



HAL
open science

Weighted majority vote using Shapley values in crowdsourcing

Tanguy Lefort, Benjamin Charlier, Alexis Joly, Joseph Salmon

► **To cite this version:**

Tanguy Lefort, Benjamin Charlier, Alexis Joly, Joseph Salmon. Weighted majority vote using Shapley values in crowdsourcing. CAp 2024 - Conférence sur l'Apprentissage Automatique, Jul 2024, Lille, France. hal-04573727

HAL Id: hal-04573727

<https://hal.science/hal-04573727v1>

Submitted on 13 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Weighted majority vote using Shapley values in crowdsourcing

Tanguy Lefort^{*1}, Benjamin Charlier², Alexis Joly³, and Joseph Salmon⁴

¹University of Montpellier, IMAG, CNRS, LIRMM, Inria

²University of Montpellier, IMAG, CNRS

³LIRMM, Inria

⁴University of Montpellier, IMAG, CNRS, Institut Universitaire de France (IUF)

May 13, 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 Turk¹. 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/>

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

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

45 In this work, we aim to propose a new weight for WMV based on the Shapley values (Shapley,
 46 1953). The Shapley value is a concept originating from cooperative game theory that has been used
 47 in various fields such as economics (Aumann, 1994), political science (Engelbrecht and Vos, 2009),
 48 statistics (Owen, 2014) and machine learning for explainability (Lundberg and Lee, 2017) or feature
 49 selection (Cohen et al., 2007). Shapley values have been used in the context of data valuation in
 50 classification (Schoch et al., 2022) and active learning (Ghorbani et al., 2022). Here, we propose to
 51 extend it to classification in a crowdsourcing setting.

52 If we consider that each worker is a feature and each task is a sample point, given a classifier, the
 53 Shapley value explains the contribution of each worker to the predicted outcome at each queried task
 54 (Molnar, 2020; Rozemberczki et al., 2022). Shapley values are used as worker importance indicators
 55 that can handle interactions between workers’ answers (Owen and Prieur, 2017; Lundberg and Lee,
 56 2017). We propose a study of their usage as interpretable weights in weighted majority votes for
 57 crowdsourcing classification tasks.

58 2 Notation and related work

59 **Notation.** We consider classical multi-class learning notation, with input in \mathcal{X} and labels in $[K] :=$
 60 $\{1, \dots, K\}$. There are n_{task} available, denoted $x_1, \dots, x_{n_{\text{task}}}$, to be labeled by n_{worker} workers. The
 61 set of n_{task} tasks with their associated true labels is $\mathcal{D} = \{(x_1, y_1^*), \dots, (x_{n_{\text{task}}}, y_{n_{\text{task}}}^*)\}$. Denote
 62 $\mathbf{Y} \in [K]^{n_{\text{task}} \times n_{\text{worker}}}$ the matrix of the workers’ answers for each task. The true labels are unob-
 63 served but crowdsourced labels are provided by the workers. We write $\mathcal{A}(x_i) = \{j \in [n_{\text{worker}}] :$
 64 $\text{worker } j \text{ labeled task } x_i\}$ the **annotators set** of a task x_i . For a task x_i and each $j \in \mathcal{A}(x_i)$, we
 65 denote $y_i^{(j)} \in [K]$ the label answered by worker j . Given an aggregation strategy **agg** (such as MV),
 66 we denote aggregated label $\hat{y}_i^{\text{agg}} \in [K]$. For any set \mathcal{S} , we write $|\mathcal{S}|$ for its cardinality. The indicator
 67 function is denoted $\mathbf{1}(\cdot)$. The matrix full of ones of size $n \times m$ is denoted $\mathbf{1}_{n \times m}$. The row of a matrix
 68 M indexed by i is denoted $M_{i,:}$ and the column indexed by j is $M_{:,j}$.

69 On released datasets, to compute performance metrics, partial true labels are made available. We
 70 denote $\mathcal{D}_{\text{train}}$ the set of tasks with their true labels unknown and $\mathcal{D}_{\text{test}}$ the set of tasks with known
 71 true labels. Note that these true labels are only used at test time. Both workers and aggregation
 72 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 $\nu_{x,f} :$
 $2^{[n_{\text{worker}}]} \rightarrow \mathbb{R}$ such that for any set of workers $S \subset [n_{\text{worker}}]$ 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 $S \subseteq [n_{\text{worker}}]$ of workers is defined – with x_S the answers of the selected workers in
 S – as

