

LABEL AMBIGUITY IN CROWDSOURCING FOR CLASSIFICATION AND EXPERT FEEDBACK

Tanguy Lefort

IMAG, Univ Montpellier, CNRS

INRIA, LIRMM,

Supervised by

Benjamin Charlier

Alexis Joly

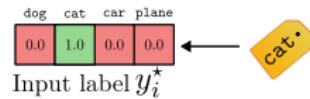
and **Joseph Salmon**



HOW TO TRAIN YOUR CLASSIFIER

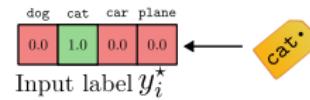
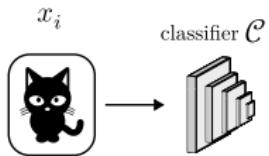
1

x_i



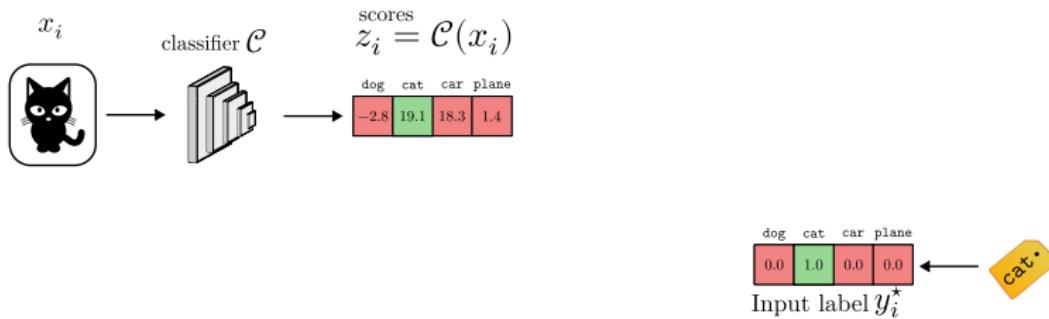
HOW TO TRAIN YOUR CLASSIFIER

1

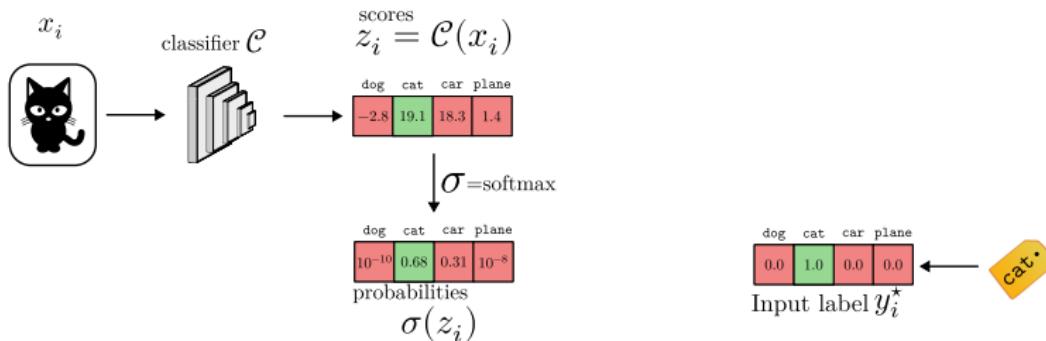


HOW TO TRAIN YOUR CLASSIFIER

1

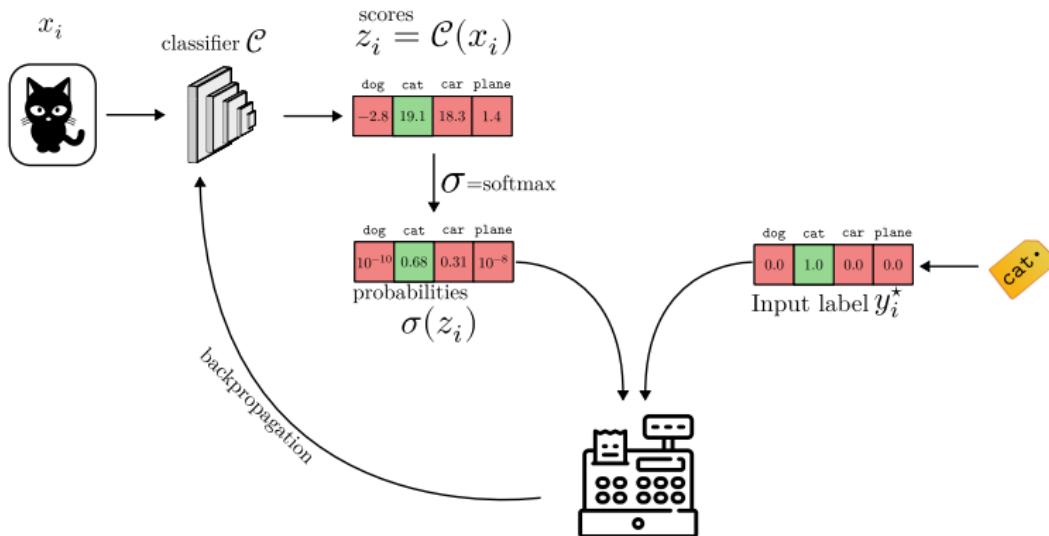


HOW TO TRAIN YOUR CLASSIFIER



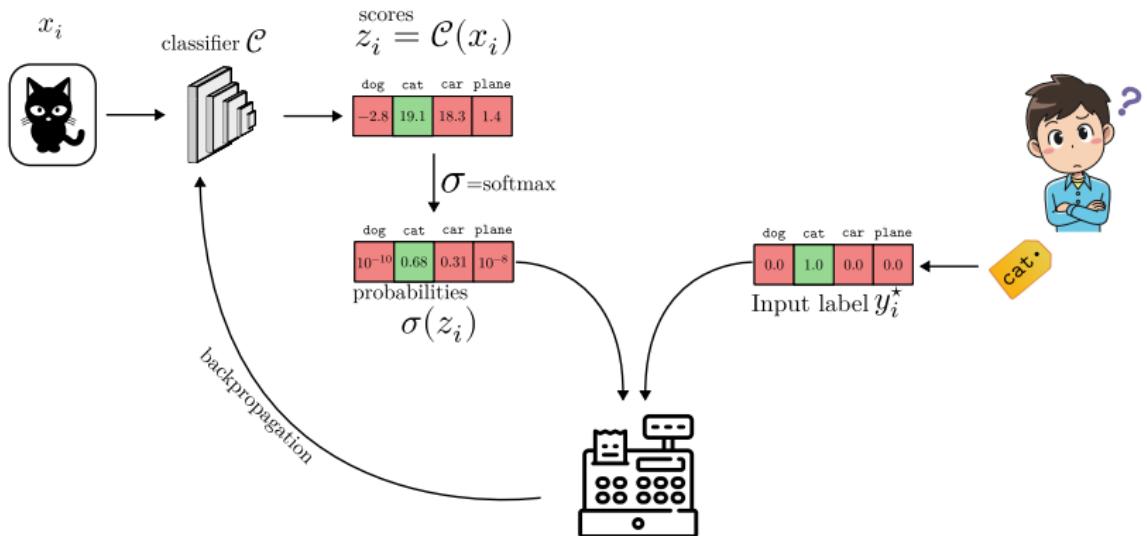
HOW TO TRAIN YOUR CLASSIFIER

1



HOW TO TRAIN YOUR CLASSIFIER

1



ASK CITIZENS TO LABEL OUR DATA

FRAMEWORK AND NOTATIONS

2

- ▶ Workers sort a given task into one of the K classes

$K = 4$

$\mathcal{A}(x_2)$

	w_1	w_2	w_3	w_4	w_5	
• 0:car	• 2:cat					
• 1:plane	• 3:dog					
x_1						
x_2						

$\mathcal{T}(w_3)$

y_i^*

ASK CITIZENS TO LABEL OUR DATA

FRAMEWORK AND NOTATIONS

2

- ▶ Workers sort a given task into one of the K classes

$K = 4$

$\mathcal{A}(x_2)$

	w_1	w_2	w_3	w_4	w_5
• 0:car					
• 1:plane					
• 2:cat	●				
• 3:dog		●			
x_1					
x_2					
$\mathcal{T}(w_3)$					

y_i^*

2

0

- ▶ $y_i^{(j)} \in [K] :=$ answer of worker j to task i
- ▶ n_{worker} workers answer n_{task} tasks

