Improve learning combining crowdsourced labels: the weighting Areas Under the Margin

Joseph Salmon
IMAG, Univ Montpellier, CNRS
Institut Universitaire de France (IUF)

Ongoing joint work with...

- Benjamin Charlier (IMAG, Univ Montpellier, CNRS)
- Alexis Joly (Inria, LIRMM, Univ Montpellier CNRS)
- **Tanguy Lefort** (IMAG, Inria, LIRMM, Univ Montpellier, CNRS)

Identify ambiguous tasks combining crowdsourced labels
by
weighting Areas Under the Margin

`https://arxiv.org/abs/2209.15380`

Mainly joint work with:

Camille Garcin (Univ. Montpellier, IMAG)
Maximilien Servajean (Univ. Paul-Valéry-Montpellier, LIRMM, Univ. Montpellier)
Alexis Joly (Inria, LIRMM, Univ. Montpellier)

and: Pl@ntNet

Pierre Bonnet (CIRAD, AMAP)
Antoine Affouard, **J-C. Lombardo**, **Titouan Lorieul**, **Mathias Chouet** (Inria, LIRMM, Univ. Montpellier)

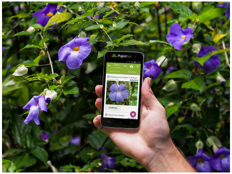
- C. Garcin et al. (2021). "Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution". *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*

Current main research topic

ML for citizen science / Pl@ntNet

Pl@ntNet

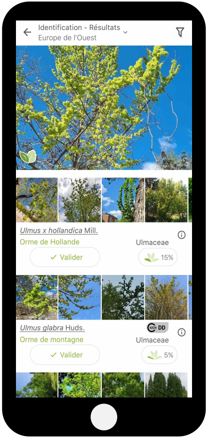
A **citizen science** platform using machine learning to help people identify plants with their mobile phones



Available on the App Store

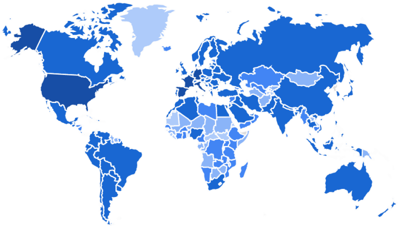
GET IT ON Google Play

Website: `https://plantnet.org/`



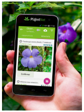
Pl@ntNet

- Start in 2011, now **25M users**
- **200+** countries
- Up to **2M** image uploaded/day
- 45 000 species
- **750M** total images
- **10 M** labeled / validated



Personal Usage


Nature, walks


Gardening


Phytotherapy

Professional Usage


Agro-ecology


Natural Areas Management


Education, animation


Tourism


Trade

Key concept of Pl@ntNet
Cooperative Learning

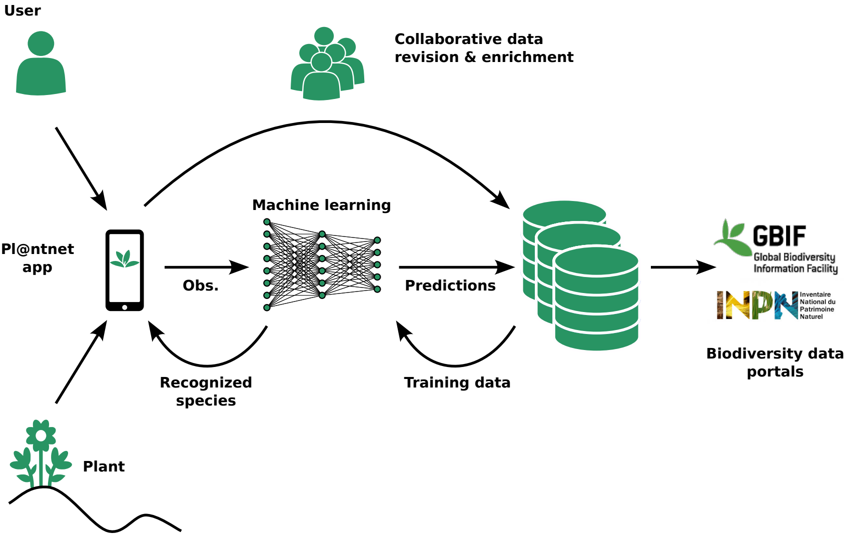


Table of Contents

A need for new benchmarks

Popular datasets limitations:

- structure of label often too simplistic (CIFAR-10, CIFAR-100)
- might be too clean (tasks easy to discriminate)
- might be too well-balanced (same number of images per class)

Motivation:
release a large-scale dataset **sharing similar features** as the Pl@ntNet dataset to foster research in plant identification $\Longrightarrow$ Pl@ntNet-300K[1]

[1] C. Garcin et al. (2021). "Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution". *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Asymetry of errors in Pl@ntNet

Asymetry of errors in Pl@ntNet

Intra-class variability: same label/species but very diverse images

Guizotia abyssinica | *Diascia rigescens* | *Lapageria rosea* | *Casuarina cunninghamiana* | *Freesia alba*

Based on pictures only, plant species are challenging to discriminate!

Asymetry of errors in Pl@ntNet
Inter-class ambiguity: different species but similar images

Cirsium *rivulare*

Cirsium *tuberosum*

Chaerophyllum *aromaticum*

Chaerophyllum *temulum*

Conostomium *kenyense*

Conostomium *quadrangulare*

Adenostyles *leucophylla*

Adenostyles *alliariae*

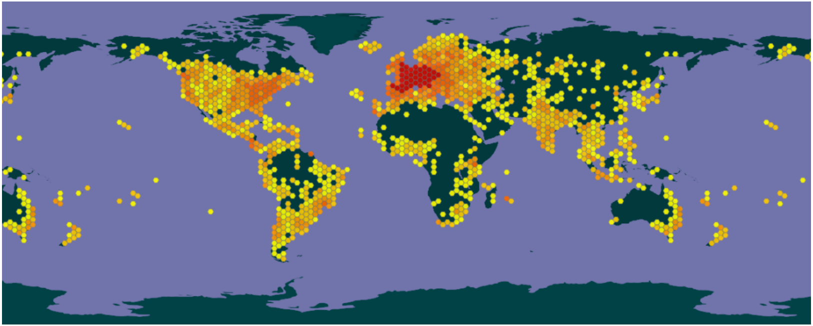
Sedum *montanum*

Sedum *rupestre*

Some species are visually similar (especially within genus)

Sampling bias

Spatial density of images collected by Pl@ntNet :



Sampling bias
Usefulness for humans

Top-5 most observed plant species in Pl@ntNet:

(a) *Prunus domestica*

(b) *Rosa chinensis*

(c) *Capsicum annuum*

(d) *Kalanchoe blossfeldiana*

(e) *Cucumis sativus*

8 548 observations

Centaurea jacea

VS.

