'} = \mathbf{0}$. Thus, at the first step we solve

$$\begin{aligned}\{\hat{\mathbf{L}}, \hat{\bar{\mathbf{D}}}\} = &\arg\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\text{vec}(\mathbf{S})\|_1 \\ &\text{subject to } \mathbf{\Omega} \circ \hat{\bar{\mathbf{R}}} = \mathbf{\Omega} \circ (\mathbf{L} + \mathbf{S}),\end{aligned} \tag{12}$$

where $\lambda > 0$ is a tunable regularization constant and $\mathbf{\Omega} \in \{0,1\}^{MK \times MK}$ is a binary mask indicating which entries of $\hat{\bar{\mathbf{R}}}$ are observed. The nuclear norm in (12) promotes low rank solutions for $\mathbf{L}$, whereas the ℓ_1 norm on the entries of $\mathbf{S}$, promotes sparse solutions. $\mathbf{S}$ is used here to capture the block diagonal matrix $\bar{\mathbf{D}}$, and any other spurious correlations in $\hat{\bar{\mathbf{R}}}$ that may arise due to the heterogeneous noise. Note that, (12) is a convex program, therefore it can be solved using off-the-shelf tools [22].

Upon estimating $\mathbf{L}$, its structure is exploited via subspace clustering [23]–[25]. Subspace clustering is used to cluster the columns of $\mathbf{L}$ into groups that correspond to each group of annotators. The proposed algorithm is tabulated in Alg. 2. Having estimated the groups of annotators, dependency-aware label aggregation approaches [19], such as the two-step label fusion in Alg. 1, can be employed.

Algorithm 2 Identifying groups of dependent annotators

1: **Input:** Annotator responses $\{g_m(\mathbf{x}_n)\}_{n=1,m=1}^{N,M}$, K, J.
2: **Output:** Estimated annotator groups $\{\hat{\mathcal{M}}_j\}_{j=1}^J$
3: Estimate co-occurrence matrices $\hat{\mathbf{R}}_{m,m'}$ $\forall m, m'$ from annotator responses, via (11)
4: Collect $\hat{\mathbf{R}}_{m,m'}$ $\forall m, m'$ in block matrix $\hat{\bar{\mathbf{R}}}$.
5: Extract $\hat{\mathbf{L}}$ from $\hat{\bar{\mathbf{R}}}$ via (12).
6: Cluster columns of $\hat{\mathbf{L}}$ via subspace clustering to obtain annotator groups.

IV. Numerical Tests

The performance of the proposed algorithm will be evaluated in this section, in synthetic and real datasets. The label aggregation algorithms considered are majority vote (denoted as *MV*), and the EM algorithm of Dawid-Skene [2] (denoted as *DS*) initialized using MV. *Group-aware DS* and *Group-aware MV*, denote the results from the hierarchical label fusion of Alg. 1 when *DS* and *MV* are used respectively, and annotator groups are estimated via Alg. 2. In all tests, label classification performance is evaluated using accuracy, given by the percentage $1/N\sum_n \mathbb{1}(\{\hat{y}_n = y_n\})$, and the Orthogonal Matching Pursuit subspace clustering algorithm [25] is used in Step 6 of Alg. 2. Experiments were performed using MATLAB [26], and results represent averages over 10 independent Monte Carlo runs.

For the synthetic data tests, N ground-truth labels $\{y_n\}_{n=1}^N$, were generated i.i.d. according to $\boldsymbol{\pi}$, i.e. $y_n \sim \boldsymbol{\pi}$, for $n = 1, \dots, N$. Annotators were then grouped into J groups and $\mathbf{\Psi}_j$ and $\{\mathbf{H}_m\}_{m=1}^M$ were generated at random. Annotator responses are generated as follows: if $y_n = k$, then the hidden variable $z_j(\mathbf{x}_n)$ is generated according to the k-th column

Dataset	MV	DS	Group-aware MV	Group-aware DS
Bluebird	75.92%	87.96%	72.22%($J=7$)	91.6% ($J=7$)
Shuttle	98.82%	97.54%	98.93%($J=3$)	99.46% ($J=3$)
CoverType	75%	48.46%	74.49% ($J=4$)	74.09% ($J=4$)

