An Efficient Algorithm for Robust Collaborative Learning

Mingda Qiao

mqiao@stanford.edu

Motivation

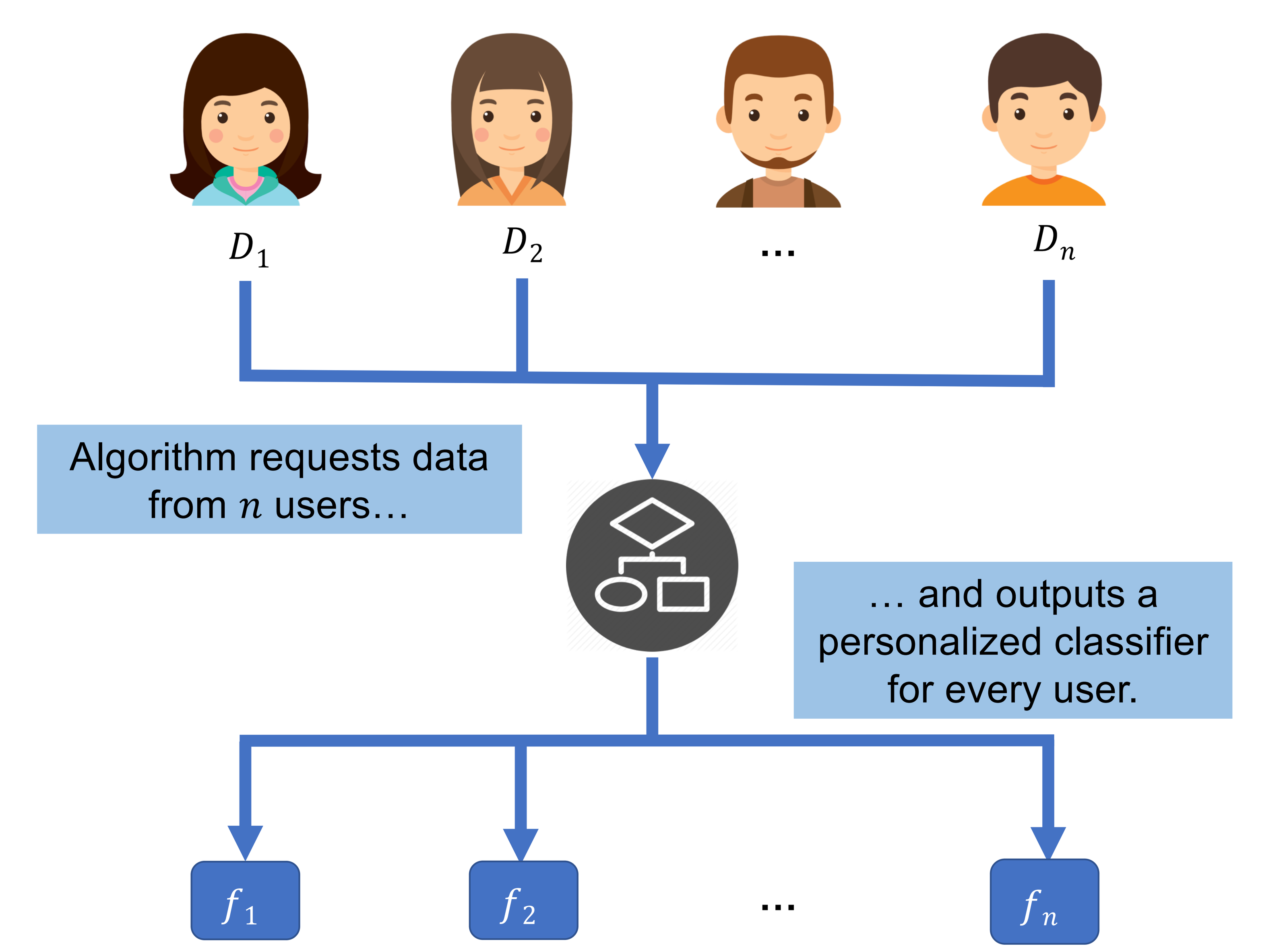
Efficient and robust learning for a diverse crowd.

Example: Personalized Spam Filtering

- Users share the same underlying ground truth…
- …but receive emails written in different languages and styles, from different senders, etc.
- Some adversarial users might maliciously give incorrect examples and labels.

Model

Robust Collaborative Learning [1,2]:



User behavior:

- Upon each request, a **truthful user** i draws $x \sim D_i$, and returns the labeled example $(x, f^*(x))$.
- No guarantee for **adversarial users**.

