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**

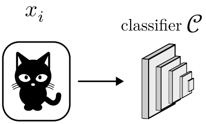
How to train your classifier
Deep learning image classification pipeline

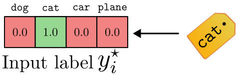
x_i

dog	cat	car	plane
0.0	1.0	0.0	0.0

Input label $y_i^\star$

How to train your classifier
Deep learning image classification pipeline

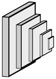




How to train your classifier
Deep learning image classification pipeline

x_i

classifier $\mathcal{C}$

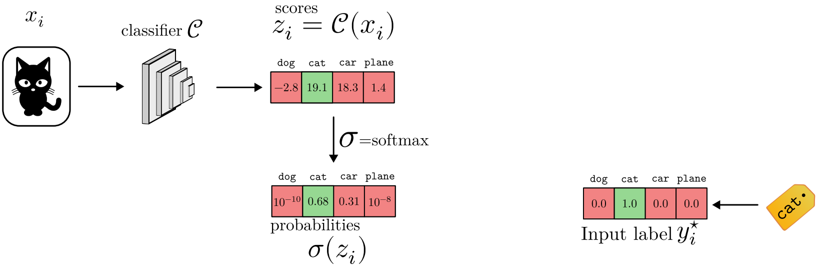


scores
$z_i = \mathcal{C}(x_i)$

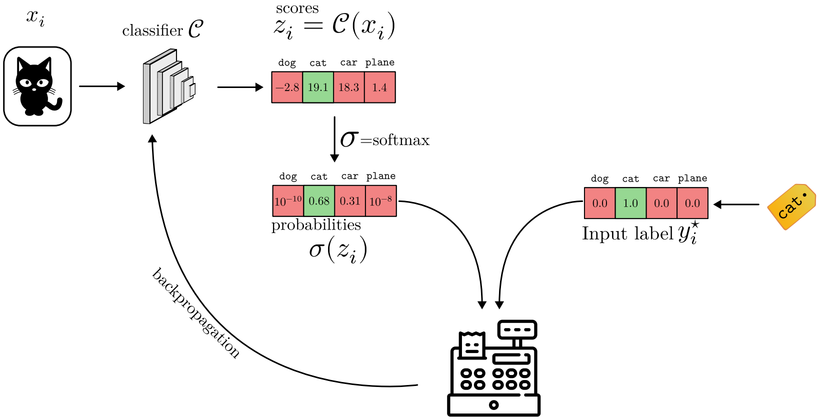
dog	cat	car	plane
-2.8	19.1	18.3	1.4

dog	cat	car	plane
0.0	1.0	0.0	0.0

Input label $y_i^\star$

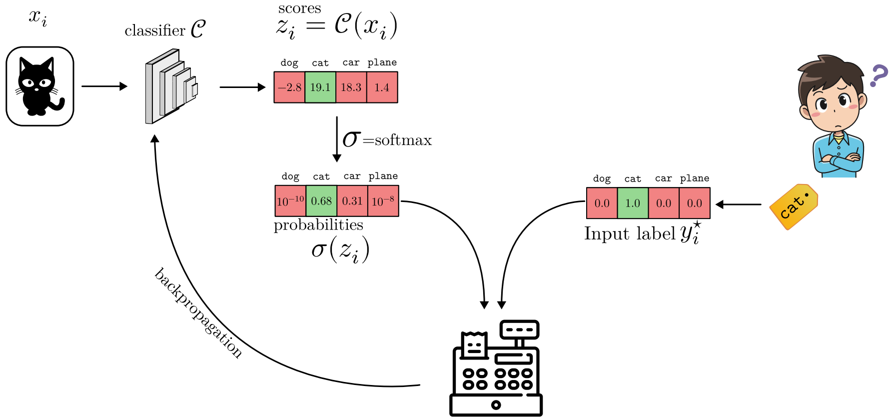


x_i
classifier $\mathcal{C}$
scores
$z_i = \mathcal{C}(x_i)$
dog cat car plane
-2.8 19.1 18.3 1.4
σ =softmax
dog cat car plane
10^{-10} 0.68 0.31 10^{-8}
probabilities
$\sigma(z_i)$
dog cat car plane
0.0 1.0 0.0 0.0
Input label $y_i^\star$
cat

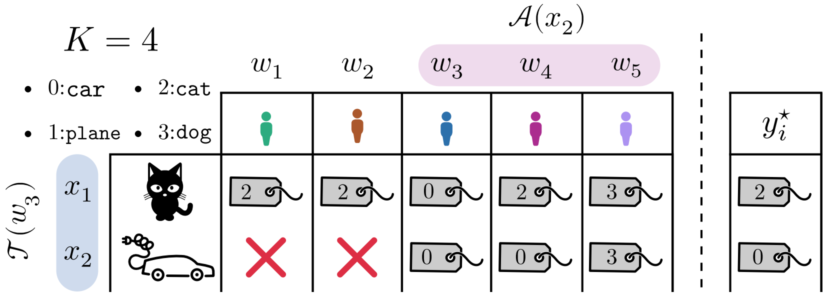

x_i
classifier $\mathcal{C}$
scores
$z_i = \mathcal{C}(x_i)$
dog cat car plane
−2.8 19.1 18.3 1.4
σ =softmax
dog cat car plane
10^{-10} 0.68 0.31 10^{-8}
probabilities
$\sigma(z_i)$
dog cat car plane
0.0 1.0 0.0 0.0
Input label $y_i^\star$
cat
backpropagation

