Weighted majority vote using Shapley values in crowdsourcing

Tanguy Lefort*[1], Benjamin Charlier[2], Alexis Joly[3], and Joseph Salmon[4]

[1]University of Montpellier, IMAG, CNRS, LIRMM, Inria
[2]University of Montpellier, IMAG, CNRS
[3]LIRMM, Inria
[4]University of Montpellier, IMAG, CNRS, Institut Universitaire de France (IUF)

March 27, 2024

Abstract

Crowdsourcing has emerged as a pivotal paradigm for harnessing collective intelligence to solve data annotation tasks. Effective label aggregation, crucial for leveraging the diverse judgments of contributors, remains a fundamental challenge in crowdsourcing systems. This paper introduces a novel label aggregation strategy based on Shapley values, a concept originating from cooperative game theory. By integrating Shapley values as worker weights into the Weighted Majority Vote label aggregation (WMV), our proposed framework aims to address the interpretability of weights assigned to workers. This aggregation reduces the complexity of probabilistic models and the difficulty of the final interpretation of the aggregation from the workers' votes. We show improved accuracy against other WMV-based label aggregation strategies. We demonstrate the efficiency of our strategy on various real datasets to explore multiple crowdsourcing scenarios.

Keywords: crowdsourcing, explainability, label aggregation, Shapley values.

1 Introduction and related work

Data annotation is a crucial step in the development of machine learning models. The quality of the annotations is a key factor in the performance of the models (Snow et al., 2008). Frequently, the annotation process is outsourced to a crowd of non-expert workers through crowdsourcing platforms such as Amazon Mechanical Turk[1]. However, the quality of the annotations can vary greatly from one worker to another (Ross et al., 2009; Ipeirotis et al., 2010; Hara et al., 2018).

To address this issue, several label aggregation strategies have been proposed in the literature. The most common approach is the majority vote (MV) strategy, which consists of selecting the label with the largest number of responses. While simple and easy to implement, MV has several limitations, such as not taking into account the reliability of the workers. Indeed, it affects the same weight to all the workers in the final aggregated label, no matter their level of expertise.

To alleviate this issue, probabilistic generative models such as DS (Dawid and Skene, 1979) or GLAD (Whitehill et al., 2009) have been proposed, relying on generative models of the votes. These models estimate the reliability of the workers and take it into account in the label aggregation process through different parameters. The DS model considers that each worker has an assigned confusion matrix – to be estimated – while GLAD models the reliability of the workers through a scalar weight and also includes the task's difficulty in the final aggregation. Such a framework is flexible enough

*tanguy.lefort@umontpellier.fr

[1]`https://www.mturk.com/`

to incorporate various sources of information, and the inference is often based on the Expectation-Maximization algorithm which is computationally expensive and sensitive to initialization. Moreover, the final interpretation of the aggregation from the workers' votes is not straightforward and the result of probabilistic models can be difficult to interpret for non-experts.

Weighted MV (WMV) strategies (Limited, 2021; Karger et al., 2011; Ma and Olshevsky, 2020) have proven to be both effective and easy to interpret. Indeed, the method's principle is straightforward: each worker is assigned a weight that represents their reliability. The aggregated label is then the label that reflects the votes of workers relative to their reliability.

In this work, we aim to propose a new weight for WMV based on the Shapley values (Shapley, 1953). The Shapley value is a concept originating from cooperative game theory that has been used in various fields such as economics (Aumann, 1994), political science (Engelbrecht and Vos, 2009), statistics (Owen, 2014) and machine learning for explainability (Lundberg and Lee, 2017) or feature selection (Cohen et al., 2007). Shapley values have been used in the context of data valuation in classification (Schoch et al., 2022) and active learning (Ghorbani et al., 2022). Here, we propose to extend it to classification in a crowdsourcing setting.

If we consider that each worker is a feature and each task is a sample point, given a classifier, the Shapley value explains the contribution of each worker to the predicted outcome at each queried task (Molnar, 2020). Shapley values are used as worker importance indicators that can handle interactions between workers' answers (Owen and Prieur, 2017; Lundberg and Lee, 2017). We propose a study of their usage as interpretable weights in weighted majority votes for crowdsourcing classification tasks.

