Weighting Areas Under the Margin in crowdsourced datasets

Tanguy Lefort, Benjamin Charlier, Alexis Joly, Joseph Salmon

Weighting Areas Under the Margin in crowdsourced datasets

Tanguy Lefort [1] & Benjamin Charlier [2] & Alexis Joly [3] & Joseph Salmon [3]

[1] *Univ. Montpellier, CNRS, IMAG, Inria, LIRMM, France tanguy.lefort@umontpellier.fr*
[2] *Univ. Montpellier, CNRS, IMAG, France benjamin.charlier@umontpellier.fr*
[4] *Inria, LIRMM, Univ. Montpellier, CNRS, France, alexis.joly@inria.fr*
[4] *Univ. Montpellier, CNRS, IMAG, IUF, France joseph.salmon@umontpellier.fr*

Résumé. En apprentissage supervisé — par exemple en classification d'images — les jeux de données modernes sont généralement étiquetés par une foule de travailleurs. Des erreurs d'étiquetage peuvent se produire en fonction de la capacité des travailleurs et de la difficulté d'identification des tâches. Certaines tâches sont intrinsèquement ambiguës et peuvent induire en erreur les travailleurs les plus experts, ce qui nuit à l'étape d'apprentissage. Dans un cadre standard d'apprentissage supervisé — avec une étiquette par tâche — l'aire sous la marge (AUM) est utilisée pour identifier les données mal étiquetées. Nous adaptons l'AUM pour identifier les tâches ambiguës dans les scénarios d'apprentissage par la foule, en introduisant l'AUM pondérée (WAUM). Le WAUM est une moyenne des AUMs pondérés par des scores dépendant de la tâche. Nous montrons que le WAUM peut aider à écarter les tâches ambiguës de l'ensemble d'apprentissage, ce qui conduit à une meilleure généralisation ou performance de calibration.

Mots-clés. Apprentissage participatif, ambiguïté des tâches

Abstract. In supervised learning — for instance in image classification — modern massive datasets are commonly labeled by a crowd of workers. Labeling errors can happen because of the workers abilities or tasks identification difficulty. Some intrinsically ambiguous tasks might fool expert workers, which could eventually be harmful to the learning step. In a standard supervised learning setting — with one label per task — the Area Under the Margin (AUM) is tailored to identify mislabeled data. We adapt the AUM to identify ambiguous tasks in crowdsourced learning scenarios, introducing the Weighted AUM (WAUM). The WAUM is an average of AUMs weighted by task-dependent scores. We show that the WAUM can help discard ambiguous tasks from the training set, leading to better generalization or calibration performance.

Keywords. Crowdsourcing, Task ambiguity

1 Introduction

Crowdsourcing labels for supervised learning has become quite common in the last two decades, notably for image classification datasets. Using a crowd of workers is fast, simple

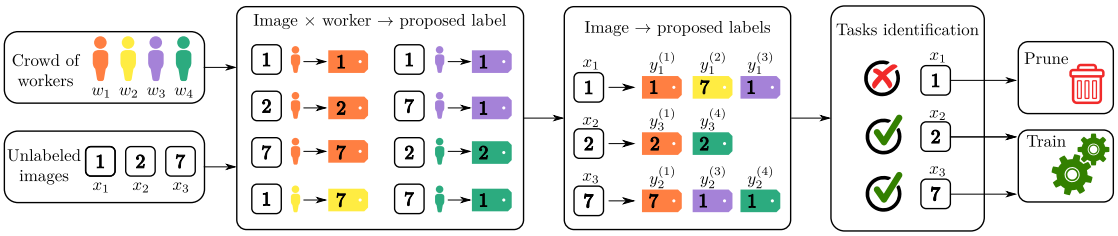


Figure 1: Learning with crowdsourcing labels: from label collection with a crowd to training on a pruned dataset. High ambiguity from either crowd workers or tasks intrinsic difficulty can lead to mislabeled data and harm generalization performance. To illustrate our notation, here the set of tasks annotated by worker w_3 is $\mathcal{T}(w_3) = \{1, 3\}$ while the set of workers annotating task x_3 is $\mathcal{A}(x_3) = \{1, 3, 4\}$.