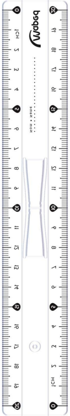
6 observations

Cenchrus agrimonioides

Sampling bias
Size

7 800 observations

Magnolia grandiflora



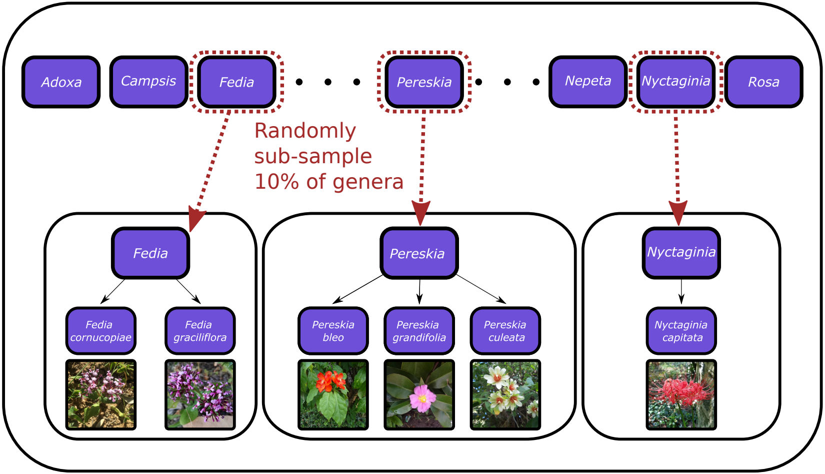
302 observations

Moehringia trinervia

Table of Contents

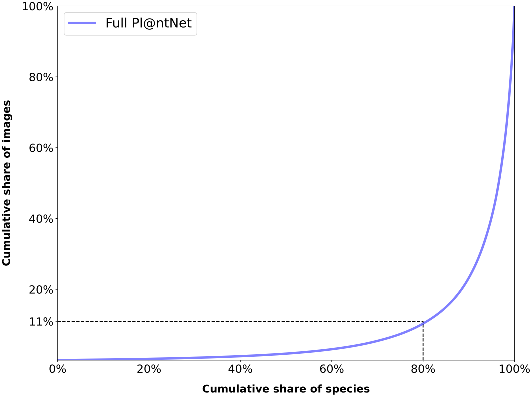
Construction of Pl@ntNet-300K

Subsampling genera preserve dataset characteristics



Sample at genus level to preserve intra-genus ambiguity (use hierarchical structure)

Long tailed distribution
Preserved with Subsampling of genera

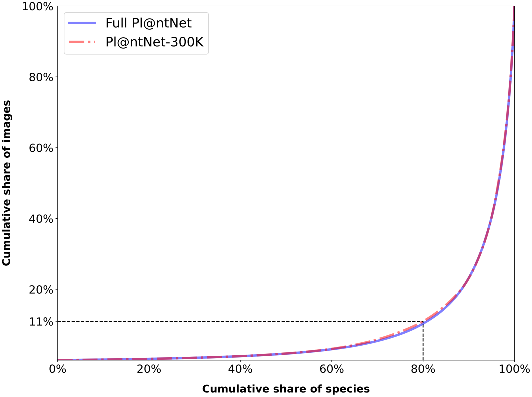


80% of species account for only 11% of images

$\Longleftrightarrow$

20% of species account for 89% of images

<u>Reminder</u>: total = 45 000 plant species (out of 300 000)



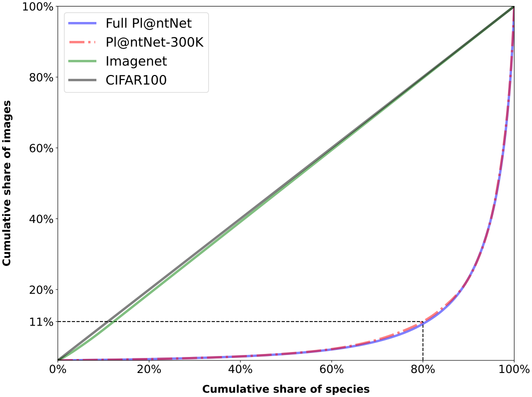
80% of species account for only 11% of images

$\Longleftrightarrow$

20% of species account for 89% of images

Reminder: total = 45 000 plant species (out of 300 000)

Long tailed distribution
Preserved with Subsampling of genera



80% of species account for only 11% of images

$\Longleftrightarrow$

20% of species account for 89% of images

<u>Reminder</u>: total = 45 000 plant species (out of 300 000)

Details on Pl@ntNet-300K

Size and links

- $306\,146$ color images
- 32 GB
- Labels: $K = 1\,081$ species
- $2\,079\,003$ volunteers "workers"

Zenodo, 1 click download

`https://zenodo.org/record/5645731`

Code to train models:

`https://github.com/plantnet/PlantNet-300K`

From workers quality to label quality

Image labeling difficulty could have a huge impact on learning:

- **Removing** very difficult tasks could be useful
  - for dataset **inspection/visualization**
  - to **clean** a dataset
  - for **training performance**[2]

  Hint: usually, such tasks are associated with mislabeling

- Next step:
  We have seen how to assert how good is a worker, but how can we assert the labeling difficulty of an image?

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

Remember: in data we trust?