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

73 In practice, this quantity has to be estimated, for instance using the **TreeSHAP** algorithm (Lundberg
 74 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:

$$\forall i \in [n_{\text{task}}], \quad \hat{y}_i = \text{WMV}(i, W) := \arg \max_{k \in [K]} \sum_{j \in \mathcal{A}(x_i)} W_{j,k} \mathbf{1}(y_i^{(j)} = k), \quad (2)$$

75 where $W \in \mathbb{R}^{n_{\text{worker}} \times K}$ is the matrix assigning the weight of worker j when answering class k . We
 76 denote $W_{j,k} \in \mathbb{R}$ the weight of worker j for class k . This weight matrix is the cornerstone of each
 77 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:

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

78 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 :

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

79 It allows us to instantiate a weight that can vary for each worker (but not per task) and it
 80 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.

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

85

- 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:

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

- M-MSR (Ma and Olshevsky, 2020): The Matrix Mean-Subsequence-Reduced strategy considers the reliability of all workers as a vector $s \in \mathbb{R}^{n_{\text{worker}}}$. 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.*:

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

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:

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

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

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

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

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

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

- 88 • KOS (Karger et al., 2011): Only set for binary classification $K = 2$, the KOS strategy comes
89 from a graph-theory perspective. The worker’s weight is estimated iteratively inspired by the
90 belief propagation algorithm (Pearl, 1986) to look at the worker agreements on neighboring
91 tasks. An edge from a worker to a task indicates that the task was answered by the worker. In
92 an EM fashion, a worker message – the reliability of worker j for task i – is stored in a matrix
93 $W \in \mathbb{R}^{n_{\text{worker}} \times n_{\text{task}}}$. Then, the task message – the likelihood of the task i to be positive – is sent
94 to the workers as a vector of $\mathbb{R}^{n_{\text{task}}}$. The final label is the sign of the weighted majority votes,
95 with the weight of worker j ’s answer to task i being equal to $W_{j,i}$.

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

101 3 Shapley label aggregation for crowdsourcing

102 3.1 Preliminaries on Shapley values

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

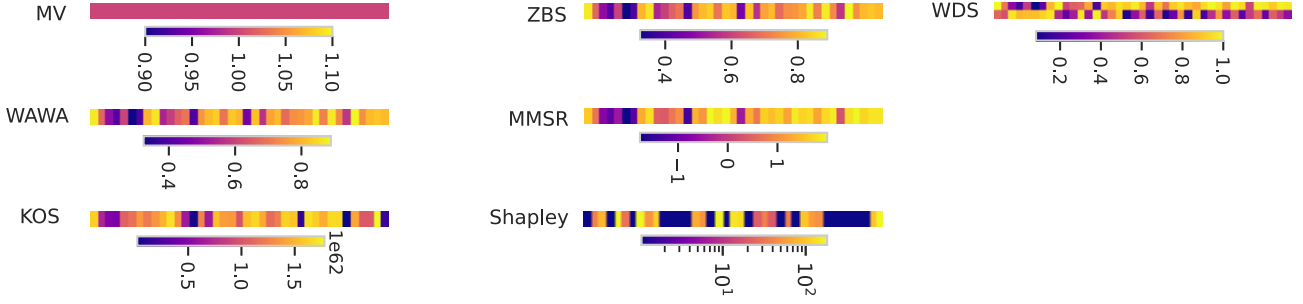


Figure 1: Example of weight matrices $W \in \mathbb{R}^{n_{\text{worker}} \times K}$ obtained for the BlueBirds ($n_{\text{worker}} = 39, K = 2$) dataset using the presented strategies and our in Algorithm 2. We transpose them for ease of readability (each row is a class, each column a worker). Only 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 $\nu_{x_i, f}(\cdot) : 2^{[n_{\text{worker}}]} \rightarrow \mathbb{R}$ 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 $[n_{\text{worker}}] \setminus \{j\}$:

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

106 When there is no ambiguity over the value function, we adopt the standard notation abuse $\phi_j(i, \nu) =$
 107 $\phi_j(i)$.

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

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

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

116 3.2 Shapley label aggregation strategy

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

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

Algorithm 2 Shapley label aggregation strategy.

