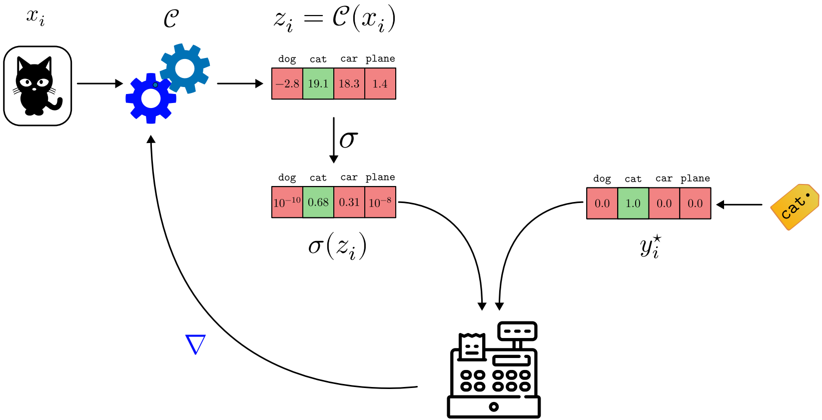
Data collection from a crowd: where is the noise coming from?

Tanguy Lefort
IMAG, Univ Montpellier, CNRS
Inria, LIRMM,

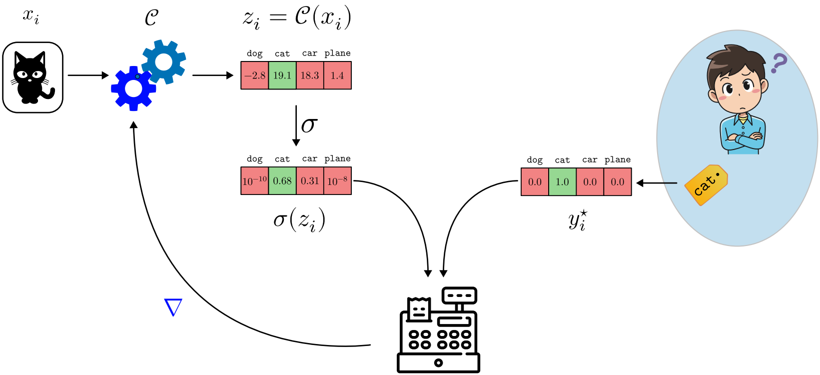
Ongoing thesis with...

- Benjamin Charlier (IMAG, Univ Montpellier, CNRS)
- Alexis Joly (INRIA, LIRMM, Univ Montpellier CNRS)
- Joseph Salmon (IMAG, Univ Montpellier, CNRS, Institut Universitaire de France (IUF))



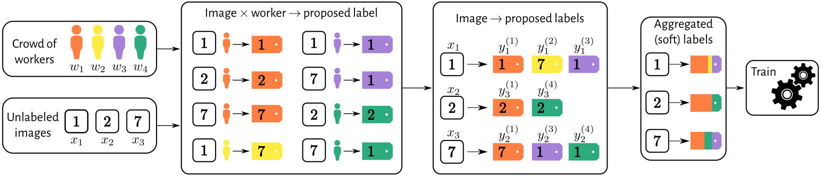
$$\mathcal{L}(\sigma(z_i), y_i^\star) = \mathrm{CE}(\sigma(z_i), y_i^\star)$$
$$= -\log\left(\sigma(z_i)_{y_i^\star}\right)$$

My big question
Where is the data coming from?

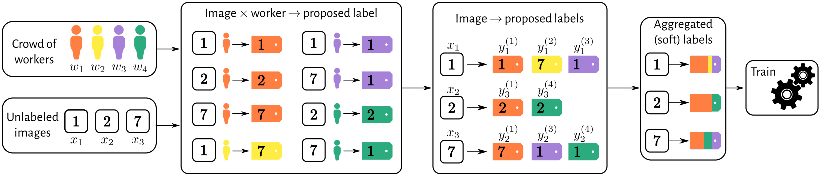


$$\begin{aligned}\mathcal{L}(\sigma(z_i), y_i^\star) &= \mathrm{CE}(\sigma(z_i), y_i^\star)\\ &= -\log\left(\sigma(z_i)_{y_i^\star}\right)\end{aligned}$$

Crowdsourcing



Crowdsourcing



Why use crowdsourcing?

