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

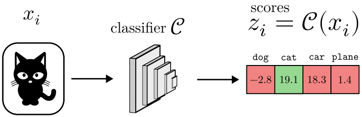
x_i

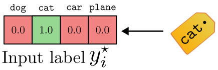
dog	cat	car	plane
0.0	1.0	0.0	0.0

Input label $y_i^\star$

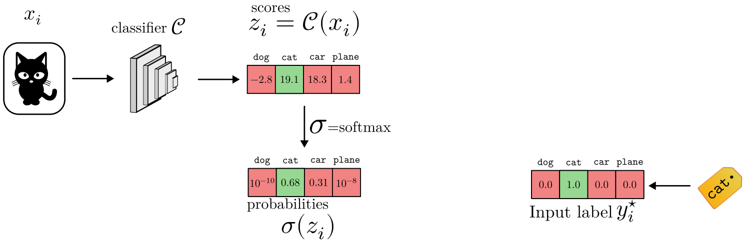
How to train your classifier

How to train your classifier

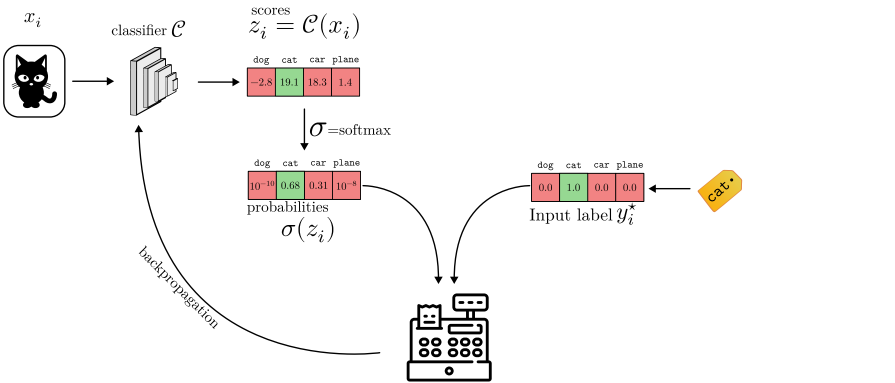




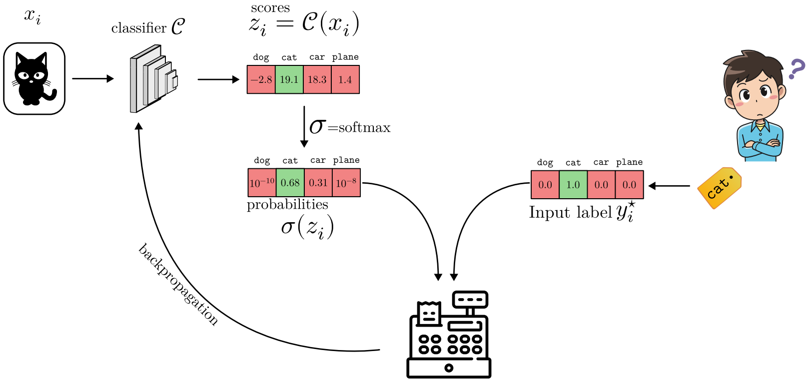
How to train your classifier



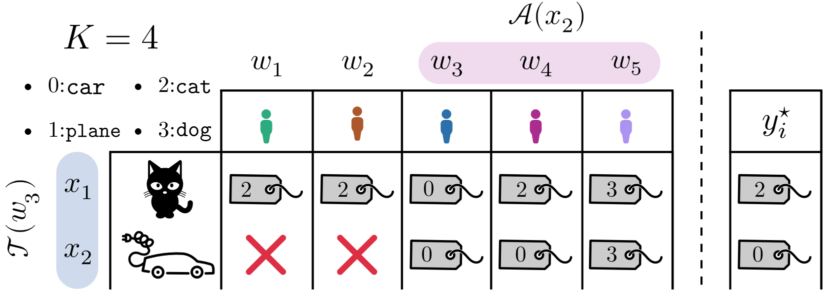
How to train your classifier



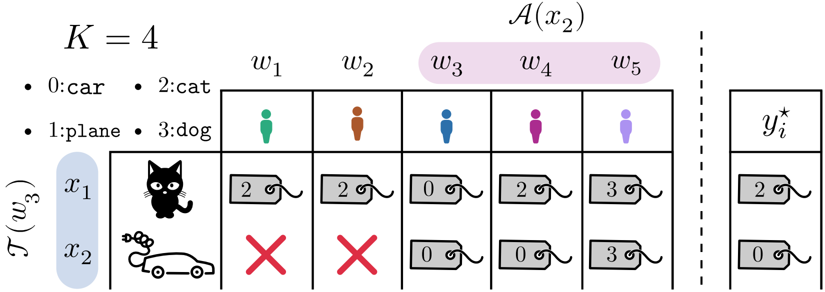
How to train your classifier



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

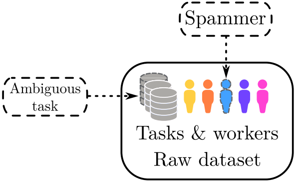


- Workers sort a given task into one of the K classes

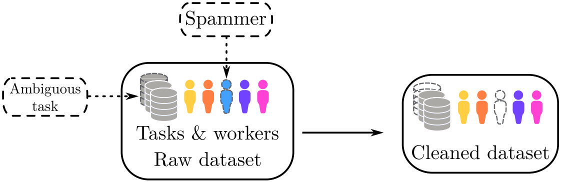


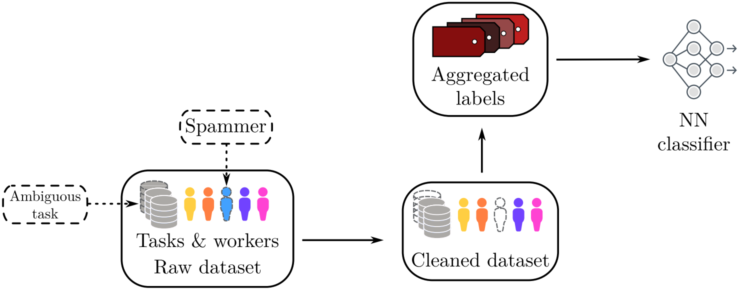
- $y_i^{(j)} \in [K] :=$ answer of worker j to task i
- n_{worker} workers answer n_{task} tasks