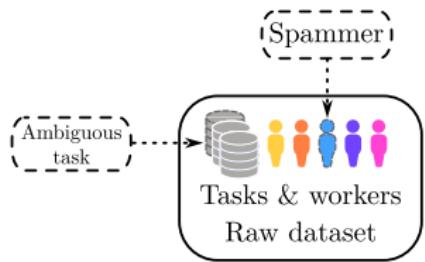
FROM THE DATA TO THE CLASSIFIER

THE PIPELINE



FROM THE DATA TO THE CLASSIFIER

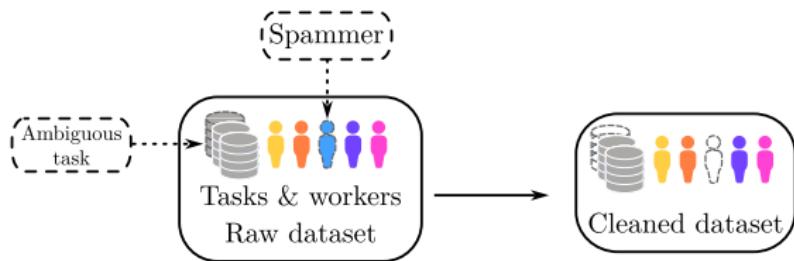
THE PIPELINE



FROM THE DATA TO THE CLASSIFIER

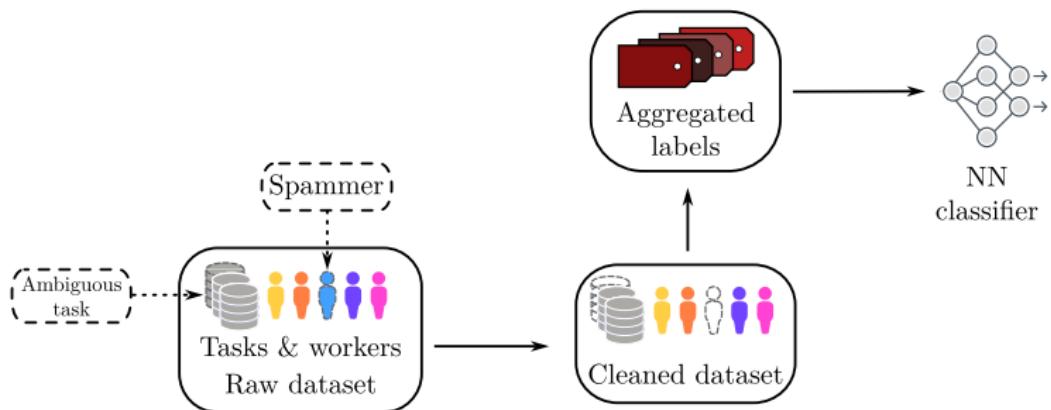
THE PIPELINE

3



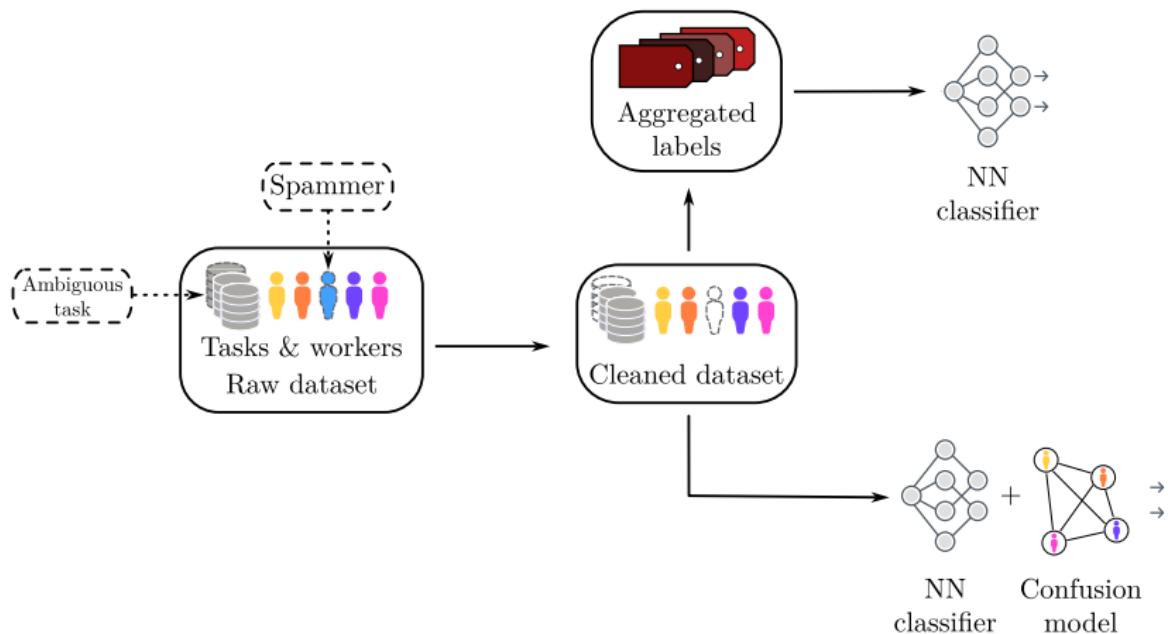
FROM THE DATA TO THE CLASSIFIER

THE PIPELINE



FROM THE DATA TO THE CLASSIFIER

THE PIPELINE



MAIN CONTRIBUTIONS

- ▶ Can we improve performance by leveraging better-quality data?

(1) T. Lefort, B. Charlier, et al. (2024a). "Identify Ambiguous Tasks Combining Crowdsourced Labels by Weighting Areas Under the Margin". In: *Transactions on Machine Learning Research*.

(2) T. Lefort, B. Charlier, et al. (2024b). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

(3) T. Lefort, A. Affouard, et al. (2024). "Cooperative learning of PI@ntNet's Artificial Intelligence algorithm: how does it work and how can we improve it?" In: *arXiv preprint arXiv:2406.03356*.

MAIN CONTRIBUTIONS

- ▶ Can we improve performance by leveraging better-quality data?

- ▶ Can we standardize crowdsourcing dataset's tools in python for reproducibility?

(1) T. Lefort, B. Charlier, et al. (2024a). "Identify Ambiguous Tasks Combining Crowdsourced Labels by Weighting Areas Under the Margin". In: *Transactions on Machine Learning Research*.

(2) T. Lefort, B. Charlier, et al. (2024b). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

(3) T. Lefort, A. Affouard, et al. (2024). "Cooperative learning of PI@ntNet's Artificial Intelligence algorithm: how does it work and how can we improve it?" In: *arXiv preprint arXiv:2406.03356*.

MAIN CONTRIBUTIONS

- ▶ Can we improve performance by leveraging better-quality data?

- ▶ Can we standardize crowdsourcing dataset's tools in python for reproducibility?

- ▶ What can we do in a large-scale setting? Application to Pl@ntNet

(1) T. Lefort, B. Charlier, et al. (2024a). "Identify Ambiguous Tasks Combining Crowdsourced Labels by Weighting Areas Under the Margin". In: *Transactions on Machine Learning Research*.

(2) T. Lefort, B. Charlier, et al. (2024b). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

(3) T. Lefort, A. Affouard, et al. (2024). "Cooperative learning of Pl@ntNet's Artificial Intelligence algorithm: how does it work and how can we improve it?" In: *arXiv preprint arXiv:2406.03356*.

MAIN CONTRIBUTIONS

- ▶ Can we improve performance by leveraging better-quality data?
 - ▶ Creation of the **WAUM**⁽¹⁾: a metric to identify ambiguous images
- ▶ Can we standardize crowdsourcing dataset's tools in python for reproducibility?
- ▶ What can we do in a large-scale setting? Application to Pl@ntNet

⁽¹⁾ T. Lefort, B. Charlier, et al. (2024a). "Identify Ambiguous Tasks Combining Crowdsourced Labels by Weighting Areas Under the Margin". In: *Transactions on Machine Learning Research*.

⁽²⁾ T. Lefort, B. Charlier, et al. (2024b). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

⁽³⁾ T. Lefort, A. Affouard, et al. (2024). "Cooperative learning of Pl@ntNet's Artificial Intelligence algorithm: how does it work and how can we improve it?" In: *arXiv preprint arXiv:2406.03356*.

MAIN CONTRIBUTIONS

- ▶ Can we improve performance by leveraging better-quality data?
 - ▶ Creation of the **WAUM**⁽¹⁾: a metric to identify ambiguous images
- ▶ Can we standardize crowdsourcing dataset's tools in python for reproducibility?
 - ▶ Creation of **peerannot** library⁽²⁾:
<https://peerannot.github.io>
- ▶ What can we do in a large-scale setting? Application to Pl@ntNet

⁽¹⁾ T. Lefort, B. Charlier, et al. (2024a). "Identify Ambiguous Tasks Combining Crowdsourced Labels by Weighting Areas Under the Margin". In: *Transactions on Machine Learning Research*.

⁽²⁾ T. Lefort, B. Charlier, et al. (2024b). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

⁽³⁾ T. Lefort, A. Affouard, et al. (2024). "Cooperative learning of Pl@ntNet's Artificial Intelligence algorithm: how does it work and how can we improve it?" In: *arXiv preprint arXiv:2406.03356*.

MAIN CONTRIBUTIONS

- ▶ Can we improve performance by leveraging better-quality data?
 - ▶ Creation of the **WAUM**⁽¹⁾: a metric to identify ambiguous images
- ▶ Can we standardize crowdsourcing dataset's tools in python for reproducibility?
 - ▶ Creation of **peerannot** library⁽²⁾:
<https://peerannot.github.io>
- ▶ What can we do in a large-scale setting? Application to Pl@ntNet
 - ▶ Creation and evaluation of a **new benchmark dataset**⁽³⁾

⁽¹⁾ T. Lefort, B. Charlier, et al. (2024a). "Identify Ambiguous Tasks Combining Crowdsourced Labels by Weighting Areas Under the Margin". In: *Transactions on Machine Learning Research*.

⁽²⁾ T. Lefort, B. Charlier, et al. (2024b). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

⁽³⁾ T. Lefort, A. Affouard, et al. (2024). "Cooperative learning of Pl@ntNet's Artificial Intelligence algorithm: how does it work and how can we improve it?" In: *arXiv preprint arXiv:2406.03356*.