How to train your classifier

Deep learning image classification pipeline

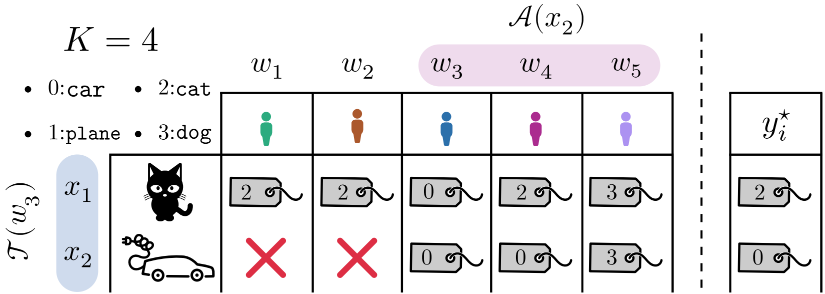


- Workers sort a given task into one of the **K classes**



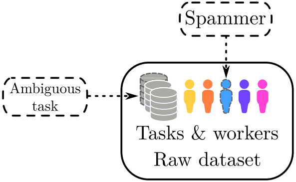
Ask citizens to label our data
Framework and notation

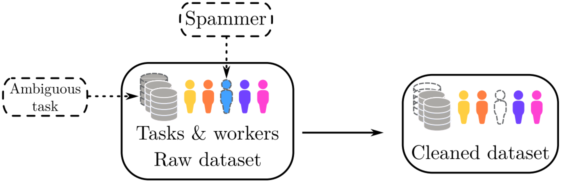
- Workers sort a given task into one of the **K classes**



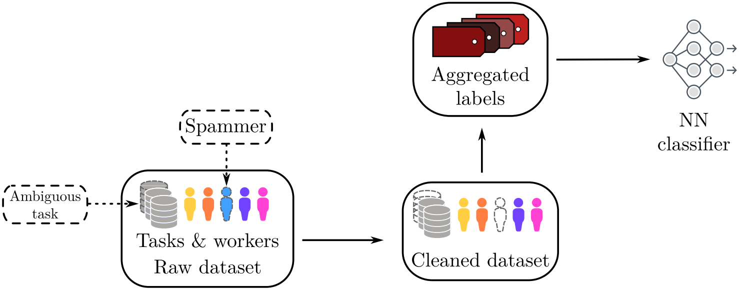
- $y_i^{(j)} \in [K] :=$ answer of worker j to task i
- n_{worker} workers answer n_{task} tasks

From the data to the classifier
The pipeline


Spammer
Ambiguous task
Tasks & workers
Raw dataset

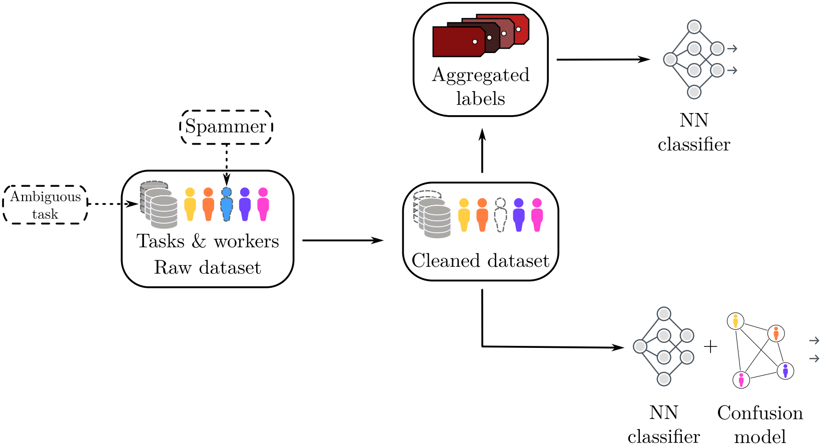


Spammer
Ambiguous task
Tasks & workers
Raw dataset
Cleaned dataset



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

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 Pl@ntNet’s Artificial Intelligence algorithm: how does it work and how can we improve it?” In: *submitted to Methods in Ecology and Evolution*.

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: *submitted to Methods in Ecology and Evolution*.

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: *submitted to Methods in Ecology and Evolution*.

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: *submitted to Methods in Ecology and Evolution*.

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: *submitted to Methods in Ecology and Evolution*.

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*.

