Crowdsourcing label noise simulation on image classification tasks

Tanguy Lefort [1,a] & Benjamin Charlier[1,b] & Joseph Salmon[1,2,c] & Alexis Joly[3,d]

[1] *IMAG, Univ. Montpellier, CNRS, Montpellier, France*
[2] *Institut Universitaire de France (IUF)* [3] *Inria, LIRMM, CNRS Montpellier*
[a] *tanguy.lefort@umontpellier.fr* [b] *benjamin.charlier@umontpellier.fr*
[c] *joseph.salmon@umontpellier.fr* [d] *alexis.joly@inria.fr*

Résumé. Il est courant de collecter des données étiquetées en ayant recourt à la production participative *crowdsourcing*. Cependant, la qualité des étiquettes dépend fortement de la difficulté des tâches et des capacités des contributeurs. Avec ces données, le manque de vérité terrain rend difficile l'évaluation de la qualité des annotations. Il existe peu de données de ce type disponibles publiquement, et encore moins sur des tâches de difficulté hétérogène avec les réponses de tous les contributeurs avant l'étape d'agrégation. Nous proposons ici un nouveau cadre de simulation de *crowdsourcing* où la qualité peut être contrôlée. Cela permet d'évaluer l'apprentissage de différentes stratégies de manière empirique à partir des étiquettes collectées. Notre objectif est de séparer les différentes sources de bruit dans l'agrégation des étiquettes: les contributeurs ne fournissant aucune information sur la vraie étiquette et ceux peu performants, utiles pour des tâches faciles.

Mots-clés. Apprentissage participatif, simulation de foule, détection de spammeur

Abstract. It is common to collect labelled datasets using crowdsourcing. Yet, labels quality depends deeply on the task difficulty and on the workers abilities. With such datasets, the lack of ground truth makes it hard to assess the quality of annotations. There are few open-access crowdsourced datasets, and even fewer that provide both heterogeneous tasks in difficulty and all workers answers before the aggregation. We propose a new crowdsourcing simulation framework with quality control. This allows us to evaluate different empirical learning strategies empirically from the obtained labels. Our goal is to separate different sources of noise: workers that do not provide any information on the true label against poorly performing workers, useful on easy tasks.

Keywords. Crowdsourcing, crowd simulation, spam detection

1 Introduction

Let us consider a supervised learning setting, with a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n_t}$ containing n_t tasks $x_i \in \mathcal{X}$ (vectorized images) with their corresponding label $y_i \in \mathcal{Y}$ (with K $\text{card}(\mathcal{Y})$ classes), commonly annotated by a crowd. With a crowd of n_w workers, it yields multiple labels $\{y_i^{(j)}\}_{j=1}^{n_w}$ for each image x_i. Learning from multiple labels is challenging (Rodrigues

et al., 2013; Khetan et al., 2018), since the ground truth is unknown and trust needs to be evaluated.

Platforms like ThePlantGame[1] expect users to label (plant) images hard to identify, possibly inducing weak estimators and very imbalanced datasets (in the number of instances per class but also in the number of proposed labels per tasks). We first need to consider more controlled crowdsourced experiments, like CIFAR-10H (Peterson et al., 2019) (relying on Amazon's Mechanical Turk[1]): the most complete open-access dataset with a control over the number of tasks annotated by each worker. Note that the images were very curated (Aitchison, 2020), thus most reaching consensus among the workers. Yet, obtaining such data is long, raises ethical or possibly legal issues, and a relevant toy simulation framework is missing.

We introduce a toy-dataset simulator that provides visually corrupted tasks and confusion-aware workers with control over the confusion using the task difficulty. Most simulations use the class-conditional assumption (Patrini et al., 2017), *i.e.,* one only observes the confused version y_{conf} of y where $\mathbb{P}_{y_{conf}|x} = \pi^\top \mathbb{P}_{y|x}$ for $\pi \in \mathbb{R}^{K\times K}$ a confusion matrix with $\pi_{k,k'} = \mathbb{P}(y_{conf} = k'|y = k)$ the probability that the true label k is confused with k'. This assumption overlooks the essential factor that is the instance-dependence, as labelling a clear image is easier than a blurry one. Whitehill et al. (2009) created difficulty-aware labelling simulations, without generating visualizable tasks, with either hard or easy instances and good or bad workers for a prescribed confusion matrix.