(3) A. Krizhevsky and G. Hinton (2009). *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto.

(4) (N.d.). `https://github.com/googlecreativelab/quickdraw-dataset`.

(5) Y. LeCun et al. (1998). “Gradient-based learning applied to document recognition”. *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Remember: in data we trust?

. . . but labeling errors are common

CIFAR10[3]

$y^\star =$ **cat**

Quickdraw[4]

$y^\star =$ **T-shirt**

MNIST[5]

$y^\star =$ **6**

---

[3] A. Krizhevsky and G. Hinton (2009). *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto.

[4] (N.d.). `https://github.com/googlecreativelab/quickdraw-dataset`.

[5] Y. LeCun et al. (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Assuming a single hard label (standard supervised settings):

- Classify data points quality with a curated set of probes[6]
- Confident learning[7]: estimate joint distribution between noisy (given) and true labels (unknown)
- Self learning[8]: train a model + extract features and similarity metric on a subset + retrain with modified weighted loss
- Representative Sampling (`CleanNet`[9]): trapping set + encoders + task similarity with constraints on loss
- Our focus here: study the learning dynamic,
  - AUM[10] (Area Under the Margin): study margin during training

---

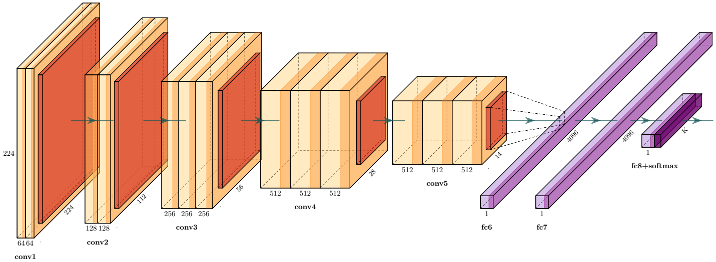
(6) S. A. Siddiqui et al. (2022). *Metadata Archaeology: Unearthing Data Subsets by Leveraging Training Dynamics*.

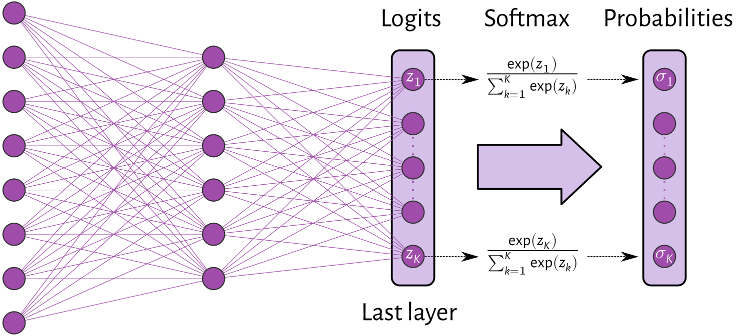
(7) C. Northcutt, L. Jiang, and I. Chuang (2021). "Confident learning: Estimating uncertainty in dataset labels". *J. Artif. Intell. Res.* 70, pp. 1373–1411.

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

(9) K.-H. Lee et al. (2018). "Cleannet: Transfer learning for scalable image classifier training with label noise". *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5447–5456.

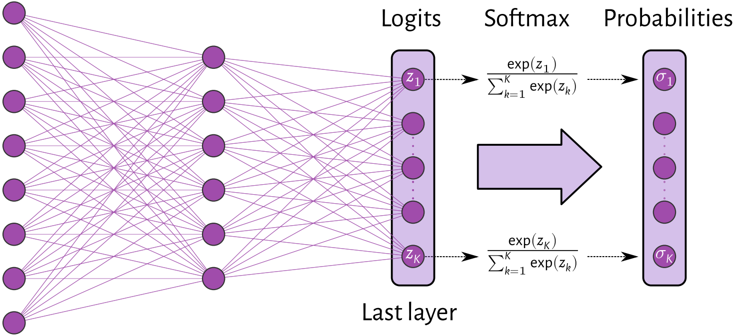
(10) G. Pleiss et al. (2020). "Identifying mislabeled data using the area under the margin ranking". *NeurIPS*.


224
64 64
conv1
224
128 128
conv2
112
256 256 256
conv3
56
512 512 512
conv4
28
512 512 512
conv5
14
4096
1
fc6
4096
1
fc7
K
1
fc8+softmax



Logits
Softmax
Probabilities
z_1
z_K
$\frac{\exp(z_1)}{\sum_{k=1}^{K}\exp(z_k)}$
$\frac{\exp(z_K)}{\sum_{k=1}^{K}\exp(z_k)}$
σ_1
σ_K
Last layer

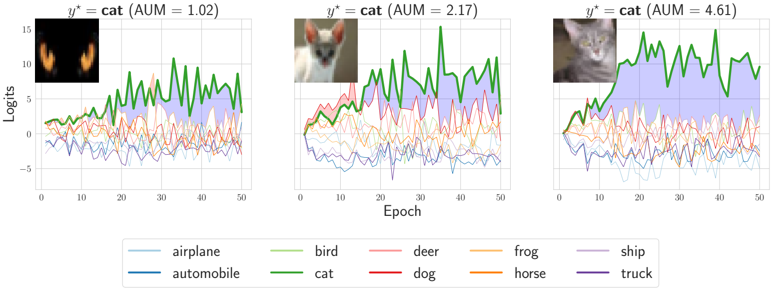
Deep learning
Notation mostly



- From an image, get a score vector $z = (z_1, \ldots, z_K)^\top \in \mathbb{R}^K$
- z_k : **score** (logit) for class k
- σ_k : **probability** (softmax) for class k
- Train for T epochs (say with SGD)

Area under the margins[11]

A step back with one label per task



For each image

- its difficulty is reflected by how quickly the network can learn to discriminate its class
- average the difference between the "true" logit value and the one associated with the most likely one along epochs

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

Settings:

- $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times [K]$ (images, labels) pairs
- Classifier: at epoch $t \in [T]$, $z^{(t)}(x_i) \in \mathbb{R}^K$ a vector of **scores** (logits)