2 Notation and related work

Notation. We consider classical multi-class learning notation, with input in $\mathcal{X}$ and labels in $[K] := \{1, \ldots, K\}$. There are $n_{\texttt{task}}$ available, denoted $x_1, \ldots, x_{n_{\texttt{task}}}$, to be labeled by $n_{\texttt{worker}}$ workers. The set of $n_{\texttt{task}}$ tasks with their associated true labels is $\mathcal{D} = \{(x_1, y_1^\star), \ldots, (x_{n_{\texttt{task}}}, y_{n_{\texttt{task}}}^\star)\}$. Denote $\mathbf{Y} \in [K]^{n_{\texttt{task}} \times n_{\texttt{worker}}}$ the matrix of the workers' answers for each task. The true labels are unobserved but crowdsourced labels are provided by the workers. We write $\mathcal{A}(x_i) = \{j \in [n_{\texttt{worker}}] :$ worker j labeled task $x_i\}$ the **annotators set** of a task x_i. For a task x_i and each $j \in \mathcal{A}(x_i)$, we denote $y_i^{(j)} \in [K]$ the label answered by worker j. Given an aggregation strategy $\texttt{agg}$ (such as MV), we denote aggregated label $\hat{y}_i^{\texttt{agg}} \in [K]$. For any set $\mathcal{S}$, we write $|\mathcal{S}|$ for its cardinality. The indicator function is denoted $\mathbb{1}(\cdot)$. The matrix full of ones of size $n \times m$ is denoted $\mathbf{1}_{n\times m}$. The row of a matrix M indexed by i is denoted $M_{i,:}$ and the column indexed by j is $M_{:,j}$.

On released datasets, to compute performance metrics, partial true labels are made available. We denote $\mathcal{D}_{\texttt{train}}$ the set of tasks with their true labels unknown and $\mathcal{D}_{\texttt{test}}$ the set of tasks with known true labels. Note that these true labels are only used at test time. Both workers and aggregation strategies do not have access to the true labels. Their goal is to recover it.

The impact of a worker on a task $x \in \mathcal{X}$, for a classifier f, is evaluated by a value function $\nu_{x,f} : 2^{[n_{\texttt{worker}}]} \to \mathbb{R}$ such that for any set of workers $S \subset [n_{\texttt{worker}}]$ and any worker $j_0 \notin S$, $\nu_{x,f}(S \cup \{j_0\}) - \nu(S)$ is the marginal contribution of worker j_0 over S. In a classification setting with output $f(x)$, the value function over a set $S \subseteq [n_{\texttt{worker}}]$ of workers is defined as

$$\nu_{x,f}(S) = \mathbb{E}[f(x)|x_S] \ . \tag{1}$$

In practice, this quantity has to be estimated, for instance using the $\texttt{TreeSHAP}$ algorithm (Lundberg et al., 2018).

Existing weighted label aggregation strategies. In this work, we focus on label aggregation strategies as weighted Majority Votes (Littlestone and Warmuth, 1994) – *i.e.* that can be written as:

$$\forall i \in [n_{\texttt{task}}], \quad \hat{y}_i = \mathrm{WMV}(i, W) := \operatorname*{arg\,max}_{k \in [K]} \sum_{j \in \mathcal{A}(x_i)} W_{j,k} \mathbb{1}(y_i^{(j)} = k), \tag{2}$$

where $W \in \mathbb{R}^{n_{\texttt{worker}} \times K}$ is the matrix assigning the weight of worker j when answering class k. We denote $W_{j,k} \in \mathbb{R}$ the weight of worker j for class k. This weight matrix is the cornerstone of each aggregation strategy. We detail below popular label aggregation strategies that fit into this framework.

- MV: The majority vote strategy assigns the label that has been chosen by the majority of the workers. It can be written as:

$$\hat{y}_i^{\mathrm{MV}} = \mathrm{WMV}(i, \mathbf{1}_{n_{\texttt{worker}} \times K}) \ . \tag{3}$$

  This is the simplest weights assignment, where all workers and all labels share the same weight.

- WAWA (Limited, 2021): This strategy, also known as the inter-rater agreement, weights each user by how much they agree with the MV labels on average. More formally, given a task i:

$$\hat{y}_i^{\mathrm{WAWA}} = \mathrm{WMV}\left(i, W\right), \quad \text{with} \quad W_{j,:} = \left(\frac{1}{|\{y_{i'}^{(j)}\}_{i'}|} \sum_{i'=1}^{n_{\mathrm{task}}} \mathbb{1}\left(y_{i'}^{(j)} = \hat{y}_{i'}^{\mathrm{MV}} \right) \right) \mathbf{1}_K \ . \tag{4}$$

  It allows us to instantiate a weight that can vary for each worker (but not per task) and it usually improves on the MV strategy.

