Labeling Sequential Data from Noisy Annotations

Tim Marrinan
Pacific Northwest National Laboratory
timothy.marrinan@pnnl.gov

Shahana Ibrahim
University of Central Florida
shahana.ibrahim@ucf.edu

Xiao Fu
Oregon State University
xiao.fu@oregonstate.edu

***Abstract*—Crowdsourcing algorithms often work under the assumption that the data samples are independent. Recent work has shown that data dependence, such as temporal correlations in sequential data, can be leveraged to improve the label quality. Existing methods that exploit this special structure rely on third-order statistics of the annotator outputs to ensure the identifiability of key latent parameters, which are costly to acquire. This work proposes an approach for integrating crowdsourced annotations under the Dawid-Skene/Hidden Markov Model (DS-HMM) for sequential data based on second-order statistics, which naturally enjoys a lower sample complexity. An effective algorithm is proposed to tackle the challenging optimization problem associated with the proposed estimator. Numerical experiments showcase the effectiveness of the data labeling paradigm.**

I. Introduction

In machine learning (ML), there is an insatiable demand for labeled training data. However, acquiring accurately labeled samples at scale is highly nontrivial. Sophistication and domain expertise from multiple annotators are often required. It is common to employ a crowd of annotators to label large datasets. Consequently, some annotators may produce inaccurate labels. *Crowdsourcing* techniques aim to integrate such noisy labels to produce more accurate annotations.

The naïve approach to integrating multiple labels in crowdsourcing is *majority voting* (MV), whose performance is limited in practical scenarios [1], [2], [3]. A more principled approach to label integration was introduced by Dawid and Skene who postulated that each annotator may have an implicit fixed probability of providing label i when shown a sample from class j [4]. Consequently Dawid-Skene (DS) model assumes a *confusion matrix* for each annotator whose entries correspond to the probabilities of the correct and incorrect annotations conditioned on the true labels. This model has led to a plethora of label integration algorithms in crowdsourcing, including iterative approaches based on expectation maximization (EM) [4], [5], [6], [7], [8], spectral approaches based on annotator's label statistics [9], [10], [11], and many developments using tensor and matrix factorization techniques [3], [2], [12], [13].

Most DS model-based approaches assume that the samples are independent, and leave potential relationships between samples unexploited. When the samples to be labeled are sequential in nature, such as frames of a video sequence or words in a sentence, such temporal dependence provides additional structural information. The work in [13] has shown that by assuming that the true labels correspond to hidden states in a Hidden Markov Model (HMM), this sequential structure can be utilized for improved label estimation. Their method is a two-stage procedure. The method combines the tensor-based moment-fitting algorithm using third-order statistics of annotator outputs [3], coupled with an EM-based refinement in order to learn the associated model parameters.

In this work, we revisit the DS model-based HMM presented in [13] for integrating crowdsourced noisy annotations for sequential data. Contrary to the existing tensor-based approach that utilizes third-order annotation statistics, we present a novel alternative that employs only second-order statistics, reducing sample complexity. In addition, we propose a *volume-minimization-based coupled nonnegative tri-factorization* (VolMinCTF) criterion to learn the relevant parameters in a single step, which avoids potential error propagation in multi-stage approaches. Furthermore, we design an efficient algorithm for handling the proposed VolMinCTF criterion with convergence guarantees. Simulations showcase the effectiveness of our approach.

II. Background

Consider a collection of T data samples, $\{\mathbf{x}_t \in \mathbb{R}^d\}_{t=1}^T$, such that each data sample belongs to one of the K classes with corresponding labels $\{y_t \in \{1, 2, \ldots, K\}\}_{t=1}^T$. That is, $y_t = k$ if $\mathbf{x}_t$ is a member of class k. Let us denote the label of tth data sample by the annotator m for $m = 1, \ldots, M$ by $f_m(\mathbf{x}_t)$. The output can be regarded as a discrete random variable whose alphabet is $\{1, 2, \ldots, K\}$. Note that $f_m(\mathbf{x}_t) \neq y_t$ often happens. Assume that sample t is assigned to a subset of annotators $\mathcal{N}_t \subseteq \{1, 2, \ldots, M\}$. Given the collection of annotator responses, $\{f_i(\mathbf{x}_t) \ : \ i \in \mathcal{N}_t\}_{t=1}^T$, the goal of a crowdsourcing algorithm is to estimate a label for each sample, $\hat{y}_t$, such that the estimate recovers the ground-truth label y_t.