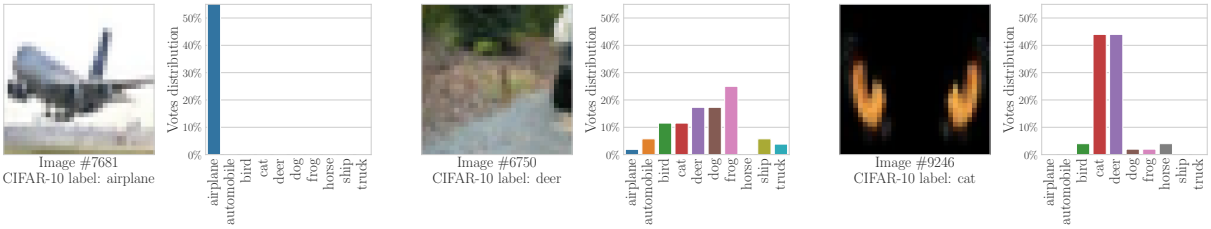
(see Fig. 1) and less expensive than using experts. Furthermore, aggregating crowdsourced labels instead of working directly with a single one enables modeling the sources of possible ambiguities and directly taking them into account at training (Aitchison, 2021). With deep neural networks nowadays common in many applications, both the architectures and data quality have a direct impact on the model performance (Müller et al., 2019; Northcutt et al., 2021b) and on calibration (Guo et al., 2017). Yet, depending on the crowd and platform's control mechanisms, the quality of the labels might be low, with possibly many mislabeled instances (Müller and Markert, 2019), leading to poor generalization (Snow et al., 2008).

Popular label aggregation schemes take into account the uncertainty related to workers' abilities: for example by estimating confusions between classes, or using a latent variable representing each worker trust (Dawid and Skene, 1979; Kim and Ghahramani, 2012; Sinha et al., 2018; Camilleri and Williams, 2019). This leads to scoring workers without taking into account the inherent difficulty of the tasks at stake. Inspired by the Item Response Theory (IRT) from Birnbaum (1968), Whitehill et al. (2009) combined both the task difficulty and the worker's ability in a feature-blind fashion for label aggregation. All the feature-blind strategies only require the labels but not the associated features[1]. For instance, GLAD (Whitehill et al., 2009) estimates a task difficulty without the actual task: its estimation only relies on the collected labels and not on the tasks themselves (in image-classification settings, this means the images are not considered for evaluating the task difficulty). In the classical supervised learning setting, the labels are said to be *hard* — *i.e.,* a Dirac mass on one class. Multiple crowdsourced labels induce *soft* labels — *i.e.,* probability distributions over the classes — for each task. Our motivation is to identify ambiguous tasks from their associated features, hence discarding hurtful tasks (such as the ones illustrated on Fig. 2b and Fig. 2b).

Recent works on data-cleaning in supervised learning (Han et al., 2019; Pleiss et al., 2020; Northcutt et al., 2021a) have shown that some images might be too corrupted or too ambiguous to be labeled by humans. Hence, one should not consider these tasks for label aggregation or learning since they might reduce generalization power.

In this work, we combine task difficulty scores with worker abilities scores, but we measure the task difficulty by incorporating feature information. We thus introduce the Weighted Area Under the Margin (WAUM), a generalization to the crowdsourcing setting of the Area Under the Margin (AUM) by Pleiss et al. (2020). The AUM is a confidence indicator in an assigned

[1]In this work we use the term task and feature interchangeably.



(a) Label `airplane` is easy to identify (unanimity among workers).

(b) Label `deer` is meaningless here, and workers are confused with all other labels.

(c) Label `cat` often confused with horns of a wild `deer`

Figure 2: Three images from `CIFAR-10H` dataset (Peterson et al., 2019): the `airplane` image (a) is easy, while the landscape (b) is ambiguous due to the image's poor quality. The last image (c) is a black cat face often perceived as the horns of a wild `deer`.

label defined for each training task. It is computed as an average of margins over scores obtained along the learning steps. The AUM reflects how a learning procedure struggles to classify a task to an assigned label. The AUM is well suited when training a neural network (where the steps are training epochs) or other iterative methods. For instance, it has led to better network calibration (Park and Caragea, 2022) using MixUp strategy (Zhang et al., 2018), *i.e.,* mixing tasks identified as simple and difficult by the AUM. Our extension of the AUM, the WAUM identifies harmful data points in crowdsourced datasets, so one can prune ambiguous tasks that degrade the generalization. It is a weighted average of workers AUM, where the weights reflect trust scores based on task difficulty and workers' ability.