Tasks & workers
Raw dataset

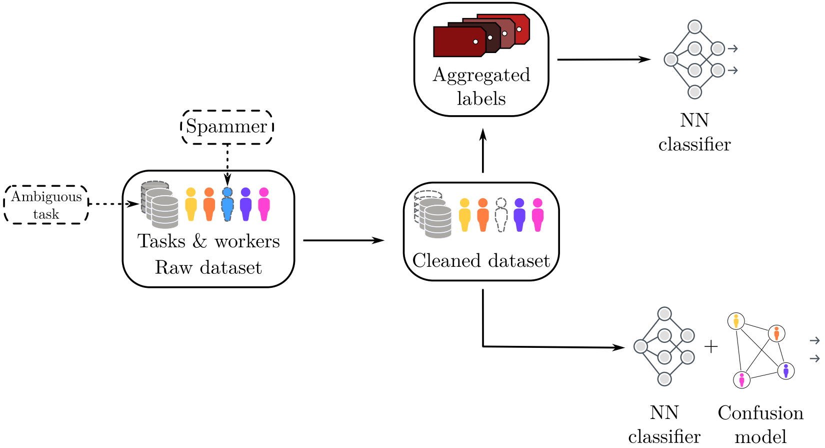


Spammer
Ambiguous task
Tasks & workers
Raw dataset


Spammer
Ambiguous task
Tasks & workers
Raw dataset
Cleaned dataset



Ambiguous task
Spammer
Tasks & workers
Raw dataset
Cleaned dataset
Aggregated labels
NN classifier



Aggregated labels
NN classifier
Spammer
Ambiguous task
Tasks & workers
Raw dataset
Cleaned dataset
+
NN classifier
Confusion model

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 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?
- 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 Pl@ntNet's Artificial Intelligence algorithm: how does it work and how can we improve it?" In: *arXiv preprint arXiv:2406.03356*.

- 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*.

- 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*.

- 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*.