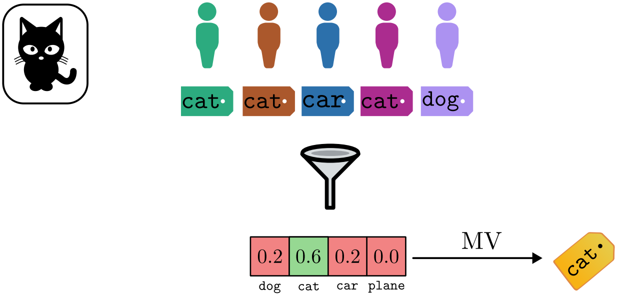
- Faster + lower cost than hiring experts
- Uncertainty obtained is valuable ⟶ data quality

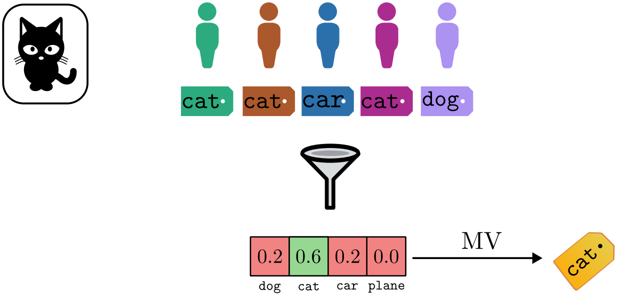
Is crowdsourcing niche? - Who uses it?

Not niche, but in the background!

- Google: Google Rewards app, Google Maps, ...
- Pl@ntnet: Plant species recognition app
- Eyewire: Map brain neurons
- Tournesol: Public interest YouTube video recommendation system
- Twitter/𝕏: Detect harmful tweets, recommendation system
- ChatGPT: Improve responses (human reinforcement learning)
- Waze, Duolingo, EDF, SNCF, TripAdvisor, Spotify, BeMyEyes, ...

Top-3 classical aggregation strategies
Majority voting (MV)

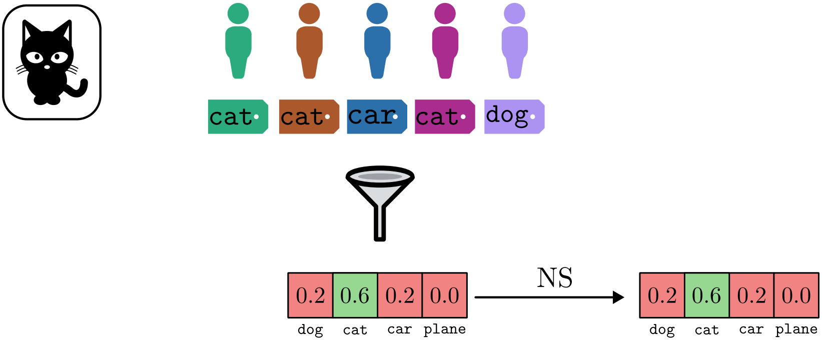




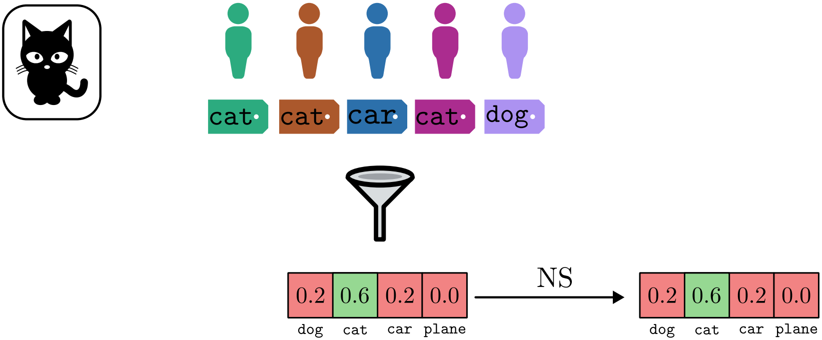
- **Pros:**
  - Easy to understand
  - Fast to run
  - One of the most studied
  - Good performance on easy tasks
- **Cons:**
  - Overly simplistic
  - No information on workers / tasks
  - Sensitive to spammers / adversarial crowds

Top-3 classical aggregation strategies

Naive soft (NS)



Top-3 classical aggregation strategies
Naive soft (NS)

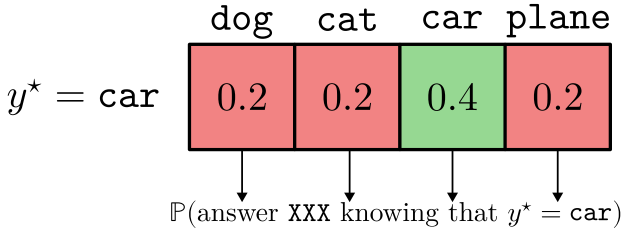


- **Pros:**
  - Easy to understand
  - Fast to run
  - Uncertainty is kept
- **Cons:**
  - No information on workers / tasks
  - Sensitive to spammers / adversarial crowds

Top-3 classical aggregation strategies
Dawid and Skene (DS)

- Knowing the true label $y^\star$ each worker answers differently.
- This answer follows a multinomial distribution.