In our framework, confusion is reflected through the colors' variability in the output images: pure colors are easy to label while blended ones are more difficult, see Figure 2. Each worker provides an answer depending on their abilities to distinguish colors. The framework allows testing different aggregations and workers evaluations strategies while knowing the ground truth. The architecture also authorizes poor labelling such as noisy workers (answering tasks at random) that we identify before label aggregation.

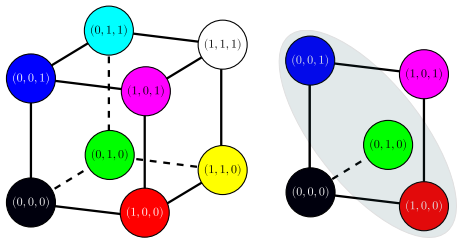


Figure 1: a) RGB graph; b) neighborhood example for $y = (0, 0, 0)$: $\mathcal{N}_y = \{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}$.

2 Crowdsourced dataset

2.1 Simulated crowdsourced data

To provide visual intuition for our simulated crowdsourced dataset, we rely on the RGB cube (Qin et al., 2019). We denote by G the unit cube graph with vertices $\mathcal{Y} = \{0, 1\}^3$, with each vertex represent an RGB color as in Figure 1. The neighborhood of $y \in \mathcal{Y}$ is defined as $\mathcal{N}_y := \{y' \in \mathcal{Y},\ \sum_{i=1}^{3} \mathbb{1}_{\{y_i \neq y'_i\}} = 1\}$ and the edges are $\{(y, y') \in \mathcal{Y} \times \mathcal{N}_y\}$. Let

[1] http://theplantgame.com/, https://www.mturk.com/

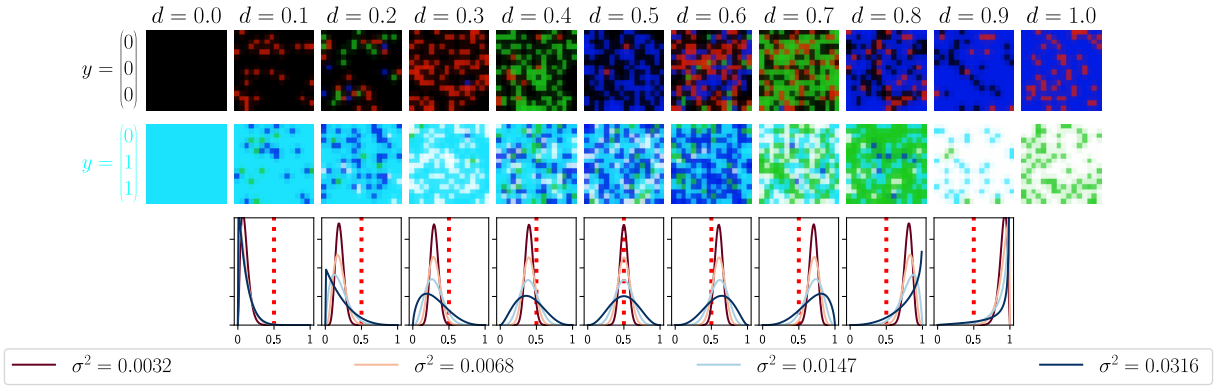


Figure 2: Outputs from Algorithm 1 with gradually increasing levels of difficulty d for true labels $(0, 0, 0)$ and $(0, 1, 1)$. The closer d is to 0, the purer the image and the simpler the labelling. The third row represents Beta confusion distributions (defined in Equation (1)) and used in Algorithm 2 for four levels of workers' abilities σ^2. The decision limit for workers between choosing y or a label in $\mathcal{N}_y$ is at 0.5.

$\Delta^n \subset \mathbb{R}^{n+1}$ be the simplex of dimension n, $\mathbf{1}_n \in \mathbb{R}^n$ the vector of ones and $A \otimes B$ the Kronecker product between two matrices A and B.