- ZBS: The Zero-Based Skill aggregation is a gradient descent (GD)-based version of the WAWA strategy. First, the labels are initialized using the MV strategy. Then, a descent step is performed on the weights to minimize the squared error between the current worker's weight and the weight assigned by the WAWA strategy. Finally, the aggregated labels are recomputed using the WMV strategy. This loop is repeated until convergence.

Algorithm 1 Zero Based Skill algorithm.

Input: $\eta > 0$ the learning rate, $t_{\max} > 0$ maximum number of iterations,
Initialize weights at step 0: $W^0 = \frac{1}{K} \mathbf{1}_{n_{\texttt{worker}} \times K}$.
for $t = 1, \dots, t_{\max}$ **do**
  Update labels: $\hat{y}_i^t = \mathrm{WMV}(i, W^{t-1})$ for $i \in [n_{\texttt{task}}]$
  Compute current accuracy by worker: $a_j = \left(\frac{1}{|\{y_{i'}^{(j)}\}_{i'}|} \sum_{i'=1}^{n_{\mathrm{task}}} \mathbb{1}\left(y_{i'}^{(j)} = \hat{y}_i^t \right) \right) \mathbf{1}_K$
  Update weights for each worker $j \in [n_{\texttt{worker}}]$: $W_{j,:}^t = W_{j,:}^{t-1} - \eta (W_{j,:}^{t-1} - a_j)$
end for
Output: $\hat{y}_i^{\mathrm{ZBS}} = \hat{y}_i^{t_{\max}}$.

- WDS (Dawid and Skene, 1979): This strategy is based on the Dawid-Skene model. A confusion matrix $\pi^{(j)} \in \mathbb{R}^{K \times K}$ is associated to each worker, such that the (k, ℓ)-entry or $\pi^{(j)}$ represents the probability for worker j to answer $\ell \in [K]$ when the unknown true label is $k \in [K]$. For instance, each diagonal term represents the ability of the worker to answer correctly the underlying label. Using this DS diagonal, we obtain a weighted majority vote denoted WDS:

$$\hat{y}_i^{\mathrm{WDS}} = \mathrm{WMV}\left(i, W\right), \quad \text{with} \quad W_{j,k} = \pi_{k,k}^{(j)} \ . \tag{5}$$

- M-MSR (Ma and Olshevsky, 2020): The Matrix Mean-Subsequence-Reduced strategy considers the reliability of all workers as a vector $s \in \mathbb{R}^{n_{\texttt{worker}}}$. Each entry s_j represents the reliability of the worker j. This strategy assumes that each worker answers independently. It also assumes that a worker is correct with probability $p_j \in [0,1]$ and the worker's probability of being wrong is uniform across classes, *i.e.*:

$$\forall (i,j) \in [n_{\texttt{task}}] \times [n_{\texttt{worker}}], \begin{cases} \mathbb{P}(y_i^{(j)} = k) = p_j & \text{if } y_i^\star = k, \\ \mathbb{P}(y_i^{(j)} = k) = \frac{1-p_j}{K-1} & \text{if } y_i^\star \neq k \end{cases} .$$

The reliability of a worker is linked to its probability of answering correctly: $s_j = \frac{K}{K-1}p_j - \frac{1}{K-1}$. This reliability can be estimated by solving a rank-one matrix completion problem defined as:

$$\mathbb{E}\left[\frac{K}{K-1}C - \frac{1}{K-1}\mathbb{1}\mathbb{1}^\top\right] = ss^\top \ ,$$

where C is the covariance matrix of the workers' answers. More precisely, given two workers $j, j' \in [n_{\texttt{worker}}]$, the covariance between them is

$$C_{j,j'} = \frac{1}{N_{j,j'}} \sum_{i=1}^{n_{\text{task}}} \mathbb{1}(y_i^{(j)} = y_i^{(j')}) \ ,$$

with $N_{j,j'}$ the number of tasks in common: $N_{j,j'} = |\{i \in [n_{\texttt{task}}] | j, j' \in \mathcal{A}(x_i)\}|$. The final label is a weighted majority vote:

$$\hat{y}_i^{\text{M-MSR}} = \text{WMV}(i, W) \quad \text{with} \quad W_{j,k} = \log \frac{(K-1)p_j}{1-p_j} \ , \tag{6}$$

where the form of the weights is derived from a maximum a posteriori formulation of the model, see (Li and Yu, 2014, Corollary 9).