EXISTING AGGREGATION STRATEGIES

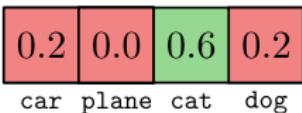
CLASSICAL AGGREGATION STRATEGY

(WEIGHTED) MAJORITY VOTES



$$\hat{y}_i^{\text{WMV}} = \operatorname{argmax}_{k \in [K]} \sum_{j \in \mathcal{A}(x_i)} \text{weight}_j \mathbb{1}(y_i^{(j)} = k)$$

For example with balanced weights:



→ cat

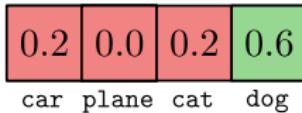
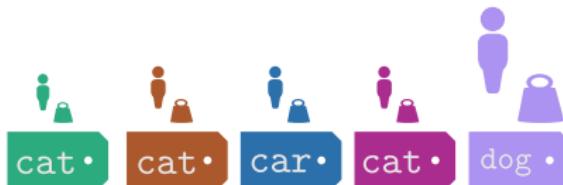
CLASSICAL AGGREGATION STRATEGY

(WEIGHTED) MAJORITY VOTES



$$\hat{y}_i^{\text{WMV}} = \operatorname{argmax}_{k \in [K]} \sum_{j \in \mathcal{A}(x_i)} \text{weight}_j \mathbb{1}(y_i^{(j)} = k)$$

For example with unbalanced weights:



→ dog

CLASSICAL AGGREGATION STRATEGY

(WEIGHTED) MAJORITY VOTES

Many existing weight choices:

- ▶ Inter worker agreement: WAWA⁽⁴⁾:

$$\text{weight}(w_j) = \text{Accuracy}(\{y_i^{(j)}\}_i, \{\hat{y}_i^{\text{MV}}\}_i)$$

- ▶ Feature importance + game theory: Shapley-value weight⁽⁵⁾
- ▶ Matrix completion: MACE⁽⁶⁾ ...

Pros: "simple" weight can scale to large datasets and be easy to interpret
Cons: Can not capture worker skills in detail

(4) <https://success.appen.com/hc/en-us/articles/202703205-Calculating-Worker-Agreement-with-Aggregate-Wawa>

(5) T. Lefort, B. Charlier, et al. (July 2024c). "Weighted majority vote using Shapley values in crowdsourcing". In: CAp 2024 - Conférence sur l'Apprentissage Automatique. Lille, France.

(6) D. Hovy et al. (2013). "Learning whom to trust with MACE". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1120–1130.

CLASSICAL AGGREGATION STRATEGY

DAWID AND SKENE⁽⁷⁾



- ▶ Introduced in a medical context (aggregate multiple diagnosis)
- ▶ Represent worker j from their pairwise confusions matrix $\pi^{(j)} \in \mathbb{R}^{K \times K}$
- ▶ Probabilistic model on their answers:

$$y^{(j)} | y^* \sim \text{Multinomial}(\pi_{y^*, \bullet}^{(j)})$$

with $\pi_{k,\ell}^{(j)} = \mathbb{P}(\text{worker } j \text{ answers } \ell \text{ with unknown truth } k)$

Pros:

- ▶ Finer modelisation
- ▶ Can use adversarial workers

Cons:

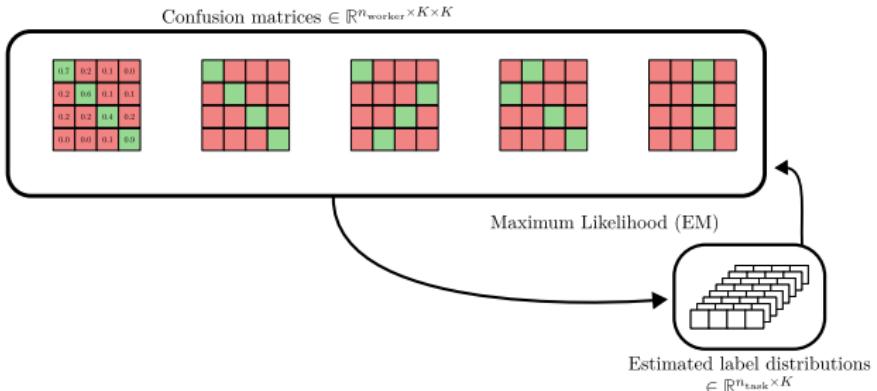
- ▶ Memory issue: $n_{\text{worker}} \times K^2$ parameters to estimate only the confusion matrices

⁽⁷⁾ A. Dawid and A. Skene (1979). "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm". In: *J. R. Stat. Soc. Ser. C. Appl. Stat.* 28.1, pp. 20–28.

CLASSICAL AGGREGATION STRATEGY

DAWID AND SKENE – MODEL

Probabilistic model → Likelihood (to maximize via the Expectation Maximization algorithm)

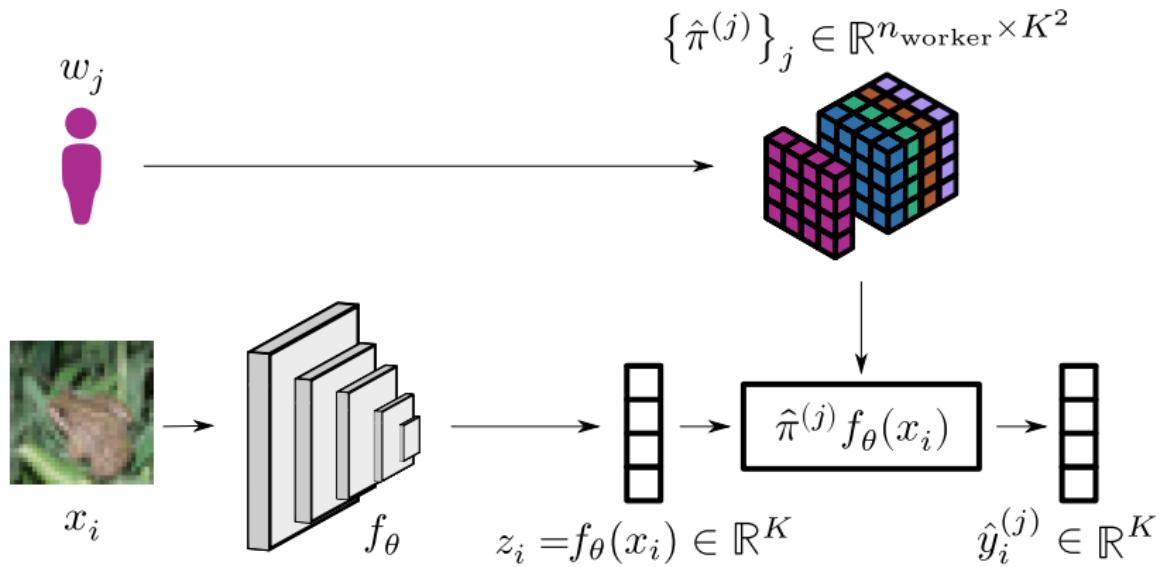


CLASSICAL DEEP-LEARNING STRATEGY

CROWDLAYER⁽⁸⁾

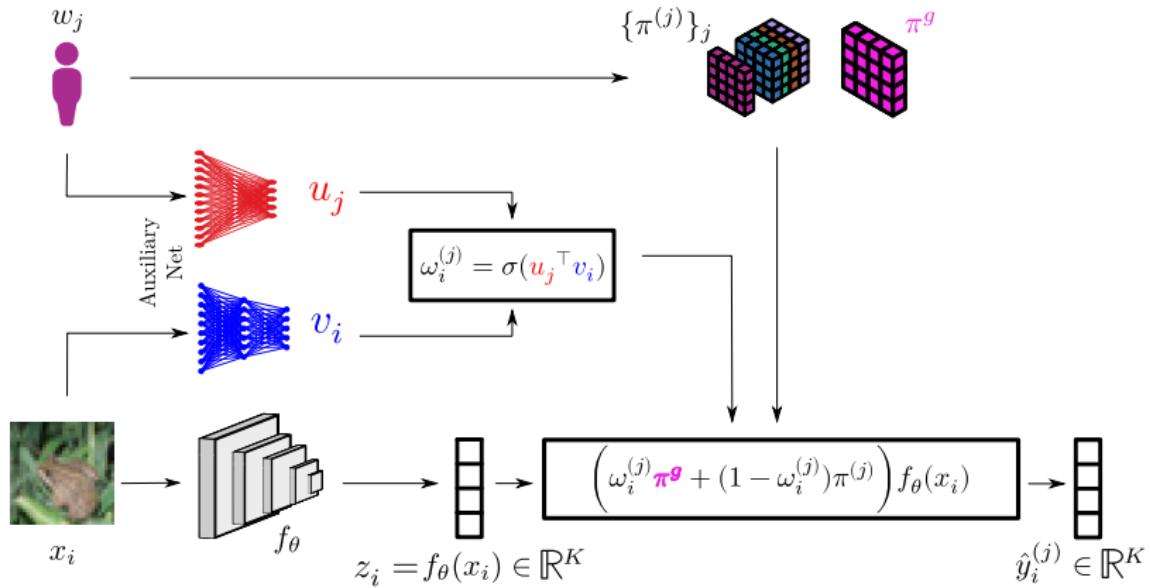


- Idea: put the DS confusion matrix in a neural network as a new layer



⁽⁸⁾ F. Rodrigues and F. Pereira (2018). "Deep learning from crowds". In: AAAI, vol. 32.

- Idea: CrowdLayer + global and local confusions



⁽⁹⁾ Z. Chu, J. Ma, and H. Wang (2021). "Learning from Crowds by Modeling Common Confusions." In: AAAI, pp. 5832–5840.