Following the DS model [4], the annotators' ability to label samples can be modeled using the so-called *confusion matrix*. Suppose that annotator m assigns a label j for any sample X from class k with an unknown but fixed probability. The nonnegative matrix, $\mathbf{A}_m \in \mathbb{R}_+^{K\times K}$, that collects these probabilities as

$$\mathbf{A}_m(j, k) \triangleq \Pr\left(f_m(X) = j \mid Y = k\right) \tag{1}$$

is referred to as the confusion matrix of annotator m. Note that we have $\mathbf{A}_m^\top \mathbf{1} = \mathbf{1}, \ \mathbf{A}_m \geq \mathbf{0}$ where $\mathbf{1} \in \mathbb{R}^K$ is an all-one vector.

In line with the Dawid-Skene (DS) model [4], let us assume that annotator responses to a data sample X are independent when conditioned on the true label Y. That is,

$$\begin{aligned}&\Pr\left(f_1(X) = j_1, f_2(X) = j_2, \ldots, f_M(X) = j_M\right)\\ &\quad = \sum_{k=1}^{K} \prod_{m=1}^{M} \Pr\left(f_m(X) = j_m \mid Y = k\right) \Pr\left(Y = k\right).\end{aligned} \tag{2}$$

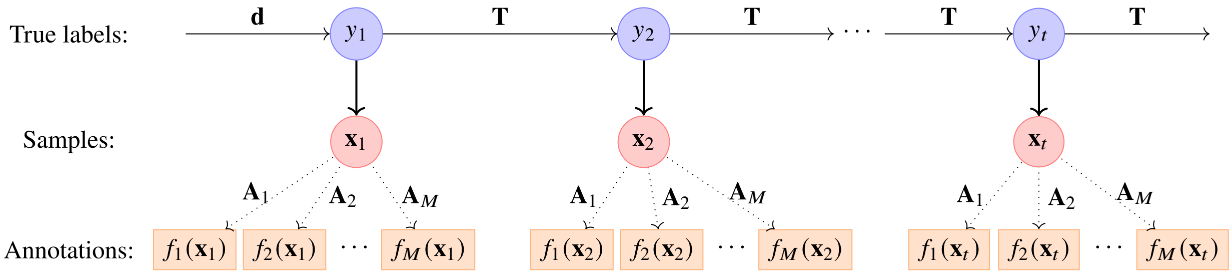

Fig. 1: Illustration of a Hidden Markov Model, $\{(\mathbf{x}_t, y_t)\}_{t=1}^T$, where the samples are the observed states and the true labels correspond to the hidden latent states. The label prior is $\mathbf{d}$ and the transition probabilities are specified by $\mathbf{T}$. The annotator responses, $f_m(\mathbf{X})$, are generated with probabilities defined by the confusion matrices, $\mathbf{A}_m$ for $m = 1, \ldots, M$.

Let us denote the vector of prior probabilities by $\mathbf{d}(k) = \Pr(Y = k)$. The label integration task under the DS-model amounts to identifying the M confusion matrices, $\mathbf{A}_m$, and the K-dimensional vector of prior probabilities, $\mathbf{d}$. After learning these model parameters, the label of each sample can be predicted by constructing a maximum *a posteriori* (MAP) estimator.

Following (2), the joint probability of the responses from any two annotators m and n can be modeled as $\mathbf{R}_{m,n}(i,j) \triangleq \Pr(f_m(\mathbf{X}) = i, f_n(\mathbf{X}) = j), i, j \in \{1, \ldots, K\}$ where

$$\mathbf{R}_{m,n} \triangleq \mathbf{A}_m \operatorname{diag}(\mathbf{d}) \mathbf{A}_n^\top \tag{3}$$

is a $K \times K$ *co-occurrence matrix*; also see [2], [12].