Average = Stability

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

Score of assigned label

Other maximum score

Dissecting the AUM

Building to the crowdsourced extension

Settings:

- $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times [K]$ (images, labels) pairs
- Classifier: at epoch $t \in [T]$, $z^{(t)}(x_i) \in \mathbb{R}^K$ a vector of **scores** (logits)

Average = Stability

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

Score of assigned label

Other maximum score

Challenging for crowdsourcing:

- No single y_i, multiple $y_i^{(j)}$: one for each worker w_j answering task x_i

Dissecting the AUM
On the way to a crowdsourced extension

Settings:

- $(x_i, y_i^{(j)})_{i\in[n_{\texttt{task}}], j\in[n_{\texttt{worker}}]}$: (task,labels) crowdsourced pairs
- Recall: $\mathcal{A}(x_i) := \{j \in [n_{\text{worker}}] : \text{ worker } j \text{ answered task } i\}$

Averaging workers AUM

$$\widetilde{\mathrm{AUM}}(x_i) = \frac{1}{|\mathcal{A}(x_i)|}\sum_{j\in\mathcal{A}(x_i)} \frac{1}{T}\sum_{t=1}^{T} \overbrace{\left[z^{(t)}_{y_i^{(j)}}(x_i) - \max_{\ell\neq y_i^{(j)}} z^{(t)}_{\ell}(x_i) \right]}^{\text{Margin between scores: content of Hinge loss}}$$

Score of assigned label by worker w_j

Other maximum score

- Multiple answers $\implies$ average each AUM (independently)

Dissecting the AUM

On the way to a crowdsourced extension

Settings:

- $(x_i, y_i^{(j)})_{i\in[n_{\mathtt{task}}], j\in[n_{\mathtt{worker}}]}$: (task,labels) crowdsourced pairs
- Recall: $\mathcal{A}(x_i) := \{j \in [n_{\mathrm{worker}}] : \text{ worker } j \text{ answered task } i\}$

$$\widetilde{\mathrm{AUM}}(x_i) = \overbrace{\frac{1}{|\mathcal{A}(x_i)|}\sum_{j\in\mathcal{A}(x_i)}}^{\text{Averaging workers AUM}} \frac{1}{T}\sum_{t=1}^{T} \overbrace{\left[\underbrace{z^{(t)}_{y_i^{(j)}}(x_i)}_{\text{Score of assigned label by worker } w_j} - \underbrace{\max_{\ell\neq y_i^{(j)}} z^{(t)}_{\ell}(x_i)}_{\text{Other maximum score}}\right]}^{\text{Margin between scores: content of Hinge loss}}$$

- Multiple answers $\Longrightarrow$ average each AUM (independently)

Reliability issue:

- Not all workers are equally gifted $\Longrightarrow$ **weight** AUM per worker

- Introduce weights $s^{(j)}(x_i)$ as the trust score in worker j for task x_i

$$\widetilde{\mathrm{AUM}}(x_i) = \overbrace{\frac{1}{S}\sum_{j\in\mathcal{A}(x_i)}}^{\text{Weighted average of AUM}} \overbrace{s^{(j)}(x_i)}^{\text{Trust score of } w_j \text{ for } x_i} \frac{1}{T}\sum_{t=1}^{T} \overbrace{\left[\underbrace{z^{(t)}_{y_i^{(j)}}(x_i)}_{\text{Score of assigned label by worker } w_j} - \underbrace{\max_{\ell\neq y_i^{(j)}} z^{(t)}_{\ell}(x_i)}_{\text{Other maximum score}} \right]}^{\text{Margin between scores: content of Hinge loss}}$$

with $S = \sum_{j\in\mathcal{A}(x_i)} s^{(j)}(x_i)$ (normalization factor)

Modifying the margin:

- Better margin (in theory, for top-k classification[12])

(12) M. Lapin, M. Hein, and B. Schiele (2016). "Loss functions for top-k error: Analysis and insights". *CVPR*, pp. 1468–1477; F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". *ICML*, pp. 10727–10735.

(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". *J. Appl. Stat.* 45.15, pp. 2800–2818.

Modifying the margin:

- Better margin (in theory, for top-k classification[12])

Change logit to softmax scores:

- avoid scale effects for scores and huge variation with multiple labels[13]

(12) M. Lapin, M. Hein, and B. Schiele (2016). “Loss functions for top-k error: Analysis and insights”. *CVPR*, pp. 1468–1477; F. Yang and S. Koyejo (2020). “On the consistency of top-k surrogate losses”. *ICML*, pp. 10727–10735.

(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”. *J. Appl. Stat.* 45.15, pp. 2800–2818.

Modifying the margin:

- Better margin (in theory, for top-k classification[12])

Change logit to softmax scores:

- avoid scale effects for scores and huge variation with multiple labels[13]

Notation:

- $\sigma(x_i) = \mathrm{softmax}(z(x_i))$ (in simplex)
- Softmax ordered: $\sigma_{[1]}(x_i) \geq \cdots \geq \sigma_{[K]}(x_i) > 0$

[12] M. Lapin, M. Hein, and B. Schiele (2016). "Loss functions for top-k error: Analysis and insights". *CVPR*, pp. 1468–1477; F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". *ICML*, pp. 10727–10735.

[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". *J. Appl. Stat.* 45.15, pp. 2800–2818.

Modifying the margin:

- Better margin (in theory, for top-k classification[12])

Change logit to softmax scores:

- avoid scale effects for scores and huge variation with multiple labels[13]

Notation:

- $\sigma(x_i) = \mathrm{softmax}(z(x_i))$ (in simplex)
- Softmax ordered: $\sigma_{[1]}(x_i) \geq \cdots \geq \sigma_{[K]}(x_i) > 0$

Weighted average of AUM

Trust score of w_j for x_i

$$\mathrm{WAUM}(x_i) := \frac{1}{S}\sum_{j\in\mathcal{A}(x_i)} s^{(j)}(x_i) \frac{1}{T}\sum_{t=1}^{T} \overbrace{\left[\sigma^{(t)}_{y_i^{(j)}}(x_i) - \sigma^{(t)}_{[2]}(x_i)\right]}^{\text{Margin between scores: content of Hinge loss}}$$

Probability of assigned label by worker w_j

2nd max. probability

---

[12] M. Lapin, M. Hein, and B. Schiele (2016). "Loss functions for top-k error: Analysis and insights". *CVPR*, pp. 1468–1477; F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". *ICML*, pp. 10727–10735.