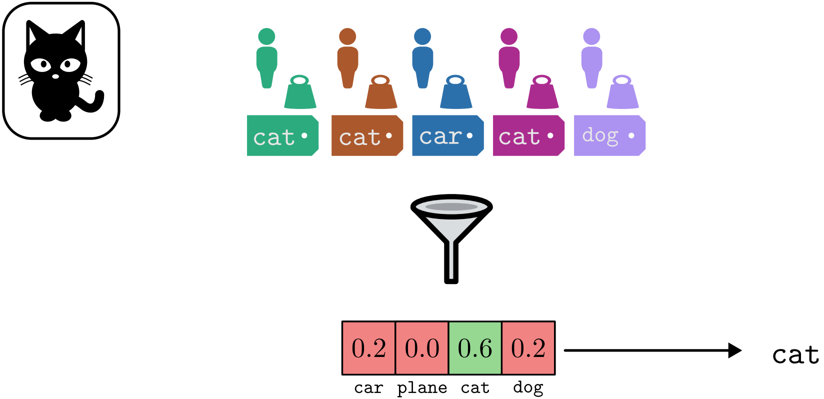
[(2)] T. Lefort, B. Charlier, et al. (2024b). “Peerannot: Classification for Crowdsourced Image Datasets with Python”. In: *Computo*.

[(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: *submitted to Methods in Ecology and Evolution*.

Existing aggregation strategies

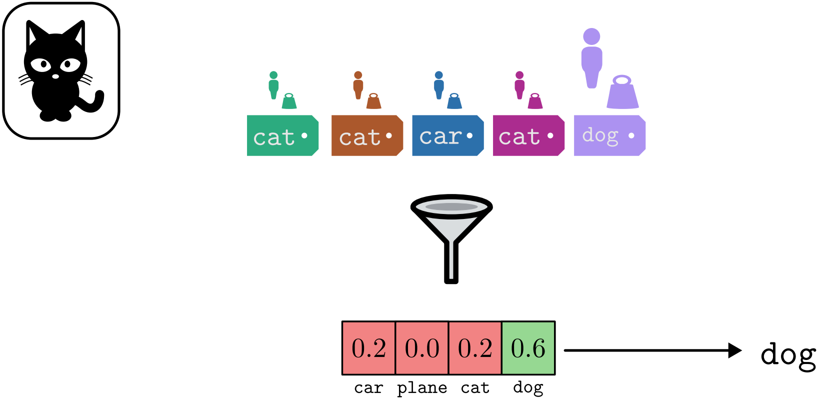
$$\hat{y}_i^{\mathrm{WMV}} = \operatorname*{argmax}_{k\in[K]} \sum_{j\in\mathcal{A}(x_i)} {}_j \mathbb{1}(y_i^{(j)} = k)$$

For example with balanced weights:



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

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:

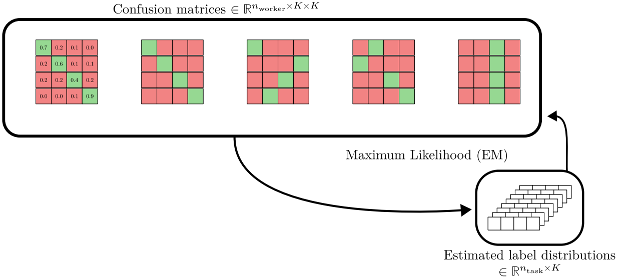
- Finer modelisation
- Can use adversarial workers

Cons:

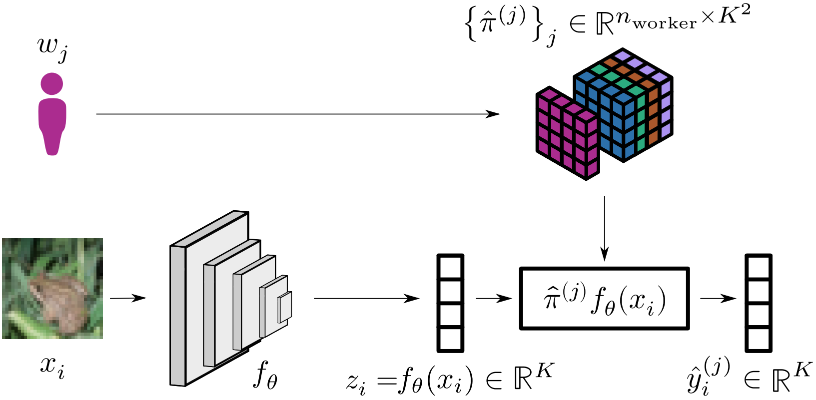
- Memory issue: $n_{\texttt{worker}} \times K^2$ parameters to estimate only the confusion matrices

[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)