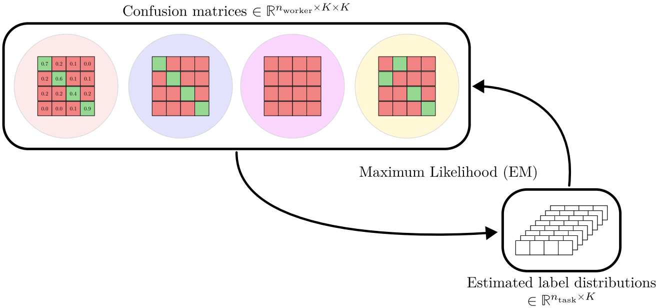
$$y^{(j)} \,|\, y^\star \sim \mathcal{M}(\pi^{(j)}_{y^\star,\bullet})$$



- Probabilistic model $\longrightarrow$ likelihood

$$\prod_{i \in [n_{\text{task}}]} \prod_{k \in [K]} \left[\rho_k \prod_{k \in [n_{\text{worker}}]} \prod_{\ell \in [K]} \left(\pi^{(j)}_{k,\ell}\right)^{\mathbb{1}_{\{y_i^{(j)} = \ell\}}} \right]^{T_{i,k}}$$

- Prevalence: $\rho_k = \mathbb{P}(y_i^\star = k)$, labels: $T_{i,k} = \mathbb{1}(y_i^\star = k)$
- Find parameters maximizing the likelihood



- **Pros:**
  - Easy to understand
  - Model worker abilities
  - Uncertainty is kept
  - Can detect spammers
  - Can use adversarial workers

- **Pros:**
  - Easy to understand
  - Model worker abilities
  - Uncertainty is kept
  - Can detect spammers
  - Can use adversarial workers
- **Cons:**
  - Memory issues: High number of classes K
  - Estimates $n_{\text{worker}} \times K^2$ coefficients (identifiability)

Spammer definition

A spammer answers independently of the true label

$$\forall(k,\ell) \in [K]^2,\ \mathbb{P}(y_i^{(j)} = k \mid y_i^\star = \ell) = \mathbb{P}(y_i^{(j)} = k)$$

- In the DS model, a spammer has a confusion matrix $\pi^{(j)}$ of rank 1.
- Distance to spammer = distance to closest rank one matrix

	dog	cat	car	plane
dog	0.7	0.2	0.1	0.0
cat	0.7	0.2	0.1	0.0
car	0.65	0.2	0.1	0.05
plane	0.75	0.15	0.1	0.0

Simulation
Hammer-Spammer dataset

- Crowd of 20 workers, 4 hammers (always right) + 16 spammers
- 2 classes, 100 tasks to label
- Everybody answers everything

Method	MV	NS	DS	GLAD
Label Recovery	0.84	0.83	1.0	1.0

- We can use adversarial workers here!

Simulation

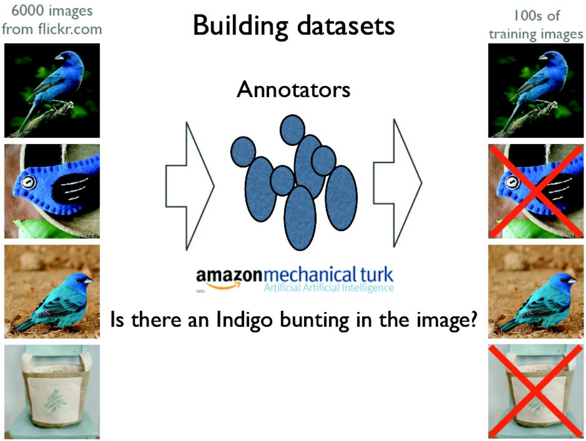
Hammer-Spammer dataset (continued)

- Crowd of 20 workers, 4 hammers (always right) + 16 spammers
- 4 classes, 100 tasks to label
- Random number of labels per task (some tasks more answered)

Method	MV	NS	DS	GLAD
Label Recovery	0.56	0.55	0.84	0.83

- Perfect recovery is no longer possible with more than 2 classes

Bluebirds dataset
A big loss for the community



Slides from http://videolectures.net/nips2010_welinder_mwc/

Method	MV	NS	DS	GLAD
Label Recovery	0.75	0.75	0.89	0.72

CIFAR-10H
History of the dataset

- Images from 80M Tiny Images web-scraped dataset to create CIFAR-10 dataset
- "We paid students to label a subset of the Tiny Images dataset[...]. The labelers were paid a fixed sum per hour spent labeling."