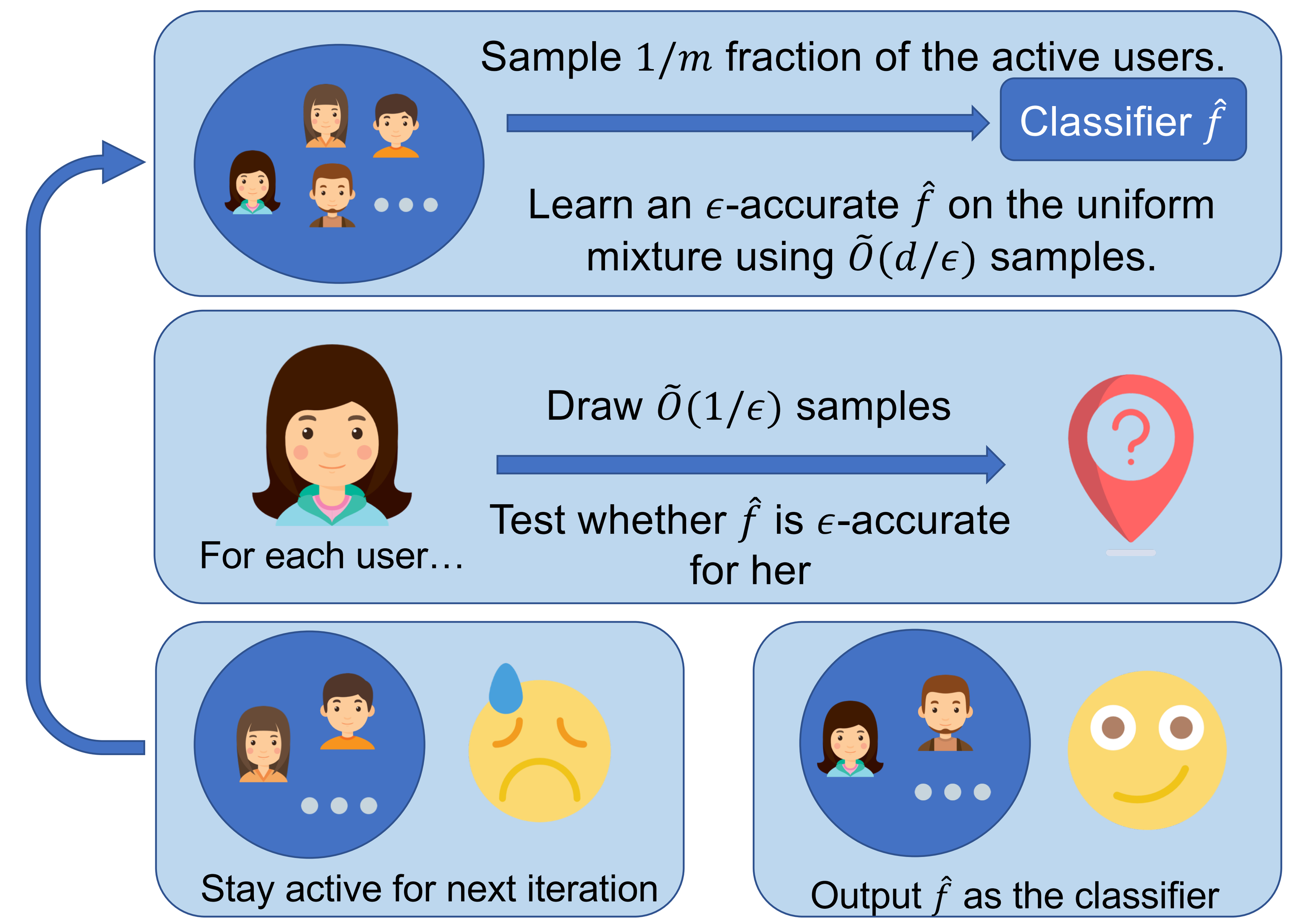
Goal: f_i is accurate on D_i for **each truthful user** i.

Known Results

- **Efficient collaboration in the absence of adversaries**: [1] gives an algorithm with the optimal $O(\log n)$ **overhead** when all users are truthful.
- **Sample-efficient collaboration even when there are adversarial users**: [2] gives an algorithm with the optimal $O(m + \log n)$ overhead when there are m adversarial users, yet the algorithm is computationally costly.

Theoretical Results

UserSample Algorithm:



Analysis:

- Each iteration assigns an accurate classifier to at least an $\Omega(1/m)$ fraction of the users with $\Omega(1)$ probability.
- After $O(m \log n)$ iterations, at most m users remain. Then, separately learn a classifier for each of them.