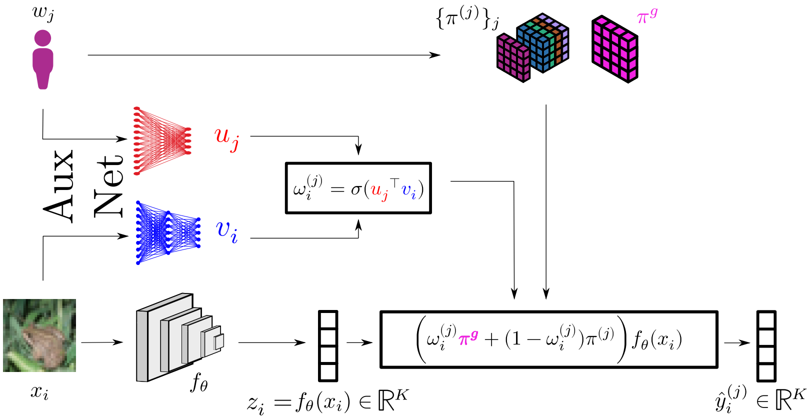


- Idea: put the DS confusion matrix in a neural network as a new layer



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

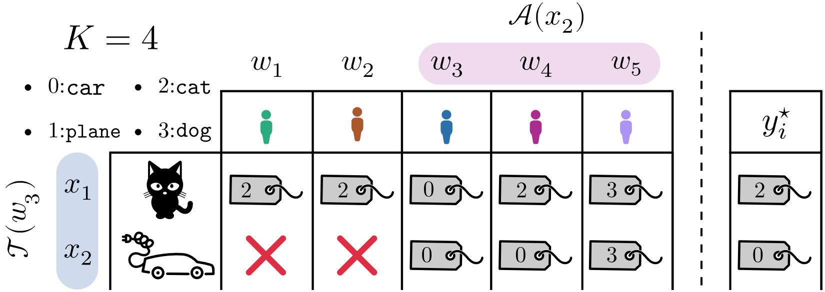
- Idea: CrowdLayer + global and local confusions



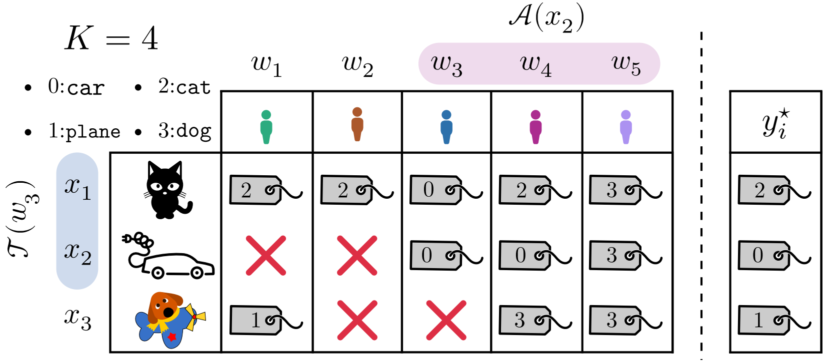
[9] Z. Chu, J. Ma, and H. Wang (2021). "Learning from Crowds by Modeling Common Confusions.". In: *AAAI*, pp. 5832–5840.

Identify ambiguous tasks in crowdsourced datasets

When images have underlying ambiguity

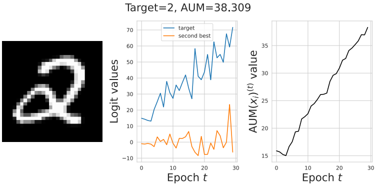


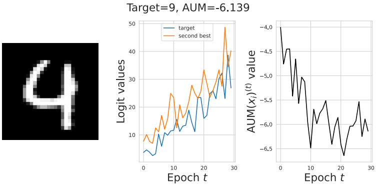
When images have underlying ambiguity



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

- 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
- Scores ordered: $\mathcal{C}(x_i)_{[1]} \geq \cdots \geq \mathcal{C}(x_i)_{[K]}$

Average = Stability

$$\mathrm{AUM}(x_i, y_i) = \frac{1}{T}\sum_{t=1}^{T} \overbrace{\left[\mathcal{C}^{(t)}(x_i)_{y_i} - \mathcal{C}^{(t)}(x_i)_{[2]} \right]}^{\text{Margin between scores: content of Hinge loss}}$$

Score of assigned label

Other maximum score

[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
- Scores ordered: $\mathcal{C}(x_i)_{[1]} \geq \cdots \geq \mathcal{C}(x_i)_{[K]}$

$$\mathrm{AUM}(x_i, y_i) = \underbrace{\frac{1}{T}\sum_{t=1}^{T}}_{\text{Average = Stability}} \overbrace{\Big[\underbrace{\mathcal{C}^{(t)}(x_i)_{y_i}}_{\text{Score of assigned label}} - \underbrace{\mathcal{C}^{(t)}(x_i)_{[2]}}_{\text{Other maximum score}}\Big]}^{\text{Margin between scores: content of Hinge loss}}$$

Challenging for crowdsourcing:

- y_i unknown

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

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
- Scores ordered: $\mathcal{C}(x_i)_{[1]} \geq \cdots \geq \mathcal{C}(x_i)_{[K]}$

$$\mathrm{AUM}(x_i, y_i) = \overbrace{\frac{1}{T}\sum_{t=1}^{T}}^{\text{Average = Stability}} \overbrace{\Big[\underbrace{\mathcal{C}^{(t)}(x_i)_{y_i}}_{\text{Score of assigned label}} - \underbrace{\mathcal{C}^{(t)}(x_i)_{[2]}}_{\text{Other maximum score}} \Big]}^{\text{Margin between scores: content of Hinge loss}}$$

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 $\hat{y}_i^{\mathrm{MV}}$

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

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[12]

With:

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

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

[12] 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 (following Servajean et al. (2017)[13]) of the form[14]:

worker skill $\times$ task difficulty

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

[13] 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.