IDENTIFY AMBIGUOUS TASKS IN CROWDSOURCED DATASETS

WHEN IMAGES HAVE UNDERLYING AMBIGUITY

$K = 4$

$\mathcal{A}(x_2)$

$\mathcal{T}(w_3)$

• 0:car • 2:cat
• 1:plane • 3:dog

	w_1	w_2	w_3	w_4	w_5	
x_1						
x_2						

y_i^*

Diagram illustrating a neural network architecture for image classification. The input layer consists of two images, x_1 (a black cat) and x_2 (a car). These are processed by two weight matrices, w_1 and w_2 , which produce feature maps of human silhouettes. These features are then processed by weight matrices w_3 through w_5 , which produce final outputs. The output for x_2 is ambiguous, represented by a row of five gray tags, each containing a different value (2, 2, 0, 2, 3). A dashed vertical line separates the input and hidden layers from the output layer, which contains a single row of three gray tags, each containing a different value (2, 0, 0).

WHEN IMAGES HAVE UNDERLYING AMBIGUITY

14

$K = 4$

$\mathcal{A}(x_2)$

	w_1	w_2	w_3	w_4	w_5
• 0:car	• 2:cat				
• 1:plane	• 3:dog				
x_1					
x_2					
x_3					

$\mathcal{T}(w_3)$

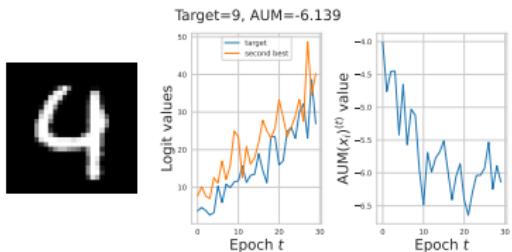
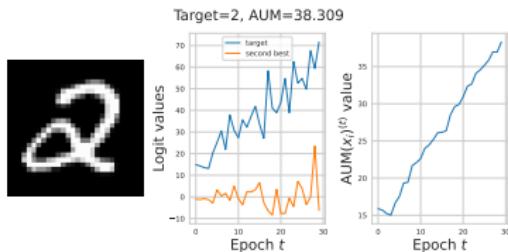
y_i^*

AMBIGUITY IN CLASSICAL SUPERVISED SETTING

AREA UNDER THE MARGIN (AUM)

Goal: identify issues in classical datasets $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times [K]$

- AUM⁽¹⁰⁾: monitor margin during training



⁽¹⁰⁾ G. Pleiss et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: *NeurIPS*.

AMBIGUITY IN CLASSICAL SUPERVISED SETTING

AREA UNDER THE MARGIN (AUM)

Goal: identify issues in classical datasets $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times [K]$

- ▶ AUM⁽¹¹⁾: monitor margin during training
- ▶ Classifier: at training epoch $t \in [T]$, $\mathcal{C}^{(t)}(x_i) \in \mathbb{R}^K$ a vector of **scores** (logits)

$$\text{AUM}(x_i, y_i) = \frac{1}{T} \sum_{t=1}^T \left[\mathcal{C}^{(t)}(x_i)_{y_i} - \max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_{\ell} \right]$$

Diagram annotations:

- Average = Stability (red bracket above the summation term)
- Margin between scores: content of Hinge loss (red bracket above the difference term)
- Score of assigned label (blue bracket under $\mathcal{C}^{(t)}(x_i)_{y_i}$)
- Other maximum score (white bracket under $\max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_{\ell}$)

⁽¹¹⁾ G. Pleiss et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: NeurIPS.

AMBIGUITY IN CLASSICAL SUPERVISED SETTING

AREA UNDER THE MARGIN (AUM)

Goal: identify issues in classical datasets $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times [K]$

- ▶ AUM⁽¹¹⁾: monitor margin during training
- ▶ Classifier: at training epoch $t \in [T]$, $\mathcal{C}^{(t)}(x_i) \in \mathbb{R}^K$ a vector of **scores** (logits)

$$\text{AUM}(x_i, y_i) = \frac{1}{T} \sum_{t=1}^T \left[\mathcal{C}^{(t)}(x_i)_{y_i} - \max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_{\ell} \right]$$

Average = Stability

Margin between scores:
content of Hinge loss

Score of assigned label

Other maximum score

The diagram illustrates the formula for AUM. It shows a large bracket under the summation sign representing the average. Inside the bracket, there are two main terms: the score of the assigned label ($\mathcal{C}^{(t)}(x_i)_{y_i}$) and the maximum score of other labels ($\max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_{\ell}$). Red annotations explain these terms: 'Average = Stability' points to the summation, 'Margin between scores: content of Hinge loss' points to the difference between the two terms, 'Score of assigned label' points to the first term, and 'Other maximum score' points to the second term.

Challenging for crowdsourcing:

- y_i unknown

⁽¹¹⁾ G. Pleiss et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: NeurIPS.

AMBIGUITY IN CLASSICAL SUPERVISED SETTING

AREA UNDER THE MARGIN (AUM)

Goal: identify issues in classical datasets $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times [K]$

- ▶ AUM⁽¹¹⁾: monitor margin during training
- ▶ Classifier: at training epoch $t \in [T]$, $\mathcal{C}^{(t)}(x_i) \in \mathbb{R}^K$ a vector of **scores** (logits)

$$\text{AUM}(x_i, y_i) = \frac{1}{T} \sum_{t=1}^T \left[\mathcal{C}^{(t)}(x_i)_{y_i} - \max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_{\ell} \right]$$

Average = Stability

Margin between scores:
content of Hinge loss

Score of assigned label

Other maximum score

The diagram illustrates the formula for AUM. It shows the overall expression $\frac{1}{T} \sum_{t=1}^T$ in a pink box. Above it, a red bracket labeled "Average = Stability" points to the summation part. Below the main expression, a blue bracket labeled "Score of assigned label" points to the term $\mathcal{C}^{(t)}(x_i)_{y_i}$. To the right, a grey bracket labeled "Other maximum score" points to the term $\max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_{\ell}$. Above the main expression, a red bracket labeled "Margin between scores: content of Hinge loss" spans the difference between the two terms.

Challenging for crowdsourcing:

- y_i unknown
 - ▶ ...so $\mathcal{C}^{(t)}(x_i)_{y_i}$ does not exist

⁽¹¹⁾ G. Pleiss et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: NeurIPS.

AMBIGUITY IN CLASSICAL SUPERVISED SETTING

AREA UNDER THE MARGIN (AUM)



Naive Extension: identify issues in concatenated datasets $\{(x_i, y_i^{(j)})\}_{i,j}$

- Plugin estimate of y_i using \hat{y}_i^{MV}

$$\widetilde{\text{AUM}}(x_i, \hat{y}_i^{\text{MV}}) = \frac{1}{T} \sum_{t=1}^T \left[\mathcal{C}^{(t)}(x_i)_{\hat{y}_i^{\text{MV}}} - \max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_\ell \right]$$

Average = Stability

Score of MV label

Margin between scores:
content of Hinge loss

Other maximum score

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

AMBIGUITY IN CLASSICAL SUPERVISED SETTING

AREA UNDER THE MARGIN (AUM)



Naive Extension: identify issues in concatenated datasets $\{(x_i, y_i^{(j)})\}_{i,j}$

- Plugin estimate of y_i using \hat{y}_i^{MV}

$$\widetilde{\text{AUM}}(x_i, \hat{y}_i^{\text{MV}}) = \frac{1}{T} \sum_{t=1}^T \left[\mathcal{C}^{(t)}(x_i)_{\hat{y}_i^{\text{MV}}} - \max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_{\ell} \right]$$

Average = Stability

Score of MV label

Margin between scores:
content of Hinge loss

Other maximum score

Which margin should be used:

- use previous work of margins' properties⁽¹²⁾

⁽¹²⁾ M. Lapin, M. Hein, and B. Schiele (2016). "Loss functions for top-k error: Analysis and insights". In: CVPR, pp. 1468–1477.

GOING TO THE CROWDSOURCING SETTING

AUMC

Naive Extension: identify issues in concatenated datasets $\{(x_i, y_i^{(j)})\}_{i,j}$

- ▶ Plugin estimate of y_i using \hat{y}_i^{MV}
- ▶ Scores ordered: $\mathcal{C}(x_i)_{[1]} \geq \dots \geq \mathcal{C}(x_i)_{[K]}$

$$\text{AUMC}(x_i, \hat{y}_i^{\text{MV}}) = \frac{1}{T} \sum_{t=1}^T \left[\mathcal{C}^{(t)}(x_i)_{\hat{y}_i^{\text{MV}}} - \mathcal{C}^{(t)}(x_i)_{[2]} \right]$$

Diagram annotations:

- Average = Stability: Points to the term $\frac{1}{T} \sum_{t=1}^T$.
- Margin between scores: margin for top-1 classification: Points to the difference $\mathcal{C}^{(t)}(x_i)_{\hat{y}_i^{\text{MV}}} - \mathcal{C}^{(t)}(x_i)_{[2]}$.
- Score of MV label: Points to the term $\mathcal{C}^{(t)}(x_i)_{\hat{y}_i^{\text{MV}}}$.
- Other maximum score: Points to the term $\mathcal{C}^{(t)}(x_i)_{[2]}$.