[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". *J. Appl. Stat.* 45.15, pp. 2800–2818.

Choosing $s^{(j)}(x_i)$:

- if $s^{(j)}(x_i) = 1$ all workers have the same weight
- if $s^{(j)}(x_i) = c_j$ the weights only depend on the worker
- DS[14] algorithm, etc.

[14] A. Dawid and A. Skene (1979). “Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm”. *J. R. Stat. Soc. Ser. C. Appl. Stat.* 28.1, pp. 20–28.

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

Choosing $s^{(j)}(x_i)$:

- if $s^{(j)}(x_i) = 1$ all workers have the same weight
- if $s^{(j)}(x_i) = c_j$ the weights only depend on the worker
- DS[14] algorithm, etc.

Our chosen worker/task score:

- Score of the form: "worker term $\times$ task term" (similar to GLAD[15])
- Estimate ability thanks to confusion matrices $\hat{\pi}^{(j)}$ (with DS)
- Use softmax scores to measure label confidence

[14] A. Dawid and A. Skene (1979). “Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm”. *J. R. Stat. Soc. Ser. C. Appl. Stat.* 28.1, pp. 20–28.

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

Choosing $s^{(j)}(x_i)$:

- if $s^{(j)}(x_i) = 1$ all workers have the same weight
- if $s^{(j)}(x_i) = c_j$ the weights only depend on the worker
- DS[14] algorithm, etc.

Our chosen worker/task score:

- Score of the form: "worker term $\times$ task term" (similar to GLAD[15])
- Estimate ability thanks to confusion matrices $\hat{\pi}^{(j)}$ (with DS)
- Use softmax scores to measure label confidence

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

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

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

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

Computing the WAUM

The pipeline summarized

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

- Estimate confusion matrices $\hat{\pi}^{(j)} \in \mathbb{R}^{K\times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute $\mathrm{AUM}(x_i, y_i^{(j)}) = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T}\left[\sigma^{(t)}_{y_i^{(j)}}(x_i) - \sigma^{(t)}_{[2]}(x_i)\right]$

- Estimate confusion matrices $\hat{\pi}^{(j)} \in \mathbb{R}^{K\times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute $\mathrm{AUM}(x_i, y_i^{(j)}) = \frac{1}{T}\sum_{t=1}^{T}\left[\sigma^{(t)}_{y_i^{(j)}}(x_i) - \sigma^{(t)}_{[2]}(x_i)\right]$
- Compute trust scores $s^{(j)}(x_i)$

- Estimate confusion matrices $\hat{\pi}^{(j)} \in \mathbb{R}^{K\times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute $\mathrm{AUM}(x_i, y_i^{(j)}) = \frac{1}{T}\sum_{t=1}^{T}\left[\sigma^{(t)}_{y_i^{(j)}}(x_i) - \sigma^{(t)}_{[2]}(x_i)\right]$
- Compute trust scores $s^{(j)}(x_i)$
- For each task compute $\mathrm{WAUM}(x_i) = \dfrac{\sum_{j\in\mathcal{A}(x_i)} s^{(j)}(x_i)\mathrm{AUM}(x_i, y_i^{(j)})}{\sum_{j'\in\mathcal{A}(x_i)} s^{(j')}(x_i)}$

Usage (for learning):

- Estimate confusion matrices $\hat{\pi}^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute $\mathrm{AUM}(x_i, y_i^{(j)}) = \frac{1}{T} \sum_{t=1}^{T} \left[\sigma^{(t)}_{y_i^{(j)}}(x_i) - \sigma^{(t)}_{[2]}(x_i) \right]$
- Compute trust scores $s^{(j)}(x_i)$
- For each task compute $\mathrm{WAUM}(x_i) = \dfrac{\sum_{j \in \mathcal{A}(x_i)} s^{(j)}(x_i) \mathrm{AUM}(x_i, y_i^{(j)})}{\sum_{j' \in \mathcal{A}(x_i)} s^{(j')}(x_i)}$

Usage (for learning):

- **Prune** x_i's with $\mathrm{WAUM}(x_i)$ below quantile q_α (say $\alpha = 0.1$)

- Estimate confusion matrices $\hat{\pi}^{(j)} \in \mathbb{R}^{K\times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute $\mathrm{AUM}(x_i, y_i^{(j)}) = \frac{1}{T}\sum_{t=1}^{T}\left[\sigma^{(t)}_{y_i^{(j)}}(x_i) - \sigma^{(t)}_{[2]}(x_i)\right]$
- Compute trust scores $s^{(j)}(x_i)$
- For each task compute $\mathrm{WAUM}(x_i) = \dfrac{\sum_{j\in\mathcal{A}(x_i)} s^{(j)}(x_i)\mathrm{AUM}(x_i, y_i^{(j)})}{\sum_{j'\in\mathcal{A}(x_i)} s^{(j')}(x_i)}$

Usage (for learning):

- **Prune** x_i's with $\mathrm{WAUM}(x_i)$ below quantile q_α (say $\alpha = 0.1$)
- Estimate **confusion matrices** $\hat{\pi}^{(j)}$ on pruned training dataset

- Estimate confusion matrices $\hat{\pi}^{(j)} \in \mathbb{R}^{K\times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute $\mathrm{AUM}(x_i, y_i^{(j)}) = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T}\left[\sigma^{(t)}_{y_i^{(j)}}(x_i) - \sigma^{(t)}_{[2]}(x_i)\right]$
- Compute trust scores $s^{(j)}(x_i)$
- For each task compute $\mathrm{WAUM}(x_i) = \dfrac{\displaystyle\sum_{j\in\mathcal{A}(x_i)} s^{(j)}(x_i)\mathrm{AUM}(x_i, y_i^{(j)})}{\displaystyle\sum_{j'\in\mathcal{A}(x_i)} s^{(j')}(x_i)}$