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

- 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[15] 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`

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

Presenting CIFAR-10H[15] 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

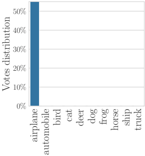
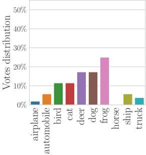


Image #6750
CIFAR-10 label: deer



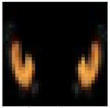
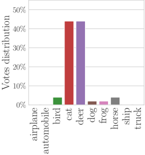


Image #9246
CIFAR-10 label: cat



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

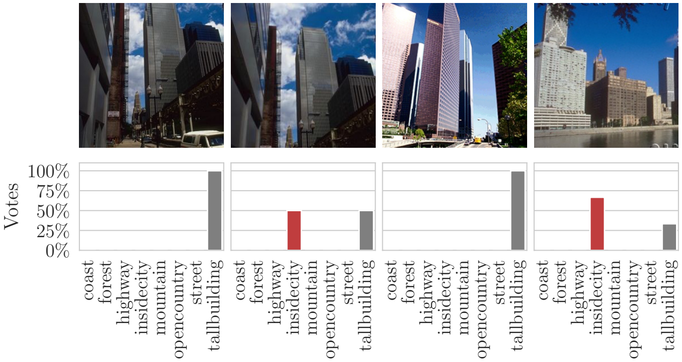
Presenting LabelMe dataset[16]

- 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`

[16] 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[16]

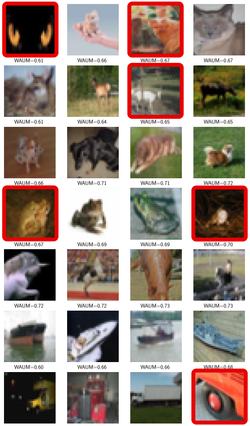
- 1000 training / 500 validation / 1188 test images
- 59 workers: each task has up to 3 votes
- 8 classes: `highway,insidecity,tallbuilding,street,forest,coast, mountain,opencountry`



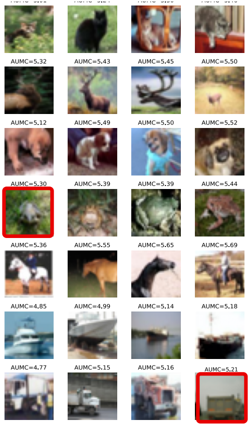
[16] F. Rodrigues, F. Pereira, and B. Ribeiro (2014). "Gaussian process classification and active learning with multiple annotators". In: *ICML*. PMLR, pp. 433–441.

QUALITATIVE RESULTS

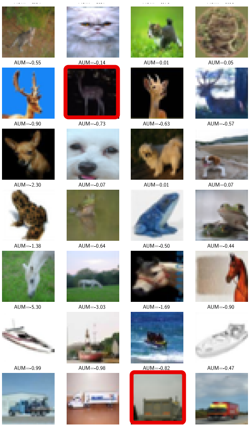
WAUM
(crowdsourcing)



AUMC
(crowdsourcing)

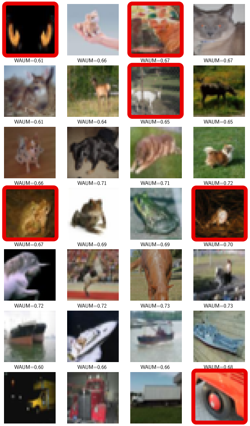


AUM
(no crowdsourcing)



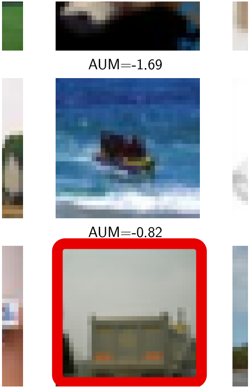
QUALITATIVE RESULTS

WAUM
(crowdsourcing)



AUMC
(crowdsourcing)

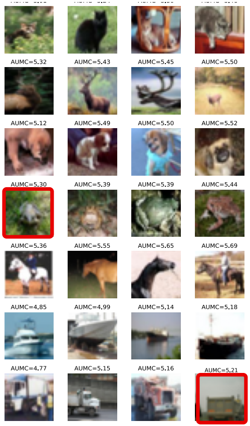
AUM
(no crowdsourcing)



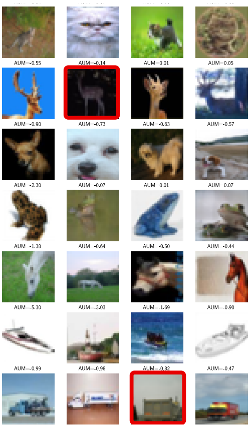
Qualitative results

WAUM
(crowdsourcing)

AUMC
(crowdsourcing)

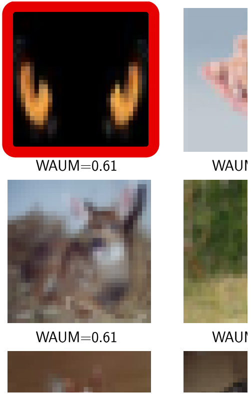


AUM
(no crowdsourcing)

