# Crowdsourcing with Arbitrary Adversaries

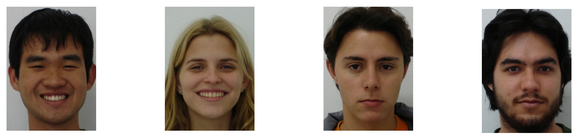
Matthäus Kleindessner, Pranjal Awasthi

RUTGERS

## Crowdsourcing

Principle: want to gather information by asking the *crowd*

**Example:** want to train a neural network that classifies whether a person looks happy or not



*happy* *happy* *not happy* *?*

⇒ need a large amount of labeled training data

On crowdsourcing platforms like Amazon Mechanical Turk™ we can hire workers to provide labels for a large number of training images for relatively little money.

Problem: workers make errors and provide wrong labels (do not know better / are lazy / behave adversarially) ⇒ need to collect redundant labels and aggregate

**Other applications of crowdsourcing:** peer grading, online rating systems

## Setup & problem formulation (binary classification)

- $n$ workers $w_1, \dots, w_n$
- i.i.d. sample of $m$ task-label pairs $((x_i, y_i))_{i=1}^m \sim D^m$
  $D$ is a joint distribution over tasks $x \in \mathcal{X}$ and ground-truth labels $y \in \{-1, +1\}$
- variables $g_{ij} \in \{0, 1\}$, $i \in [m]$, $j \in [n]$: $g_{ij} = 1 \Leftrightarrow w_j$ is presented with $x_i$ and provides an estimate $w_j(x_i) \in \{-1, +1\}$
- $A \in \{-1, 0, +1\}^{m \times n}$ with $A_{ij} = w_j(x_i)$ if $g_{ij} = 1$ and $A_{ij} = 0$ if $g_{ij} = 0$

ASSUMPTION: each worker $w_j$ follows (probabilistic or deterministic) strategy such that $w_j(x_i)$ only depends on $x_i$

DEFINITIONS:

$$\varepsilon_{w_j}(x, y) := \Pr_{w_j | (x,y)}[w_j(x) \neq y \,|\, (x, y)] \quad \text{... conditional error probability}$$

$$\varepsilon_{w_j} := \Pr_{(x,y)\sim D, w_j}[w_j(x) \neq y] = \mathbb{E}_{(x,y)\sim D}[\varepsilon_{w_j}(x, y)] \quad \text{... unconditional error prob.}$$

**Questions of interest:**

(i) Given only $A$, how can we estimate $y_1, \dots, y_m$?

(ii) Given only $A$, how can we estimate $\varepsilon_{w_1}, \dots, \varepsilon_{w_n}$?

(iii) If we can choose $g_{ij}$ (in advance or adaptively), how should we choose it?

**One-coin model:** $\varepsilon_{w_j}(x, y)$ constant on $\mathcal{X} \times \{-1, +1\}$, i.e. $\varepsilon_{w_j}(x, y) \equiv \varepsilon_{w_j}$, for all $j \in [n]$ ⇒ (i) to (iii) have been studied extensively (e.g., [1, 2, 3, 4, 5])

**Our contribution:** study (ii) in an extension of the one-coin model: we only assume that there exists (unknown) $L \subseteq [n]$ with $L \geq \frac{n}{2} + 2$ such that $\varepsilon_{w_j}(x, y) \equiv \varepsilon_{w_j}$ for $j \in L$ (our extension allows almost half of the workers to be perfectly colluding adversaries) / answer (i) by making use of existing strategies for the one coin-model

Face images taken from `http://fei.edu.br/~cet/facedatabase.html`.