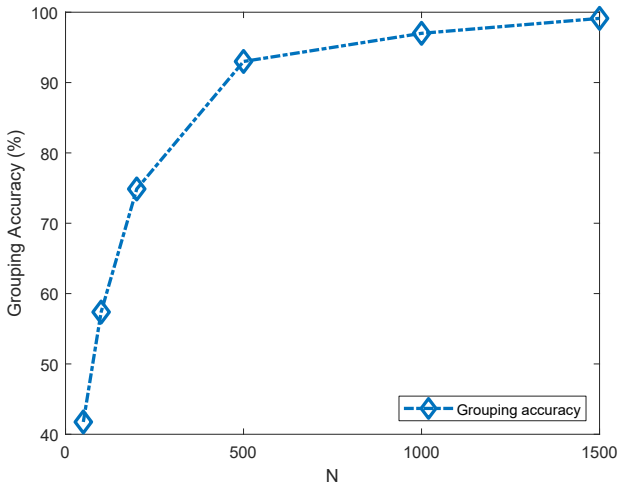
TABLE I: Classification accuracy for real datasets.

of $\mathbf{\Psi}_j$. Finally, if $z_j(\mathbf{x}_n) = k'$ and $m \in \mathcal{M}_j$ $g_m(\mathbf{x}_n)$ is generated according to the k'-th column of $\mathbf{H}_m$. Results for a synthetic dataset with $K = 3$ classes, $M = 40$ synthetically generated annotators, belonging to $J = 4$ groups, with 10 annotators per group, are shown in Fig. 1, as the number of data N increases. Specifically Fig. 1a shows the grouping accuracy, i.e. the percentage of correctly grouped annotators, of Alg. 2. Clearly, as N grows, the sample estimates of co-occurrence matrices become more accurate yielding more accurate groupings. For the same dataset, Fig. 1b depicts the percentage of correctly classified data, when taking the group structure of the annotators into account versus ignoring it, as in *DS* [2] and *MV*. Here we see that even though the number of data increases, the performance of *DS* and *MV* does not improve, with *MV* exhibiting more robust behavior than *DS* since it does not make any explicit conditional independence assumptions. On the other hand, as the grouping accuracy increases, so does the classification accuracy of *Group-aware DS* and *Group-aware MV*, corroborating the claim that annotator dependencies should be taken into account. Another interesting observation is that when the grouping accuracy is low, the two-stage label fusion of Alg. 1 can yield worse results than *DS* or *MV*.

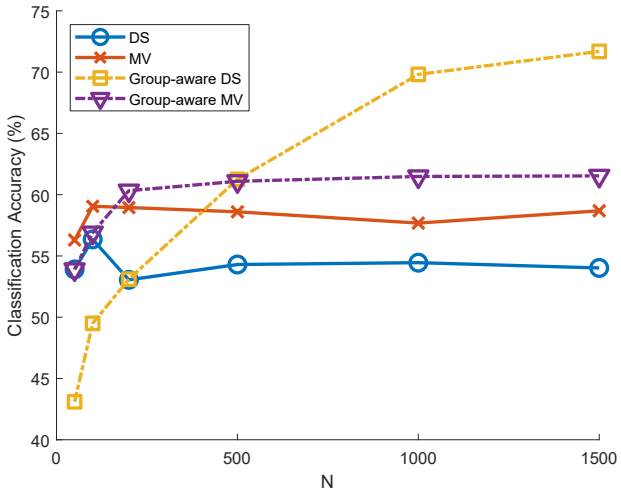
The performance of Alg. 2 is further evaluated on 3 real datasets: the bluebird dataset [27], with $N = 108, M = 39, K = 2$, the Shuttle dataset [28], with $N = 58,000, M = 15, K = 7$, and the CoverType dataset [28] with $N = 581,012, M = 15, K = 7$. Bluebird is a crowdsourcing dataset, where annotators are tasked with classifying images of birds into two classes. Shuttle and CoverType are machine learning datasets from the UCI database. For these datasets, a collection of $M = 15$ classification algorithms from MATLAB act as annotators. These algorithms are trained on randomly selected subsets of the datasets, and then provided labels for the entire dataset. Table I shows the classification accuracy for the 3 real datasets. For *Group-aware DS* and *Group-aware MV* the parentheses indicate the number of groups J that yielded the specific result. The real data results show similar trends to the synthetic data ones, namely improved performance when consider annotator groups for the crowdsourced classification task.

V. Conclusions

This contribution showcased an algorithm for identifying groups of dependent annotators in crowdsourcing, using the moments of annotator responses. Simulated tests in synthetic and real datasets showcased the effectiveness of the proposed method, and the importance of incorporating annotator dependencies in the crowdsourcing task.



(a) Grouping accuracy



(b) Classification accuracy

Fig. 1: Simulated tests on a synthetic dataset with $K = 3$ classes, $M = 40$ annotators and $J = 4$ groups.