We generate n_t tasks following Algorithm 1, that provides an image x, a ground truth label y and a difficulty level d as follows. Each task/image has pixels whose color is a combination of the true color y and alternative ones from $\mathcal{N}_y$. For a difficulty level $d \in [0, 1]$, if a sample drawn from a Bernoulli distribution $\mathcal{B}er(d)$ is 0, then the pixel value remains y. Otherwise, a Dirichlet distribution with parameter $(1/3, 1/3, 1/3)$ provides a sampled distribution over the neighborhood $\mathcal{N}_y$.

Algorithm 1: Generate one RGB image with visual label corruption.

Data: $d \in [0, 1]$: task difficulty level; y: task label; $W \in \mathbb{N}^*$: width of image.
Result: x: image; $\mathcal{N}_y$: neighborhood; $\nu_{\cdot|y}$: confusion distribution.

$x \leftarrow \mathbb{1}_{W^2} \otimes y \in [0, 1]^{W^2 \times 3}$ `// Vectorized image of constant` y
$\nu_{\cdot|y} \in \Delta^2 \sim \mathcal{D}ir\,(1/3, 1/3, 1/3)$
for $k = 1, \ldots, W^2$ **do** `// For each pixel`
  $switch \sim \mathcal{B}er(d)$
  if $switch = 1$ **then**
    $(q_{y'})_{y' \in \mathcal{N}_y} \sim \mathcal{D}ir\left(\nu_{\cdot|y}\right)$ `// Corruption distribution`
    $x[k] \leftarrow \arg\max_{y' \in \mathcal{N}_y}(q_{y'})$

The pixel color is then chosen as the most likely one in $\mathcal{N}_y$ with respect to the former distribution. To sample balanced difficulties and control the maximum level, a candidate difficulty distribution is $\mathcal{U}(0, d_{\max})$.

The workers' answers are implemented using Algorithm 2. For each image x_i, $i \in [n_t]$ to label, the worker must decide whether the true label y_i is the dominant color in x_i. To model this decision, a random variable with support $[0, 1]$ (the confusion-difficulty

Algorithm 2: Simulate one worker answer for a fixed task

Data: y: label; d: difficulty; $\nu_{\cdot|y}$: confusion distribution; $\{(\sigma^{(j)}_{y\leftrightarrow y'})^2\}_{y,y'}$ worker abilities to distinguish y from $y' \in \mathcal{N}_y$.

Result: Worker answer $y^{(j)}$

$confu^{(j)} \sim \mathcal{B}eta\left(\mu = d, \sigma^2 = \max_{y\in\mathcal{N}_{y'}} \left(\sigma^{(j)}_{y\leftrightarrow y'}\right)^2\right)$ // Equation (1)

if $conf u^{(j)} < \frac{1}{2}$ **then** // Worker answers true color

  $y^{(j)} = y$

else // The worker is wrong

  $(alt_{y'})_{y'\in\mathcal{N}_y} \sim \left(\mathcal{B}eta\left(\mu = (1-d)\nu_{y'|y}, \sigma^2 = \max_{\ell\in\mathcal{N}_y\setminus\{y'\}} \left(\sigma^{(j)}_{y'\leftrightarrow\ell}\right)^2\right)\right)_{y'\in\mathcal{N}_y}$

  $y^{(j)} = \arg\max_{y'\in\mathcal{N}_y} alt_{y'}$ // Choose worker's most present color in $\mathcal{N}_y$

range) reflects the prior expectations on the worker, *i.e.,* their abilities to distinguish the different colors (third row of Figure 2). We have considered Beta distributions with mean $\mu = d_i$ and variance σ^2 depending on their abilities. If their confusion (the draw from the Beta distribution) is above 1/2 (Figure 3) then a majority of pixels are considered from $\mathcal{N}_{y_i}$. Then we model the worker's decision among the colors from $\mathcal{N}_{y_i}$ with another Beta distribution, the only changes are the mean, now at $(1-d_i)\nu_{y'|y_i}$ to reflect the proportion of each alternative color, and the variance that no longer takes into account y_i but only its possible confusions.