## Our approach

For estimating error probabilities:

Key observation:

$$\Pr_{(x,y)\sim D, w_j, w_k}[w_j(x) = w_k(x)] = 1 - \varepsilon_{w_j} - \varepsilon_{w_k} + 2\varepsilon_{w_j}\varepsilon_{w_k} + 2\operatorname{Cov}_{(x,y)\sim D}[\varepsilon_{w_j}(x, y), \varepsilon_{w_k}(x, y)].$$

Agreement probabilities $\Pr[w_j(x) = w_k(x)]$ can be easily estimated from $A$:

$$\Pr[w_j(x) = w_k(x)] \approx \frac{\sum_{i=1}^m g_{ij} g_{ik} \mathbb{1}\{A_{ij} = A_{ik}\}}{\sum_{i=1}^m g_{ij} g_{ik}} =: p_{jk}$$

⇒ suggests to solve system of equations (in unknowns $\varepsilon_l$, $l \in [n]$, $c_{jk}$, $1 \leq j < k \leq n$)

$$1 - \varepsilon_j - \varepsilon_k + 2\varepsilon_j\varepsilon_k + 2c_{jk} = p_{jk}, \quad 1 \leq j < k \leq n. \tag{1}$$

**Identifiability:** in general, the system (1) is not identifiable. If $w_j$ follows the one-coin model, then $c_{jk} \triangleq \operatorname{Cov}_{(x,y)\sim D}[\varepsilon_{w_j}(x, y), \varepsilon_{w_k}(x, y)] = 0$, $\forall k \neq j$.

ASSUMPTION: *there exists $L \subseteq [n]$ with $|L| \geq n/2 + 2$ such that for all $j \in L$, the worker $w_j$ follows the one-coin model with $\varepsilon_{w_j}(x, y) \equiv \varepsilon_{w_j} < 1/2$.*

We show that under this assumption, the system (1) has at most one solution.

**Approximate solution:** if $p_{jk}$ in (1) are not exactly true agreement probabilities, there might be no solution of (1) in accordance with our assumption at all.

We prove that if estimates $p_{jk}$ are not too bad, we can solve (1) together with our assumption approximately, and our approximate solution is guaranteed to be close to true error probabilities $\varepsilon_{w_1}, \dots, \varepsilon_{w_n}$ and covariances $\operatorname{Cov}[\varepsilon_{w_j}(x, y), \varepsilon_{w_k}(x, y)]$, $j < k$.

How?? We guess three workers $w_{i_1}, w_{i_2}, w_{i_3}$ that we suspect to follow the one coin-model. Then we can solve (1).

- Guess was correct ⇒ solution approximately satisfies our assumption and guarantee holds
- Guess was wrong ⇒ two possibilities: (i) solution approximately satisfies our assumption; then guarantee still holds (ii) solution does not approximately satisfy our assumption; then we know our guess was wrong

For predicting ground-truth labels:

We take weighted majority votes in which the weights depend on our estimates of the error probabilities. We propose two ways:

- $\hat{y}_i = \operatorname{sign}\left\{\sum_{l \in \hat{L}} \ln\left((1 - \hat{\varepsilon}_{w_l})/\hat{\varepsilon}_{w_l}\right) \cdot A_{il}\right\}$ ... involves only those workers that we believe to follow the one-coin model / Alg. 1 in experiments
- $\hat{y}_i = \operatorname{sign}\left\{\sum_{l \in [n]} (1 - 2\hat{\varepsilon}_{w_l}) \cdot A_{il}\right\}$ ... all workers / Alt-Alg. 1 in experiments

## Theoretical results

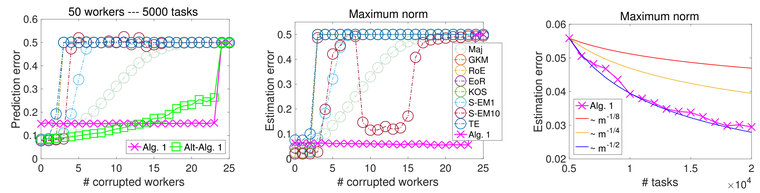
**Theorem** *Assume that $\frac{n}{2} + 2$ of the workers follow the one-coin model with error probabilities not greater than $\gamma_{\mathrm{TR}} < \frac{1}{2}$ and that we are given estimates $p_{jk}$ of $\Pr[w_j(x) = w_k(x)]$. If $|\Pr[w_j(x) = w_k(x)] - p_{jk}| \leq \beta$ for all $j \neq k$ and $\beta$ sufficiently small, we can compute estimates $\hat{\varepsilon}_{w_1}, \dots, \hat{\varepsilon}_{w_n}$ of $\varepsilon_{w_1}, \dots, \varepsilon_{w_n}$ such that*

