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.

**Motivation**

**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.

**Model**

Robust Collaborative Learning [1, 2]:

Algorithm requests data from $n$ users...

... and outputs a personalized classifier for every user.

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 $i$.

**Theoretical Results**

**UserSample Algorithm:**
- Learn an $\epsilon$-accurate $f$ on the uniform mixture using $\tilde{O}(d/\epsilon)$ samples.
- Draw $\tilde{O}(1/\epsilon)$ samples.
- Test whether $f$ is $\epsilon$-accurate for her.
- Stay active for next iteration.
- Output $f$ as the classifier.

**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.

**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.

**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 $\{1, 2, ..., d\}$; (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

**References**