This work is a condensed version of Lefort et al. (2022). The full article provides more results on simulations and real datasets. We also show to impact of the pruning hyperparameter introduced by the WAUM and consider cases where pruning might be harmful. Finally, we introduce the `peerannot` library available at `https://github.com/peerannot/peerannot` that was used to generate the results.

2 Weighted Area Under the Margin

2.1 Definitions, notation, and construction

We consider classical multi-class learning notation, with input in $\mathcal{X}$ and labels in $[K] := \{1, \dots, K\}$. The set of tasks is written as $\mathcal{X}_{\texttt{train}} = \{x_1, \dots, x_{n_{\texttt{task}}}\}$, and we assume there are $n_{\texttt{task}}$ *i.i.d* tasks and labels $\{(x_1, y_1^\star), \dots, (x_{n_{\texttt{task}}}, y_{n_{\texttt{task}}}^\star)\}$ with underlying distribution denoted by $\mathbb{P}$. The true labels $(y_i^\star)_{i\in[n_{\texttt{task}}]}$ are unobserved but crowdsourced labels are provided by $n_{\texttt{worker}}$ workers $(w_j)_{j\in[n_{\texttt{worker}}]}$. We write $\mathcal{A}(x_i) = \{j \in [n_{\texttt{worker}}] : \text{worker } w_j \text{ labeled task } x_i\}$ the **annotators set**[2] of a task x_i and $\mathcal{T}(w_j) = \{i \in [n_{\texttt{task}}] : \text{worker } w_j \text{ answered task } x_i\}$ the **tasks set** for a worker w_j. For a task x_i and each $j \in \mathcal{A}(x_i)$, we denote $y_i^{(j)} \in [K]$ the

[2]As illustrated in Fig. 1, the size of the annotators and tasks sets might not be fixed, and the standard supervised setting is recovered when $|\mathcal{A}(x_i)| = 1$ for all $i \in [n_{\texttt{task}}]$.

label answered by worker w_j and we call soft label any vector $\hat{y}_i$ in the standard simplex $\Delta_{K-1} = \{p \in \mathbb{R}^K, \sum_{k=1}^K p_k = 1, p_k \geq 0\}$. For any set $\mathcal{S}$, we write $|\mathcal{S}|$ for its cardinality. The training set has task-wise and worker-wise formulations:

$$\mathcal{D}_{\texttt{train}} = \bigcup_{i=1}^{n_{\texttt{task}}} \left\{ \left(x_i, \left(y_i^{(j)}\right)\right) \text{ for } j \in \mathcal{A}(x_i) \right\} = \bigcup_{j=1}^{n_{\texttt{worker}}} \underbrace{\left\{ \left(x_i, \left(y_i^{(j)}\right)\right) \text{ for } i \in \mathcal{T}(w_j) \right\}}_{\mathcal{D}_{\texttt{train}}^{(j)}} \quad . \tag{1}$$

DS model. The Dawid and Skene (DS) model (Dawid and Skene, 1979) aggregates answers and evaluates the workers' confusion matrix to observe where their expertise lies exactly. The confusion matrix of worker w_j is denoted by $\pi^{(j)} \in \mathbb{R}^{K \times K}$ and reflects individual error-rates between pairs of labels: $\pi_{\ell,k}^{(j)} = \mathbb{P}(y_i^{(j)} = k | y_i^\star = \ell)$ represents the probability that worker w_j gives label k to a task whose true label is ℓ. The model assumes that the probability for a task x_i to have true label $y_i^\star = \ell$ follows a multinomial distribution with probabilities $\pi_{\ell,\bullet}^{(j)}$ for each worker, independently of $\mathcal{X}_{\texttt{train}}$ (feature-blind). In practice, DS estimates are obtained thanks to the EM algorithm to output estimated confusion matrices $(\pi^{(j)})_{j \in [n_{worker}]}$.