Usage (for learning):

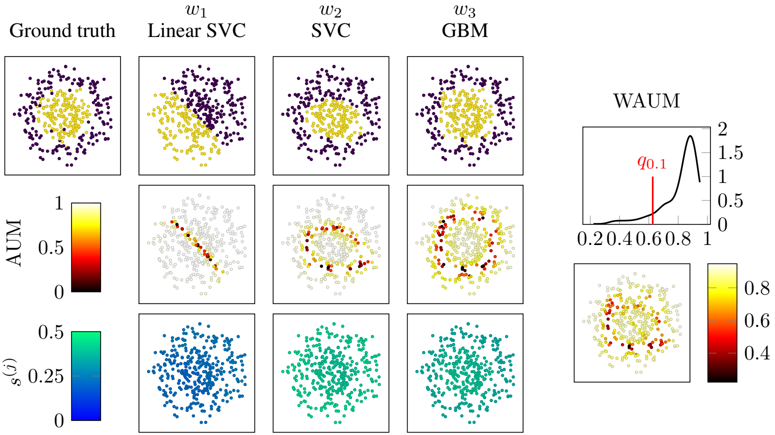
- **Prune** x_i's with $\mathrm{WAUM}(x_i)$ below quantile q_α (say $\alpha = 0.1$)
- Estimate **confusion matrices** $\hat{\pi}^{(j)}$ on pruned training dataset
- Get **soft labels**: normalize $\hat{y}_i = \left(\displaystyle\sum_{j\in\mathcal{A}(x_i)} \hat{\pi}^{(j)}_{k,k}\mathbb{1}_{\{y_i^{(j)}=k\}}\right)_{k\in[K]} \in \mathbb{R}^K$

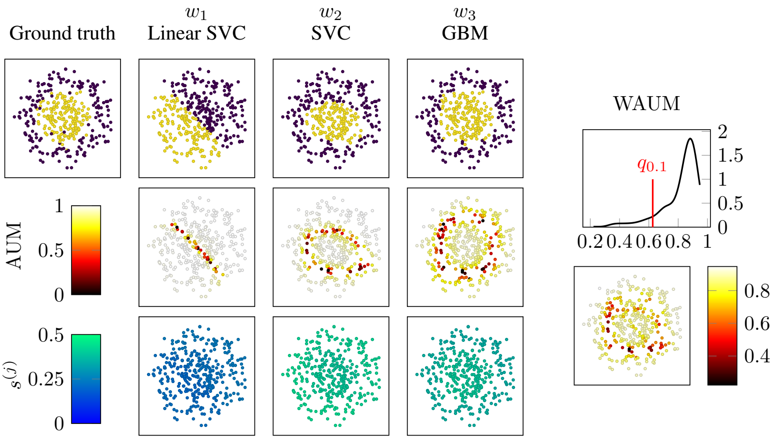
- Estimate confusion matrices $\hat{\pi}^{(j)} \in \mathbb{R}^{K\times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute $\mathrm{AUM}(x_i, y_i^{(j)}) = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T}\left[\sigma^{(t)}_{y_i^{(j)}}(x_i) - \sigma^{(t)}_{[2]}(x_i)\right]$
- Compute trust scores $s^{(j)}(x_i)$
- For each task compute $\mathrm{WAUM}(x_i) = \dfrac{\displaystyle\sum_{j\in\mathcal{A}(x_i)} s^{(j)}(x_i)\mathrm{AUM}(x_i, y_i^{(j)})}{\displaystyle\sum_{j'\in\mathcal{A}(x_i)} s^{(j')}(x_i)}$

Usage (for learning):

- **Prune** x_i's with $\mathrm{WAUM}(x_i)$ below quantile q_α (say $\alpha = 0.1$)
- Estimate **confusion matrices** $\hat{\pi}^{(j)}$ on pruned training dataset
- Get **soft labels**: normalize $\hat{y}_i = \left(\displaystyle\sum_{j\in\mathcal{A}(x_i)} \hat{\pi}^{(j)}_{k,k}\mathbb{1}_{\{y_i^{(j)}=k\}}\right)_{k\in[K]} \in \mathbb{R}^K$
- **Train** a classifier on the pruned dataset (with soft labels)

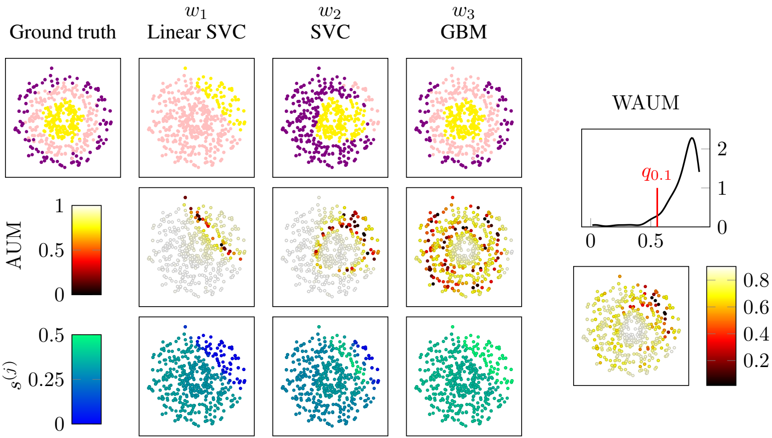
Simulation with circles
Binary setting



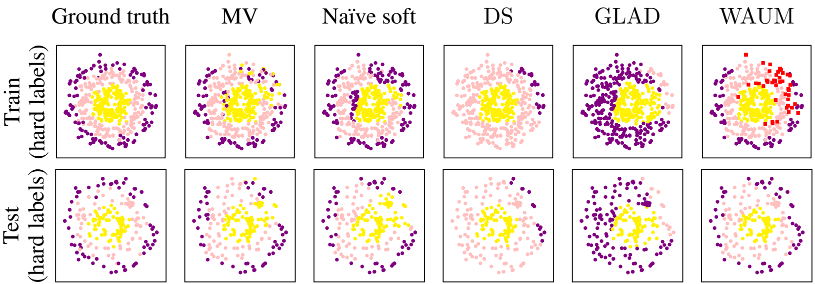


- Workers = simulated classifiers (answering 500 tasks)
- Normalized trust scores
- Neural Network: 3-dense layers' artificial neural network (30, 20, 20)