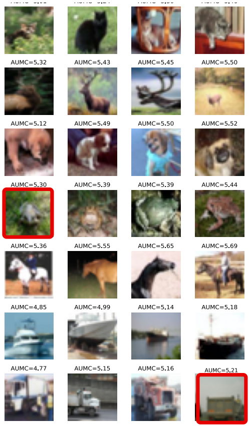


QUALITATIVE RESULTS

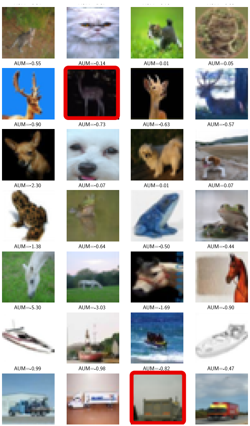
WAUM (crowdsourcing)



AUMC (crowdsourcing)

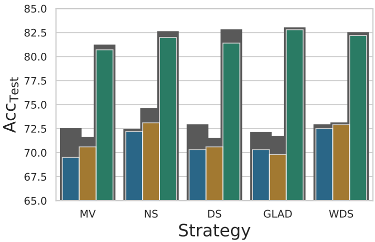


AUM (no crowdsourcing)

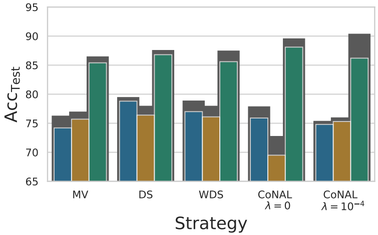


Ablation study

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!

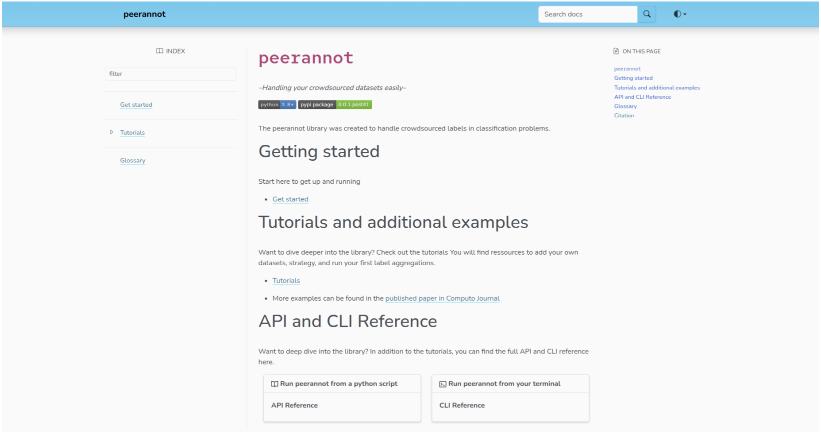
The peerannot library

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`



- **Handle large datasets**: we implemented on-the-fly queries to avoid storing all data in memory (json data format)

- **Handle large datasets**: we implemented on-the-fly queries to avoid storing all data in memory (json data format)
- CLI (Command Line Interface) for **efficient pipelines running jobs**

- **Handle large datasets**: we implemented on-the-fly queries to avoid storing all data in memory (json data format)
- CLI (Command Line Interface) for **efficient pipelines running jobs**
- **More identification metrics** and aggregation strategies for classification

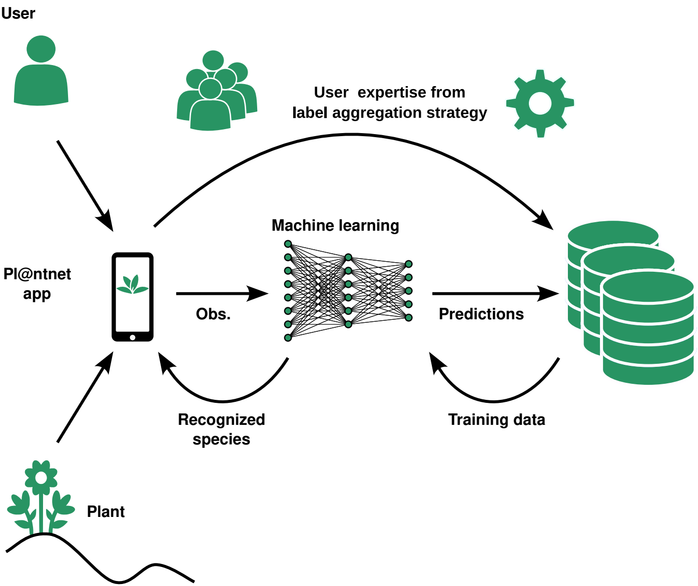
Peerannot library

Comparison with existing library CrowdKit

- **Handle large datasets**: we implemented on-the-fly queries to avoid storing all data in memory (json data format)
- CLI (Command Line Interface) for **efficient pipelines running jobs**
- **More identification metrics** and aggregation strategies for classification
- **Seamless integration** with `PyTorch` pipelines:
  - directly train `Torchvision` classifiers on the data
  - keep the same framework end-to-end
  - support top-k and calibration metrics at evaluation time