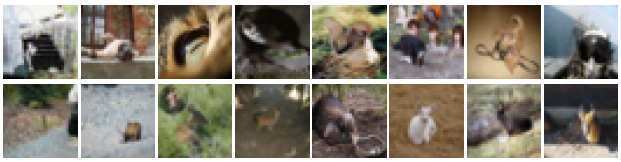
Ambiguous tasks identification with the AUM. Pleiss et al. (2020) have introduced the AUM in the standard learning setting (*i.e.,* $|\mathcal{A}(x_i)| = 1$ for all $i \in [n_{\texttt{task}}]$). Given a training task and a label $(x, y) \in \mathcal{D}_{\texttt{train}}$, let $z^{(t)}(x) \in \mathbb{R}^K$ be the logit score vector at epoch $t \leq T$ when learning a neural network on $\mathcal{D}_{\texttt{train}}$ (where T is the number of training epochs). We use the notation $z_{[1]}^{(t)}(x) \geq \cdots \geq z_{[K]}^{(t)}(x)$ for sorting $(z_1^{(t)}(x), \ldots, z_K^{(t)}(x))$ in non-increasing order. Let us denote $\sigma^{(t)}(x) := \sigma(z^{(t)}(x))$ the softmax output of the scores at epoch t. Sorting the probabilities in decreasing order such that $\sigma_{[1]}^{(t)}(x) \geq \cdots \geq \sigma_{[K]}^{(t)}(x)$, the AUM reads:

$$\text{AUM}\left(x, y; \mathcal{D}_{\texttt{train}}\right) = \frac{1}{T} \sum_{t=1}^{T} \left[\sigma_y^{(t)}(x) - \sigma_{[2]}^{(t)}(x)\right] \quad . \tag{2}$$

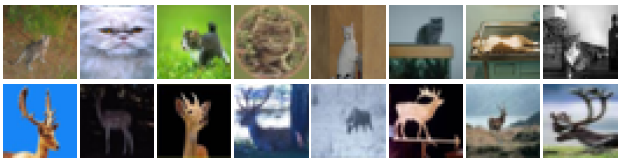
We write $\text{AUM}\left(x, y\right)$ instead of $\text{AUM}\left(x, y; \mathcal{D}_{\texttt{train}}\right)$ when the training set is clear from the context. Pleiss et al. (2020) use an average of margins over logit scores, whereas we instead consider the average of margin after a softmax step in Eq. (2). We have adapted the original AUM relying on logit scores by applying a softmax step. This tempers scaling issues as advocated by Ju et al. (2018) in ensemble learning. Moreover, we consider the margin introduced by Yang and Koyejo (2020) instead. Indeed, the corresponding hinge loss has better theoretical properties than the one used in the original AUM, especially in top-k settings[3] (Lapin et al., 2016; Yang and Koyejo, 2020; Garcin et al., 2022).

During the training phase, the AUM keeps track of the difference between the score assigned to the proposed label and the score assigned to the second-largest one. It has been introduced to detect mislabeled observations in a dataset: the higher the AUM, the more confident the prediction is in the assigned label. Hence, the lower the AUM, the more likely the label is wrong. Finally, note that the AUM computation depends on the chosen neural

[3]For top-k, consider $\sigma_{[k+1]}^{(t)}(x)$ instead of $\sigma_{[2]}^{(t)}(x)$ in Eq. (2).



(a) 8 worst images with our proposed WAUM



(b) 8 worst images with original AUM as in Pleiss et al. (2020)

Figure 3: `CIFAR-10H`: 8 worst images detected for the `cat` (first row) and `deer` (second row) labels in `CIFAR-10`. (a) the worst AUMs for the original method by Pleiss et al. (2020), training on the test set of `CIFAR-10`; (b) the worst WAUMs with our proposed method training on `CIFAR-10H`. Both are computed using a Resnet-18.

network and on its initialization: pre-trained architectures could be used, yet any present bias would transfer to the AUM computation.