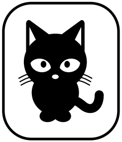
(2) T. Lefort, B. Charlier, et al. (2024b). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

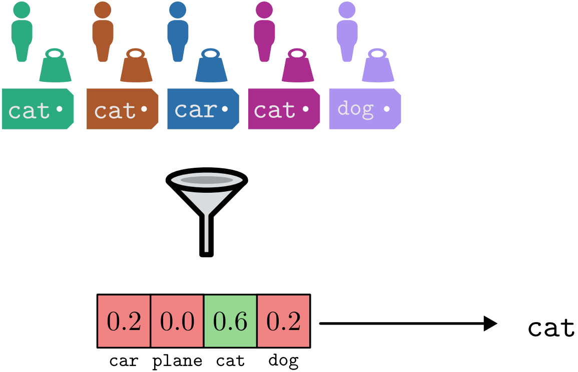
(3) T. Lefort, A. Affouard, et al. (2024). "Cooperative learning of Pl@ntNet's Artificial Intelligence algorithm: how does it work and how can we improve it?" In: *arXiv preprint arXiv:2406.03356*.

Existing aggregation strategies

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

For example with balanced weights:

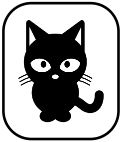


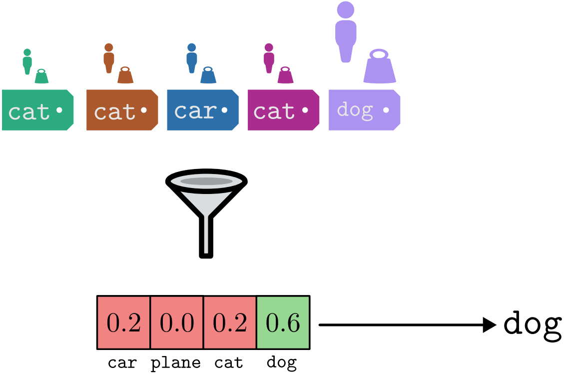


Classical aggregation strategy
(Weighted) Majority Votes

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

For example with unbalanced weights:





Many existing weight choices:

- Inter worker agreement: WAWA[4]:

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

- 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.

- 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:

$$y^{(j)} | y^\star \sim \mathrm{Multinomial}(\pi^{(j)}_{y^\star, \bullet})$$

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

Pros:

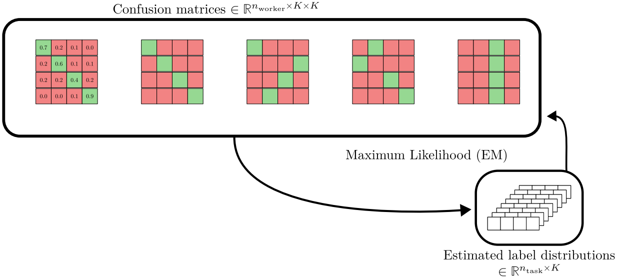
- Finer modelisation
- Can use adversarial workers

Cons:

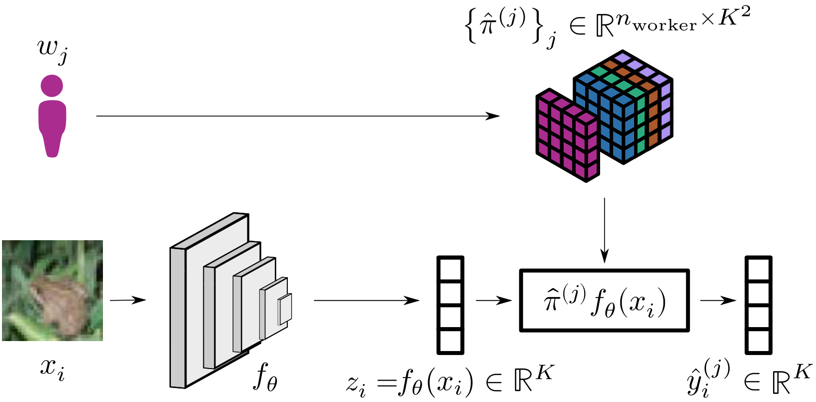
- Memory issue: $n_{\texttt{worker}} \times K^2$ parameters to estimate only the confusion matrices

[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.