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

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

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

129 3.3 Implementation

130 To compute Shapley values, we use the `Shap` library (Lundberg and Lee, 2017). We choose an XG-
131 BOOST classifier (Chen and Guestrin, 2016) as the classifier f . In practice, the value function $\nu_{x,f}$
132 evaluated at a set $S \subseteq [n_{\text{worker}}]$ defined in Equation (1) is estimated using the `TreesHAP` algorithm
133 (Lundberg et al., 2018). Label aggregation strategies are implemented in Python using the `crowd-kit`²
134 or `peerannot` (Lefort et al., 2023) libraries. The XGBOOST classifier is known to have an extensive
135 number of hyperparameters to tune. To choose them, we first use the `optuna` (Akiba et al., 2019)
136 library to tune over a 3-fold cross-validation of best hyperparameters for the set of tasks and label
137 $\{(x_i, \hat{y}_i^0)_{i \in [n_{\text{task}}]}\}$. This random search includes the trees' depth, learning rate, the number of trees,
138 the minimum child weight, the subsampling proportion and regularization parameter. These best
139 parameters are then used in Algorithm 2 to iteratively train the XGBOOST model with the current
140 label estimates. Note that this hyperparameter search can be costly in computation time.

141 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 $\mathcal{D}_{\text{test}} = \{(x_i, y_i^*)\}_{i=1}^{n_{\text{test}}}$ 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 $(\hat{y}_i^{\text{agg}})_{i=1}^{n_{\text{test}}}$ over the total number of tasks in $\mathcal{D}_{\text{test}}$ with ground truth in $y^* = (y_i^*)_{i=1}^{n_{\text{test}}}$:

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

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

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

²<https://github.com/Toloka/crowd-kit>

142 These metrics are evaluated on several real datasets. The BlueBirds dataset (Welinder et al., 2010)
 143 is a binary classification ($K = 2$) dataset with $n_{\text{task}} = 108$ tasks and $n_{\text{worker}} = 39$ workers. Workers
 144 were asked to identify if there was a blue bird of the species Indigo Bunting in the presented image.
 145 The Temporal Ordering (Temp) (Snow et al., 2008) dataset is a binary classification dataset with
 146 $n_{\text{task}} = 462$ tasks and $n_{\text{worker}} = 76$ workers. Workers were presented with sentences with events and
 147 asked if the event presented in the first sentence occurred before the one in the second sentence. The
 148 LabelMe dataset (Rodrigues and Pereira, 2018) consists of $n_{\text{task}} = 1000$ images shown to $n_{\text{worker}} = 77$
 149 workers. The task was to classify the image into one of the $K = 8$ classes. Finally, the Music
 150 dataset (Rodrigues et al., 2014) is a music genre classification for $n_{\text{task}} = 700$ samples annotated by
 151 $n_{\text{worker}} = 44$ workers. There are $K = 10$ different music genres to be assigned to each task.

152 4 Results

153 4.1 Performance on real datasets

154 We evaluate the Shapley label aggregation strategy on several real datasets. From Table 1, we see that
 155 using Shapley-based weights in the WMV strategy outperforms other strategies in terms of accuracy
 156 and F1 score. Note that the KOS strategy can not be applied to the datasets considered with $K > 2$
 157 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

158 Note that the Music dataset is known to be more challenging than the other three datasets due to
 159 high variability in the workers’ answers. This is reflected in the Shapley aggregation as the weights
 160 used are based on the impacts of the workers’ answers in the current aggregation, and if the worker’s
 161 answers are less reliable, so is their interpretation.

162 4.2 More information on workers

163 Using the Shapley values as workers’ contribution, we can also provide more information on the
 164 workers’ reliability. Let us explore the BlueBirds dataset Shapley weights as an example. From
 165 Figure 2, we see that worker 34 has the best overall contribution to the final label. Note that this
 166 order of contribution given by Shapley values is in agreement with the accuracy of the workers even
 167 though Shapley values are not directly linked to the accuracy of a model. Indeed, as we know the
 168 ground truth, we can compute the accuracy of each worker. The accuracy of worker 0 is 0.80, worker
 169 34 is 0.79 and worker 33 is 0.44 (random answers). The worker 22 – not represented in Figure 2 as
 170 they are not a main contributor – has an accuracy of 0.42 – worse than a random guess – and an
 171 average absolute contribution of 0.03 to the final label. This worker is indeed not contributing to the
 172 final label given the poor quality of their answers.