The parametrization of the Beta distribution is done as follows. Let $\alpha, \beta > 0$, then for $\sigma^2 < \mu(1-\mu)$:

$$X \sim \mathcal{B}eta(\mu, \sigma^2) \quad (1)$$
$$\Leftrightarrow X \sim \mathcal{B}eta\left(\alpha = \mu\left(\tfrac{\mu(1-\mu)}{\sigma^2} - 1\right), \beta = \tfrac{1-\mu}{\mu}\alpha\right).$$

In practice, given parameters (μ, σ^2), we use $\left(\mu, \min(\sigma^2, \sigma^2_{\max} - \epsilon)\right)$ with $\epsilon > 0$ small and $\sigma^2_{\max}$ *s.t.* $\alpha > 1$ to avoid U-shaped or degenerated distributions that would not reflect a confusion behavior.

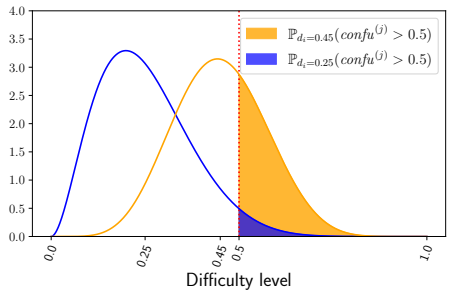
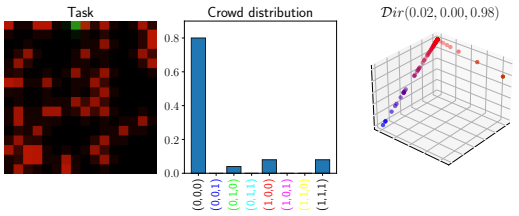


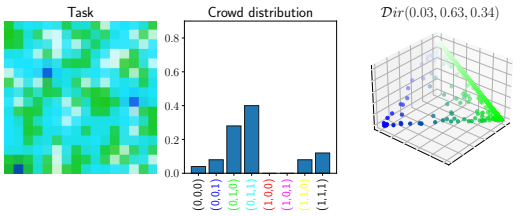
Figure 3: Probability of confusion, *i.e.,* to answer a label in $\mathcal{N}_{y_i}$ depending on the difficulty of the task ($d_i = 0.25$ and 0.45) for a fixed confusion level $\sigma^2 = 0.015$.

2.2 From crowd to learnable labels

To learn human decisions from crowd labels we use soft labels (*i.e.,* the distribution of votes) known to provide better calibrated architectures (Guo et al., 2017). Besides, we have access to the most confusing color: $\arg\max_{y'\in\mathcal{N}_y} \nu_{y'|y}$ that we call the second-best choice. In replicating a human prediction, we thus consider the Second-Best Accuracy (SBA), that is "the second label predicted is the second-best



(a) $y_i = (0, 0, 0)$, $d_i = 0.2$, $\nu_{\cdot|y_i}$ is concentrated near $(1, 0, 0)$, not $(0, 1, 0)$ nor $(0, 0, 1)$. Except spammers, there were no votes in $\mathcal{N}_{y_i}$.



(b) $y_i = (0, 1, 1)$, $d_i = 0.45$, $\nu_{\cdot|y_i}$ is concentrated near $(0, 1, 0)$ mostly. We see the crowd hesitated with the second-best choice $(0, 1, 0)$.

Figure 4: Labelling for $n_w = 25$ workers: 6 spammers and 19 workers from Algorithm 2. We display 1000 draws from $\nu_{\cdot|y_i}$ the confusion distribution.

choice". This evaluates the model capacity to take into account the confusion in its prediction the way humans would when presented a task.

2.3 Identifying spammers in crowd

Here, we call spammer a worker that answers any label randomly, without looking at the task given i.e., $\mathbb{P}(y_i^{(j)} = c | y_i = c') = \mathbb{P}(y_i^{(j)} = c)$ for $c, c' \in [K]$. In our simulations we use $\mathbb{P}(y_i^{(j)} = c) = K^{-1}$. A strategy proposed by Raykar et al. (2012) to detect spammers is to compute a *spam score* $s^{(j)} \in [0, 1]$ from the confusion matrix $\pi^{(j)}$ of each worker. Using $\hat{u}_j = \arg\min_{u_j} \left\| \pi^{(j)} - \mathbf{1}_K u_j^\top \right\|_F^2 \; s.t \; u_j^\top \mathbf{1}_K = 1$:

$$s^{(j)} = \left\| \pi^{(j)} - \mathbf{1}_K \hat{u}_j^\top \right\|_F^2 = \frac{1}{K(K-1)} \sum_{c<c'} \sum_{k \in [K]} \left(\pi_{ck}^{(j)} - \pi_{c'k}^{(j)} \right)^2 \quad . \tag{2}$$