Probabilistic model $\longrightarrow$ Likelihood (to maximize via the Expectation Maximization algorithm)

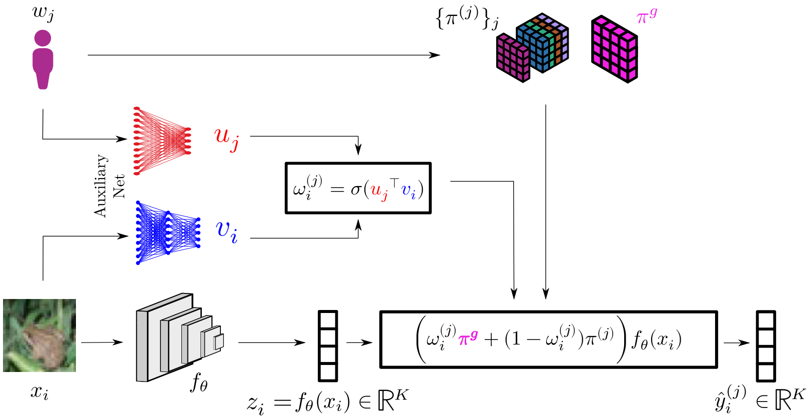


- Idea: put the DS confusion matrix in a neural network as a new layer



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

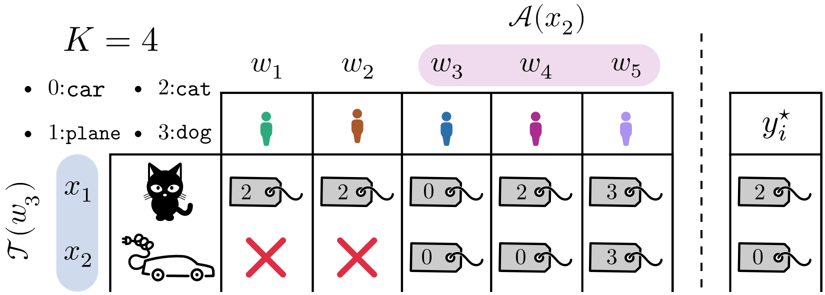
- Idea: CrowdLayer + global and local confusions



[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



WHEN IMAGES HAVE UNDERLYING AMBIGUITY

$K = 4$

- 0:car
- 1:plane
- 2:cat
- 3:dog

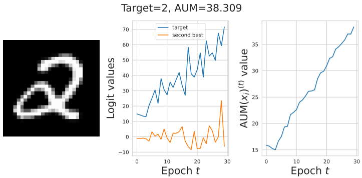
$\mathcal{A}(x_2)$

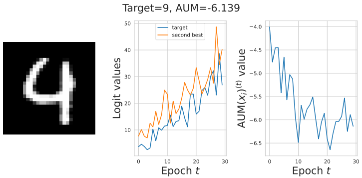
$\mathcal{T}(w_3)$		w_1	w_2	w_3	w_4	w_5	$y_i^\star$
x_1		2	2	0	2	3	2
x_2		✗	✗	0	0	3	0
x_3		1	✗	✗	3	3	1

Ambiguity in classical supervised setting
Area Under the Margin (AUM)

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

- AUM[10]: monitor margin during training





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

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

- 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)

$$\mathrm{AUM}(x_i, y_i) = \underbrace{\frac{1}{T}\sum_{t=1}^{T}}_{\text{Average = Stability}} \overbrace{\left[\underbrace{\mathcal{C}^{(t)}(x_i)_{y_i}}_{\text{Score of assigned label}} - \underbrace{\max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_\ell}_{\text{Other maximum score}} \right]}^{\text{Margin between scores: content of Hinge loss}}$$

[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 $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times [K]$

- 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)

$$\mathrm{AUM}(x_i, y_i) = \overbrace{\frac{1}{T}\sum_{t=1}^{T}}^{\text{Average = Stability}} \overbrace{\Big[\underbrace{\mathcal{C}^{(t)}(x_i)_{y_i}}_{\text{Score of assigned label}} - \underbrace{\max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_\ell}_{\text{Other maximum score}} \Big]}^{\text{Margin between scores: content of Hinge loss}}$$

Challenging for crowdsourcing:

- y_i unknown

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

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)

$$\mathrm{AUM}(x_i, y_i) = \overbrace{\frac{1}{T}\sum_{t=1}^{T}}^{\text{Average = Stability}} \overbrace{\left[\underbrace{\mathcal{C}^{(t)}(x_i)_{y_i}}_{\text{Score of assigned label}} - \underbrace{\max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_\ell}_{\text{Other maximum score}}\right]}^{\text{Margin between scores: content of Hinge loss}}$$

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*.