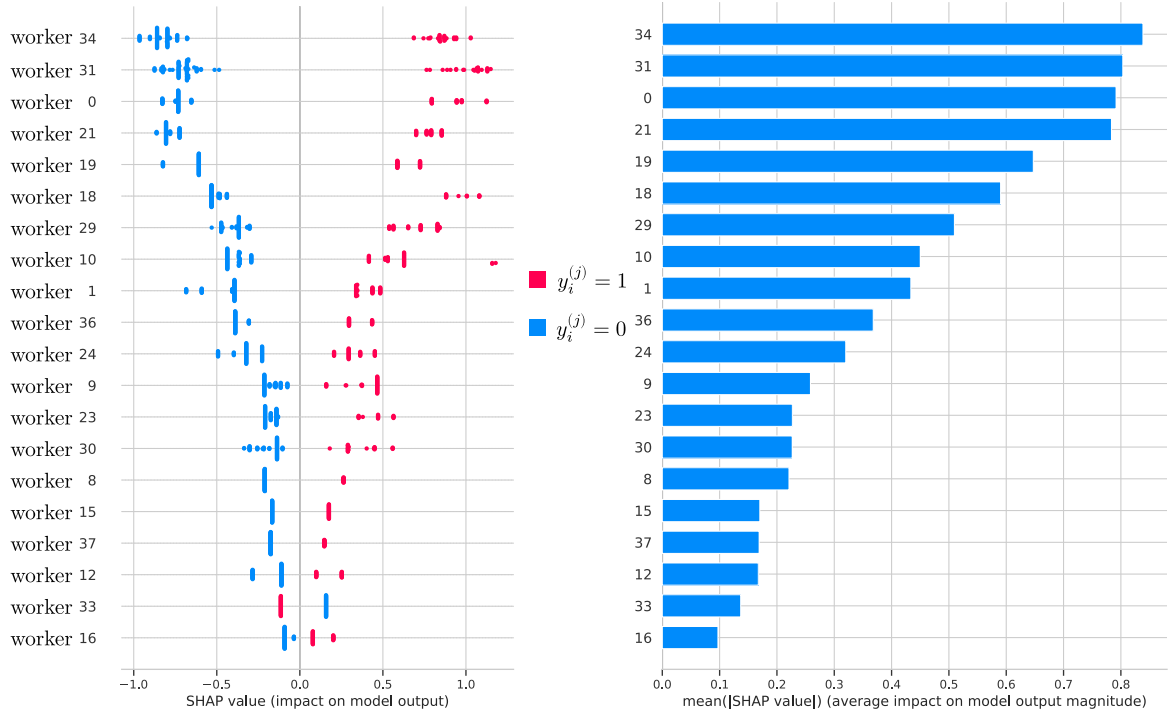


Figure 2: Summary of the main contributing workers for the BlueBirds dataset using Shapley values. Left: Worker 34 has the highest contribution to the final label – either for class 0 or 1, followed by worker 31. Worker 18 has better identification for class 1 than class 0. Right: Average impact of each 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.

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

176 5 Conclusion

177 We introduced a new label aggregation strategy based on Shapley values for crowdsourcing classifi-
 178 cation tasks. In the framework of weighted majority votes, we used the Shapley values as workers’
 179 weights to aggregate the labels. We showed that this strategy outperforms other weighted major-
 180 ity vote strategies on real datasets in terms of accuracy and F1 score. Moreover, we discussed how
 181 Shapley-based skills can be used to explore workers’ reliability. However, this strategy is limited by the
 182 scalar value used to aggregate the labels in the WMV strategy. Not unlike most other WMV strate-
 183 gies, it does not take into account per-class skills. An extension of this work would be to consider
 184 multidimensional skills based on Shapley values for each worker, allowing for a per-class contribution
 185 to the final label and a finer estimation of workers’ skills. Also, the impact of the model used should
 186 be studied. However, a previous study should be conducted to only test models that can handle very
 187 sparse categorical data – as workers typically answer only a few tasks – and for which we can pro-
 188 vide reliable Shapley values. This study focuses on weight majority vote strategies, other aggregation
 189 strategies – such as Dawid and Skene (1979) – often outperform WMV strategies with enough votes.