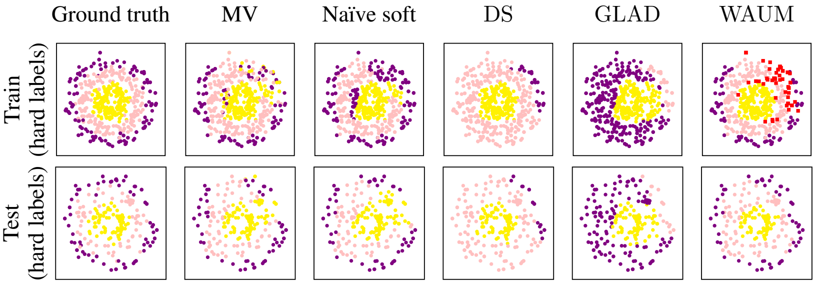
Simulation with circles
Three classes



- 3 classes with 250 tasks per class
- Normalized trust scores
- Neural Network: 3-dense layers' artificial neural network (30, 20, 20)



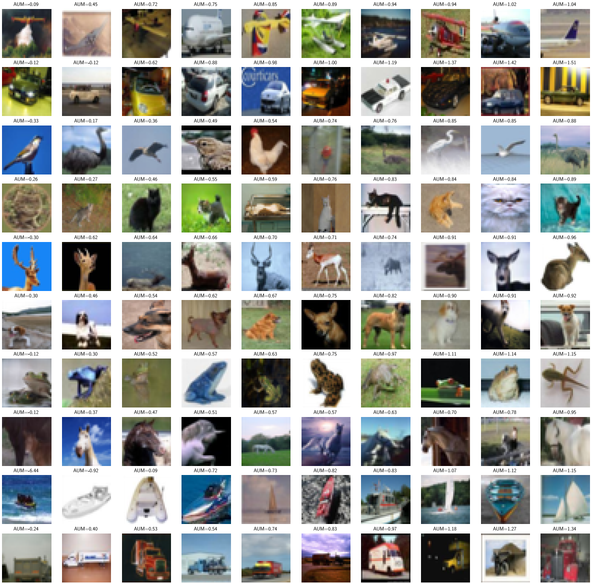
Ground truth
MV
Naïve soft
DS
GLAD
WAUM
Train (hard labels)
Test (hard labels)



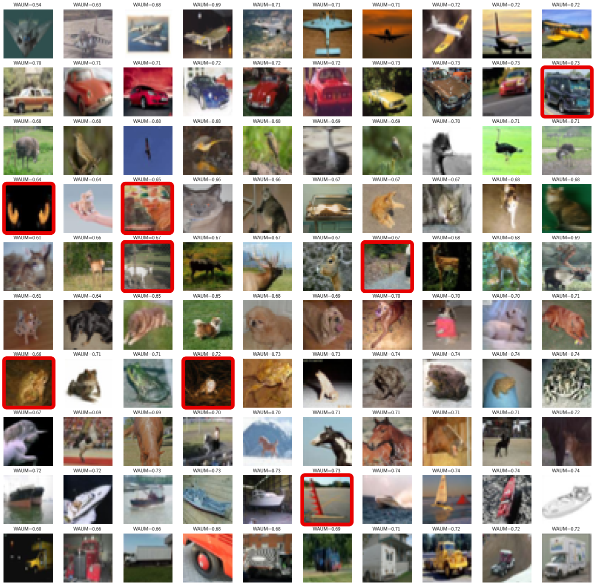
	MV	Naive soft	DS	GLAD	WAUM($\alpha = 0.1$)
Test accuracy	0.727	0.697	0.753	0.578	0.806

Results on CIFAR10H

Improved mislabeled detections: worst AUM/WAUM



Results on CIFAR10H
Improved mislabeled detections: worst AUM/WAUM



Bokeh application of the AUM/WAUM to the CIFAR10H dataset.
(see horse, cat and deer for instance)

Generalization performance and calibration error (with a Resnet-18):

Aggregation method	Test accuracy (on CIFAR10-train)	ECE (expected calibration error)
MV	69.533 ± 0.84	0.175 ± 0.01
Naive soft	72.149 ± 2.74	0.132 ± 0.03
DS (vanilla)	70.268 ± 0.93	0.173 ± 0.01
DS (spam identification)	70.053 ± 0.81	0.174 ± 0.01
GLAD	66.569 ± 8.48	0.173 ± 0.01
WAUM	**72.747** ± 1.93	**0.124** ± 0.01

Remark: ECE[16] Expected Calibration Error, the smaller the better

[16] C. Guo et al. (2017). "On calibration of modern neural networks". *ICML*, p. 1321.

"Can I use the WAUM in my framework?"

Ablation study (LabelMe)

Aggregation method	Test Accuracy	ECE
WDS	85.6	0.162
WAUM + WDS	**87.1**	**0.129**

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

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

"Can I use the WAUM in my framework?"

Ablation study (LabelMe)

Aggregation method	Test Accuracy	ECE
WDS	85.6	0.162
WAUM + WDS	**87.1**	**0.129**
GLAD[(17)]	87.1	**0.119**
WAUM + GLAD	**87.6**	0.123

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

[(18)] Z. Chu, J. Ma, and H. Wang (2021). “Learning from Crowds by Modeling Common Confusions.”. *AAAI*, pp. 5832–5840.

Aggregation method	Test Accuracy	ECE
WDS	85.6	0.162
WAUM + WDS	**87.1**	**0.129**
GLAD(17)	87.1	**0.119**
WAUM + GLAD	**87.6**	0.123
CoNAL(18) (lambda=0)	88.1	0.119
WAUM + CoNAL(lambda=0)	**89.2**	**0.108**

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

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

"Can I use the WAUM in my framework?"