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

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

$$\widetilde{\mathrm{AUM}}(x_i, \hat{y}_i^{\mathrm{MV}}) = \underbrace{\frac{1}{T}\sum_{t=1}^{T}}_{\text{Average = Stability}} \overbrace{\left[\underbrace{\mathcal{C}^{(t)}(x_i)_{\hat{y}_i^{\mathrm{MV}}}}_{\text{Score of MV label}} - \underbrace{\max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_\ell}_{\text{Other maximum score}} \right]}^{\text{Margin between scores: content of Hinge loss}}$$

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

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

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

$$\widetilde{\mathrm{AUM}}(x_i, \hat{y}_i^{\mathrm{MV}}) = \underbrace{\frac{1}{T}\sum_{t=1}^{T}}_{\text{Average = Stability}} \overbrace{\left[\underbrace{\mathcal{C}^{(t)}(x_i)_{\hat{y}_i^{\mathrm{MV}}}}_{\text{Score of MV label}} - \underbrace{\max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_\ell}_{\text{Other maximum score}} \right]}^{\text{Margin between scores: content of Hinge loss}}$$

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.

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

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

Average = Stability

$$\mathrm{AUMC}(x_i, \hat{y}_i^{\mathrm{MV}}) = \frac{1}{T}\sum_{t=1}^{T} \overbrace{\left[\mathcal{C}^{(t)}(x_i)_{\hat{y}_i^{\mathrm{MV}}} - \mathcal{C}^{(t)}(x_i)_{[2]} \right]}^{\text{Margin between scores: margin for top-1 classification}}$$

Score of MV label

Other maximum score

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 $K-1$)

$$\mathrm{WAUM}(x_i) := \overbrace{\frac{1}{S}\sum_{j\in\mathcal{A}(x_i)}}^{\text{Weighted average of AUM}} \overbrace{s^{(j)}(x_i)}^{\text{Trust score of } w_j \text{ for } x_i} \overbrace{\frac{1}{T}\sum_{t=1}^{T}}^{\text{Average = Stability}} \overbrace{\left[\underbrace{\sigma^{(t)}_{y_i^{(j)}}(x_i)}_{\text{Probability of assigned label by worker } w_j} - \underbrace{\sigma^{(t)}_{[2]}(x_i)}_{\text{Second maximum probability}}\right]}^{\text{Margin between scores}}$$

[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.

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.

Our chosen worker/task score:

- Consider a score of the form[14]: worker skill $\times$ task difficulty[15]

$$s^{(j)}(x_i) = \Big\langle \underbrace{\mathrm{diag}(\hat{\pi}^{(j)})}_{\text{Worker } j \text{ overall ability}} \;\Big|\; \underbrace{\sigma^{(T)}(x_i)}_{\text{Difficulty of task } i} \Big\rangle \in [0,1]$$

[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.

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

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

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

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

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

Usage (for learning):

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

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

Usage (for learning):

- **Prune** x_i's with $\mathrm{WAUM}(x_i)$ 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

2006

80 million images creation

- labels from web crawling
- 32×32 quality

2009

CIFAR-10 creation:

- 10 classes and 60K images
- 50K train / 10k test
- Students paid to filter out mislabeled images

Problem

- votes have been lost
- errors have been found

2019

CIFAR-10H creation

- Switch original train/test sets
- Pay 2571 workers to label images
- $\simeq$ 50 votes per image

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

2006

80 million images creation

- labels from web crawling
- 32×32 quality

2009

CIFAR-10 creation:

- 10 classes and 60K images
- 50K train / 10k test
- Students paid to filter out mislabeled images

Problem

- votes have been lost
- errors have been found

2019

CIFAR-10H creation

- Switch original train/test sets
- Pay 2571 workers to label images
- $\simeq$ 50 votes per image

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

Image #7681
CIFAR-10 label: airplane

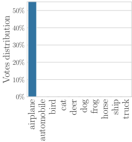


Image #6750
CIFAR-10 label: deer

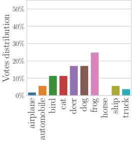
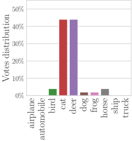