A. Dependent data

In many cases, data samples may exhibit temporal dependencies, i.e., the probability of the tth sample, $\mathbf{x}_t$, having label $y_t = k$ depends on the value of y_{t-1}. This might occur in tasks that involve sequential data such as annotating frames of a video sequence or part-of-speech labeling in natural language processing. In this work, we revisit the DS-model based hidden Markov model (DS-HMM) presented in [13], for such sequential data labeling paradigm. To be specific, the model assumes that the sequence of labels $y_1, y_2, \ldots, y_T$ forms a stationary, time-homogeneous, discrete-time Markov chain, with transition matrix $\mathbf{T} \in \mathbb{R}^{K \times K}$ and the corresponding transition probabilities $\mathbf{T}(j,k) = \Pr(y_t = j \mid y_{t-1} = k)$. Here, the sequence $\{(\mathbf{x}_t, y_t)\}_{t=1}^T$ follows a hidden Markov model (HMM) where the data samples $\mathbf{x}_t$'s are the observed states and the true labels y_t's correspond to hidden latent states—see Fig. 1. The model was shown useful in dealing with sequential data [13]. Nonetheless, the approach in [13] needs third-order statistics of the annotator outputs, which can be sample-costly. The work in [13] also employs two-stage algorithms, yet such approaches may suffer from error propagation between the stages. In this work, we propose a one-stage alternative using sample-efficient second-order statistics.

III. Proposed Approach

A. Pairwise Statistics-based Modeling

Consider the joint probability that annotator m returns label i for sample $\mathbf{x}_t$ after returning label l for sample $\mathbf{x}_{t-1}$. Let us assume that these joint probabilities form the elements of the *consecutive-step co-occurrence matrices* (CSCO) defined as

$$\mathbf{C}_{m,n}^{(t,t-1)}(i,l) \triangleq \Pr(f_m(\mathbf{x}_t) = i, f_n(\mathbf{x}_{t-1}) = l), \tag{4}$$

as the sequential analog of the co-occurrence matrices from (3). Since the Markov chain is stationary and time-homogeneous, these joint probabilities do not depend on the value of t. Hence, we drop the superscript t in the notations for the sequel. According to Bayes rule and the definitions of the confusion matrices $\mathbf{A}_m$, the prior vector $\mathbf{d}$, and the transition matrix $\mathbf{T}$, by letting $\boldsymbol{\Theta} \triangleq \mathbf{T} \operatorname{diag}(\mathbf{d})$, the CSCO matrices can be written as follows:

$$\mathbf{C}_{m,n}^{(t,t-1)} = \mathbf{A}_m \boldsymbol{\Theta} \mathbf{A}_n^\top. \tag{5}$$

If all CSCO matrices $\mathbf{C}_{m,n}^{(t,t-1)}, \forall m, n \in \{1, \ldots, M\}$ are available (in practice, $\mathbf{C}_{m,n}^{(t,t-1)}$ are estimated using consecutive samples co-labeled by annotators m and n), one can construct an augmented co-occurrence matrix of size $MK \times MK$ as follows:

$$\begin{aligned}\mathbf{C}^{(t,t-1)} &= \begin{bmatrix} \mathbf{C}_{1,1}^{(t,t-1)} & \mathbf{C}_{1,2}^{(t,t-1)} & \cdots & \mathbf{C}_{1,M}^{(t,t-1)} \\ \mathbf{C}_{2,1}^{(t,t-1)} & \mathbf{C}_{2,2}^{(t,t-1)} & \cdots & \mathbf{C}_{2,M}^{(t,t-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{M,1}^{(t,t-1)} & \mathbf{C}_{M,2}^{(t,t-1)} & \cdots & \mathbf{C}_{M,M}^{(t,t-1)} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_1^\top & \mathbf{A}_2^\top & \cdots & \mathbf{A}_M^\top \end{bmatrix}^\top \boldsymbol{\Theta} \begin{bmatrix} \mathbf{A}_1^\top & \mathbf{A}_2^\top & \cdots & \mathbf{A}_M^\top \end{bmatrix}. \end{aligned} \tag{6}$$

Hence, the label integration task for sequential data boils down to learning a nonnegative tri-factorization model under DS-HMM.