Issue:

- Lose all worker-related information
- Sensitive to poorly performing workers

Weighted Areas Under the Margins: identify issues in concatenated datasets $\{(x_i, y_i^{(j)})\}_{i,j}$

- ▶ Scale effects in the scores discarded, need normalization⁽¹³⁾

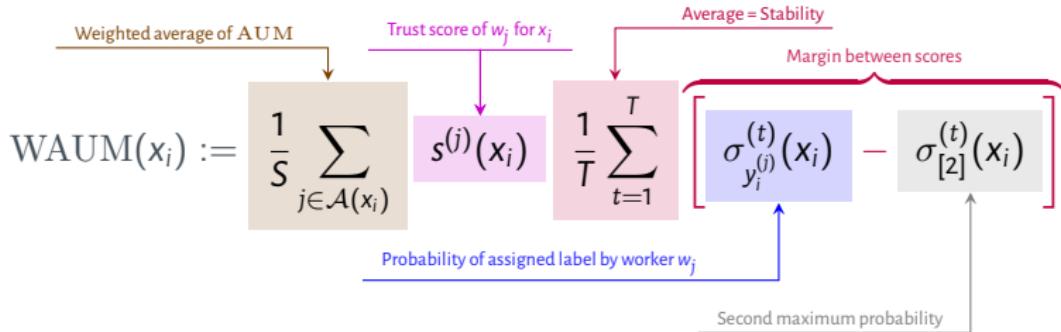
⁽¹³⁾ C. Ju, A. Bibaut, and M. van der Laan (2018). "The relative performance of ensemble methods with deep convolutional neural networks for image classification". In: *J. Appl. Stat.* 45.15, pp. 2800–2818.

Weighted Areas Under the Margins: identify issues in concatenated datasets $\{(x_i, y_i^{(j)})\}_{i,j}$

- Scale effects in the scores discarded, need normalization⁽¹³⁾

With:

- $\sigma(x_i) = \sigma(\mathcal{C}(x_i)) \in \Delta_{K-1}$ (simplex of dim $K - 1$)



⁽¹³⁾ C. Ju, A. Bibaut, and M. van der Laan (2018). "The relative performance of ensemble methods with deep convolutional neural networks for image classification". In: *J. Appl. Stat.* 45.15, pp. 2800–2818.

WEIGHTS IN THE WAUM

LEVERAGE BOTH TASKS AND LABELS



Our chosen worker/task score:

- Consider a score of the form⁽¹⁴⁾: worker skill \times task difficulty⁽¹⁵⁾

⁽¹⁴⁾J. Whitehill et al. (2009). "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise". In: *NeurIPS*. vol. 22.

⁽¹⁵⁾M. Servajean et al. (2017). "Crowdsourcing thousands of specialized labels: A Bayesian active training approach". In: *IEEE Transactions on Multimedia* 19.6, pp. 1376–1391.

WEIGHTS IN THE WAUM

LEVERAGE BOTH TASKS AND LABELS



Our chosen worker/task score:

- Consider a score of the form⁽¹⁴⁾: worker skill \times task difficulty⁽¹⁵⁾

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

Worker j overall ability Difficulty of task i

⁽¹⁴⁾ J. Whitehill et al. (2009). "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise". In: *NeurIPS*. vol. 22.

⁽¹⁵⁾ M. Servajean et al. (2017). "Crowdsourcing thousands of specialized labels: A Bayesian active training approach". In: *IEEE Transactions on Multimedia* 19.6, pp. 1376–1391.

COMPUTING THE WAUM

THE PIPELINE SUMMARIZED

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\text{worker}}]$

COMPUTING THE WAUM

THE PIPELINE SUMMARIZED

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\text{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$

COMPUTING THE WAUM

THE PIPELINE SUMMARIZED

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\text{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute all WAUM(x_i) during training

COMPUTING THE WAUM

THE PIPELINE SUMMARIZED

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\text{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute all WAUM(x_i) during training

COMPUTING THE WAUM

THE PIPELINE SUMMARIZED

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\text{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute all $\text{WAUM}(x_i)$ during training

Usage (for learning):

- **Prune** x_i 's with $\text{WAUM}(x_i)$ below quantile q_α (say $\alpha = 0.01$)
- **Estimate confusion matrices** $\hat{\pi}^{(j)}$ on pruned training dataset

COMPUTING THE WAUM

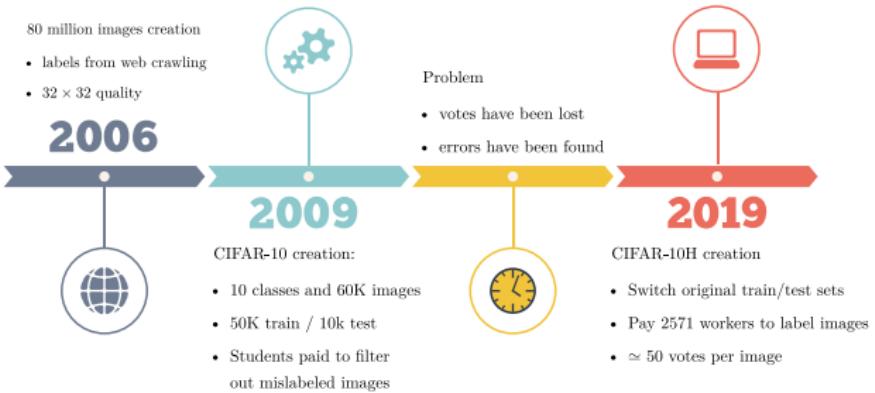
THE PIPELINE SUMMARIZED

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\text{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute all $\text{WAUM}(x_i)$ during training

Usage (for learning):

- **Prune** x_i 's with $\text{WAUM}(x_i)$ below quantile q_α (say $\alpha = 0.01$)
- **Estimate confusion matrices** $\hat{\pi}^{(j)}$ on pruned training dataset
- **Aggregate labels** and **train** a classifier on the newly pruned dataset

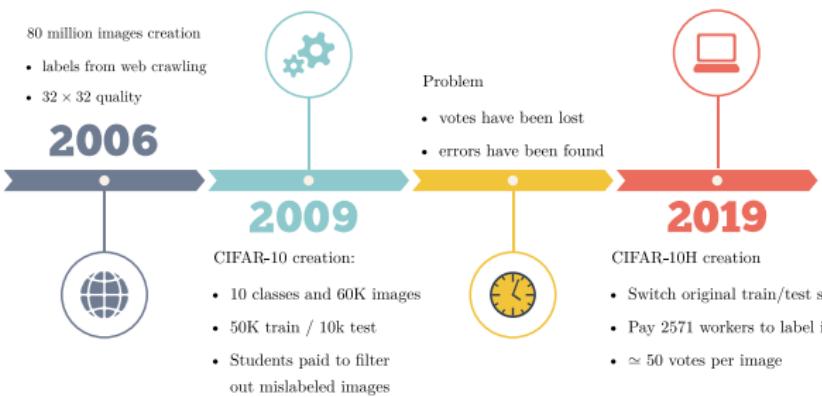
PRESENTING CIFAR-10H⁽¹⁶⁾ DATASET



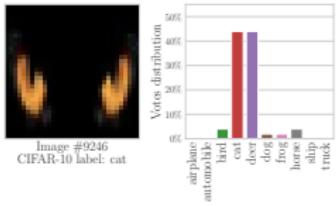
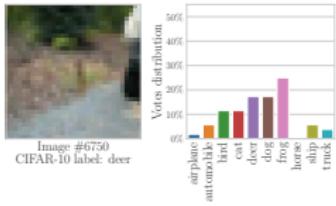
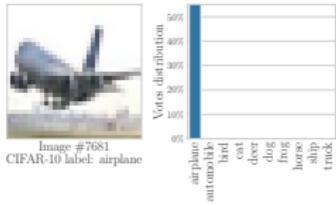
Labels: cat, dog, car, plane, bird, horse, frog, deer, ship, truck

⁽¹⁶⁾J. C. Peterson et al. (2019). "Human Uncertainty Makes Classification More Robust". In: ICCV, pp. 9617–9626.

PRESENTING CIFAR-10H⁽¹⁶⁾ DATASET



Labels: cat, dog, car, plane, bird, horse, frog, deer, ship, truck



⁽¹⁶⁾J. C. Peterson et al. (2019). "Human Uncertainty Makes Classification More Robust". In: ICCV, pp. 9617–9626.

PRESENTING LABELME DATASET⁽¹⁷⁾



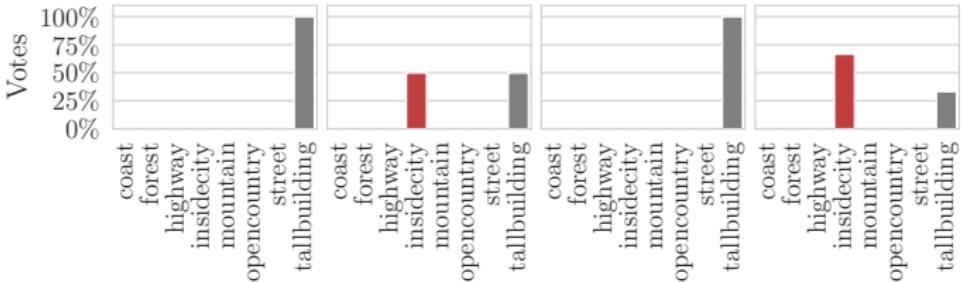
- ▶ 1000 training / 500 validation / 1188 test images
- ▶ 59 workers: each task has up to 3 votes
- ▶ 8 classes: highway, insidecity, tallbuilding, street, forest, coast, mountain, opencountry

⁽¹⁷⁾ F. Rodrigues, F. Pereira, and B. Ribeiro (2014). "Gaussian process classification and active learning with multiple annotators". In: ICML. PMLR, pp. 433–441.