- KOS (Karger et al., 2011): Only set for binary classification $K = 2$, the KOS strategy comes from a graph-theory perspective. The worker's weight is estimated iteratively inspired by the belief propagation algorithm (Pearl, 1986) to look at the worker agreements on neighboring tasks. An edge from a worker to a task indicates that the task was answered by the worker. In an EM fashion, a worker message – the reliability of worker j for task i – is stored in a matrix $W \in \mathbb{R}^{n_{\texttt{worker}} \times n_{\texttt{task}}}$. Then, the task message – the likelihood of the task i to be positive – is sent to the workers as a vector of $\mathbb{R}^{n_{\texttt{task}}}$. The final label is the sign of the weighted majority votes, with the weight of worker j's answer to task i being equal to $W_{j,i}$.

Note that depending on the strategy, the weights $W_{j,k}$ might not be upper-bounded. Indeed, the KOS strategy does not have an upper bound on the weights for instance. If the weights are not upper-bounded, the more a worker answers following other workers, the more weight they will accumulate. In Figure 1 we show how each strategy leads to different weights and scales. The weights are computed for the BlueBirds dataset (Welinder et al., 2010) presented in more detail in Section 3.4.

3 Shapley label aggregation for crowdsourcing

3.1 Preliminaries on Shapley values

Shapley values have been used to quantify the contribution of individual features in machine learning models' prediction (Molnar, 2020). In the context of crowdsourcing, we propose to use Shapley values to quantify the contribution of each worker to the final label aggregation.

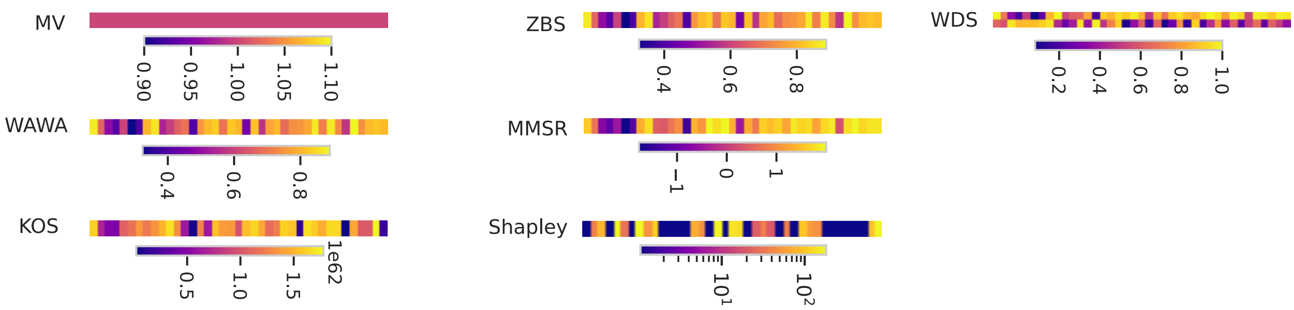

Figure 1: Example of weight matrices $W \in \mathbb{R}^{n_{\texttt{worker}} \times K}$ obtained for the BlueBirds ($n_{\texttt{worker}} = 39, K = 2$) dataset using the presented strategies and our in Algorithm 2. We transpose them for ease of readability (each row is a class, each column a worker). Only WDS takes into account the class label. For KOS, we show the weight in absolute value, averaged over the tasks (as the weights are task-dependent). However, this transformation does not affect the scale of magnitude of weights.

Definition 1. *Given a set of n_{worker} workers, a classifier f, a task $x_i \in \mathcal{X}$ and a value function $\nu_{x_i,f}(\cdot) : 2^{[n_{worker}]} \to \mathbb{R}$ the Shapley value of a worker j for a task i is defined as the average marginal contribution of the worker j to every subset of $[n_{worker}] \setminus \{j\}$:*

$$\phi_j(i, \nu) = \sum_{S \subseteq [n_{worker}] \setminus \{j\}} \frac{|S|!(n - |S| - 1)!}{n!} \left[\nu_{x_i,f}(S \cup \{j\}) - \nu_{x_i,f}(S)\right] \quad . \tag{7}$$

When there is no ambiguity over the value function, we adopt the standard notation abuse $\phi_j(i, \nu) = \phi_j(i)$.