$$|\varepsilon_{w_i} - \hat{\varepsilon}_{w_i}| \leq C(\gamma_{\mathrm{TR}}) \cdot \beta^{1/4}, \quad i = 1, \dots, n.$$

⇒ if every worker is presented with every task, it follows from Hoeffding's inequality that $|\varepsilon_{w_i} - \hat{\varepsilon}_{w_i}| \leq C(\gamma_{TR}) \cdot (\ln(n^2/\delta)/m)^{1/8}$ with probability at least $1 - \delta$

## Experiments

### Synthetic data

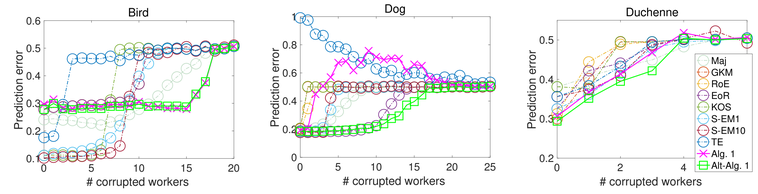



*Prediction error as a function of the number of corrupted workers. Estimation error as a function of # corrupted workers and # tasks, respectively.*

### Real data

| Data set | Maj | GKM [1] | RoE [2] | EoR [2] | KOS [3] | S-EM1 [4] | S-EM10 [4] | TE [5] | Alg. 1 | Alt-Alg. 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| Bird | 0.240 | 0.277 | 0.277 | 0.277 | 0.111 | 0.111 | 0.101 | 0.175 | 0.296 | 0.277 |
| Dog | 0.188 | 0.202 | 0.183 | 0.187 | 0.206 | 0.183 | 0.177 | 0.991 | 0.192 | 0.185 |
| Duchenne | 0.380 | 0.300 | 0.312 | 0.325 | 0.381 | 0.325 | 0.356 | 0.356 | 0.306 | 0.293 |
| RTE | 0.256 | 0.492 | 0.493 | 0.117 | 0.400 | 0.161 | 0.102 | 0.210 | 0.363 | 0.290 |
| Temp | 0.097 | 0.564 | 0.569 | 0.056 | 0.067 | 0.067 | 0.062 | 0.071 | 0.199 | 0.058 |
| Web | 0.121 | 0.024 | 0.042 | 0.101 | 0.037 | 0.093 | 0.051 | 0.995 | 0.030 | 0.061 |

*Prediction error on data sets commonly used in the literature (A is sparse!).*




*Prediction and estimation error as a function of # corrupted workers.*

## Open questions

- task-dependent error probabilities for all workers
- error rate of our model / our estimator ($m^{-1/2}$ rather than $m^{-1/8}$ ??)
- multiclass classification problems
- adaptive task assignment

### References

[1] A. Ghosh, S. Kale, and P. McAfee. Who moderates the moderators? Crowdsourcing abuse detection in user-generated content. In *Conference on Electronic Commerce (EC)*, 2011.

[2] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi. Aggregating crowdsourced binary ratings. In *World Wide Web Conference (WWW)*, 2013.

[3] D. Karger, S. Oh, and D. Shah. Efficient crowdsourcing for multi-class labeling. In *ACM Sigmetrics*, 2013.

[4] Y. Zhang, X. Chen, D. Zhou, and M. Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research*, 17(102):1–44, 2016.

[5] T. Bonald and R. Combes. A minimax optimal algorithm for crowdsourcing. In *Neural Information Processing Systems (NIPS)*, 2017.