PRESENTING LABELME DATASET⁽¹⁷⁾



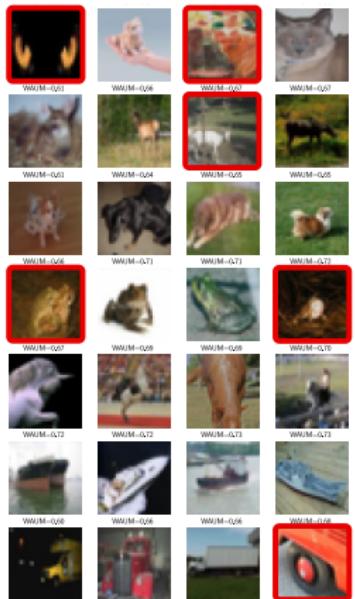
- ▶ 1000 training / 500 validation / 1188 test images
- ▶ 59 workers: each task has up to 3 votes
- ▶ 8 classes: highway, insidecity, tallbuilding, street, forest, coast, mountain, opencountry



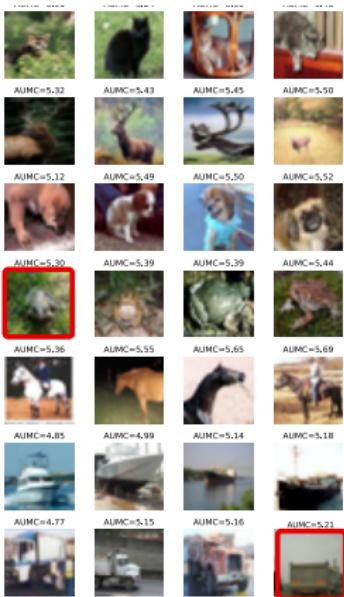
⁽¹⁷⁾ F. Rodrigues, F. Pereira, and B. Ribeiro (2014). "Gaussian process classification and active learning with multiple annotators". In: ICML. PMLR, pp. 433–441.

QUALITATIVE RESULTS

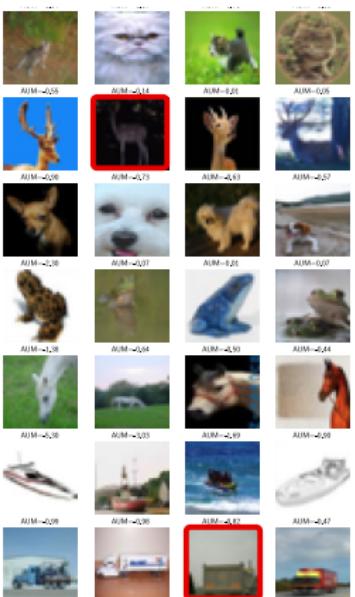
WAUM
(crowdsourcing)



AUMC
(crowdsourcing)



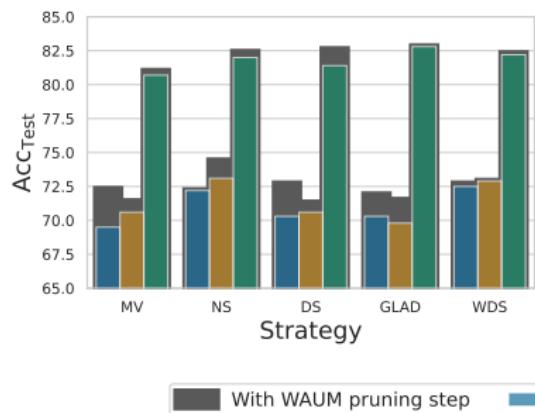
AUM
(no crowdsourcing)



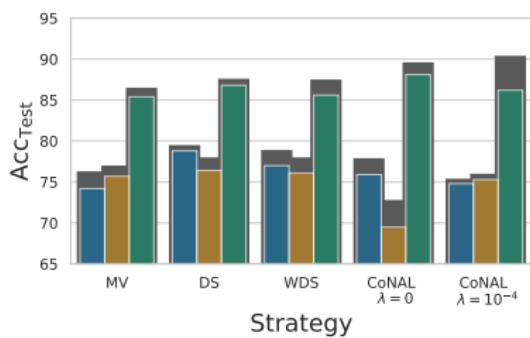
ABLATION STUDY



CIFAR-10H



LabelMe





In short

- ▶ Introduced the WAUM to find ambiguous images
- ▶ Better quality data can improve performance



In short

- ▶ Introduced the WAUM to find ambiguous images
- ▶ Better quality data can improve performance

Towards large-scale problems

- ▶ DS model and confusion matrices do not scale
- ▶ What is currently done in large-scale settings?
- ▶ Can we evaluate their performance?



In short

- ▶ Introduced the WAUM to find ambiguous images
- ▶ Better quality data can improve performance

Towards large-scale problems

- ▶ DS model and confusion matrices do not scale
- ▶ What is currently done in large-scale settings?
- ▶ Can we evaluate their performance?
 - ▶ **To evaluate we need data and code that scale!**

THE PEERANNOT LIBRARY

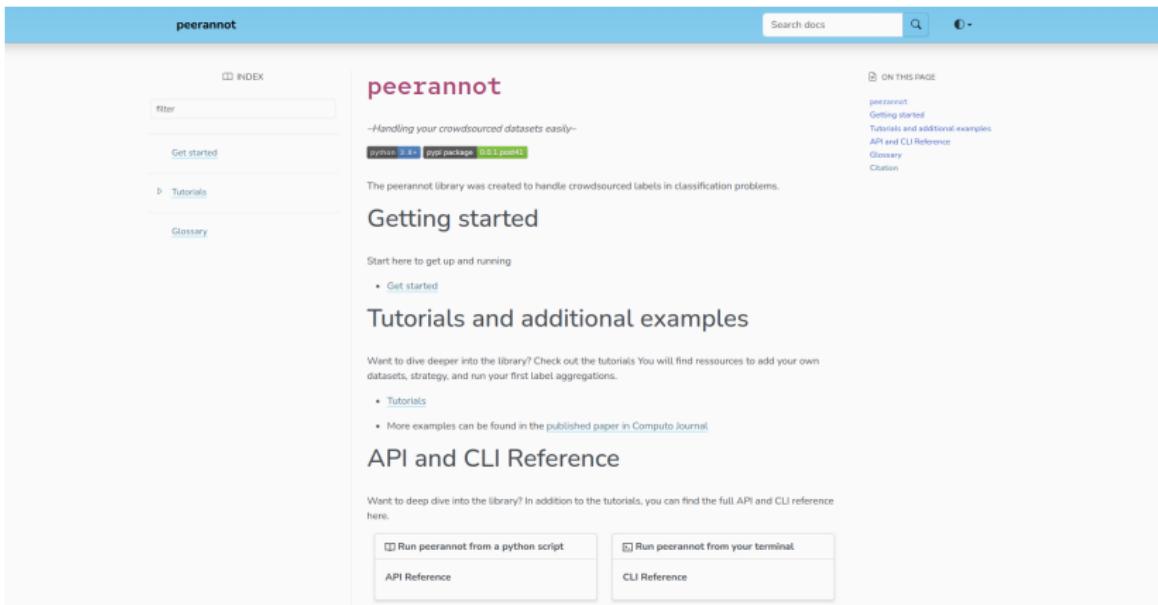
PEERANNOT LIBRARY

HANDLE CROWDSOURCED DATA IN CLASSIFICATION

- Python library for small and large crowdsourced datasets

```
pip install peerannot
```

- Documentation available at: <https://peerannot.github.io>



The screenshot shows the official documentation for the peerannot library. The top navigation bar includes a search bar and links for "Search docs", "ON THIS PAGE", and "peerannot". The main content area features a sidebar with links for "INDEX", "Get started", "Tutorials", and "Glossary". The main content area has several sections: "peerannot" (with a brief description), "Getting started" (with a link to "Get started"), "Tutorials and additional examples" (with a link to "Tutorials and additional examples"), and "API and CLI Reference" (with links to "Run peerannot from a python script", "Run peerannot from your terminal", "API Reference", and "CLI Reference").

peerannot

Search docs

ON THIS PAGE

peerannot

Getting started

Tutorials and additional examples

API and CLI Reference

Glossary

Citation

INDEX

Get started

Tutorials

Glossary

peerannot

Handling your crowdsourced datasets easily

python 0.8.1 | pip package 0.8.1.post45

The peerannot library was created to handle crowdsourced labels in classification problems.

Getting started

Start here to get up and running

- Get started

Tutorials and additional examples

Want to dive deeper into the library? Check out the tutorials. You will find resources to add your own datasets, strategy, and run your first label aggregations.

- Tutorials
- More examples can be found in the published paper in [Computo Journal](#).

API and CLI Reference

Want to deep dive into the library? In addition to the tutorials, you can find the full API and CLI reference here.