B. Proposed Algorithm: VolMinCTF

Our goal is to estimate the DS-HMM parameters, $\mathbf{A}_m$ for all m and $\boldsymbol{\Theta}$, using the relation (6). Towards this task, we propose a volume-minimization-based coupled matrix tri-factorization (VolMinCTF) to jointly estimate the model parameters with or without completely observing $\mathbf{C}$. Let $\boldsymbol{\Omega} \subseteq \{1, \ldots, M\} \times \{1, \ldots, M\}$ be a subset of the blocks in the augmented CSCO matrix $\mathbf{C}$, where $(m,n) \in \boldsymbol{\Omega}$, if $\mathbf{C}_{m,n}$ has been observed. Since each $\mathbf{C}_{m,n}$ describes a joint probability distribution—see (4))—we use the Kullback–Leibler (KL) divergence as the fitting term in our objective function. Consequently, we propose the following coupled factorization criterion:

$$\begin{aligned}\min_{\boldsymbol{\Theta},\{\mathbf{A}_m\}} \quad & \sum_{(m,n)\in\boldsymbol{\Omega}} D_{KL}(\mathbf{C}_{m,n} \mid \mathbf{A}_m\boldsymbol{\Theta}\mathbf{A}_n^\top) + \lambda \log\det(\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \delta\mathbf{I}) \\ & \text{s.t. } \mathbf{A}_m \geq \mathbf{0}, \mathbf{1}^\top\mathbf{A}_m = \mathbf{1}^\top, \text{ for all } m = 1,\ldots,M, \\ & \quad \mathbf{1}^\top\boldsymbol{\Theta}\mathbf{1} = 1, \end{aligned} \tag{7}$$

where $D_{KL}(\boldsymbol{X}|\boldsymbol{Y}) \triangleq \sum_{i,j} x_{ij} \log \frac{x_{ij}}{y_{ij}}$ denotes the KL divergence, $\text{vol}(\boldsymbol{\Theta}) = \log\det(\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \delta\mathbf{I})$ is a volume regularizer [14], and $\lambda \geq 0$ is the regularization hyper-parameter. The volume regularizer helps to find the minimum-volume data-enclosing simplex that leads to identifying latent factors of the nonnegtive trifactorization model (6)—in [15], [16], [14], [17], [18], [19], [20]. Note that the constraints are used to respect the probability simplex constraints on the columns of $\mathbf{A}_m$, $\mathbf{T}$, and $\mathbf{d}$.

Since (7) is not convex, we propose to solve it inexactly by minimizing a convex upper-bound of the cost function using the block successive upperbound minimization (BSUM) framework [19], [21], [22]. The majorizer is computed in two steps. First, we find a convex majorizer for the data-fidelity term, i.e., the term that sums all KL divergences. Let τ index the iterations. At step τ, suppose that we have an estimate of $\{\mathbf{A}_m^{(\tau)}\}_{m=1}^M$ and $\boldsymbol{\Theta}^{(\tau)}$ from the previous iteration. Define

$$v_{m,n,i,j}^{(\tau)}(r,s) \triangleq \frac{\mathbf{A}_m^{(\tau)}(i,r)\boldsymbol{\Theta}^{(\tau)}(r,s)\mathbf{A}_n^{(\tau)}(j,s)}{\sum_{r,s=1}^K \mathbf{A}_m^{(\tau)}(i,r)\boldsymbol{\Theta}^{(\tau)}(r,s)\mathbf{A}_n^{(\tau)}(j,s)}, \tag{8}$$

so that $\sum_{r,s=1}^K v_{m,n,i,j}^{(\tau)}(r,s) = 1$, and let

$$x(r,s) \triangleq \frac{\mathbf{A}_m(i,r)\boldsymbol{\Theta}(r,s)\mathbf{A}_n(j,s)}{v_{m,n,i,j}^{(\tau)}(r,s)}. \tag{9}$$

With these definitions, one can represent each summand in the KL divergence term in (7) as a scalar function defined with respect to an element of the unknown groundtruth second-order statistics, i.e., $d_{KL}(\cdot|\cdot) : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$, such that

$$\begin{aligned} & d_{KL}\left(\mathbf{C}_{m,n}(i,j) \mid \sum_{r,s=1}^K v_{m,n,i,j}^{(\tau)}(r,s)x(r,s)\right) \\ &= \mathbf{C}_{m,n}(i,j)\log\mathbf{C}_{m,n}(i,j) \\ &- \mathbf{C}_{m,n}(i,j)\log\left(\sum_{r,s=1}^K \mathbf{A}_m(i,r)\boldsymbol{\Theta}(r,s)\mathbf{A}_n(j,s)\right). \end{aligned} \tag{10}$$

The negative logarithmic function is convex. Hence, by Jensen's inequality, we can compute a locally tight convex majorizer:

$$\begin{aligned} & \sum_{r,s=1}^K v_{m,n,i,j}^{(\tau)}(r,s) d_{KL}\left(\mathbf{C}_{m,n}(i,j) \mid x(r,s)\right) \\ &= \sum_{r,s=1}^K v_{m,n,i,j}^{(\tau)}(r,s)\mathbf{C}_{m,n}(i,j)\log v_{m,n,i,j}^{(\tau)}(r,s)\mathbf{C}_{m,n}(i,j) \\ &- v_{m,n,i,j}^{(\tau)}(r,s)\mathbf{C}_{m,n}(i,j)\log\mathbf{A}_m(i,r) \\ &- v_{m,n,i,j}^{(\tau)}(r,s)\mathbf{C}_{m,n}(i,j)\log\boldsymbol{\Theta}(r,s) \\ &- v_{m,n,i,j}^{(\tau)}(r,s)\mathbf{C}_{m,n}(i,j)\log\mathbf{A}_n(j,s). \end{aligned} \tag{11}$$

At iteration τ, the first term on the R.H.S. of (11) is constant with respect to the optimization variables. Hence, we can construct a locally tight convex majorizer for the data-fidelity term of the objective function (7), that is separable in $\mathbf{A}_m, \mathbf{A}_n$, and $\boldsymbol{\Theta}$.

The volume-regularizer $\log\det(\boldsymbol{\Theta}^\top\boldsymbol{\Theta}+\delta\mathbf{I})$ is concave. Hence, we can find a convex surrogate via the Taylor expansion. At step τ, let $F^{(\tau)} = (\boldsymbol{\Theta}^{(\tau)\top}\boldsymbol{\Theta}^{(\tau)} + \delta\mathbf{I})^{-1}$. Then we obtain

$$\log\det(\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \delta\mathbf{I}) \leq \text{Tr}(F^{(\tau)}\boldsymbol{\Theta}^\top\boldsymbol{\Theta}) + \log\det((F^{(\tau)})^{-1}) - K, \tag{12}$$

where equality holds when $\boldsymbol{\Theta} = \boldsymbol{\Theta}^{(\tau)}$—see the detailed derivation in [14]. The last two terms on the R.H.S. of (12) are constant with respect to the optimization variables. Hence, by combining (11) and (12), we can majorize (7) as follows

$$\min_{\{\mathbf{A}_m\}_{m=1}^M,\boldsymbol{\Theta}} g(\{\mathbf{A}_m\}_{m=1}^M) + h(\boldsymbol{\Theta}) + \beta \tag{13}$$

where g, h, and β collect terms with $\{\mathbf{A}_m\}$, $\boldsymbol{\Theta}$, and constants, respectively.