Image #9246
CIFAR-10 label: cat



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

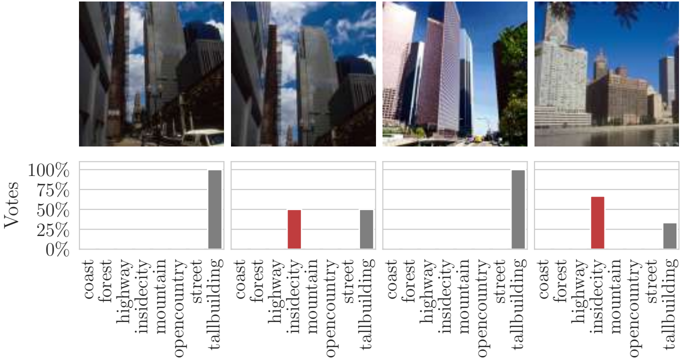
Presenting LabelMe dataset[17]

- 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]

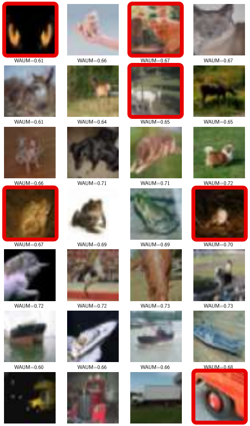
- 1000 training / 500 validation / 1188 test images
- 59 workers: each task has up to 3 votes
- 8 classes: highway, insidecity, tallbuilding, street, forest, coast, mountain, opencountry



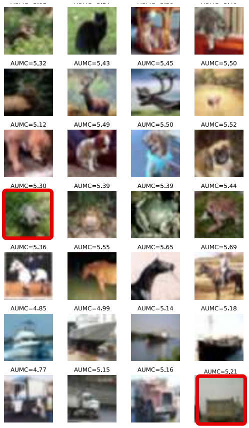
[17] F. Rodrigues, F. Pereira, and B. Ribeiro (2014). "Gaussian process classification and active learning with multiple annotators". In: *ICML*. PMLR, pp. 433–441.

QUALITATIVE RESULTS

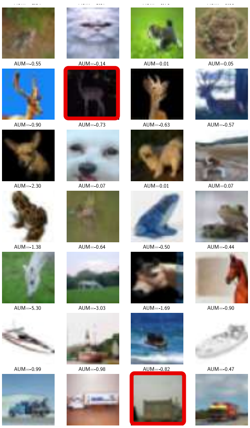
WAUM
(crowdsourcing)

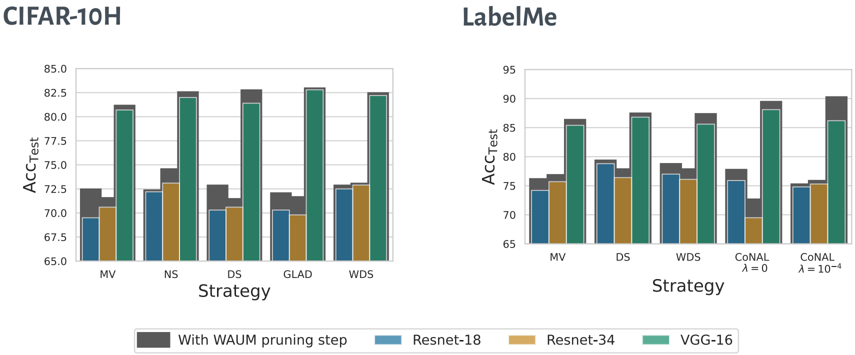


AUMC
(crowdsourcing)



AUM
(no crowdsourcing)




CIFAR-10H
LabelMe
85.0
82.5
80.0
77.5
75.0
72.5
70.0
67.5
65.0
$\mathrm{Acc}_{\mathrm{Test}}$
MV
NS
DS
GLAD
WDS
Strategy
95
90
85
80
75
70
65
$\mathrm{Acc}_{\mathrm{Test}}$
MV
DS
WDS
CoNAL $\lambda = 0$
CoNAL $\lambda = 10^{-4}$
Strategy
With WAUM pruning step
Resnet-18
Resnet-34
VGG-16