WAUM. The AUM is defined in a standard supervised setting with (hard) labels: we now adapt it to crowdsourced frameworks to improve the identification of hard tasks. Let $s^{(j)}(x_i) \in [0,1]$ be a trust factor in the answer of worker w_j for task x_i. The WAUM is then defined as:

$$\mathrm{WAUM}(x_i) = \frac{1}{\sum_{j' \in \mathcal{A}(x_i)} s^{(j')}(x_i)} \sum_{j \in \mathcal{A}(x_i)} s^{(j)}(x_i) \mathrm{AUM}\big(x_i, y_i^{(j)} s\big) \ . \tag{3}$$

It is a weighted average of AUMs over each worker's answer with a per task weighting score $s^{(j)}(x_i)$ based on workers' abilities. This score considers the impact of the AUM for each answer since it is more informative if the AUM indicates uncertainty for an expert than for a non-expert.

The scores $s^{(j)}$ are obtained *à la* Servajean et al. (2017): each worker has an estimated confusion matrix $\hat{\pi}^{(j)} \in \mathbb{R}^{K \times K}$. Note that the vector $\mathrm{diag}(\hat{\pi}^{(j)}) \in \mathbb{R}^K$ represents the probability for worker w_j to answer correctly to each task. With a neural network classifier, we estimate the probability for the input $x_i \in \mathcal{X}_{\texttt{train}}$ to belong in each category by $\sigma^{(T)}(x_i)$, *i.e.,* the probability estimate at the last epoch. As a trust factor, we propose the inner product between the diagonal of the confusion matrix and the softmax vector:

$$s^{(j)}(x_i) = \left\langle \mathrm{diag}(\hat{\pi}^{(j)}), \sigma^{(T)}(x_i) \right\rangle \in [0,1] \ . \tag{4}$$

The scores control the weight of each worker in Eq. (3). This choice of weight is inspired by the bilinear scoring system of GLAD (Whitehill et al., 2009), as detailed hereafter. The closer to one, the more we trust the worker for the given task. In GLAD, the trust score is modeled as the product $\alpha_j \beta_i$, with $\alpha_j \in \mathbb{R}$ (resp. $\beta_i \in (0, +\infty)$) representing worker ability (resp. task difficulty). In Eq. (4), the diagonal of the confusion matrix $\hat{\pi}^{(j)}$ represents the worker's ability and the softmax the task difficulty. Hence, the score $s^{(j)}(x_i)$ can be seen as a multidimensional version of GLAD's trust score.

Dataset pruning. Our procedure (Algorithm 1) proceeds as follows. We initialize our method by estimating the confusion matrices for all workers. For each worker w_j, the AUM is computed for its labeled tasks, and so is its worker-dependent trust scores $s^{(j)}(x_i)$ with Eq. (4). The WAUM in Eq. (3) is then computed for each task. The most ambiguous tasks, the ones whose WAUM are below a threshold, are then discarded, and the associated pruned dataset $\mathcal{D}_{\text{pruned}}$ is output.

We consider for the threshold a quantile of order $\alpha \in [0, 1]$ of the WAUM scores. The hyperparameter α (proportion of training data points pruned) can be chosen on a validation set, yet choosing $\alpha \in \{0.1, 0.05, 0.01\}$ has led to satisfactory results in all our experiments.

Algorithm 1 WAUM (Weighted Area Under the Margin).

Input: $\mathcal{D}_{\texttt{train}}$: tasks and crowdsourced labels,
$\alpha \in [0, 1]$: proportion of training points pruned
$T \in \mathbb{N}$: number of epochs
`Est`: Estimation procedure for the confusion matrices
Initialization: Get confusion matrix $\{\hat{\pi}^{(j)}\}_{j\in[n_{\texttt{worker}}]}$ from `Est`
Train a neural network for T epochs on $\mathcal{D}_{\texttt{train}}$
for $j \in [n_{worker}]$ **do**
  Get $\text{AUM}(x_i, y_i^{(j)}; \mathcal{D}_{\texttt{train}})$ using Eq. (2) for $i \in \mathcal{T}(w_j)$
  Get **trust scores** $s^{(j)}(x_i)$ using Eq. (4) for $i \in \mathcal{T}(w_j)$
for *each task* $x \in \mathcal{X}_{train}$ **do**
  Compute $\text{WAUM}(x)$ using Eq. (3)
Get q_α $(\text{WAUM}(x_i))_{i\in[n_{\texttt{task}}]}$, α-**quantile threshold**
$\mathcal{D}_{\text{pruned}} = \left\{ \left(x_i, \left(y_i^{(j)}\right)_{j\in\mathcal{A}(x_i)}\right) : \text{WAUM}(x_i) \geq q_\alpha, x_i \in \mathcal{X}_{\texttt{train}} \right\}$
Result: $\mathcal{D}_{\text{pruned}}$