Theorem: UserSample has an overhead of $O(m \log n)$, which is near-optimal up to a log factor.

Open: Efficient algorithm with optimal overhead?

Experiments

Setting:

- $n = 200$ users, among which $m = 2$ users are adversarial.
- Adversarial behavior: flip the correct label.
- Hypothesis class has VC-dimension $d = 500$.

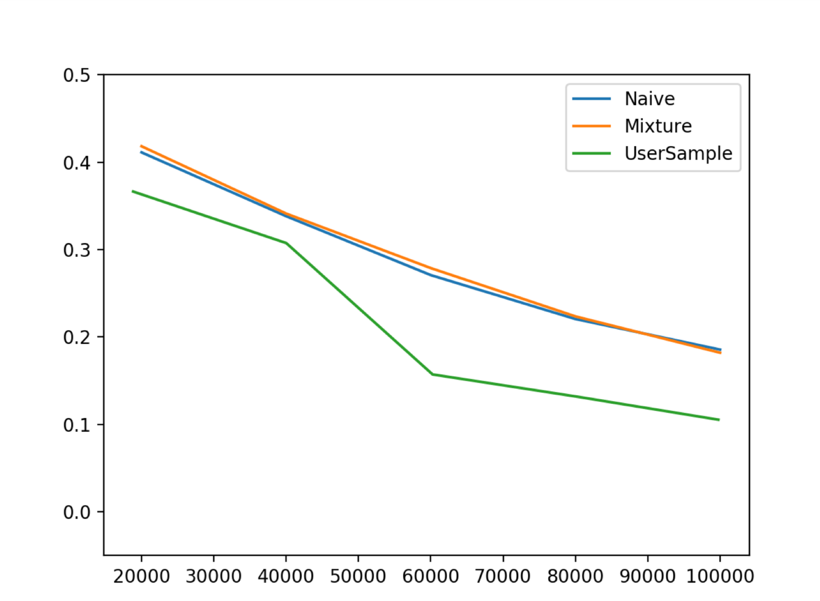
Ground truth: (a) random binary function over $\{1,2,\dots,d\}$; (b) random linear classifier over $\mathbb{R}^d$.

Data distribution: (a) uniform over a random subset of size d_0; (b) Gaussian over a random d_0-dimensional subspace of $\mathbb{R}^d$. Different users have different d_0.

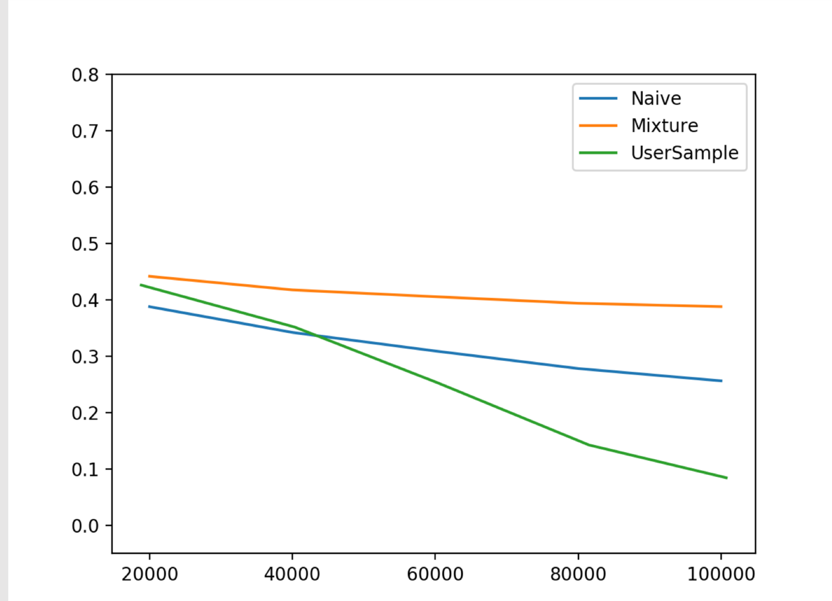
Methods:

- Naïve: learn a classifier for every user separately
- Mixture: directly learn the uniform mixture distribution
- UserSample: the proposed algorithm

Empirical Results



(a) Binary functions



(b) Linear functions

X-axis: number of training examples. Y-axis: largest testing error among all truthful users. Both averaged over 10 trials.

References

[1] Avrim Blum, Nika Haghtalab, Ariel D. Procaccia, and Mingda Qiao. Collaborative PAC learning. NeurIPS 2017.

[2] Mingda Qiao. Do outliers ruin collaboration? ICML 2018.