Run peerannot from a python script

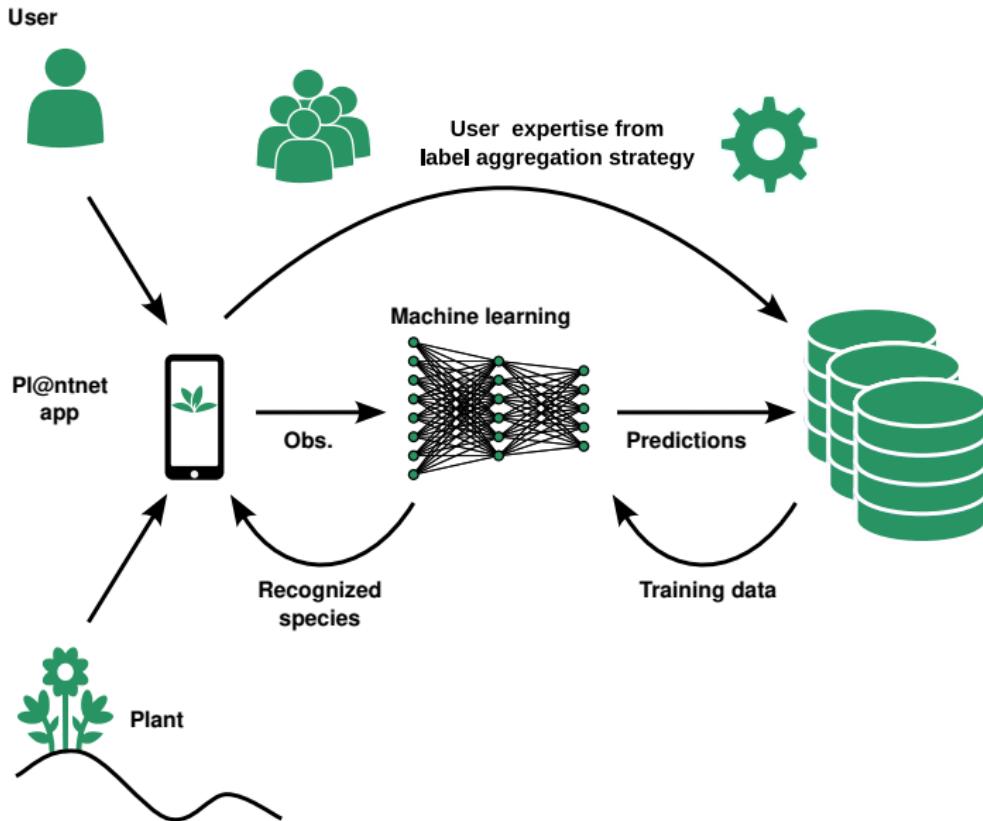
Run peerannot from your terminal

API Reference

CLI Reference

CROWDSOURCING IN LARGE SCALE: THE CASE OF PL@NTNET

PRESENTING PL@NTNET PIPELINE



REALEASING A NEW DATASET



- ▶ South Western European flora obs since 2017
- ▶ $n_{\text{worker}} \simeq 823\,000$ users answered more than $K \simeq 11000$ species
- ▶ $n_{\text{task}} \simeq 6\,700\,000$ observations
- ▶ 9 000 000 votes casted
- ▶ **Imbalance:** 80% of observations are represented by 10% of total votes

REALEASING A NEW DATASET

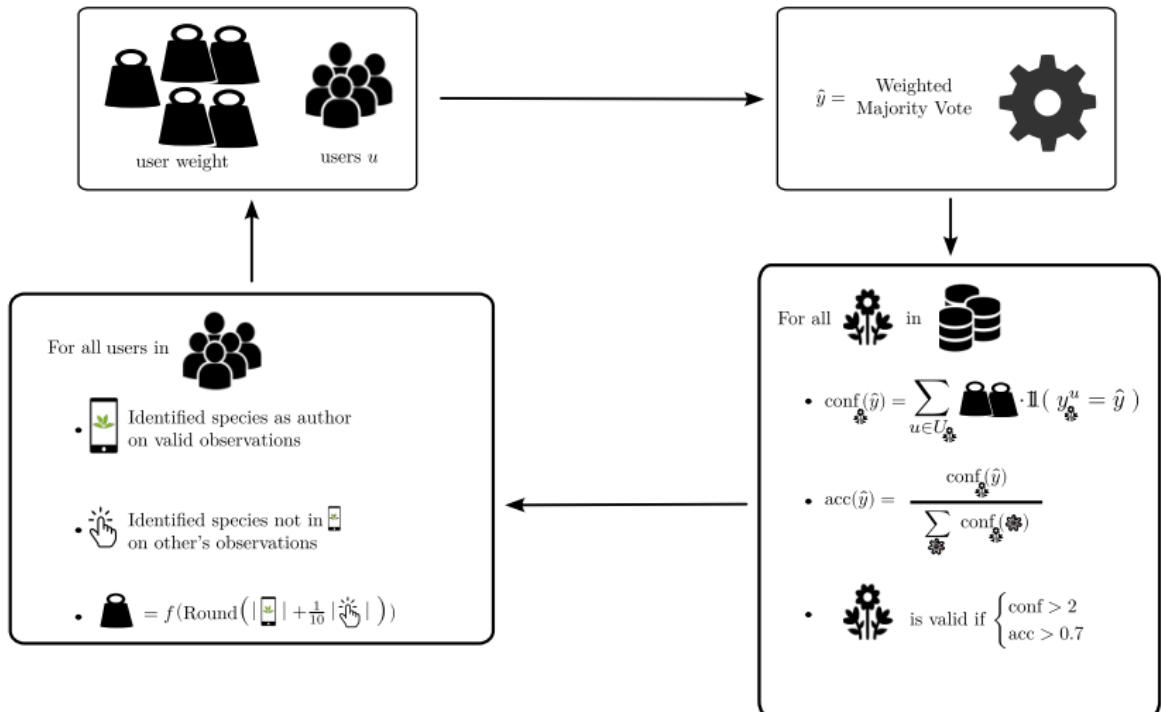


- ▶ South Western European flora obs since 2017
- ▶ $n_{\text{worker}} \simeq 823\,000$ users answered more than $K \simeq 11000$ species
- ▶ $n_{\text{task}} \simeq 6\,700\,000$ observations
- ▶ 9 000 000 votes casted
- ▶ **Imbalance:** 80% of observations are represented by 10% of total votes

- ▶ Extraction of 98 experts (TelaBotanica + expert knowledge)

- ▶ <https://zenodo.org/records/10782465>

PL@NTNET AGGREGATION STRATEGY

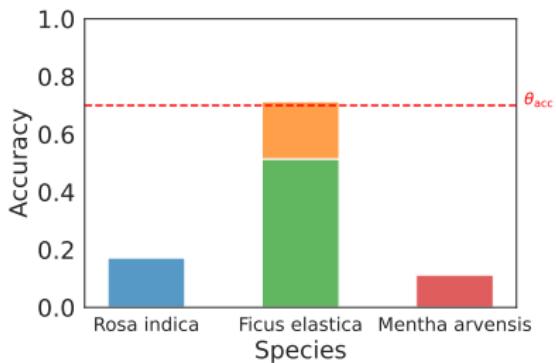
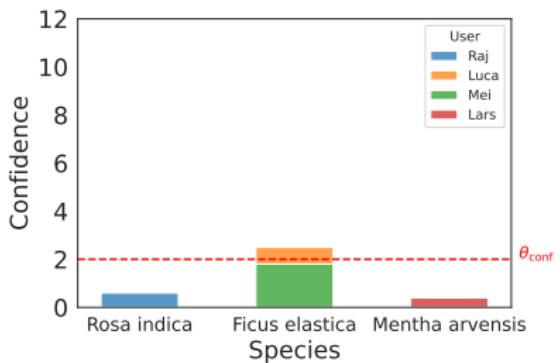


PL@NTNET AGGREGATION STRATEGY

EXAMPLES



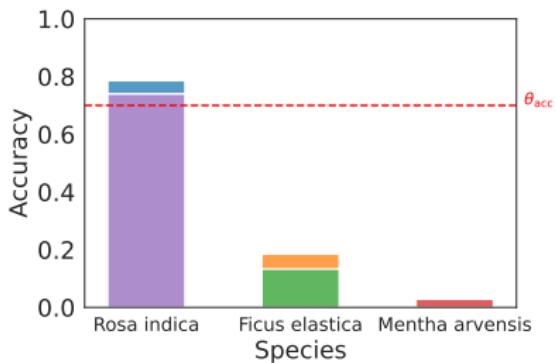
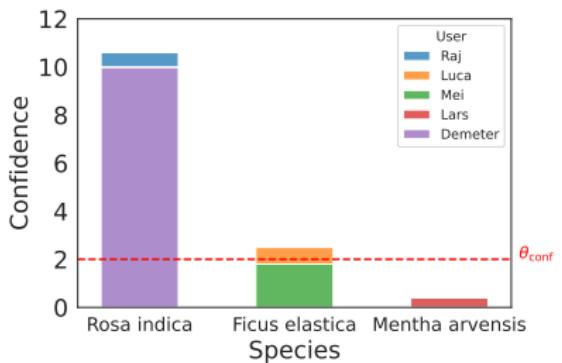
Initial setting