2.2 Label aggregation and classifier training.

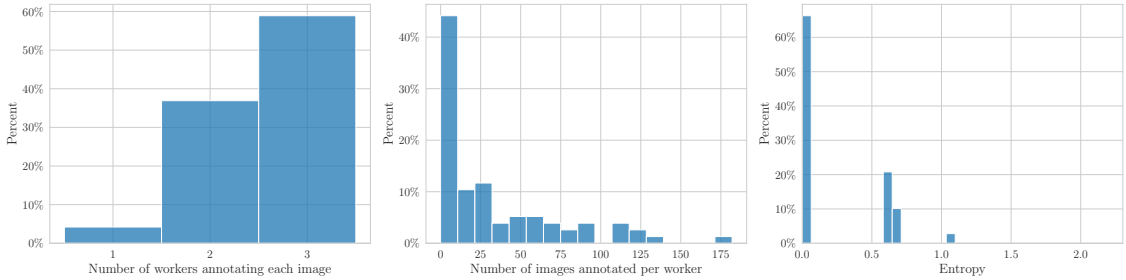
Once a pruned dataset $\mathcal{D}_{\text{pruned}}$ has been obtained thanks to the WAUM, one can create soft labels through an aggregation step, and use them to train another classifier. Aggregated soft labels contain information regarding human uncertainty, and could often be less noisy than NS labels. They can help improve model calibration (Wen et al., 2021; Zhong et al., 2021), a property useful for interpretation (Jiang et al., 2012; Kumar et al., 2019). Concerning the classifier training, note that it can differ from the one used to compute the WAUM. We train a neural network whose architecture is adapted dataset per dataset and that can differ from the one used in Algorithm 1 (it is the case for instance for the `LabelMe` dataset).

For an aggregation technique agg, we write the full training method WAUM + agg and instantiate several choices below. By default, our aggregation strategy is a weighted version of DS, coined WAUM + WDS. It weights votes according to each worker's confidence as follows. First, it estimates confusion matrices $\{\hat{\pi}^{(j)}\}_{j\in[n_{\texttt{worker}}]}$ with DS applied to $\mathcal{D}_{\text{pruned}}$. Then, it computes soft labels $(\hat{y}_i^{\text{WDS}})_{k\in[K]}$ for all tasks $x_i \in \mathcal{X}_{\text{pruned}}$ by weighting labels with workers' confidence: $\hat{y}_i^{\text{WDS}} = \frac{\tilde{y}_i}{\sum_{k\in[K]}(\tilde{y}_i)_k}$ with $\tilde{y}_i = \Big(\sum_{j\in\mathcal{A}(x_i)} \hat{\pi}^{(j)}_{k,k} \mathbb{1}_{\{y_i^{(j)}=k\}}\Big)_{k\in[K]}$ for all $x_i \in \mathcal{X}_{\text{pruned}}$.

3 Experiments

Metrics investigated After training with aggregated labels, we report two performance metrics on a test set $\mathcal{D}_{\texttt{test}}$: top-1 accuracy and expected calibration error (ECE) (with $M = 15$ bins as in Guo et al. (2017)). We also report the training accuracy $\text{Acc}_{\texttt{train}}(y^\star, \hat{y}) = \frac{1}{|\mathcal{D}_{\texttt{train}}|} \sum_{i=1}^{|\mathcal{D}_{\texttt{train}}|} \mathbb{1}_{\{\text{argmax}\, \hat{y}_i = y_i^\star\}}$, the accuracy of the aggregation method on the training set's true labels. The training accuracy is computed on $\mathcal{D}_{\text{pruned}}$ for the WAUM since tasks detected ambiguous are not labeled.