CIFAR-10H

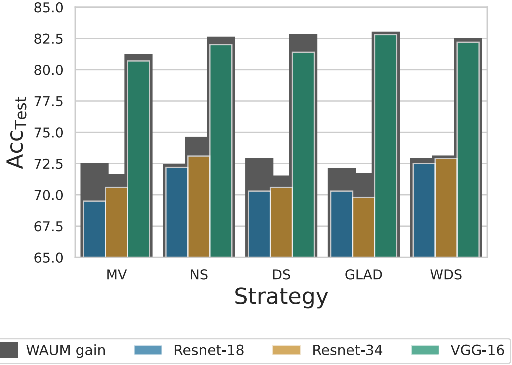
History of the dataset

- Images from 80M Tiny Images web-scraped dataset to create CIFAR-10 dataset
- "We paid students to label a subset of the Tiny Images dataset[...]. The labelers were paid a fixed sum per hour spent labeling."
- "Since each image in the dataset already comes with a noisy label (the search term used to find the image), all we needed the labelers to do was to filter out the mislabeled images."
- "Furthermore, we personally verified every label submitted by the labelers."

CIFAR-10H
History of the dataset

- Images from 80M Tiny Images web-scraped dataset to create CIFAR-10 dataset
- "We paid students to label a subset of the Tiny Images dataset[...]. The labelers were paid a fixed sum per hour spent labeling."
- "Since each image in the dataset already comes with a noisy label (the search term used to find the image), all we needed the labelers to do was to filter out the mislabeled images."
- "Furthermore, we personally verified every label submitted by the labelers."
- Reproduce the crowdsourcing step with CIFAR-10H with 2571 workers on $10,000$ tasks $\longrightarrow$ **511,400** labels collected (workers paid 1$ 50)

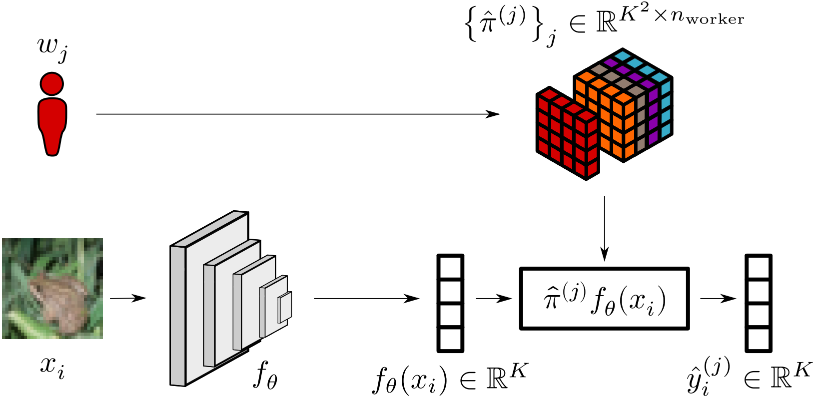
- All aggregation strategies have over 99.2% recovering label accuracy $\longrightarrow$ one of the largest public crowdsourced datasets but **too clean**
- But performance on test tasks **after** training a model may vary!



Non-aggregation-based strategies

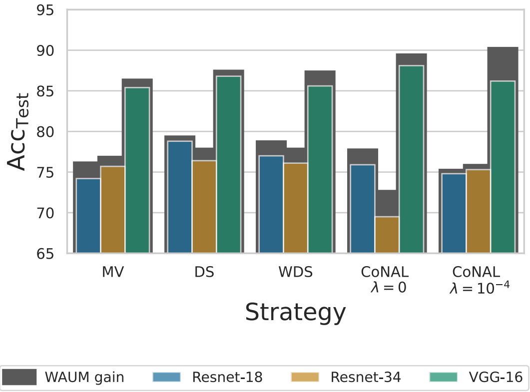
A quick look into the deep learning world

- Not all crowdsourcing strategies rely on aggregating labels
- … but they rely on adapting the DS model most of the time



Non-aggregation-based strategies
Results on LabelMe dataset

- LabelMe dataset: 1000 tasks, 77 workers, 8 (overlapping) classes
- Between 1 and 3 labels per task (very few!)



- PeerAnnot library: `https://peerannot.github.io/`
- API and CLI (in Python or directly in your terminal, or a mix)

```
for strat in [MV, NS, DS, GLAD]:
  ! peerannot aggregate ./my_dataset/ -s ${strat}
```

- 3 modules: `aggregate`, `aggregate-deep`, and `identify`
- Allow to aggregate, train, and explore datasets (reproducibility!)
- Paper online: `https://tanglef.github.io/computo_2023`

Conclusion - What I actually do

My big question

Should we learn from every image scrapped?

- How to detect issues not in workers, but in tasks
- Developed the WAUM statistic (seen in previous figures) that improves models' performance