Shapley values satisfy the following properties – given a task x and a classifier f:

- Symmetry: if $\nu_{x,f}(S \cup \{p\}) = \nu_{x,f}(S \cup \{q\})$ for all set $S \subseteq [n_{\texttt{worker}}] \setminus \{p, q\}$, then $\phi_p = \phi_q$.
- Null worker: if $\nu_{x,f}(S \cup \{p\}) = \nu_{x,f}(S)$ for $S \subseteq [n_{\texttt{worker}}]$ then $\phi_p = 0$.
- Additivity: for two value functions $\nu^1_{x,f}$ and $\nu^2_{x,f}$, $\phi_j(i, \nu^1_{x,f} + \nu^2_{x,f}) = \phi_j(i, \nu^1_{x,f}) + \phi_j(i, \nu^2_{x,f})$.
- Efficiency: $\sum_{j=1}^{n_{\texttt{worker}}} \phi_j(\nu) = \nu_{x,f}([n_{\texttt{worker}}])$.

Where the Shapley value is interesting for a crowdsourcing problem, is that if a worker does not help the classifier to predict the label, then its Shapley value will be close to zero. And, two workers with similar contributions will obtain similar Shapley values.

3.2 Shapley label aggregation strategy

We introduce the following label aggregation algorithm based on Shapley values. It is based on the Expectation-Maximization procedure where we iteratively estimate the labels and the workers' skills until convergence – *e.g.* stabilization of the skills. Given a current estimation of the labels and a classifier f, we consider the skill of each worker as their total contribution to the prediction. The contribution of a worker $j_0 \in [n_{\texttt{worker}}]$ on a single task $i_0 \in [n_{\texttt{task}}]$ is given as $|\phi_{j_0}(i_0)| \in \mathbb{R}_+$.

First, note that in Algorithm 2 the participation of each worker is linked to their total contribution. There is no upper bound on the skill estimation with contrib as we value a worker who answers multiple times. However, if they answer many labels randomly, their Shapley value is close to zero and their total contribution is low. As it can be seen in Figure 1, as the Shapley values can be used for feature importance in prediction, it can also identify which workers are the most important for the final label aggregation and could be the center of more analysis. And, at the same time, it can identify which workers are not contributing to the final label.

Algorithm 2 Shapley label aggregation strategy.

Input: classifier f, $t_{\max} > 0$ maximum number of iterations
Initialize labels with MV: $\hat{y}_i^0 = \mathrm{WMV}(i, \mathbf{1}_{n_{\texttt{worker}} \times K})$ for each task $i \in [n_{\texttt{task}}]$
for $t = 0, \dots, t_{\max} - 1$ **do**
  Train classifier f on $\{(\mathbf{Y}, \hat{y}_i^t)_{i \in [n_{\texttt{task}}]}\}$ (workers' answers and current aggregated labels)
  Compute Shapley values' total contribution of each worker $j \in [n_{\texttt{worker}}]$:

$$\mathrm{contrib}(j) = \sum_{i=1}^{n_{\texttt{task}}} |\phi_j(i)| \ .$$

  Update weights: $W_{j,:}^t = \mathrm{contrib}(j) \mathbf{1}_K$ for each worker $j \in [n_{\texttt{worker}}]$
  Update labels with WMV: $\hat{y}_i^{t+1} = \mathrm{WMV}(i, W^t)$ for $i \in [n_{\texttt{task}}]$.
end for
Output: $\hat{y}_i^{\mathrm{shapley}} = \hat{y}_i^{t_{\max}}$.

3.3 Implementation

To compute Shapley values, we use the `Shap` library (Lundberg and Lee, 2017). We choose an XGBOOST classifier (Chen and Guestrin, 2016) as the classifier f. In practice, the value function $\nu_{x,f}$ evaluated at a set $S \subseteq [n_{\texttt{worker}}]$ defined in Equation (1) is estimated using the `TreeSHAP` algorithm (Lundberg et al., 2018). Label aggregation strategies are implemented in Python using the `crowd-kit`[2] or `peerannot` (Lefort et al., 2023) libraries. The XGBOOST classifier is known to have an extensive number of hyperparameters to tune. To choose them, we first use the `optuna` (Akiba et al., 2019) library to tune over a 3-fold cross-validation of best hyperparameters for the set of tasks and label $\{(x_i, \hat{y}_i^0)_{i \in [n_{\texttt{task}}]}\}$. This random search includes the trees' depth, learning rate, the number of trees, the minimum child weight, the subsampling proportion and regularization parameter. These best parameters are then used in Algorithm 2 to iteratively train the XGBOOST model with the current label estimates. Note that this hyperparameter search can be costly in computation time.