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

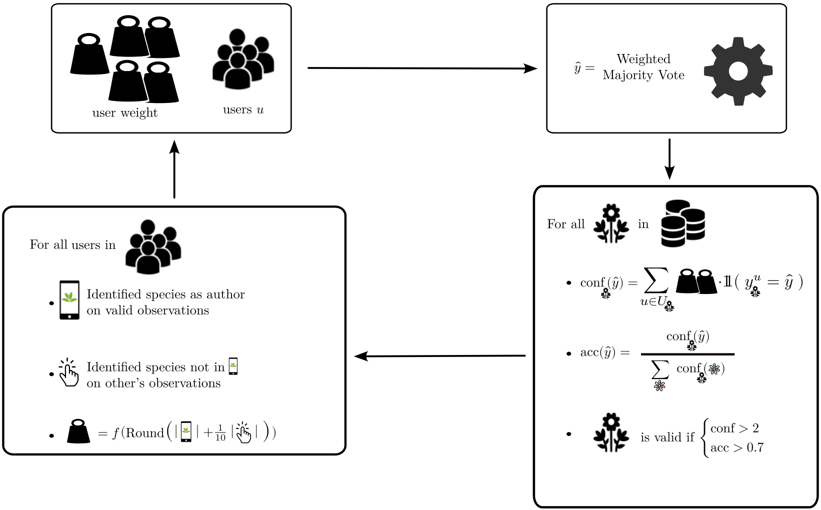
Realeasing a new dataset

- 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

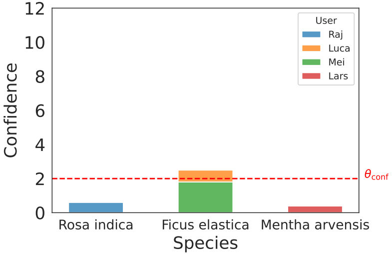
- Extraction of 98 experts (TelaBotanica + expert knowledge)

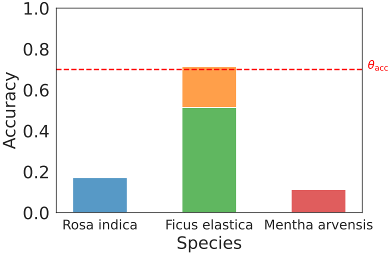
- `https://zenodo.org/records/10782465`

PL@NTNET AGGREGATION STRATEGY

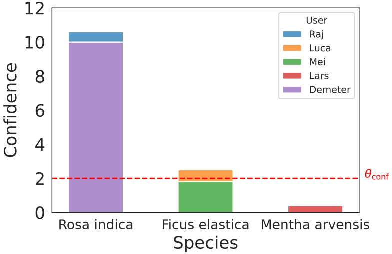


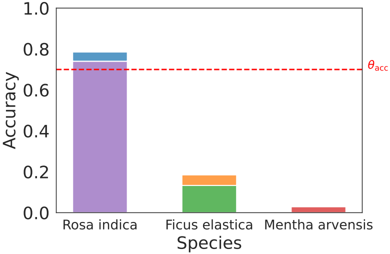
Initial setting



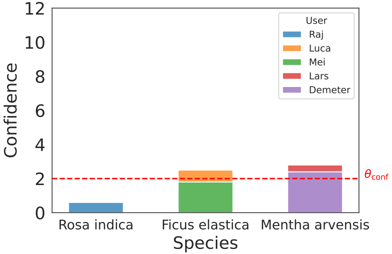


Label switch

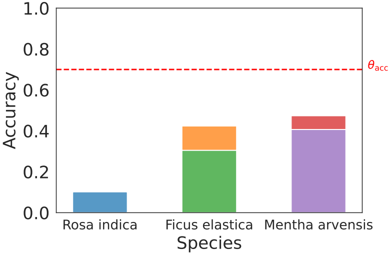




Invalidate



User
Raj
Luca
Mei
Lars
Demeter
Confidence
12
10
8
6
4
2
0
θ_{conf}
Rosa indica
Ficus elastica
Mentha arvensis
Species



Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
θ_{acc}
Rosa indica
Ficus elastica
Mentha arvensis
Species

- **Majority Vote** (MV)

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

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

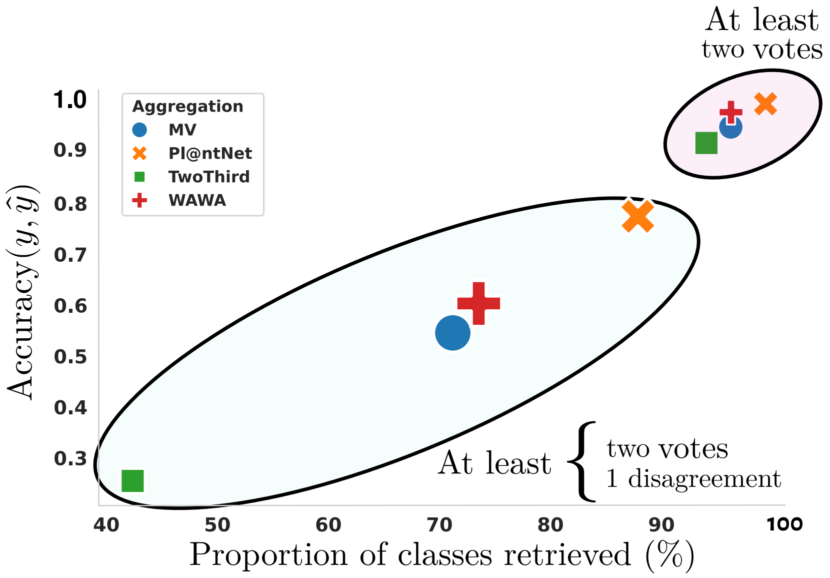
Compared strategies

- **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-expert users
- Could invalidate bad quality observation