This is simply the sum of the unbiased variance of each column of $\pi^{(j)}$. The closer $s^{(j)}$ is to 0, the more likely worker j is a spammer. However, $\pi^{(j)}$ in Equation (2) can not be computed without knowing y_i. The DS model, proposed by Dawid et al. (1979), estimates the confusion matrices without access to y_i by maximizing the likelihood with unknowns $\pi^{(j)}$'s, indicators and prevalence $\rho \in \Delta^7$: $\prod_{i \in [n_t]} \prod_{k \in [K]} \left\{ \rho_k \prod_{j \in [n_w]} \prod_{\ell \in [K]} (\pi_{k\ell}^{(j)}) \right\}^{\mathbb{1}_{\{y_i = k\}}}$.

3 Experiments

We compare the impact of the workers abilities in the labelling aggregation using two settings with a different number of workers. First, we estimate $\hat{\pi}^{(j)}$ with the DS model. Then, from Equation (2), we get $\hat{s}^{(j)}$ the worker's estimated spam score, and train a k-means ($k = 2$) on the scores to separate the spammers (simulated from Section 2.3) from workers (simulated with Algorithm 2). From Table 1, when only a few workers can be trusted, removing spammers drastically improves the learning procedure.

Table 1: Top-1, Top-2 and SBA accuracies with logistic regression on simulated crowd data taking $n_t = 500$, $W = 64$ and $d_{\max} = 0.6$ using soft labels with and without spam removal. Images are smoothed using a Gaussian filter with standard deviation 0.5. The levels of workers abilities are those from Figure 2. Test size is 30%.

	$n_w = 25$ with 17 spammers			$n_w = 100$ with 88 spammers		
	Top-1	Top-2	SBA	Top-1	Top-2	SBA
soft labels	0.35	0.53	0.19	0.19	0.39	0.17
remove spams + soft labels	**0.71**	**0.88**	**0.38**	**0.80**	**0.92**	**0.42**

Figure 5: CIFAR-10H dataset: example of tasks labelled by workers 1098, 2160 and 2156 identified as spammers. Answered labels do not match the obvious true label.

Running the spammer detection on the CIFAR-10H dataset, we obtain that there are 19 spammers (out of 2571) that could be removed from the crowd (*e.g.,* Figure 5).
Acknowledgment: Work supported by the Chaire CaMeLOt ANR-20-CHIA-0001-01.

References

Dawid, AP and AM Skene (1979). Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *J. R. Stat. Soc. Ser. C. Appl. Stat.* 28.1, pp. 20–28.
Whitehill, J et al. (2009). Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. *NIPS*. Vol. 22. Curran Associates, Inc.
Raykar, VC and S Yu (2012). Eliminating Spammers and Ranking Annotators for Crowdsourced Labeling Tasks. *J. Mach. Learn. Res.* 13, pp. 491–518.
Rodrigues, F, F Pereira, and B Ribeiro (2013). Learning from multiple annotators: distinguishing good from random labelers. *Pattern Recognition Letters* 34.12, pp. 1428–1436.
Guo, C et al. (2017). On calibration of modern neural networks. *ICML*. PMLR, p. 1321.
Patrini, G et al. (2017). Making deep neural networks robust to label noise: A loss correction approach. *CVPR*, pp. 1944–1952.
Khetan, A, ZC Lipton, and A Anandkumar (2018). Learning From Noisy Singly-labeled Data. *ICLR*.
Peterson, JC, RM Battleday, and TL Griffithsand O Russakovsky (2019). Human Uncertainty Makes Classification More Robust. ICCV, pp. 9617–9626.
Qin, Y et al. (2019). A Multi-class Classification Algorithm Based on Hypercube. 2019 IEEE DDCLS, pp. 406–409.
Aitchison, L (2020). A statistical theory of cold posteriors in deep neural networks. *ICLR*.