PL@NTNET AGGREGATION STRATEGY EXAMPLES



Label switch

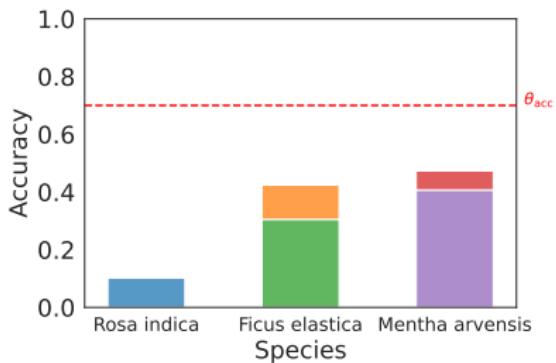
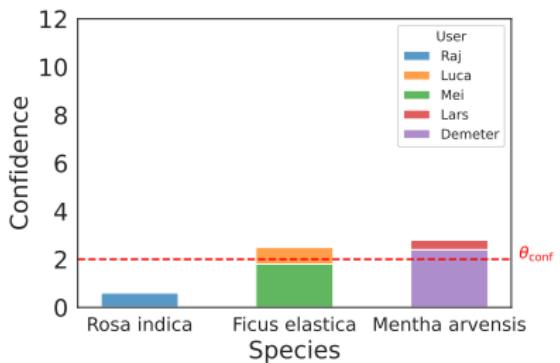


PL@NTNET AGGREGATION STRATEGY

EXAMPLES



Invalidate



COMPARED STRATEGIES

- ▶ **Majority Vote (MV)**

COMPARED STRATEGIES

- ▶ **Majority Vote (MV)**
- ▶ **Worker agreement with aggregate (WAWA)**

$$\text{weight}(w_j) = \text{Accuracy}(\{y_i^{(j)}\}_i, \{\hat{y}_i^{\text{MV}}\}_i)$$

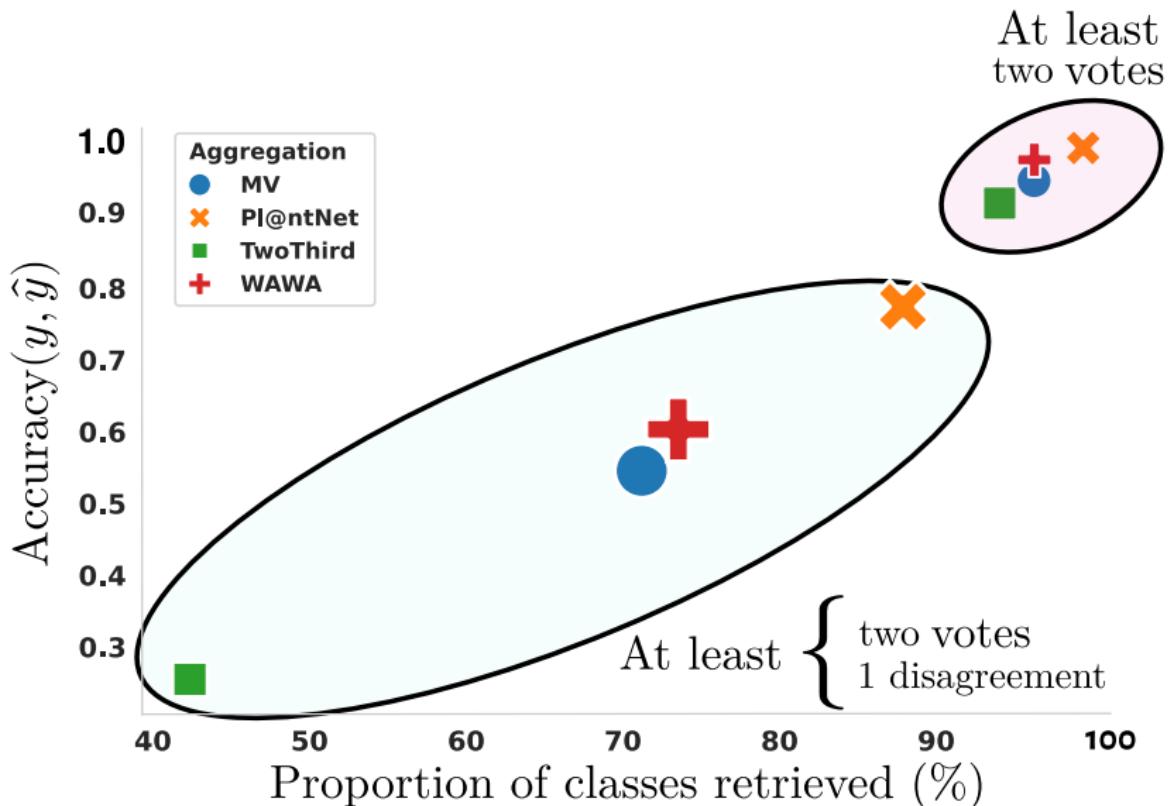
COMPARED STRATEGIES

- ▶ **Majority Vote (MV)**
- ▶ **Worker agreement with aggregate (WAWA)**

$$\text{weight}(w_j) = \text{Accuracy}(\{y_i^{(j)}\}_i, \{\hat{y}_i^{\text{MV}}\}_i)$$

- ▶ **TwoThird** (from iNaturalist pipeline)
 - Need 2 votes
 - 2/3 of agreements

RESULTS



Why?

- ▶ More data
- ▶ Could correct non-xpert users
- ▶ Could invalidate bad quality observation

Why?

- ▶ More data
- ▶ Could correct non-xpert users
- ▶ Could invalidate bad quality observation

Main danger

- ▶ Redundancy: users are already guided by AI predictions

STRATEGIES TO INTEGRATE THE AI VOTE

- ▶ **AI as worker:** naive integration
- ▶ **AI fixed weight:** weight=1.7 to invalidate two new users by $< \theta_{\text{conf}}$
- ▶ **AI invalidating:** fixed weight but can only invalidate observations
- ▶ **AI confident:** fixed weight on data with $\mathbb{P}(\text{predicted species}) > \theta_{\text{score}}$

STRATEGIES TO INTEGRATE THE AI VOTE

- ▶ **AI as worker:** naive integration
- ▶ **AI fixed weight:** weight=1.7 to invalidate two new users by $< \theta_{\text{conf}}$
- ▶ **AI invalidating:** fixed weight but can only invalidate observations
- ▶ **AI confident:** fixed weight on data with $\mathbb{P}(\text{predicted species}) > \theta_{\text{score}}$

⇒ confident AI with $\theta_{\text{score}} = 0.7$ performs best...
but invalidating AI could be preferred for safety ⇐

CONCLUSION

CONCLUSION AND PERSPECTIVES

KEY POINTS



In short:

- ▶ **Identifying ambiguous data** in crowdsourced datasets
- ▶ Creation of the **peerannot library** to run reproducible experiments
- ▶ Release a **new large scale dataset**
- ▶ **Evaluation and improvements** of the Pl@ntNet crowdsourcing setting

In short:

- ▶ Identifying ambiguous data in crowdsourced datasets
- ▶ Creation of the **peerannot library** to run reproducible experiments
- ▶ Release a new large scale dataset
- ▶ Evaluation and improvements of the Pl@ntNet crowdsourcing setting

Perspectives:

- ▶ Need for better data collection: **recommendation system**
- ▶ Extend the library for **multiclass** classification and **regression**



In short:

- ▶ Identifying ambiguous data in crowdsourced datasets
- ▶ Creation of the **peerannot library** to run reproducible experiments
- ▶ Release a new large scale dataset
- ▶ Evaluation and improvements of the Pl@ntNet crowdsourcing setting

Perspectives:

- ▶ Need for better data collection: **recommendation system**
- ▶ Extend the library for **multiclass** classification and **regression**

Thank you!

-  Chu, Z., J. Ma, and H. Wang (2021). "Learning from Crowds by Modeling Common Confusions.". In: *AAAI*, pp. 5832–5840.
-  Dawid, A. and A. Skene (1979). "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm". In: *J. R. Stat. Soc. Ser. C. Appl. Stat.* 28.1, pp. 20–28.
-  Hovy, D. et al. (2013). "Learning whom to trust with MACE". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1120–1130.
-  Ju, C., A. Bibaut, and M. van der Laan (2018). "The relative performance of ensemble methods with deep convolutional neural networks for image classification". In: *J. Appl. Stat.* 45.15, pp. 2800–2818.
-  Lapin, M., M. Hein, and B. Schiele (2016). "Loss functions for top-k error: Analysis and insights". In: *CVPR*, pp. 1468–1477.
-  Lefort, T., A. Affouard, et al. (2024). "Cooperative learning of PI@ntNet's Artificial Intelligence algorithm: how does it work and how can we improve it?" In: *arXiv preprint arXiv:2406.03356*.

-  Lefort, T., B. Charlier, et al. (2024a). "Identify Ambiguous Tasks Combining Crowdsourced Labels by Weighting Areas Under the Margin". In: *Transactions on Machine Learning Research*.
-  — (2024b). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.
-  — (July 2024c). "Weighted majority vote using Shapley values in crowdsourcing". In: *CAp 2024 - Conférence sur l'Apprentissage Automatique*. Lille, France.
-  Peterson, J. C. et al. (2019). "Human Uncertainty Makes Classification More Robust". In: *ICCV*, pp. 9617–9626.
-  Pleiss, G. et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: *NeurIPS*.
-  Rodrigues, F. and F. Pereira (2018). "Deep learning from crowds". In: *AAAI*. Vol. 32.
-  Rodrigues, F., F. Pereira, and B. Ribeiro (2014). "Gaussian process classification and active learning with multiple annotators". In: *ICML*. PMLR, pp. 433–441.

REFERENCES III

-  Servajean, M. et al. (2017). "Crowdsourcing thousands of specialized labels: A Bayesian active training approach". In: *IEEE Transactions on Multimedia* 19.6, pp. 1376–1391.
-  Whitehill, J. et al. (2009). "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise". In: *NeurIPS*. Vol. 22.