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

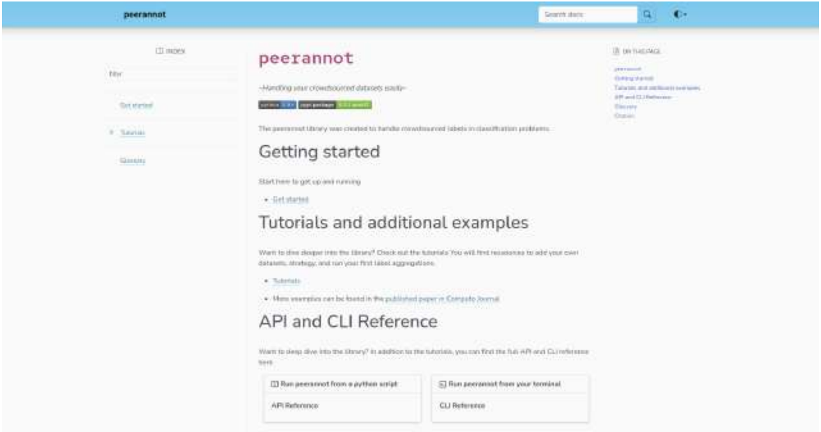
Peerannot library

Handle crowdsourced data in classification

- Python library for small and large crowdsourced datasets

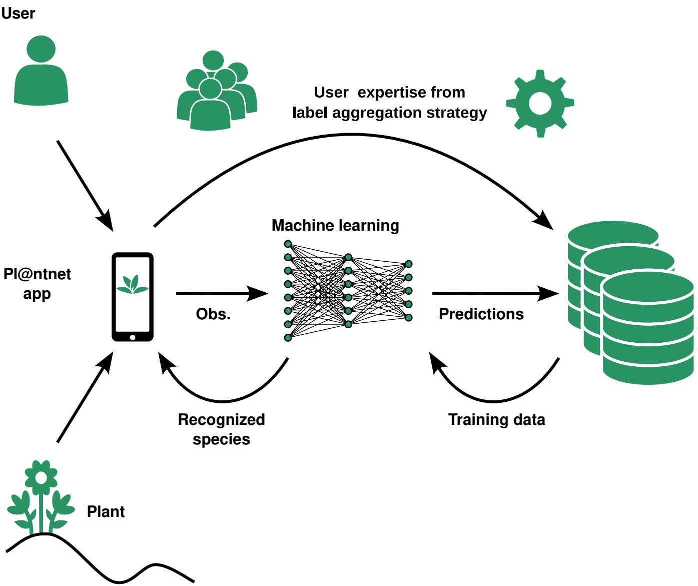
```
pip install peerannot
```

- Documentation available at: `https://peerannot.github.io`



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

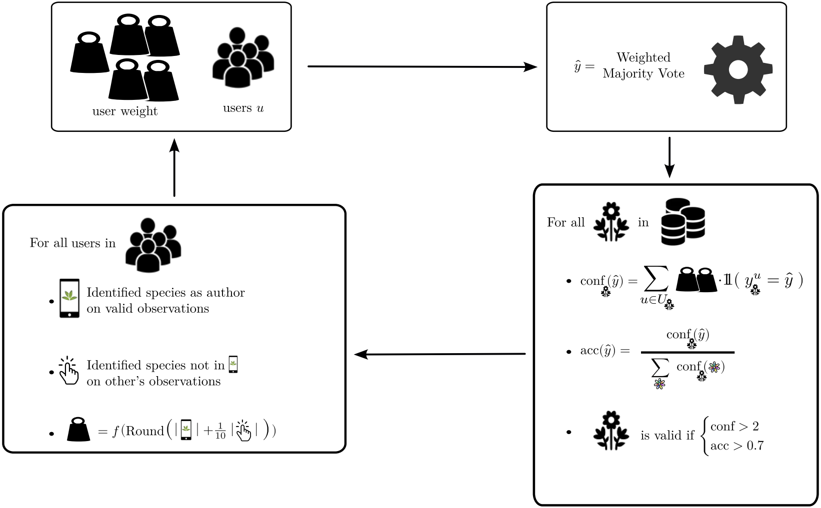
Realeasing a new dataset

- 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

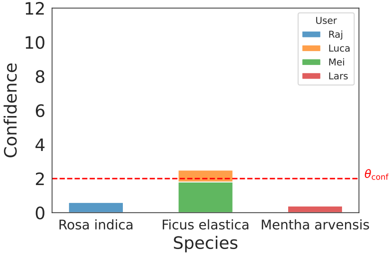
- Extraction of 98 experts (TelaBotanica + expert knowledge)