3.4 Evaluation metrics

We evaluate the performance of the Shapley label aggregation strategy using the accuracy and the F1 score. More precisely, each of the real datasets considered provides a – partially known – ground truth. This test set is denoted $\mathcal{D}_{\texttt{test}} = \{(x_i, y_i^\star)\}_{i=1}^{n_{\texttt{test}}}$ and is used to evaluate the accuracy of the label aggregation strategies. This ground truth is not used during the aggregation, only at evaluation time. The accuracy of the aggregation strategy `agg` is the proportion of correctly predicted labels $(\hat{[y]}_i^{\texttt{agg}})_{i=1}^{n_{\texttt{test}}}$ over the total number of tasks in $\mathcal{D}_{\texttt{test}}$ with ground truth in $y^\star = (y_i^\star)_{i=1}^{n_{\texttt{test}}}$:

$$\mathrm{Accuracy}(\hat{y}^{\texttt{agg}}, y^\star) = \frac{1}{n_{\texttt{test}}} \sum_{i=1}^{n_{\texttt{test}}} \mathbb{1}(\hat{y}_i^{\texttt{agg}} = y_i^\star) \ .$$

We take into account the possible class imbalance by presenting a macro-average F1 score. This score is commonly used to evaluate the balance between precision and recall in classification tasks. It provides a measure of the quality of the label aggregation strategy when dealing with imbalanced datasets. Denoting respectively TP_k, FP_k and FN_k the true positives, false positives and false negatives related to the class $k \in [K]$, the macro averaged F1-score writes

$$\mathrm{F1} = \frac{1}{K} \sum_{k=1}^{K} \frac{\mathrm{TP}_k}{\mathrm{TP}_k + 0.5(\mathrm{FN}_k + \mathrm{FP}_k)} \ .$$

[2] `https://github.com/Toloka/crowd-kit`

These metrics are evaluated on several real datasets. The BlueBirds dataset (Welinder et al., 2010) is a binary classification ($K = 2$) dataset with $n_{\texttt{task}} = 108$ tasks and $n_{\texttt{worker}} = 39$ workers. Workers were asked to identify if there was a blue bird of the species Indigo Bunting in the presented image. The Temporal Ordering (Temp) (Snow et al., 2008) dataset is a binary classification dataset with $n_{\texttt{task}} = 462$ tasks and $n_{\texttt{worker}} = 76$ workers. Workers were presented with sentences with events and asked if the event presented in the first sentence occurred before the one in the second sentence. The LabelMe dataset (Rodrigues and Pereira, 2018) consists of $n_{\texttt{task}} = 1000$ images shown to $n_{\texttt{worker}} = 77$ workers. The task was to classify the image into one of the $K = 8$ classes. Finally, the Music dataset (Rodrigues et al., 2014) is a music genre classification for $n_{\texttt{task}} = 700$ samples annotated by $n_{\texttt{worker}} = 44$ workers. There are $K = 10$ different music genres to be assigned to each task.

4 Results

4.1 Performance on real datasets

We evaluate the Shapley label aggregation strategy on several real datasets. From Table 1, we see that using Shapley-based weights in the WMV strategy outperforms other strategies in terms of accuracy and F1 score. Note that the KOS strategy can not be applied to the datasets considered with $K > 2$ as it is only suited for binary classification tasks.

Table 1: Accuracy and F1 Score of the WMV-based label aggregation strategies over 4 real datasets: BlueBirds, Temp, LabelMe and Music. We obtain equal or better performance in accuracy and F1 score for 3 out of the 4 datasets.