(17) I. Shumailov et al. (2024). "AI models collapse when trained on recursively generated data". In: *Nature* 631.8022, pp. 755–759.

Why?

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

Main danger

- Model collapse(17): users are already guided by AI predictions

(17) I. Shumailov et al. (2024). "AI models collapse when trained on recursively generated data". In: *Nature* 631.8022, pp. 755–759.

Strategies to integrate the AI vote

- AI **as worker**: naive integration

Strategies to integrate the AI vote

- AI **as worker**: naive integration
- AI **fixed weight:**
  - weight fixed to 1.7
  - can invalidate two new users but is not self-validating

- AI **as worker**: naive integration
- AI **fixed weight:**
  - weight fixed to 1.7
  - can invalidate two new users but is not self-validating
- AI **invalidating:**
  - weight fixed to 1.7
  - can only invalidate observation

Strategies to integrate the AI vote

- AI **as worker**: naive integration
- AI **fixed weight:**
  - weight fixed to 1.7
  - can invalidate two new users but is not self-validating
- AI **invalidating:**
  - weight fixed to 1.7
  - can only invalidate observation
- AI **confident:**
  - weight fixed to 1.7
  - can participate if confidence in prediction high enough (θ_{score})

Strategies to integrate the AI vote

- AI **as worker**: naive integration
- AI **fixed weight:**
  - weight fixed to 1.7
  - can invalidate two new users but is not self-validating
- AI **invalidating:**
  - weight fixed to 1.7
  - can only invalidate observation
- AI **confident:**
  - weight fixed to 1.7
  - can participate if confidence in prediction high enough (θ_{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 **multilabel** 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 **multilabel** classification and **regression**

Thank you!

References I

- 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.
- 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: *submitted to Methods in Ecology and Evolution*.

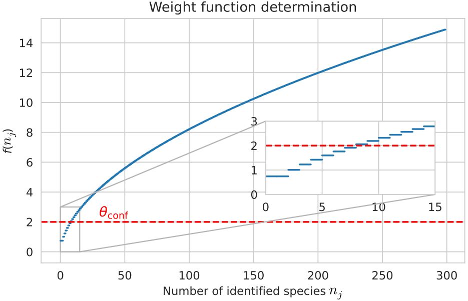
References II

- 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

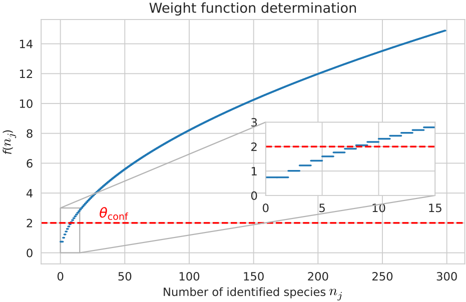
- Servajean, M. et al. (2017). "Crowdsourcing thousands of specialized labels: A Bayesian active training approach". In: *IEEE Transactions on Multimedia* 19.6, pp. 1376–1391.
- Shumailov, I. et al. (2024). "AI models collapse when trained on recursively generated data". In: *Nature* 631.8022, pp. 755–759.
- Whitehill, J. et al. (2009). "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise". In: *NeurIPS*. Vol. 22.

$$f(n_j) = n_j^{\alpha} - n_j^{\beta} + \gamma \text{ with } \begin{cases} \alpha & = 0.5 \\ \beta & = 0.2 \\ \gamma & \simeq 0.74 \end{cases}$$



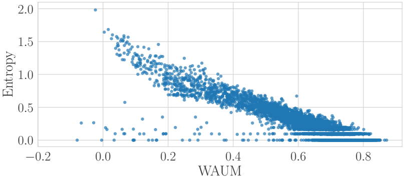
- With 8 identified species one becomes self-validating

$$f(n_j) = n_j^{\alpha} - n_j^{\beta} + \gamma \text{ with } \begin{cases} \alpha & = 0.5 \\ \beta & = 0.2 \\ \gamma & \simeq 0.74 \end{cases}$$

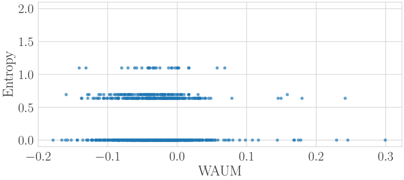


- With 8 identified species one becomes self-validating
- But observations can be invalidated at any time in the future

CIFAR-10H



LabelMe



- Entropy is irrelevant with few votes per task