Future work involves tailoring the clustering methods for the task at hand, developing methods for identifying the number of groups J, utilizing more general dependency models, as well as performance analysis of the proposed method.

REFERENCES

[1] J. Howe, "The rise of crowdsourcing," *Wired Magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[2] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Applied Statistics*, pp. 20–28, 1979.

[3] A. Jaffe, B. Nadler, and Y. Kluger, "Estimating the accuracies of multiple classifiers without labeled data." in *AISTATS*, vol. 2, San Diego, CA, 2015, p. 4.

[4] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan, "Spectral methods meet EM: A provably optimal algorithm for crowdsourcing," in *Advances in Neural Information Processing Systems*, 2014, pp. 1260–1268.

[5] P. A. Traganitis, A. Pagès-Zamora, and G. B. Giannakis, "Blind multiclass ensemble classification," *IEEE Transactions on Signal Processing*, vol. 66, no. 18, pp. 4737–4752, Sept 2018.

[6] S. Ibrahim, X. Fu, N. Kargas, and K. Huang, "Crowdsourcing via pairwise co-occurrences: Identifiability and algorithms," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d´ Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 7847–7857.

[7] S. Ibrahim and X. Fu, "Crowdsourcing via annotator co-occurrence imputation and provable symmetric nonnegative matrix factorization," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 4544–4554. [Online]. Available: https://proceedings.mlr.press/v139/ibrahim21a.html

[8] A. T. Nguyen, B. Wallace, J. J. Li, A. Nenkova, and M. Lease, "Aggregating and predicting sequence labels from crowd annotations," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 299–309.

[9] P. Ruiz, P. Morales-Álvarez, R. Molina, and A. Katsaggelos, "Learning from crowds with variational Gaussian processes," *Pattern Recognition*, vol. 88, pp. 298–311, April 2019. [Online]. Available: http://decsai.ugr.es/vip/files/journals/1-s2.0-S0031320318304060-main.pdf

[10] P. A. Traganitis and G. B. Giannakis, "Unsupervised ensemble classification with sequential and networked data," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

[11] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *Journal of Machine Learning Research*, vol. 13, no. 16, pp. 491–518, 2012.

[12] P. A. Traganitis and G. B. Giannakis, "Identifying spammers to boost crowdsourced classification," in *46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[13] A. Jaffe, E. Fetaya, B. Nadler, T. Jiang, and Y. Kluger, "Unsupervised ensemble learning with dependent classifiers," in *Artificial Intelligence and Statistics*, 2016, pp. 351–360.

[14] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi, "Community-based bayesian aggregation models for crowdsourcing," in *Proceedings of the 23rd International Conference on World Wide Web*, ser. WWW '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 155–164. [Online]. Available: https://doi.org/10.1145/2566486.2567989

[15] S. Jagabathula, L. Subramanian, and A. Venkataraman, "Identifying unreliable and adversarial workers in crowdsourced labeling tasks," *Journal of Machine Learning Research*, vol. 18, no. 93, pp. 1–67, 2017.

[16] Q. Ma and A. Olshevsky, "Adversarial crowdsourcing through robust rank-one matrix completion," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 21 841–21 852.

[17] P. A. Traganitis and G. B. Giannakis, "Detecting adversaries in crowdsourcing," in *2021 IEEE International Conference on Data Mining (ICDM)*, 2021, pp. 1373–1378.

[18] S. Ibrahim, X. Fu, N. Kargas, and K. Huang, "Crowdsourcing via pairwise co-occurrences: Identifiability and algorithms," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019, pp. 7847–7857.

[19] P. A. Traganitis and G. B. Giannakis, "Blind multi-class ensemble learning with dependent classifiers," in *Proc. of the 26th European Signal Processing Conference*, Rome, Italy, Sep 2018.

[20] H. Chen, B. Chen, and P. K. Varshney, "A new framework for distributed detection with conditionally dependent observations," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1409–1419, March 2012.

[21] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, Jun. 2011.

[22] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx.

[23] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Process. Magazine*, vol. 28, no. 2, pp. 52–68, 2010.

[24] P. A. Traganitis and G. B. Giannakis, "Sketched subspace clustering," *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1663–1675, 2018.

[25] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, "Greedy feature selection for subspace clustering." *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2487–2517, 2013.

[26] MATLAB, *version 9.9.0 (R2020b)*. Natick, Massachusetts: The MathWorks Inc., 2020.

[27] P. Welinder, S. Branson, P. Perona, and S. J. Belongie, "The multidimensional wisdom of crowds," in *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 2010, pp. 2424–2432.

[28] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml