Permutation-Invariant Consensus over Crowdsourced Labels

Michael Giancola, Randy Paffenroth, Jacob Whitehill
Worcester Polytechnic Institute
Worcester, MA, USA
{ mjgiancola, rcpaffenroth, jrwhitehill } @ wpi.edu

Abstract

This paper introduces a novel crowdsourcing consensus model and inference algorithm – which we call PICA (Permutation-Invariant Crowdsourcing Aggregation) – that is designed to recover the ground-truth labels of a dataset while being invariant to the class permutations enacted by the different annotators. This is particularly useful for settings in which annotators may have systematic confusions about the meanings of different classes, as well as clustering problems (e.g., dense pixel-wise image segmentation) in which the names/numbers assigned to each cluster have no inherent meaning. The PICA model is constructed by endowing each annotator with a doubly-stochastic matrix (DSM), which models the probabilities that an annotator will perceive one class and transcribe it into another. We conduct simulations and experiments to show the advantage of PICA compared to two baselines (Majority Vote, and an "unpermutation" heuristic) for three different clustering/labeling tasks. We also explore the conditions under which PICA provides better inference accuracy compared to a simpler but related model based on right-stochastic matrices. Finally, we show that PICA can be used to crowdsource responses for dense image segmentation tasks, and provide a proof-of-concept that aggregating responses in this way could improve the accuracy of this labor-intensive task.

Introduction

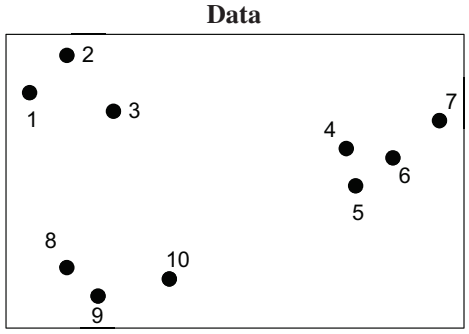
In many crowdsourcing scenarios, annotators are asked to view some kind of data (an image, video, text passage, etc.) and to label it as belonging to some specific class from a set of mutually exclusive classes. For example, annotators in a facial expression labeling task might be requested to view a set of face images and to label each face as displaying one of a finite set of basic facial emotions – e.g., anger, surprise, fear, joy, etc. (Ekman 1992). Based on the raw labels, a variety of existing crowdsourcing consensus algorithms could be used to try to infer simultaneously both the ground-truth labels of the face images and also each individual labeler's accuracy score. However, what if some annotators had a *systematic confusion* in the form of a *permutation* of what the different classes meant – e.g., the annotator perceives *fear* when in fact the subject in the image is *surprised*, and the annotator perceives *surprise* when the subject is actually in *fear* (Jack et al. 2009).[1] In this case, the annotators' labels might still carry valuable information about which face images display the same facial expression. However, in order to accurately infer the ground-truth, the consensus algorithm would need to "un-permute" the labels assigned to the images according to the particular – and *a priori* unknown – misunderstanding of that labeler.

In other crowdsourcing settings – e.g., asking annotators to *cluster* a dataset – there may exist multiple ground-truth labelings that are all equally valid and are equivalent modulo a permutation of the class labels. For instance, when asking multiple annotators to perform dense image segmentation and to assign each pixel a cluster label, the "name" assigned to each cluster (1, 2, 3, etc.) may not carry any inherent meaning – what is important is which pixels belong to the same class. In this case, it would be useful to be able to aggregate over multiple annotators' votes while ignoring the particular name they assign to each class.

In this paper, we present a crowdsourcing consensus algorithm called PICA (Permutation-Invariant Crowdsourcing Aggregation), which is based on doubly-stochastic matrices (DSMs) and the Sinkhorn-Knopp algorithm (Adams and Zemel 2011; Sinkhorn 1964; Sinkhorn and Knopp 1967). PICA is designed to perform *permutation-invariant* simultaneous inference of the ground-truth labels and the individual annotators' *style* of labeling. In contrast to previous work on label aggregation for clustering settings (see section below), our method can benefit from harnessing a smaller number of degrees of freedom to reduce overfitting, and it can model class-specific accuracies.

The chief **contributions** of our paper are the following: (1) We propose and derive an inference algorithm for a novel crowdsourcing consensus algorithm ("PICA") that is designed to be invariant to annotator-specific class permutations. (2) We assess the performance, in terms of accuracy of recovering the ground-truth labels, of PICA for three different kinds of clustering and labeling tasks – including a dense pixel-wise image segmentation – and show that PICA significantly outperforms Majority Vote, as well as a simple "unpermutation" heuristic. (3) We explore the conditions under

[1] This phenomenon has been observed particularly among annotators whose cultural background differs from that of the subjects whose faces they are observing; see Jack et al. 2009.



Labels

	1	2	3	4	5	6	7	8	9	10
Annotator 1	1	1	1	2	2	2	2	3	3	3
Annotator 2	1	1	1	2	2	2	1	3	3	2
Annotator 3	3	3	2	1	1	1	1	2	2	2
Majority Vote	1	1	1	2	2	2	**1**	3	3	**2**
PICA	1	1	1	2	2	2	2	3	3	3
Ground-truth	1	1	1	2	2	2	2	3	3	3

Figure 1: Toy example: Entries give the cluster which each annotator assigns to the given point. While the Majority Vote heuristic makes several mistakes (bolded), the proposed PICA algorithm recovers all the labels correctly.

which PICA also outperforms the related (but simpler) consensus model by Smyth et al. (1995); in particular, we show that the advantage can be substantial in settings in which one class occurs relatively rarely compared to the others.

Toy example: Clustering

To convey the intuition behind the problem we are tackling, suppose three annotators were asked to cluster the points shown below into three clusters, and that the labels they assigned to the points are given in Figure 1. In particular, notice that, while annotators 1 and 2 agree mostly on which points are assigned to clusters 1, 2, and 3, respectively, annotator 3 has "permuted" 1,2,3 for 3,1,2. Moreover, while annotator 1's labels are perfect, annotators 2 and 3 both make a labeling error.

If we attempted to recover the true clustering simply by computing the Majority Vote for each example, we would make several inference errors due to differing – and arbitrary – permutations that the labelers might have when "naming" the different clusters (1, 2, and 3). In contrast, if we could somehow infer each annotator's "style", then we could recover the ground-truth with higher accuracy. Indeed, when applying the PICA consensus algorithm proposed in this paper, we can recover the ground-truth labels with perfect accuracy, while Majority Vote achieves only 80% correct.

Related Work

Since about 10 years ago, when crowdsourcing services such as Amazon Mechanical Turk and Crowdflower started to become popular, a wide variety of algorithms have been developed for aggregating multiple annotators' labels to arrive at a consensus opinion about the ground-truth of a labeled dataset. Many of these algorithms are unsupervised learning algorithms based on the Expectation-Maximization framework and seminal work by Dawid and Skene (1979), whereby the annotators' accuracies as well as the ground-truth labels of the dataset are estimated iteratively. While early work on crowdsourcing consensus focused on binary and multiple-choice labeling tasks (Whitehill et al. 2009; Raykar et al. 2010; Smyth et al. 1995), consensus algorithms now exist for more complex label types such bounding boxes in images (Salek, Bachrach, and Key 2013), taxonomies and trees (Bragg, Weld, and others 2013; Sun et al. 2015), and open-ended text responses (Lin, Weld, and others 2012). Moreover, such algorithms are sometimes able to achieve higher label inference accuracy by exploiting more detailed information about the annotators and the task itself, such as example difficulty (Whitehill et al. 2009), task-dependent biases (Kamar, Kapoor, and Horvitz 2015), and schools of thought among labelers (Tian and Zhu 2012).

The specific issue of how to conduct crowdsourcing consensus for *clusterings* has also received significant research attention, possibly due to the prevalent use of clustering for data visualization and data pre-processing (Strehl and Ghosh 2002; Alush and Goldberger 2012; Kaminsky and Goldberger 2016; Gomes et al. 2011; Yi et al. 2012; Vinayak and Hassibi 2016).

Some of these works frame the clustering problem as a binary classification task, whereby each example is compared to another example to assess whether or not they belong to the same cluster. They then attempt to infer the set of all possible binary comparisons from a small crowdsourced subset of comparisons. For example, the approach by Yi et al. (2012) learns a distance metric from the sparse set of pairwise similarity scores. Vinayak and Hassibi (2016) compare two methods for collecting information about which edges are contained in the graph of all binary comparisons: random edge queries (one edge is observed) and random triangle queries (a triplet of edges is observed). They show that collecting triangle queries can result in more reliable edge information. The work of Gomes et al. (2011) is concerned with how to capture the possibly different dimensions (e.g., size, color, shape) along which different annotators assign examples to clusters, into a "master" clustering model. Their model is optimized using a variational Bayes method.

In terms of application domain (specifically dense image segmentation), the two prior works most similar to ours are (Alush and Goldberger 2012) and (Kaminsky and Goldberger 2016). In these works, the authors frame the clustering problem as a binary classification task, whereby each example is compared to every other example to assess whether or not they belong to the same cluster. By estimating the ground-truth binary decision value for each of the $d(d-1)/2$ unordered comparisons (for d examples), and by taking such annotator parameters into account as overall skill (Alush and Goldberger 2012) and granularity (Kaminsky and Goldberger 2016), the ground-truth cluster labels of every example can be inferred. In both of these papers, the authors applied their technique to a dense image segmentation task, whereby the annotator partitions the set of pixels in an image into disjoint sets that represent different objects. Compared

to other segmentation algorithms, their consensus-based approach yielded improved accuracy.

In our work, we also tackle (among others) the image segmentation problem using crowdsourcing consensus. However, rather than conduct inference over the $d(d-1)/2$ unordered comparisons, we directly optimize the cluster labels assigned to the individual examples as an n-way classification problem (for n clusters). Our model endows each annotator with his/her own *style* matrix that specifies how the ground-truth label assignment is mapped to the annotator's own labels. This allows our model to capture *cluster-specific* accuracy characteristics – e.g., when performing dense image segmentation of a sky, some annotators might be better at distinguishing certain kinds of clouds than others. Moreover, our algorithm generalizes beyond consensus over clusterings: it can be used for inferring ground-truth for multiple-choice questions in which each annotator may permute his/her labels according to some fixed transformation matrix. In this sense, our algorithm offers a way of conducting *permutation-invariant* crowdsourcing consensus.

Finally, we note that our proposed PICA method is similar to the much older method by Smyth et al. (1995); however, they did not explicitly test their model for clustering applications or permutation-invariant labeling settings. We discuss differences between and empirically compare the two models in greater detail later in the paper.

Notation

We use Futura font to denote random variables and Roman font to denote random draws of these variables. For instance, S and z are random variables, and S and z are random draws, respectively. We use upper-case letters to denote matrices/vectors and lower-case letters to denote scalars. For example, S is a matrix and z is a scalar.

PICA: Model Description

Consider a dataset of d examples. Each example has a ground-truth label in the finite set Ω. Let $n = |\Omega|$. Our goal is to use crowdsourcing to determine the ground-truth label of each example j – which we represent with random variable $\mathsf{z}_j \in \Omega$ – for every example in our dataset. To do this, we collect a vector $\mathsf{L}^{(i)} \in (\Omega \cup \{\epsilon\})^d$ of labels from each of m different labelers, where $\mathsf{L}_j^{(i)}$ is the label given by annotator i to example j. If labeler i did not annotate example j, then we define $\mathsf{L}_j^{(i)} = \epsilon$. Together, the annotators' label vectors define an indexed set $\mathcal{L} = \{\mathsf{L}^{(1)}, \mathsf{L}^{(2)}, \ldots, \mathsf{L}^{(m)}\}$. $\mathcal{L}$ is the set of *observed labels*.

We wish to model how different annotators who are labeling the same example may *perceive* the same latent category but *transcribe* the perceived category into different labels. For example, in clustering tasks, different annotators may use different names, numbers, or colors to denote the same set of clusters. We can model the process by which each annotator transcribes his/her perception into a label via a $n \times$n matrix, which we call the *style* of the annotator. In particular, our model endows each annotator i with an $n \times n$ matrix $\mathsf{S}^{(i)}$, where entry $\mathsf{S}_{zl}^{(i)} \in (0,1)$ gives the probability that the annotator perceives an example to belong to class z but transcribes it as $l \in \Omega$. In addition to style, we also model each annotator's *accuracy*. In particular, we define random variable $\mathsf{a}^{(i)} \in (0,1)$ as the probability that annotator i *perceives* the correct category of some example. When conducting crowdsourcing consensus over all the labelers to infer the ground-truth labels, it is important to take both style and accuracy into account.

Likelihood Model

We define the probability that annotator i assigns label $l \in \Omega$ to example j, given the ground-truth label z, style matrix S, and accuracy a as:

$$\begin{aligned} p(\mathsf{L}_j^{(i)} = l \mid \mathsf{z}_j = z, \mathsf{S}^{(i)} = S, \mathsf{a}^{(i)} = a) \\ \doteq \; a \times S_{zl} + (1-a) \times \frac{\sum_{z' \neq z} S_{z'l}}{n-1} \\ = \; \left(S \begin{bmatrix} a & \frac{1-a}{n-1} & \cdots & \\ \frac{1-a}{n-1} & a & \frac{1-a}{n-1} & \cdots \\ \vdots & & \ddots & \\ \frac{1-a}{n-1} & & & a \end{bmatrix} \right)_{zl} \end{aligned} \tag{1}$$

The intuition is that there are two situations in which an annotator i would respond with some particular label l: either (i) they correctly perceived the example as belonging to class z but transcribed it into l (i.e., $a \times S_{zl}$), or (ii) they incorrectly perceived some other class z' but still transcribed it into l (i.e., $(1-a) \times \sum_{z' \neq z} S_{z'l}$). Notice that, in either case, it is possible that $l = z$. We assume that the probability of an incorrect perception $z' \neq z$ is distributed uniformly over all $(n-1)$ incorrect classes.

Probability distribution over permutations

While a permutation matrix containing only 0s and 1s is able to model an annotator's transformation from perceptions onto observed labels (i.e., their "style" of labeling), they are difficult to optimize since they belong to a discrete set. Following the work of Adams and Zemel (2011), we utilize doubly-stochastic matrices (DSMs), i.e., matrices comprising non-negative real numbers (probabilities) in which every row sums to 1 (i.e., the matrix is right-stochastic) and every column sums to 1 (i.e., the matrix is also left-stochastic). DSMs can be considered as "differentiable relaxations of permutation matrices" (Adams and Zemel 2011), and this enables us to use continuous optimization methods such as gradient descent and Expectation-Maximization. DSMs offer the additional advantage that they can model a "soft" style whereby an annotator "usually" transcribes z into l but may sometimes transcribe it into l', etc.

Simplifying the Likelihood Model

Notice that, in Equation 1, the style matrix S is right-multiplied by an accuracy matrix containing a on the diagonal and $\frac{1-a}{n-1}$ everywhere else. The accuracy matrix can eas-

ily be verified to be a DSM. Moreover, by making use of a simple lemma, we can simplify our model:

Lemma 1. *Let A and B be two arbitrary DSMs. Then AB is also a DSM.*

Proof. See appendix. □

We can therefore "fold" the accuracy matrix into the style matrix, so that the latter expresses both the permutation and accuracy of the annotator. This enables us to simplify the likelihood model to be:

$$p(\mathsf{L}_j^{(i)} = l \mid \mathsf{z}_j = z, \mathsf{S}^{(i)} = S) = S_{zl} \tag{2}$$

Inference

Given the observed labels $\mathcal{L}$ and our simplified likelihood model (Equation 2), we can use Expectation-Maximization to optimize over the style matrices of all the annotators and infer the ground-truth of each example. However, the constraint that each style matrix be doubly-stochastic requires special handling, which we describe in the M-Step section below.

E-Step: In this step, we compute the posterior probability distributions of $\mathsf{z}_j \in \Omega \ \forall j \in \{1, \ldots, n\}$ given $\mathsf{S}^{(1)}, \ldots, \mathsf{S}^{(m)}$ from the last M-Step and the observed labels $\mathcal{L}$.

$$\begin{aligned} p(\mathsf{z}_j = z_j \mid \mathsf{L}^{(1)} = L^{(1)}, \ldots, \mathsf{L}^{(m)} = L^{(m)}, \\ \mathsf{S}^{(1)} = S^{(1)}, \ldots, \mathsf{S}^{(m)} = S^{(m)}) \\ \propto \quad p(z_j) \prod_{i : L_j^{(i)} \neq \epsilon} S^{(i)}_{z_j, L_j^{(i)}} \end{aligned}$$

where $S^{(i)}_{z_j, L_j^{(i)}}$ is the z_jth row and $L_j^{(i)}$th column of $S^{(i)}$.

M-Step: In this step, we maximize the auxiliary function Q, defined as the expectation of the joint log-likelihood of the observed labels and ground-truth labels, with respect to the posterior distribution of each z_j computed in the last E-Step, denoted $\widetilde{p}$.

$$\begin{aligned} & Q(S^{(1)}, \ldots, S^{(m)}) \\ &= E[\log p(L^{(1)}, \ldots, L^{(m)}, \\ &\qquad z_1, \ldots, z_d | S^{(1)}, \ldots, S^{(m)})] \\ &= E\left[\log \prod_j \left(p(z_j) \prod_{i : L_j^{(i)} \neq \epsilon} p(L_j^{(i)} | z_j, S^{(i)}) \right)\right] \\ &= \sum_{ij} \sum_{z_j} \log \left(S^{(i)}_{z_j, L_j^{(i)}} \right) \widetilde{p}(z_j) + \text{const.} \end{aligned}$$

Prior on $S^{(i)}$: In some settings, we may also wish to add a regularization term to "push" the style matrices towards the identity permutation; this can be useful if most labelers are assumed to use some "default" permutation (identity). To do so, we can add to Q an additional term $\frac{\gamma}{n^2} \|S^{(i)} - I\|^2_{\text{Fr}}$ where γ specifies the strength of the regularization, I is the identity matrix with the same dimensions as $S^{(i)}$, and $\| \cdot \|^2_{\text{Fr}}$ is the squared Frobenius norm.

Optimizing over Doubly-Stochastic Matrices

If S were simply a real-valued matrix, we could simply use a gradient ascent method (stochastic, conjugate, etc.) to find values of $S^{(1)}, \ldots, S^{(m)}$ that maximize Q. However, the constraint that each style matrix is doubly-stochastic matrix (DSM) requires a more specialized optimization method to ensure that the result of each gradient update remains on the Birkhoff polytope of DSMs. In particular, we utilize a method called Sinkhorn Propagation (Adams and Zemel 2011), which relies on Sinkhorn Normalization (Sinkhorn 1964), described below.

Sinkhorn Normalization

Sinkhorn Normalization is a method of transforming a square matrix A (which we call the *parameterizing matrix*) to a DSM. The algorithm iteratively performs row and column normalizations on A. Sinkhorn (1964) showed that this iterative normalization necessarily converges to a DSM if all the entries of A are strictly positive. The row and column normalization functions are given below:

$$R_{i,j}(M) = \frac{M_{i,j}}{\sum_{k=1}^n M_{i,k}} \qquad C_{i,j}(M) = \frac{M_{i,j}}{\sum_{k=1}^n M_{k,j}}$$

Sinkhorn Propagation

Sinkhorn Propagation (SinkProp) utilizes Sinkhorn Normalization to optimize functions of DSMs (Adams and Zemel 2011). Starting with a strictly positive parameterizing matrix A, we perform alternating rounds (row, column, row, etc.) of Sinkhorn Normalization to arrive at a DSM T. In particular, we define the SinkProp function SP as:

$$\text{SP}^s(A) = \begin{cases} A & \text{if } s = 0 \\ C(R(\text{SP}^{s-1}(A))) & \text{otherwise} \end{cases} \tag{3}$$

where s is the number of Sinkhorn iterations and R and C are the row and column normalization functions applied element-wise to the matrix A. (In practice, for the experiments in this paper we found that $s = 25$ rounds was sufficient.)

After computing $T = \text{SP}^s(A)$ for some s, we can compute the gradient of f with respect to the parameterizing matrix A by back-propagating the gradient through the row and column normalizations. Since R and C are matrix-valued functions, we vectorize them so that the Jacobian of each function can be represented by a 2-D matrix (rather than a tensor). We thus apply the chain rule as:

$$\frac{\partial f}{\partial A} = \frac{\partial f}{\partial \text{vec}[C]} \frac{\partial \text{vec}[C]}{\partial \text{vec}[R]} \frac{\partial \text{vec}[R]}{\partial \text{vec}[C]} \cdots \frac{\partial \text{vec}[R]}{\partial \text{vec}[A]} \tag{4}$$

where

$$\frac{\partial f}{\partial \text{vec}[C]} = \begin{bmatrix} \frac{\partial f}{\partial C_{11}} & \cdots & \frac{\partial f}{\partial C_{1n}} & \frac{\partial f}{\partial C_{21}} & \cdots & \frac{\partial f}{\partial C_{nn}} \end{bmatrix}$$

$$\frac{\partial \text{vec}[C]}{\partial \text{vec}[R]} = \begin{bmatrix} \frac{\partial C_{11}}{\partial R_{11}} & \cdots & \frac{\partial C_{1n}}{\partial R_{11}} & \frac{\partial C_{21}}{\partial R_{11}} & \cdots & \frac{\partial C_{nn}}{\partial R_{11}} \\ \vdots & & & & & \\ \frac{\partial C_{11}}{\partial R_{1n}} & & & & & \\ \frac{\partial C_{11}}{\partial R_{21}} & & & \ddots & & \vdots \\ \vdots & & & & & \\ \frac{\partial C_{11}}{\partial R_{2n}} & & & & & \\ \vdots & & & & & \\ \frac{\partial C_{11}}{\partial R_{nn}} & & & \cdots & & \frac{\partial C_{nn}}{\partial R_{nn}} \end{bmatrix}$$

$$\frac{\partial C_{ij}}{\partial R_{xy}} = \delta(i,x)\left(\frac{\delta(j,y)}{\sum_{k=1}^{n} R_{ky}} - \frac{R_{xy}}{(\sum_{k=1}^{n} R_{ky})^2}\right)$$

Here, $\delta(\cdot)$ is the Kronecker delta function. The matrix $\frac{\partial R}{\partial C}$ has a similar structure to $\frac{\partial C}{\partial R}$. The gradient expression is the same as well, with indices transposed appropriately.

Optimization in Log-Space

To optimize $S^{(1)}, \ldots, S^{(m)}$ to maximize Q during the M-Step, we first set the parameterizing matrix A to I for each $S^{(i)}$, where I is the $n \times n$ identity matrix. Although A is required to be strictly positive to guarantee convergence of Sinkhorn Normalization (Sinkhorn 1964), the SinkProp backpropagation does not guarantee that A will stay strictly positive after the gradient update. Hence, to maintain positivity, we conduct the optimization in log-space, and set $A' = \exp(A)$ (element-wise exponentiation) as the starting value for each $S^{(i)}$.

Inference of Ground-Truth

After conducting Expectation-Maximization, we take the final probability distribution, from the last E-Step, of each z_j as the estimated label for each example j.

Comparison to RSM-based Model

PICA is based on doubly-stochastic matrices (DSMs), which are a special case of right-stochastic matrices (RSMs) – non-negative matrices in which the rows, but not necessarily the columns, sum to 1. RSMs are therefore also capable of modeling the same kinds of "styles" that our proposed DSM-based model can model. RSM-based models for crowdsourcing consensus algorithms were first developed by Smyth et al. (1995). Compared to RSM-based models, DSMs directly encode how each annotator may permute his/her perception of the class into a different label. DSMs can have an advantage over RSMs due to a smaller number of degrees of freedom: whereas an $n \times n$ RSM has $n(n-1)$ free parameters, DSMs have only $(n-1)^2$, which for small n (i.e., a small number of classes or clusters) can be substantial in relative terms. When collecting only a modest number of labels per labeler, this reduction in free parameters can lead to better estimation of each annotator's style matrix and more accurate inference of the ground-truth labels.

	Labels									
	1	2	3	4	5	6	7	8	9	10
Annotator 1	1	1	1	2	2	2	2	3	3	3
Annotator 2	1	3	3	2	2	2	2	3	3	1
Annotator 3	1	2	1	2	1	3	1	1	2	3
Heuristic (L=1)	1	1	**3**	2	2	2	2	3	3	**1**
Heuristic (L=2)	1	1	1	2	2	2	2	**1**	3	3
Heuristic (L=3)	**2**	1	1	2	2	**3**	2	**2**	**1**	3
PICA	1	1	1	2	2	2	2	3	3	3
Ground-truth	1	1	1	2	2	2	2	3	3	3

Table 1: With labels from inaccurate annotators, PICA perfectly recovers the ground-truth labels while the Unpermutation Heuristic makes mistakes regardless of the choice of leader. Incorrectly inferred labels are shown in bold.

Comparison to Unpermutation Heuristic

An alternative way[2] to find consensus over crowdsourced labels despite annotators' permutations is to "un-permute" the labels based on similarities between annotators. For example, if for some subset of examples Annotator 1 uses label A and Annotator 2 uses label B, it is likely A and B refer to the same class. Based on this idea, we constructed the following algorithm, which we refer to as the *Unpermutation Heuristic* (UH):

- Randomly pick an annotator i as the "leader".
- For each annotator $j \neq i$
  - For each class $k = 1, ..., n$ according to i's labels
    - Find the symbol used most frequently by j to express class k.
  - Un-permute j's labels based on the inferred permutation.
- Conduct majority vote over the unpermuted labels of all annotators.

To illustrate how PICA and the Unpermutation Heuristic differ, recall the toy example presented in Figure 1, but consider a new set of observed labels given in Table 1. Annotator 1 is the same as in the previous example (perfect accuracy, identity style). Annotator 2 and 3 also have an identity style, but have low accuracy. PICA is able to identify this and again recovers the ground-truth labels with perfect accuracy. In contrast, the Unpermutation Heuristic misinterprets the low accuracy of Annotators 2 and 3 as a style transformation and makes several mistakes.

In addition, it is important to note that the performance of the Unpermutation Heuristic is highly sensitive to the choice of the "leader". This is not surprising; if the leader is highly inaccurate, unpermuting the other annotators' labels based on their labels will propagate the error. The results of the Unpermutation Heuristic for each possible choice of leader are given in Table 1. The accuracy of the Unpermutation Heuristic for each leader is 80%, 90%, and 60% respectively, while PICA scores 100%.

[2]Suggested anonymously by a reviewer; thanks.

Experiments

To evaluate our proposed PICA model, we performed three experiments: (1) a Mechanical Turk-based experiment on clustering text passages; (2) an abstract labeling simulation; and (3) a dense (pixel-wise) image segmentation experiment.

Permutation-invariant accuracy measurement: Since the cluster labels in our experiments have no inherent meaning (e.g., we could swap the labels "1" and "2" without changing any semantics of the clusters), we computed accuracy of the estimated cluster labels as the *maximum* accuracy, over all $n!$ permutations, of the estimates w.r.t. ground-truth cluster labels.

Optimization details: For all but the image segmentation experiment, we conducted Expectation-Maximization until $Q^{(k)} - Q^{(k-1)} < 10^{-4}$, where $Q^{(k)}$ is the value of Q at the kth iteration. For the image segmentation experiment, we used a tolerance of 10^{-5}. The code for the PICA model implementation as well as the experimental analyses is available at https://github.com/mjgiancola/MQP.

Text Passage Clustering

We designed a text passage clustering task which required annotators to cluster 6 passages into three groups (see Figure 2). This experiment was conducted on real annotators from Amazon Mechanical Turk. The workers were not told how many passages to put into each group, or by what characteristics to cluster them. They were told simply to "Determine how to group the passages based on any similarities and differences that you can identify". Of the six text passages, two were in English, two were in Italian, and two were in Russian. We expected that most people would cluster by language, as the content of the passages were selected to be totally unrelated. (The passages were selected from Wikipedia articles on disparate topics such as mongoose, jet pack, Italian soccer players, etc.) See Figure 2 for the task description that we posed on Mechanical Turk.

We had 25 workers complete our task, one of whom didn't label one of the passages, resulting in 149 labels. Qualitatively, we observed that, while the majority of workers clustered by language, there was some noise in the data due to people clustering with some other reasoning in mind. We assessed accuracy based on a randomly chosen subset (without replacement) of just 3 (out of 25) annotators, and then repeated the experiment 100 times to obtain an average performance estimate. Hence, the accuracy statistics we report reflect an annotation scenario in which only a small number of annotators provide labels, and the job of the aggregator is to infer the ground-truth labels despite the annotators' differing styles.

Results: The proposed PICA model (based on DSMs) achieved an average (over all 100 samplings) of 93% accuracy, with average cross-entropy (computed between the probability distributions of the z_j w.r.t. ground-truth labels) of 2.54. In contrast, the Unpermutation Heuristic achieved an accuracy of only 90.5%, and simple Majority Vote scored 89%. This provides a simple proof-of-concept on labels from real human annotators that permutation-invariant consensus algorithms can be useful. The best performing model, however, was actually the RSM approach (Smyth et al. 1995), which achieved an accuracy of 96% and cross-entropy of 1.65; this shows that the lower number of free parameters in DSMs may not always be decisive.

Instructions

Below is a collection of passages of text in different languages. The objective of this HIT is to collect the passages into three groups. Determine how to group the passages based on any similarities and differences that you can identify. To complete this HIT:

- Read **all** the passages.
- Decide how to group the passages and go back to select a group for each passage.

Passages

1. Mongoose is the popular English name for 29 of the 34 species in the 14 genera of the family Herpestidae, which are small feliform carnivorans native to southern Eurasia and mainland Africa.
   ◯ Group 1 ◯ Group 2 ◯ Group 3
2. Nicola Ventola (Grumo Appula, 24 maggio 1978), un ex calciatore italiano, di ruolo attaccante.
   ◯ Group 1 ◯ Group 2 ◯ Group 3
   ...
6. L'oceano Indiano e un oceano della Terra. In particolare, sia per superficie che per volume, tra i cinque oceani della Terra e il terzo.
   ◯ Group 1 ◯ Group 2 ◯ Group 3

Figure 2: The text passage clustering task we posted on Amazon Mechanical Turk. The annotators' job was to group the different text passages into clusters.

Rare Class Simulation

This simulation was designed to highlight the difference between our DSM-based model and a simpler RSM-based model (e.g., (Smyth et al. 1995)). Recall that, once $n-1$ rows of an $n \times n$ DSM have been identified, then the last row can be inferred unambiguously. With RSMs, on the other hand, all of the rows must be inferred from data independently (since there is no constraint that each column sums to 1). This difference can be decisive in a setting in which one of the classes (from $\{1, \ldots, n\}$) occurs only rarely in the dataset, so that few of the observed labels provide any information on the values of each $S^{(i)}$.

For this simulation, we generated 100 examples, each of which was assigned a label $z_j \in \Omega = \{$ 'a', 'b', 'c' $\}$, with $p(\text{'a'}) = 0.5$, $p(\text{'b'}) = 0.45$, $p(\text{'c'}) = 0.05$. We then simulated 100 annotators who each have a random accuracy $a^{(i)}$ sampled uniformly from $[0.75, 1)$. Each annotator had a random permutation of the identity for their style matrix. For each annotator, we collected 10 labels, sampled randomly (without replacement) from the 100 total examples, giving a total of 1000 observed labels. As in the Text Passage Cluster-

(a) Original (b) Ground-truth (c) Noisy labeler

Figure 3: A dense (pixel-wise) image segmentation task on one of the images ("couple") from the ADE20K dataset (Zhou et al. 2017). Sub-figures a-c show (in order) the original image, the ground-truth segmentation, the segmentation collected from one of the simulated noisy labelers.

ing experiment, results were averaged over 100 independent trials.

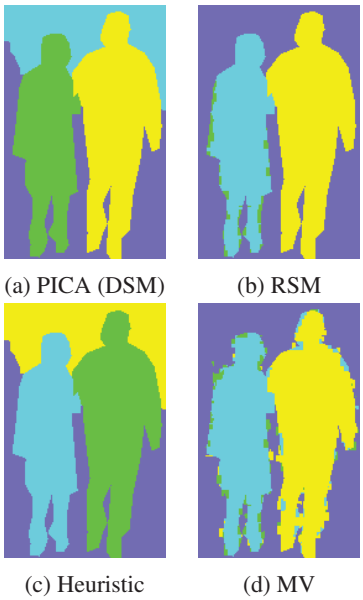
Results: The DSM-based PICA algorithm achieved a percent correct of 88% and 4.01 cross entropy, while the RSM-based model scored 79% with 6.32 cross entropy. As a baseline, the Unpermutation Heuristic scored 62% and Majority Vote scored 47%. These results show an example of how the more constrained DSM-based PICA algorithm can yield higher accuracy than the simpler (and less constrained) RSM-based approach.

Dense Image Segmentation

In this experiment we explored whether PICA could reconstruct a dense (pixel-wise) image segmentation from multiple noisy segmentations. In our experiment, we used an image and ground-truth segmentation (Figure 3a and 3b)[3] from the ADE20K dataset (Zhou et al. 2017), which in turn included images from the LabelMe dataset (Russell et al. 2008). Based on the ground-truth segmentation, we then generated segmentations for 10 simulated annotators by permuting the class labels and adding noise (see Figure 3c for an example). The input to our model thus consisted of $10 \times \texttt{numPixels}$ labels, where each of the 10 segmentations corresponded to the labels of a single annotator, and each pixel corresponded to an individual example (i.e. pixel j in image i is label $L_j^{(i)}$). We generated a segmentation dataset for four different images: "couple", "flag", "light', and "people"; the "couple" image is shown in Figure 3a; the others are not shown due to space limitations. Together, these data enable us to assess whether PICA can infer each annotator's different "style" in assigning pixels to colors and combine them across annotators to yield aggregated labels that are more accurate.

Results: Accuracy in inferring the ground-truth clustering labels w.r.t. the ground-truth of the four approaches – PICA (based on DSMs), RSM, the Unpermutation Heuristic, and Majority Vote – are shown in Table 2. Graphical results just for the "couple" image are also shown in Figure 4. As with the previous experiments, we compute the %-correct accuracy and cross-entropy as the *maximum* over all $n!$ permutations, where n is the number of class labels. Hence, the accuracies of the models are not penalized due to just the arbitrary coloring assigned to each cluster in the image.

[3]Figure 3 and 4 are best viewed in color.



(a) PICA (DSM) (b) RSM

(c) Heuristic (d) MV

Figure 4: Results of the crowdsourced image segmentation task. Sub-figures a-d show (in order) the inferred ground-truth (over all 10 noisy labelers) using the proposed DSM-based PICA algorithm, the RSM-based algorithm (Smyth et al. 1995), the Unpermutation Heuristic, and simple Majority Vote.

For the "couple" image, PICA was able to reconstruct the original segmentation with 100% accuracy. In contrast, the RSM model achieved 80% accuracy relative to the original segmentation, as it was unable to distinguish the foreground and background of the image. A simple majority vote still achieved 77% accuracy, but generates a segmentation with a lot of noise. For this example, the Unpermutation Heuristic achieved the same level of performance as PICA – this was expected as the simulated annotators were highly accurate.

For all four images that we tested, the results are shown in Table 2. For three of the four images, PICA performed the best in terms of %-correct clustering accuracy (fraction of pixels assigned to the correct cluster, after taking the maximum over all $n!$ permutations), tying with the Unpermutation Heuristic twice. For the fourth image, PICA tied with RSM. In terms of cross-entropy, the results were more mixed, and both RSM and PICA achieved the best cross-entropy in two of the four images.

While this is simply a proof-of-concept, it illustrates how crowdsourcing – and principled algorithms for aggregating over crowdsourced labels – can be used for image segmentation tasks. After investigating annotation consistency between segmentations, Zhou et al. (2017) discussed the important sources of error. The first was varying levels of segmentation quality from different annotators. The second was ambiguities in object naming (i.e. one annotator labeled a

Model	Image							
	"couple"		"flag"		"light"		"people"	
	%	C.E.	%	C.E.	%	C.E.	%	C.E.
PICA	**100**	65.5	**99.4**	**.077**	**99.8**	**.020**	**99.0**	48.2
RSM	80.4	**13.9**	98.7	.139	99.1	.040	**99.0**	**39.6**
UH	**100**	–	99.2	–	**99.8**	–	98.9	–
MV	77.5	–	81.5	–	88.9	–	86.0	–

Table 2: Accuracy of the proposed DSM-based PICA model compared to the RSM-based model (Smyth et al. 1995) and the Unpermutation Heuristic (UH) for aggregating over multiple annotators' dense (pixel-wise) image segmentations. We applied the algorithms to four different images ("couple", "flag", "light", and "people"). We provide the results of Majority Vote as a baseline. Accuracy was measured using %-correct accuracy in assigning pixels to clusters, as well as cross-entropy.

truck as "car"). As shown empirically, the proposed PICA model can construct a segmentation which eliminates the noise in the given data. The object naming issue would be eliminated entirely, as PICA is invariant to each annotator's style.

Conclusion & Future Work

This paper introduced a model for crowdsourcing consensus – PICA – which provides a degree of invariance to permutations of class labels, provided that there is enough data from the annotators to recover each annotator's "style" of labeling. We showed that this invariance made our model well-suited to clustering tasks. Moreover, we identified a particular regime – annotation tasks in which some classes may occur only rarely – in which doubly-stochastic matrix (DSM)-based PICA can provide better performance than a simpler right-stochastic matrix (RSM)-based model. In other settings, the empirical results suggest that DSM and RSM seem to perform equally well.

Future research: Permuting one class label for another is an instance of how a labeler may have a particular "style", over which a crowdsourcing consensus algorithm must marginalize when inferring ground-truth labels. But there are other settings in which style may be more complex. For instance, in transcription tasks in which labelers generate not just a fixed-length label but an entire sentence, each annotator may tend to use different words or symbols to mean the same thing. It would be interesting to explore how more general forms of style can be modeled.

More technically, an interesting area of future inquiry is exploring the relationship between the number of annotators and examples in some labeling task. One would expect that the number of annotators and examples have an inverse relationship in the sense that having more annotators can compensate for having fewer labels per annotator. But is there a limit to the extent that more examples can make up for a lack of annotators (and vice-versa)? For the particular case of PICA, it would be interesting to investigate how this issue interacts with the relative benefits/drawbacks of using DSM-based versus RSM-based models.

Appendix: Proofs and Derivations

Lemma 1: Dot Product of two DSMs is a DSM

Let A and B be two arbitrary DSMs.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \vdots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ & & \vdots & \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}$$

Column j of AB has the form:

$$AB_{*j} = \begin{bmatrix} a_{11}b_{1j} + a_{12}b_{2j} + \dots + a_{1n}b_{nj} \\ a_{21}b_{1j} + a_{22}b_{2j} + \dots + a_{2n}b_{nj} \\ \vdots \\ a_{n1}b_{1j} + a_{n2}b_{2j} + \dots + a_{nn}b_{nj} \end{bmatrix}$$

Taking the sum of the elements in column j, we can reorder the terms and find:

$$\begin{aligned} & b_{1j}(a_{11} + \dots + a_{n1}) \\ & + b_{2j}(a_{12} + \dots + a_{n2}) \\ & + \dots \\ & + b_{nj}(a_{1n} + \dots + a_{nn}) \\ & = b_{1j} + b_{2j} + \dots + b_{nj} \\ & = 1 \end{aligned}$$

A similar argument shows that each row of AB sums to 1.

Expectation-Maximization Gradient Derivations

Our derivations below make use of the conditional independence encoded in the graphical model in Figure 5.

E-Step: In this step, we compute the posterior probabilities of $\mathsf{z}_j \in \Omega \; \forall j \in \{1, \dots, n\}$ given the S values from the last M-Step and the set of all observed labels:

$$\begin{aligned} & p(\mathsf{z}_j = z_j \mid \mathsf{L}^{(1)} = L^{(1)}, \dots, \mathsf{L}^{(m)} = L^{(m)}, \\ & \qquad \mathsf{S}^{(1)} = S^{(1)}, \dots, \mathsf{S}^{(m)} = S^{(m)}) \\ = \; & p(z_j \mid L^{(1)}, \dots, L^{(m)}, S^{(1)}, \dots, S^{(m)}) \\ \propto \; & p(z_j) p(L^{(1)}, \dots, L^{(m)} \mid S^{(1)}, \dots, S^{(m)}, z_j) \\ = \; & p(z_j) \prod_{i : L_j^{(i)} \neq \epsilon} p(L_j^{(i)} \mid z_j, S^{(i)}) \\ = \; & p(z_j) \prod_{i : L_j^{(i)} \neq \epsilon} S^{(i)}_{z, L_j^{(i)}} \end{aligned}$$

where we note that $p(z_j | S^{(1)}, \dots, S^{(m)}) = p(z_j)$ by conditional independence assumptions from the graphical model.

M-Step: In this step, we maximize the auxiliary function Q, defined as the expectation of the joint log-likelihood of

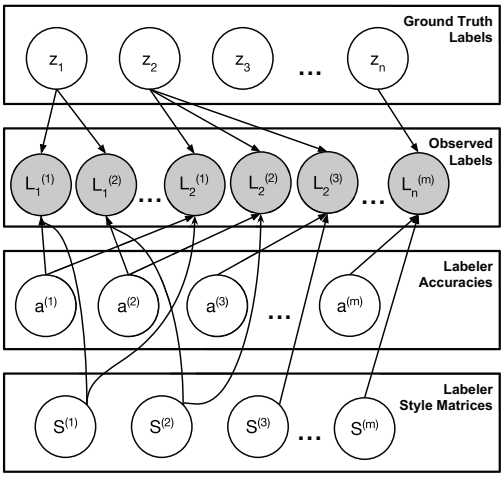


Figure 5: Graphical model of ground-truth, observed labels, styles, and accuracies. Only the observed labels (shaded) are observed. The accuracies $\mathsf{a}^{(1)}, \ldots, \mathsf{a}^{(m)}$ can be "folded" into the style matrices due to Lemma 1.

the observed and ground-truth labels given the parameters $S^{(1)}, \ldots, S^{(m)}$, with respect to the posterior probabilities of z computed in the last E-Step, denoted by probability distribution $\widetilde{p}$.