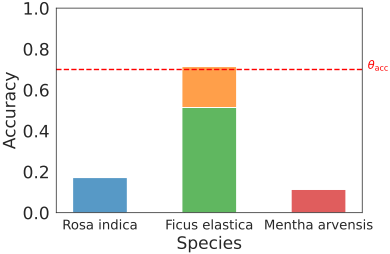
- `https://zenodo.org/records/10782465`

Pl@ntNet aggregation strategy

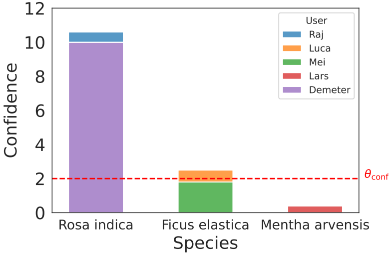


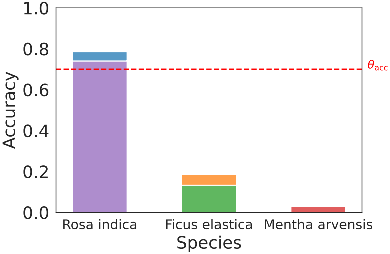
Initial setting





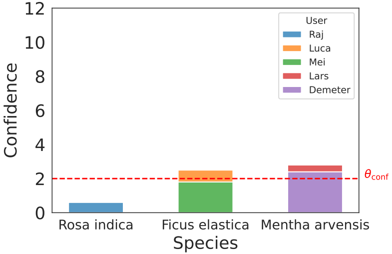
Label switch

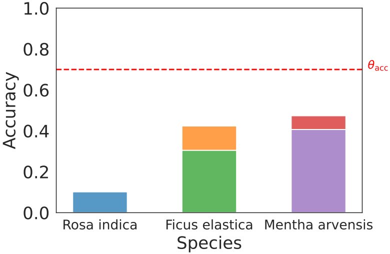




Pl@ntNet aggregation strategy
Examples

Invalidate





- **Majority Vote** (MV)

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

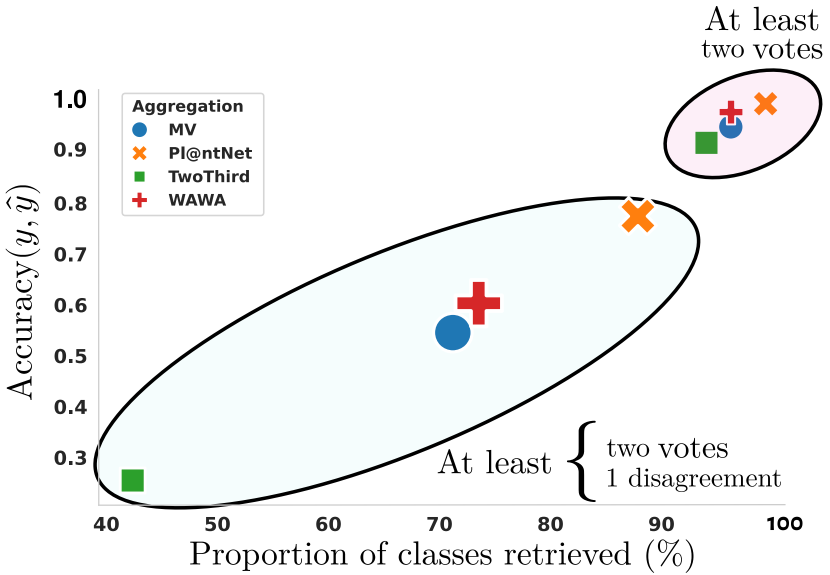
$$\mathrm{weight}(w_j) = \mathrm{Accuracy}(\{y_i^{(j)}\}_i, \{\hat{y}_i^{\mathrm{MV}}\}_i)$$

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

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

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

Results


At least
two votes
Aggregation
MV
Pl@ntNet
TwoThird
WAWA
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Accuracy$(y, \hat{y})$
At least
two votes
1 disagreement
40
50
60
70
80
90
100
Proportion of classes retrieved (%)

Why?

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

Integrating the AI vote

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 $\mathbb{P}(\text{predicted species}) > \theta_{\text{score}}$

$\Longrightarrow$ confident AI with $\theta_{\text{score}} = 0.7$ performs best…
but invalidating AI could be preferred for safety $\Longleftarrow$

CONCLUSION

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 Pl@ntNet’s Artificial Intelligence algorithm: how does it work and how can we improve it?” In: *arXiv preprint arXiv:2406.03356*.

References II

- 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.