Ablation study (LabelMe)

Aggregation method	Test Accuracy	ECE
WDS	85.6	0.162
WAUM + WDS	**87.1**	**0.129**
GLAD[17]	87.1	**0.119**
WAUM + GLAD	**87.6**	0.123
CoNAL[18] (lambda=0)	88.1	0.119
WAUM + CoNAL(lambda=0)	**89.2**	**0.108**
CoNAL(lambda=1e-4)	86.2	0.135
WAUM + CoNAL(lambda=1e-4)	**90.0**	**0.099**

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

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

"Can I use the WAUM in my framework?"
Ablation study (Music dataset)

Aggregation method	Test Accuracy	ECE
WDS	60.2	0.348
WAUM + WDS	**63.1**	**0.377**
GLAD[17]	61.5	0.361
WAUM + GLAD	**61.5**	**0.355**
CoNAL[18] (lambda=0)	64.2	0.340
WAUM + CoNAL(lambda=0)	**64.5**	**0.265**
CoNAL(lambda=1e-4)	64.2	0.361
WAUM + CoNAL(lambda=1e-4)	**64.4**	**0.274**

[17] J. Whitehill et al. (2009). "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise". *NeurIPS*. vol. 22.
[18] Z. Chu, J. Ma, and H. Wang (2021). "Learning from Crowds by Modeling Common Confusions.". *AAAI*, pp. 5832–5840.

Take home message(s)

- Citizen science challenges: many and varied (need more attention)

Take home message(s)

- Citizen science challenges: many and varied (need more attention)
- Crowdsourcing / Label uncertainty: helpful for **data curation**

Take home message(s)

- Citizen science challenges: many and varied (need more attention)
- Crowdsourcing / Label uncertainty: helpful for **data curation**
- Improved **data quality** $\Rightarrow$ **improved learning** performance

Conclusion

Take home message(s)

- Citizen science challenges: many and varied (need more attention)
- Crowdsourcing / Label uncertainty: helpful for **data curation**
- Improved **data quality** $\Rightarrow$ **improved learning** performance
- Toolbox: **`https://peerannot.github.io/`**
- Some benchmarks: **`https://benchopt.github.io/`**

Conclusion

Take home message(s)

- Citizen science challenges: many and varied (need more attention)
- Crowdsourcing / Label uncertainty: helpful for **data curation**
- Improved **data quality** $\Rightarrow$ **improved learning** performance
- Toolbox: **`https://peerannot.github.io/`**
- Some benchmarks: **`https://benchopt.github.io/`**

Future work

- Release a Pl@ntnet crowdsourced dataset (**2M workers**)
- Leverage gamification for more quality labels **`theplantgame.com`**

Contact:

Joseph Salmon

joseph.salmon@umontpellier.fr

https://josephsalmon.eu

Github: @josephsalmon

Mastodon: @josephsalmon@sigmoid.social

- (N.d.). `https://github.com/googlecreativelab/quickdraw-dataset`.
- Chu, Z., J. Ma, and H. Wang (2021). "Learning from Crowds by Modeling Common Confusions.". *AAAI*, pp. 5832–5840.
- Dawid, A. and A. Skene (1979). "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm". *J. R. Stat. Soc. Ser. C. Appl. Stat.* 28.1, pp. 20–28.
- Garcin, C. et al. (2021). "Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution". *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.
- Guo, C. et al. (2017). "On calibration of modern neural networks". *ICML*, p. 1321.
- Han, J., P. Luo, and X. Wang (2019). "Deep self-learning from noisy labels". *ICCV*, pp. 5138–5147.
- Ju, C., A. Bibaut, and M. van der Laan (2018). "The relative performance of ensemble methods with deep convolutional neural networks for image classification". *J. Appl. Stat.* 45.15, pp. 2800–2818.

References II

Krizhevsky, A. and G. Hinton (2009). *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto.

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

LeCun, Y. et al. (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, K.-H. et al. (2018). "Cleannet: Transfer learning for scalable image classifier training with label noise". *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5447–5456.

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

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

Siddiqui, S. A. et al. (2022). *Metadata Archaeology: Unearthing Data Subsets by Leveraging Training Dynamics*.

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

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

An alternative: GLAD[19]

Generative model of Labels, Abilities, and Difficulties

- DS assumption: errors only come from workers (no task modeling)

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

- DS assumption: errors only come from workers (no task modeling)

GLAD: incorporating task difficulty

Model labeling errors as a function of worker ability and task difficulty:

- worker j has an ability $\alpha_j \in \mathbb{R}$
- task i has a difficulty $\beta_i \in \mathbb{R}_+^\star$

$$\mathbb{P}\left(y_i^{(j)} = y_i^\star | \alpha_j, \beta_i\right) = \frac{1}{1 + e^{-\alpha_j \beta_i}}$$

Note: assume uniform errors on other labels

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

ECE[20]

Expected Calibration Error

For $x \in \mathcal{X}_{\texttt{train}} = \{x_1, \dots, x_{n_{\texttt{task}}}\}$, let $\sigma(x) \in \Delta_{K-1}$ (softmax output)

Split $[0,1]$ into $M(=15)$ bins $I_1, \dots, I_M$ of size $\frac{1}{M}$: $I_m = (\frac{m-1}{M}, \frac{m}{M}]$, for $m \in [M]$

Denote $B_m = \{x \in \mathcal{X}_{\texttt{train}} : \sigma_{[1]}(x) \in I_m\}$ the tasks whose predicted probabilities are in the m-th bin

Define **accuracy** and **confidence**:

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}_{\{\sigma_{[1]}(x_i) = y_i\}} \quad \text{and} \quad \mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \sigma_{[1]}(x_i) \quad .$$

Then, the Expected Calibration Error (ECE) reads:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n_{\texttt{task}}} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right| \quad .$$

Perfect calibrattion : $\mathrm{ECE} = 0$ (accuracy = confidence for each subset B_m)

[20] C. Guo et al. (2017). "On calibration of modern neural networks". *ICML*, p. 1321.