Update for $\mathbf{A}_m$: The update of $\mathbf{A}_m$ for $m = 1,\ldots,M$ in (13) can be computed in closed form, which is obtained via setting the partial gradient w.r.t. each element of $\mathbf{A}_m$ to zero:

$$\begin{aligned} \mathbf{A}_p^{(\tau+1)}(q,w) = \sum_{n \text{ s.t. } (p,n)\in\boldsymbol{\Omega}} \sum_{j,s=1}^K & \frac{v_{p,n,q,j}^{(\tau)}(w,s)\mathbf{C}_{p,n}(q,j)}{\mu_p} \\ & + \frac{v_{n,p,j,q}^{(\tau)}(s,w)\mathbf{C}_{n,p}(j,q)}{\mu_p}, \end{aligned} \tag{14}$$

where the μ_p is chosen such that $\mathbf{1}^\top\mathbf{A}_p^{(\tau)} = \mathbf{1}^\top$. Note that the indices in the first and second terms of (14) are reversed.

Update for $\boldsymbol{\Theta}$: The update for $\boldsymbol{\Theta}$ cannot be computed in closed form, so we use a lightweight first-order method to minimize $h(\boldsymbol{\Theta})$. Let $\tilde{\boldsymbol{\Theta}}^{(0)} = \boldsymbol{\Theta}^{(\tau)}$. Then the sequence of iterates

$$\tilde{\boldsymbol{\Theta}}^{(u+1)} = \text{Proj}_{\boldsymbol{\Phi}}(\tilde{\boldsymbol{\Theta}}^{(u)} - \alpha^{(u)}\nabla h(\tilde{\boldsymbol{\Theta}}^{(u)})) \tag{15}$$

converges, where u denotes the iteration index, $\boldsymbol{\Phi} = \{\mathbf{X} \in \mathbb{R}^{K\times K} : \mathbf{X} \geq \mathbf{0},\ \mathbf{1}^\top\mathbf{X}\mathbf{1} = 1\}$ and $\alpha^{(u)} > 0$ is step-size parameter of the form $\alpha^{(u)} = a/u$ for some constant $a > 0$ [23]. Projection operation $\text{Proj}_{\boldsymbol{\Phi}}(\boldsymbol{\Theta})$ onto the feasible set $\boldsymbol{\Phi}$ can be accomplished easily by projecting a vectorized version of the argument $\mathbf{\Theta}$ onto the (K^2-1)-dimensional unit simplex defined by the convex hull of the canonical basis vectors, $\text{conv}\{\mathbf{e}_1,\ldots,\mathbf{e}_{K^2}\}$. Finally, the optimization variable $\boldsymbol{\Theta}$ is updated with the converged value $\tilde{\boldsymbol{\Theta}}^*$ of the sequence of iterates (15), i.e., $\boldsymbol{\Theta}^{(\tau+1)} = \tilde{\boldsymbol{\Theta}}^*$.

BSUM Algorithm: It is straightforward to check that the objective function in (13) satisfies the assumptions of the BSUM method [21], and thus by alternating between updates of $\{\mathbf{A}_m^{(\tau)}\}_{m=1}^M$ and $\boldsymbol{\Theta}^{(\tau)}$ (with all other variables are held fixed) the sequence of iterates will produce a stationary point of (7).

Remark: The work in [19] deals with a similar tri-factorization problem in the context of topic modeling. However, their method does not consider missing co-occurrences and also adopts a computationally costly, second-order optimization procedure for handling $\boldsymbol{\Theta}$ subproblem.

IV. Numerical Evaluation

Settings: To demonstrate the effectiveness of the proposed method, we consider a part-of-speech (POS) tagging experiment, where $M = 10$ classifiers were trained using the Natural Language Toolkit (NLTK) [24] on subsets of the Brown coprus [25] to provide POS tags of text[1]. The number of tags (classes) is $K = 12$ and the classifiers provided POS tags for all words in the Penn Treebank corpus [26], which contains $T = 1,00,676$ words in $3,914$ sentences. The data is randomly divided into 10 sets of roughly 391 sentences for cross-validation, which amounts to approximately 10k words per set. In each trial, the validation set is used to tune hyper-parameters, while the remaining 90% of the data is used to for training/testing (since the problem is unsupervised). The reported results are the average of the 10-fold cross-validation. Each sample is only labeled by two annotators and each annotator co-labels with only two others.