Strategy	**BlueBirds** ($K = 2$)		**Temp** ($K = 2$)		**LabelMe** ($K = 8$)		**Music** ($K = 10$)	
	Accuracy	**F1 Score**	**Accuracy**	**F1 Score**	**Accuracy**	**F1 Score**	**Accuracy**	**F1 Score**
MV	0.759	0.742	0.939	0.938	0.769	0.765	0.711	0.744
WAWA	0.759	0.742	**0.945**	**0.945**	0.770	0.766	0.797	0.801
ZBS	0.648	0.623	0.943	0.943	0.774	0.769	**0.800**	**0.804**
WDS	0.759	0.736	**0.945**	**0.945**	0.736	0.724	0.794	0.797
KOS	0.722	0.678	0.569	0.384	—	—	—	—
M-MSR	0.639	0.578	0.924	0.922	0.767	0.761	0.742	0.744
Shapley	**0.805**	**0.794**	**0.945**	**0.945**	**0.777**	**0.762**	0.760	0.765

4.2 More information on workers

Using the Shapley values as workers' contribution, we can also provide more information on the workers' reliability. Let us explore the BlueBirds dataset Shapley weights as an example. From Figure 2, we see that worker 34 has the best overall contribution to the final label. Note that this order of contribution given by Shapley values is in agreement with the accuracy of the workers even though Shapley values are not directly linked to the accuracy of a model. Indeed, as we know the ground truth, we can compute the accuracy of each worker. The accuracy of worker 0 is 0.80, worker 34 is 0.79 and worker 33 is 0.44 (random answers). The worker 22 – not represented in Figure 2 as they are not a main contributor – has an accuracy of 0.42 – worse than a random guess – and an average absolute contribution of 0.03 to the final label. This worker is indeed not contributing to the final label given the poor quality of their answers.

Worker 18 has a better identification for class 1 than class 0. However, as we use a single scalar value that is class-blind in Algorithm 2 to aggregate the label in the WMV, this asymmetrical contribution is not taken into account. This is a limitation of the current Shapley label aggregation strategy.

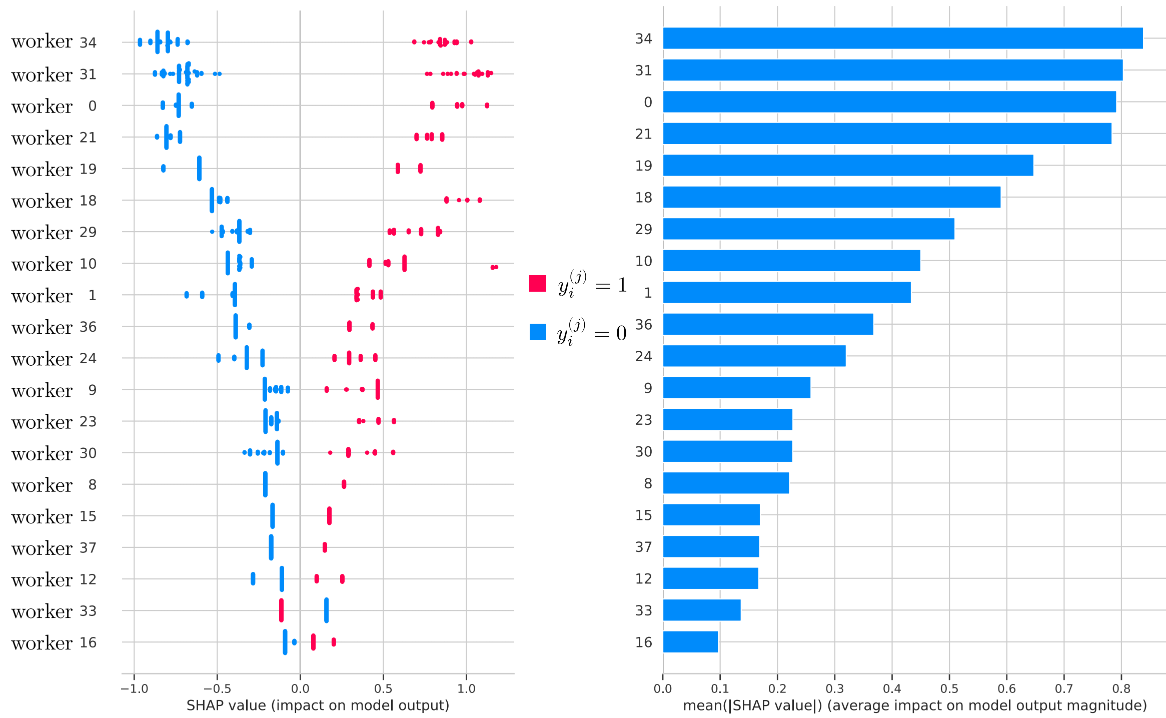

Figure 2: Summary of the main contributing workers for the BlueBirds dataset using Shapley values. Left: Worker 34 has the highest contribution to the final label – either for class 0 or 1, followed by worker 31. Worker 18 has better identification for class 1 than class 0. Right: Average impact of each worker on the final predicted label by the XGBOOST classifier. This worker's contribution scalar value – used in Algorithm 2 weight $W_{j,:}$ – does not allow to differentiate between classes.