LabelMe dataset This dataset consists in classifying 1 000 images in $K = 8$ categories. In total 77 workers are reported in the dataset (though only 59 of them answered any task at all!). Each task has between 1 and 3 labels. A validation set of 500 images and a test set of 1 188 images are available. The architecture used for training is a VGG-16 combined with two dense layers as described in Rodrigues and Pereira (2018). The VGG-16 backbone classifier is pre-trained on `Imagenet` with data augmentation using random flipping, shearing and dropout. Adam optimizer with a learning rate set to 0.005 is used during the 1 000 training epochs. For the WAUM computation, 500 epochs are used with a pre-trained Resnet-50 (it differs from the modified VGG used later for training) and the same optimization settings. Contrary to the modified VGG-16, the Resnet-50 could be fully pre-trained. The general stability of pre-trained Resnets, thanks to the residuals connections, allows us to compute the WAUM with way fewer epochs (each being also with a lower computational cost) compared to VGGs (He et al., 2016). The hyperparameter α is set to 0.01. Experiments were executed with Nvidia RTX 2080 and Quadro T2000 GPUs. Additional coding details at available at `https://github.com/peerannot/peerannot/`. Two other real datasets, `CIFAR-10H` and `Music`, are available in Lefort et al. (2022).



(a) Feedback effort per task (b) Load per worker distribution (c) Naive soft labels, entropy distribution

Figure 4: `LabelMe`: dataset visualization

We observe in Tab. 1 that the WAUM improves the final test accuracy when combined with the CoNAL network. CoNAL was specifically tailored for `LabelMe`. Hence, by modeling a common confusion between classes, pruning most ambiguous tasks with the WAUM, CoNAL improves the classifier generalization performance and calibration in comparison to simple strategies. Combined with our WAUM, additional gains are obtained on both metrics.

Table 1: `LabelMe`: generalization performance by crowdsourcing strategy (here $\alpha = 0.01$)

Aggregation method	$\text{Acc}_{\texttt{test}}$	ECE	$\text{Acc}_{\texttt{train}}$
MV	85.4 ± 1.0	0.136 ± 0.01	76.1
NS	86.1 ± 1.0	0.138 ± 0.01	76.9
DS	86.8 ± 0.5	0.123 ± 0.01	79.7
GLAD	87.1 ± 0.9	0.119 ± 0.01	77.6
CrowdLayer	85.4 ± 4.2	0.142 ± 0.04	–
CoNAL($\lambda = 0$)	88.1 ± 1.0	0.119 ± 0.01	–
CoNAL($\lambda = 10^{-4}$)	86.2 ± 6.4	0.135 ± 0.06	–
WAUM + WDS	87.1 ± 0.8	0.129 ± 0.01	74.4
WAUM+CoNAL($\lambda = 0$)	89.2 ± 1.0	0.108 ± 0.01	–
WAUM+CoNAL($\lambda = 10^{-4}$)	$\mathbf{90.0} \pm 0.8$	$\mathbf{0.099} \pm 0.01$	–

4 Conclusion and future work

In this paper, we investigate crowdsourcing aggregation models and how judging systems may impact generalization performance. Most models consider the ambiguity from the workers' perspective (very few consider the difficulty of the task itself) and evaluate workers on hard tasks that might be too ambiguous to be relevant, leading to a performance drop. Using a popular model (DS), we develop the WAUM, a flexible feature-aware metric that can identify hard tasks and improves generalization performance. It also yields a fairer evaluation of workers' abilities and supports recent research on data pruning in supervised datasets. Independently of pruning, the WAUM allows identifying early the images that need extra labeling efforts, or that cannot be correctly labeled at all.

Extension of the WAUM to more general learning tasks (*e.g.,* top-k classification) would be natural, including labeling tasks sequentially. Indeed, the WAUM could help to identify tasks requiring additional expertise and guide how to allocate more experts/workers for such identified tasks. Future works could adapt the WAUM to imbalanced crowdsourced datasets to identify potentially too ambiguous images that naturally occur in open platforms like Pl@ntNet[4].

Last but not least, on the dataset side, we believe that the community would benefit from releasing a challenging dataset (such as the one by Garcin et al. (2021) for instance) tailored to learn in crowdsourcing settings. Indeed, a dataset with the following properties could greatly foster future research in the field: a varying number of labels per worker, a high number of classes, and a subset with ground truth labels to test generalization performance.
Acknowledgment: Work supported by the Chaire CaMeLOt ANR-20-CHIA-0001-01.