Baselines: To highlight how temporal dependency affects the results, the sequential methods have also been compared with methods that assume i.i.d. samples. For all methods, the estimated confusion matrices, label prior, and transition matrix (in the case of sequential methods) are provided as input along with the sentence indices and observed annotator label sequences to the Viterbi algorithm, which returns a MAP prediction of the label sequence $\{\hat{y}_t\}_{t=1}^{T}$ [27]. As a baseline, we compute the oracle PMFs by assuming either i.i.d. or sequential samples. These two methods denoted as `MAP-oracle(i.i.d.)` and `MAP-oracle(sequential)`, respectively, represent the best possible MAP predictions for the available annotations. The i.i.d. methods considered include the `MomentMatching` [3], the `MultiSPA` method [2], and the proposed `VolMinCTF` initialized by the `MultiSPA` algorithm. The i.i.d. version of the proposed method discards the transition matrix before running the Viterbi algorithm. Including this comparison should provide intuition as to whether the estimate of the transition matrix is reasonable, if the data follows the Markov chain assumption. For sequential methods, we include the two techniques from the work [13], denoted as `AO-ADMM` and `AO-ADMM+EM`. We also include the proposed method with `MultiSPA` initialization, and a version where the output is further refined using the EM method via the Baum-Welch algorithm [28].

Results: Results from the 10-fold cross-validation experiment are shown in Table I. Notably, the difference between the oracle PMFs when assuming i.i.d. samples or sequential samples (i.e., `MAP-oracle(i.i.d)` versus `MAP-oracle(sequential)`) is small, suggesting that the assumption of temporal dependence is not strongly supported for this dataset. Nonetheless, we see that the techniques that utilize the proposed VolMinCTF are robust to this potential model-mismatch and out-perform all other methods. In particular, the VolMinCTF followed by EM refinement (denoted as `VolMinCTF+EM`) provides the best label predictions, with an accuracy only 5% below the sequential oracle method `MAP-oracle(sequential)`. In addition, one can note that all three techniques based on the VolMinCTF finish in the top three positions in every category. Recall that we employ only two annotators to label each sample and each annotator only labels with two others in this simulation. Hence, these results illustrate that the proposed approaches are well-suited for cases where annotations are limited for each data sample. Other baselines does not perform well under this challenging scenario. For example, the `MultiSPA` method performs poorly, possibly due to the small number of annotators co-labeling, which leads to low-quality estimates of pairwise co-occurrence matrices. Despite the low-accuracy of these estimates, the `MultiSPA` method still provides a reasonable initialization for `VolMinCTF`. Finally, as expected, the `MomentMatching`, `AO-ADMM`, and `AO-ADMM+EM` methods all suffer from the lack third-order statistics [13], even with the EM refinement.

TABLE I: Results of 10-fold cross validation on the POS labeling experiment where 10 annotators each co-label with 2 other annotators, and only 2 annotators label each sample.

Method	Accuracy	F_1	Precision	Recall
`MAP-oracle(i.i.d.)`	0.7797	0.7675	0.8424	0.7060
`MomentMatching`	0.5052	0.4626	0.5148	0.4216
`MultiSPA`	0.1530	0.1513	0.1636	0.1423
`VolMinCTF(i.i.d.)`	0.6963	0.5802	0.6108	0.5531
`MAP-oracle(sequentia`	0.7844	0.7707	0.8189	0.7281
`AO-ADMM`	0.4530	0.4212	0.4522	0.3946
`AO-ADMM+EM`	0.4834	0.4535	0.4859	0.4263
`VolMinCTF`	0.6973	0.5763	0.5965	0.5580
`VolMinCTF+EM`	0.7306	0.6511	0.6796	0.6257