5 Conclusion

We introduced a new label aggregation strategy based on Shapley values for crowdsourcing classification tasks. In the framework of weighted majority votes, we used the Shapley values as workers' weights to aggregate the labels. We showed that this strategy outperforms other weighted majority vote strategies on real datasets in terms of accuracy and F1 score. Moreover, we discussed how Shapley-based skills can be used to explore workers' reliability. However, this strategy is limited by the scalar value used to aggregate the labels in the WMV strategy. Not unlike most other WMV strategies, it does not take into account per-class skills. An extension of this work would be to consider multidimensional skills based on Shapley values for each worker, allowing for a per-class contribution to the final label and a finer estimation of workers' skills.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902.

Aumann, R. J. (1994). Economic applications of the shapley value. In *Game-theoretic methods in general equilibrium analysis*, pages 121–133. Springer.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Cohen, S., Dror, G., and Ruppin, E. (2007). Feature selection via coalitional game theory. *Neural computation*, 19(7):1939–1961.

Dawid, A. and Skene, A. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *J. R. Stat. Soc. Ser. C. Appl. Stat.*, 28(1):20–28.

Engelbrecht, G. and Vos, A. (2009). On the use of the shapley value in political conflict resolution. *Scientia Militaria: South African Journal of Military Studies*, 37(1).

Ghorbani, A., Zou, J., and Esteva, A. (2022). Data shapley valuation for efficient batch active learning. In *2022 56th Asilomar Conference on Signals, Systems, and Computers*, pages 1456–1462. IEEE.

Hara, K., Adams, A., Milland, K., Savage, S., Callison-Burch, C., and Bigham, J. P. (2018). A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–14.

Ipeirotis, P. G., Provost, F., and Wang, J. (2010). Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67.

Karger, D., Oh, S., and Shah, D. (2011). Iterative learning for reliable crowdsourcing systems. *Advances in neural information processing systems*, 24.

Lefort, T., Charlier, B., Joly, A., and Salmon, J. (2023). Peerannot: classification for crowdsourced image datasets with Python. working paper or preprint.

Li, H. and Yu, B. (2014). Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086*.

Limited, A. (2021). Calculating worker agreement with aggregate (wawa).

Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and computation*, 108(2):212–261.

Lundberg, S. M., Erion, G. G., and Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Ma, Q. and Olshevsky, A. (2020). Adversarial crowdsourcing through robust rank-one matrix completion. In *NeurIPS*, volume 33, pages 21841–21852.

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Owen, A. B. (2014). Sobol'indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251.

Owen, A. B. and Prieur, C. (2017). On shapley value for measuring importance of dependent inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):986–1002.

Pearl, J. (1986). *Fusion, Propagation, and Structuring in Belief Networks*, page 139–188. Association for Computing Machinery, New York, NY, USA, 1 edition.

Rodrigues, F. and Pereira, F. (2018). Deep learning from crowds. In *AAAI*, volume 32.

Rodrigues, F., Pereira, F., and Ribeiro, B. (2014). Gaussian process classification and active learning with multiple annotators. In *ICML*, pages 433–441. PMLR.

Ross, J., Zaldivar, A., Irani, L., and Tomlinson, B. (2009). Who are the turkers? worker demographics in amazon mechanical turk. *Department of Informatics, University of California, Irvine, USA, Tech. Rep*, 49.

Schoch, S., Xu, H., and Ji, Y. (2022). Cs-shapley: class-wise shapley values for data valuation in classification. *Advances in Neural Information Processing Systems*, 35:34574–34585.

Shapley, L. S. (1953). A value for n-person games. In Kuhn, H. W. and Tucker, A. W., editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, Princeton.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. (2008). Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics.

Welinder, P., Branson, S., Belongie, S., and Perona, P. (2010). The Multidimensional Wisdom of Crowds. In *NIPS*.

Whitehill, J., Wu, T., Bergsma, J., Movellan, J., and Ruvolo, P. (2009). Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NeurIPS*, volume 22.