$$
\begin{aligned}
&Q(S^{(1)}, \ldots, S^{(m)}) \\
&= E\left[\log p(\mathcal{L} = L, \mathsf{z} = z \mid \mathsf{S} = S)\right] \\
&= E\left[\log p(L, z \mid S)\right] \\
&= E\left[\log \prod_j \left(p(z_j) \prod_i p(L_j^{(i)} \mid z_j, S^{(i)}) \right) \right] \\
&\quad \text{since } L_j^{(i)} \text{ are cond. indep.} \\
&= \sum_j E\left[\log p(z_j)\right] + \sum_{ij} E\left[\log \left(L_j^{(i)} \mid z_j, S^{(i)} \right)\right] \\
&= C + \sum_{ij} E[\log \left(L_j^{(i)} \mid z_j, S^{(i)} \right)] \\
&\quad \text{since } \sum_j E[\log p(z_j)] \text{ is const. w.r.t. each } S^{(i)} \\
&= \sum_{ij} \sum_{z_1, \ldots, z_d} \log \left(p(L_j^{(i)} | z_j, S^{(i)}) \right) \widetilde{p}(z_1, \ldots, z_d) \\
&= \sum_{ij} \sum_{z_j} \log \left(p(L_j^{(i)} | z_j, S^{(i)}) \right) \widetilde{p}(z_j) \\
&\quad \sum_{\substack{z_1, \ldots, z_{j-1} \\ z_{j+1}, \ldots, z_d}} \widetilde{p}(z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_d) \\
&= \sum_{ij} \sum_{z_j} \log \left(p(L_j^{(i)} | z_j, S^{(i)}) \right) \widetilde{p}(z_j) \\
&= \sum_{ij} \sum_{z_j} \log \left(S^{(i)}_{z, L_j^{(i)}} \right) \widetilde{p}(z_j)
\end{aligned}
$$

To take the first derivative of Q, we vectorize each $S^{(i)}$:

$$
\frac{\partial Q}{\partial \text{vec}[S^{(i)}]} = \begin{bmatrix} \frac{\partial Q}{\partial S_{11}^{(i)}} & \cdots & \frac{\partial Q}{\partial S_{1n}^{(i)}} & \frac{\partial Q}{\partial S_{21}^{(i)}} & \cdots & \frac{\partial Q}{\partial S_{nn}^{(i)}} \end{bmatrix}
$$

where

$$
\begin{aligned}
\frac{\partial Q}{\partial S_{xy}^{(i)}} &= \sum_{j : L_j^{(i)} \neq \epsilon} \sum_{z_j} \frac{\partial}{\partial S_{xy}^{(i)}} [\log \left(S^{(i)}_{z_j, L_j^{(i)}} \right)] \widetilde{p}(z_j) \\
&= \sum_{j : L_j^{(i)} \neq \epsilon} \frac{\partial}{\partial S_{xy}^{(i)}} [\log \left(S^{(i)}_{x, L_j^{(i)}} \right)] \widetilde{p}(x) \\
&= \sum_{j : L_j^{(i)} \neq \epsilon} \delta(L_j^{(i)}, y) \frac{\partial}{\partial S_{xy}^{(i)}} [\log \left(S_{xy}^{(i)} \right)] \widetilde{p}(x) \\
&= \sum_{j : L_j^{(i)} \neq \epsilon} \delta(L_j^{(i)}, y) \frac{\widetilde{p}(x)}{S_{xy}^{(i)}}
\end{aligned}
$$

Text Passage Clustering Data

We collected passages on Wikipedia for our text clustering experiment. Since we wanted passages in multiple languages, we used English, Russian, and Italian Wikipedia articles. The passages we used came from the following articles: Mongoose and Jet Pack (English), Nicola Ventola and Indian Ocean (Italian), and Ludwig van Beethoven and G-Eazy (Russian).

Image Segmentation Data

We used an image and segmentation from the ADE20K dataset (Zhou et al. 2017). The image was originally from the LabelMe dataset (Russell et al. 2008). To reduce time costs, we cropped sections of the image to use. Also, to simplify the results, we regenerated our own segmentations which ignored a few class labels (i.e. small objects in the background).

References

Adams, R. P., and Zemel, R. S. 2011. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*.

Alush, A., and Goldberger, J. 2012. Ensemble segmentation using efficient integer linear programming. *IEEE transactions on pattern analysis and machine intelligence* 34(10):1966–1977.

Bragg, J.; Weld, D. S.; et al. 2013. Crowdsourcing multi-label classification for taxonomy creation. In *First AAAI conference on human computation and crowdsourcing*.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics* 20–28.

Ekman, P. 1992. An argument for basic emotions. *Cognition & emotion* 6(3-4):169–200.

Gomes, R. G.; Welinder, P.; Krause, A.; and Perona, P. 2011. Crowdclustering. In *Advances in neural information processing systems*, 558–566.

Jack, R. E.; Blais, C.; Scheepers, C.; Schyns, P. G.; and Caldara, R. 2009. Cultural confusions show that facial expressions are not universal. *Current Biology* 19(18):1543–1548.

Kamar, E.; Kapoor, A.; and Horvitz, E. 2015. Identifying and accounting for task-dependent bias in crowdsourcing. In *Third AAAI Conference on Human Computation and Crowdsourcing*.

Kaminsky, O., and Goldberger, J. 2016. Combining clusterings with different detail levels. In *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*, 1–6. IEEE.

Lin, C. H.; Weld, D.; et al. 2012. Crowdsourcing control: Moving beyond multiple choice. *arXiv preprint arXiv:1210.4870*.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *Journal of Machine Learning Research* 11(Apr):1297–1322.

Russell, B. C.; Torralba, A.; Murphy, K. P.; and Freeman, W. T. 2008. Labelme: a database and web-based tool for image annotation. *International journal of computer vision* 77(1-3):157–173.

Salek, M.; Bachrach, Y.; and Key, P. 2013. Hotspotting-a probabilistic graphical model for image object localization through crowdsourcing. In *AAAI*.

Sinkhorn, R., and Knopp, P. 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics* 21(2):343–348.

Sinkhorn, R. 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics* 35(2):876–879.

Smyth, P.; Fayyad, U. M.; Burl, M. C.; Perona, P.; and Baldi, P. 1995. Inferring ground truth from subjective labelling of venus images. In *Advances in neural information processing systems*, 1085–1092.

Strehl, A., and Ghosh, J. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* 3(Dec):583–617.

Sun, Y.; Singla, A.; Fox, D.; and Krause, A. 2015. Building hierarchies of concepts via crowdsourcing. In *IJCAI*, 844–853.

Tian, Y., and Zhu, J. 2012. Learning from crowds in the presence of schools of thought. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 226–234. ACM.

Vinayak, R. K., and Hassibi, B. 2016. Crowdsourced clustering: Querying edges vs triangles. In *Advances in Neural Information Processing Systems*, 1316–1324.

Whitehill, J.; Wu, T.-f.; Bergsma, J.; Movellan, J. R.; and Ruvolo, P. L. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, 2035–2043.

Yi, J.; Jin, R.; Jain, S.; Yang, T.; and Jain, A. K. 2012. Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *Advances in neural information processing systems*, 1772–1780.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.