V. Conclusion

Integrating noisy crowdsourced annotations to produce high-quality labels is a crucial bottleneck in training large-scale supervised machine learning algorithms. In this work we proposed a crowdsourcing technique using the DS-HMM paradigm to exploit temporal dependence in sequential data. The VolMinCTF is a one-stage approach that provides high-quality sample-efficient MAP estimates of the labels, relying only on second-order statistics. The numerical evaluations provide a proof-of-concept and demonstrate that the proposed method can outperform existing techniques. The VolMinCTF has the added benefit of being able to recover the latent parameters of the DS-HMM with incomplete knowledge of the CSCO matrices, which means that not all annotators need to label all samples or with all other annotators. Future work will make the relationship between annotator workload and identifiability explicit.

Acknowledgements: This work is supported in part by the National Science Foundation (NSF) under project NSF IIS-2007836.

[1] Matlab code for the experiments is available at https://github.com/marrintp. The data was generously provided by Traganitis and Giannakis [13].

References

[1] D. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., vol. 24, 2011.

[2] S. Ibrahim, X. Fu, N. Kargas, and K. Huang, "Crowdsourcing via pairwise co-occurrences: Identifiability and algorithms," *Advances in neural information processing systems*, vol. 32, pp. 7847–7857, 2019.

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

[4] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 28, no. 1, pp. 20–28, 1979.

[5] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds." *Journal of machine learning research*, vol. 11, no. 4, 2010.

[6] A. Ghosh, S. Kale, and P. McAfee, "Who moderates the moderators? Crowdsourcing abuse detection in user-generated content," in *Proceedings of the 12th ACM conference on Electronic commerce*, 2011, pp. 167–176.

[7] Q. Liu, J. Peng, and A. T. Ihler, "Variational inference for crowdsourcing," *Advances in neural information processing systems*, vol. 25, 2012.

[8] D. R. Karger, S. Oh, and D. Shah, "Budget-optimal task allocation for reliable crowdsourcing systems," *Operations Research*, vol. 62, no. 1, pp. 1–24, 2014.

[9] ——, "Efficient crowdsourcing for multi-class labeling," in *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*, 2013, pp. 81–92.

[10] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan, "Spectral methods meet em: A provably optimal algorithm for crowdsourcing," *Advances in neural information processing systems*, vol. 27, 2014.

[11] A. Jaffe, B. Nadler, and Y. Kluger, "Estimating the accuracies of multiple classifiers without labeled data," in *Artificial Intelligence and Statistics*. PMLR, 2015, pp. 407–415.

[12] 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*, vol. 139, 2021, pp. 4544–4554.

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

[14] X. Fu, K. Huang, B. Yang, W.-K. Ma, and N. D. Sidiropoulos, "Robust volume minimization-based matrix factorization for remote sensing and document clustering," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6254–6268, 2016.

[15] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications." *IEEE Signal Process. Mag.*, vol. 36, no. 2, pp. 59–80, 2019.

[16] N. Gillis, *Nonnegative Matrix Factorization*. SIAM, 2020.

[17] C.-H. Lin, W.-K. Ma, W.-C. Li, C.-Y. Chi, and A. Ambikapathi, "Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5530–5546, 2015.

[18] T. Marrinan and N. Gillis, "Hyperspectral unmixing with rare endmembers via minimax nonnegative matrix factorization," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1015–1019.

[19] K. Huang, X. Fu, and N. Sidiropoulos, "Learning hidden Markov models from pairwise co-occurrences with application to topic modeling," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2068–2077.

[20] ——, "Anchor-free correlated topic modeling: Identifiability and algorithm," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 1794–1802.

[21] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[22] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 57–77, 2015.

[23] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[24] S. Bird, "NLTK: the natural language toolkit," in *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 2006, pp. 69–72.

[25] W. N. Francis and H. Kucera, "Brown Corpus Manual," Department of Linguistics, Brown University, Providence, Rhode Island, US, Tech. Rep., 1979. [Online]. Available: http://icame.uib.no/brown/bcm.html

[26] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The Penn treebank: Annotating predicate argument structure," in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[27] G. D. Forney, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[28] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.