190 References

- 191 Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation
192 hyperparameter optimization framework. *CoRR*, abs/1907.10902.
- 193 Aumann, R. J. (1994). Economic applications of the shapley value. In *Game-theoretic methods in*
194 *general equilibrium analysis*, pages 121–133. Springer.
- 195 Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the*
196 *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD
197 '16, pages 785–794, New York, NY, USA. ACM.
- 198 Cohen, S., Dror, G., and Ruppin, E. (2007). Feature selection via coalitional game theory. *Neural*
199 *computation*, 19(7):1939–1961.
- 200 Dawid, A. and Skene, A. (1979). Maximum likelihood estimation of observer error-rates using the EM
201 algorithm. *J. R. Stat. Soc. Ser. C. Appl. Stat.*, 28(1):20–28.
- 202 Engelbrecht, G. and Vos, A. (2009). On the use of the shapley value in political conflict resolution.
203 *Scientia Militaria: South African Journal of Military Studies*, 37(1).
- 204 Ghorbani, A., Zou, J., and Esteva, A. (2022). Data shapley valuation for efficient batch active learning.
205 In *2022 56th Asilomar Conference on Signals, Systems, and Computers*, pages 1456–1462. IEEE.
- 206 Hara, K., Adams, A., Milland, K., Savage, S., Callison-Burch, C., and Bigham, J. P. (2018). A data-
207 driven analysis of workers’ earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI*
208 *conference on human factors in computing systems*, pages 1–14.
- 209 Ipeirotis, P. G., Provost, F., and Wang, J. (2010). Quality management on amazon mechanical turk.
210 In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67.
- 211 Karger, D., Oh, S., and Shah, D. (2011). Iterative learning for reliable crowdsourcing systems. *Ad-*
212 *vances in neural information processing systems*, 24.
- 213 Lefort, T., Charlier, B., Joly, A., and Salmon, J. (2023). Peerannot: classification for crowdsourced
214 image datasets with Python. working paper or preprint.
- 215 Li, H. and Yu, B. (2014). Error rate bounds and iterative weighted majority voting for crowdsourcing.
216 *arXiv preprint arXiv:1411.4086*.
- 217 Limited, A. (2021). Calculating worker agreement with aggregate (wawa).
- 218 Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and*
219 *computation*, 108(2):212–261.
- 220 Lundberg, S. M., Erion, G. G., and Lee, S.-I. (2018). Consistent individualized feature attribution for
221 tree ensembles. *arXiv preprint arXiv:1802.03888*.
- 222 Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances*
223 *in neural information processing systems*, 30.
- 224 Ma, Q. and Olshevsky, A. (2020). Adversarial crowdsourcing through robust rank-one matrix com-
225 pletion. In *NeurIPS*, volume 33, pages 21841–21852.
- 226 Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

- 227 Owen, A. B. (2014). Sobol’indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantifi-*
228 *cation*, 2(1):245–251.
- 229 Owen, A. B. and Prieur, C. (2017). On shapley value for measuring importance of dependent inputs.
230 *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):986–1002.
- 231 Pearl, J. (1986). *Fusion, Propagation, and Structuring in Belief Networks*, page 139–188. Association
232 for Computing Machinery, New York, NY, USA, 1 edition.
- 233 Rodrigues, F. and Pereira, F. (2018). Deep learning from crowds. In *AAAI*, volume 32.
- 234 Rodrigues, F., Pereira, F., and Ribeiro, B. (2014). Gaussian process classification and active learning
235 with multiple annotators. In *ICML*, pages 433–441. PMLR.
- 236 Ross, J., Zaldivar, A., Irani, L., and Tomlinson, B. (2009). Who are the turkers? worker demographics
237 in amazon mechanical turk. *Department of Informatics, University of California, Irvine, USA, Tech.*
238 *Rep*, 49.
- 239 Rozemberczki, B., Watson, L., Bayer, P., Yang, H.-T., Kiss, O., Nilsson, S., and Sarkar, R. (2022).
240 The shapley value in machine learning. *arXiv preprint arXiv:2202.05594*.
- 241 Schoch, S., Xu, H., and Ji, Y. (2022). Cs-shapley: class-wise shapley values for data valuation in
242 classification. *Advances in Neural Information Processing Systems*, 35:34574–34585.
- 243 Shapley, L. S. (1953). A value for n-person games. In Kuhn, H. W. and Tucker, A. W., editors,
244 *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, Princeton.
- 245 Snow, R., O’Connor, B., Jurafsky, D., and Ng, A. (2008). Cheap and fast - but is it good? evaluating
246 non-expert annotations for natural language tasks. In *Conference on Empirical Methods in Natural*
247 *Language Processing*, pages 254–263. Association for Computational Linguistics.
- 248 Welinder, P., Branson, S., Belongie, S., and Perona, P. (2010). The Multidimensional Wisdom of
249 Crowds. In *NIPS*.
- 250 Whitehill, J., Wu, T., Bergsma, J., Movellan, J., and Ruvolo, P. (2009). Whose vote should count
251 more: Optimal integration of labels from labelers of unknown expertise. In *NeurIPS*, volume 22.