References

Aitchison, L. (2021). A statistical theory of cold posteriors in deep neural networks. In *ICLR*.

Birnbaum, A. L. (1968). Some latent trait models and their use in inferring an examinee's ability. *Statistical theories of mental test scores*.

[4] `https://plantnet.org/en/`

Camilleri, M. P. and Williams, C. K. (2019). The extended Dawid-Skene model. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 121–136. Springer.

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.

Garcin, C., Joly, A., Bonnet, P., Affouard, A., Lombardo, J.-C., Chouet, M., Servajean, M., Lorieul, T., and Salmon, J. (2021). Pl@ntnet-300k: a plant image dataset with high label ambiguity and a long-tailed distribution. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Garcin, C., Servajean, M., Joly, A., and Salmon, J. (2022). Stochastic smoothing of the top-k calibrated hinge loss for deep imbalanced classification. In *ICML*.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. (2017). On calibration of modern neural networks. In *ICML*, page 1321.

Han, J., Luo, P., and Wang, X. (2019). Deep self-learning from noisy labels. In *ICCV*, pages 5138–5147.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, pages 770–778.

Jiang, X., Osl, M., Kim, J., and Ohno-Machado, L. (2012). Calibrating predictive model estimates to support personalized medicine. *J. Am. Med. Inform. Assoc.*, 19(2):263–274.

Ju, C., Bibaut, A., and van der Laan, M. (2018). The relative performance of ensemble methods with deep convolutional neural networks for image classification. *J. Appl. Stat.*, 45(15):2800–2818.

Kim, H.-C. and Ghahramani, Z. (2012). Bayesian classifier combination. In *AISTATS*, volume 22, pages 619–627.

Kumar, A., Liang, P. S., and Ma, T. (2019). Verified uncertainty calibration. In *NeurIPS*, volume 32.

Lapin, M., Hein, M., and Schiele, B. (2016). Loss functions for top-k error: Analysis and insights. In *CVPR*, pages 1468–1477.

Lefort, T., Charlier, B., Joly, A., and Salmon, J. (2022). Identify ambiguous tasks combining crowdsourced labels by weighting areas under the margin. *arXiv preprint arXiv:2209.15380*.

Müller, R., Kornblith, S., and Hinton, G. E. (2019). When does label smoothing help? *NeurIPS*, 32.

Müller, N. M. and Markert, K. (2019). Identifying mislabeled instances in classification datasets. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Northcutt, C., Jiang, L., and Chuang, I. (2021a). Confident learning: Estimating uncertainty in dataset labels. *J. Artif. Intell. Res.*, 70:1373–1411.

Northcutt, C. G., Athalye, A., and Mueller, J. (2021b). Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Park, S. Y. and Caragea, C. (2022). On the calibration of pre-trained language models using mixup guided by area under the margin and saliency. In *ACML*, pages 5364–5374.

Peterson, J. C., Battleday, R. M., Griffiths, T. L., and Russakovsky, O. (2019). Human uncertainty makes classification more robust. In *ICCV*, pages 9617–9626.

Pleiss, G., Zhang, T., Elenberg, E. R., and Weinberger, K. Q. (2020). Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*.

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

Servajean, M., Joly, A., Shasha, D., Champ, J., and Pacitti, E. (2017). Crowdsourcing thousands of specialized labels: A Bayesian active training approach. *IEEE Transactions on Multimedia*, 19(6):1376–1391.

Sinha, V. B., Rao, S., and Balasubramanian, V. N. (2018). Fast Dawid-Skene: A fast vote aggregation scheme for sentiment classification. *arXiv preprint arXiv:1803.02781*.

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.

Wen, Y., Jerfel, G., Muller, R., W. Dusenberry, M., Snoek, J., Lakshminarayanan, B., and Tran, D. (2021). Combining ensembles and data augmentation can harm your calibration. In *ICLR*.

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.

Yang, F. and Koyejo, S. (2020). On the consistency of top-k surrogate losses. In *ICML*, pages 10727–10735.

Zhang, H., Cissé, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *ICLR*.

Zhong, Z., Cui, J., Liu, S., and Jia, J. (2021). Improving calibration for long-tailed recognition. In *CVPR*, pages 16489–16498.