Adversarial Learning from Crowds

Pengpeng Chen[1,3], Hailong Sun[2,3], Yongqiang Yang[1,3], Zhijun Chen[1,3]

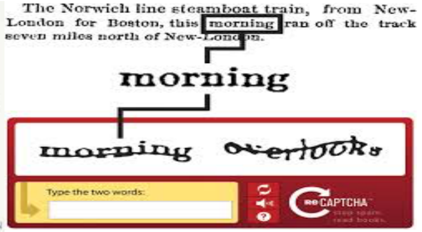
[1]SKLSDE Lab, School of Computer Science and Engineering, Beihang University, China

[2]School of Software, Beihang University, Beijing, China

[3]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, China

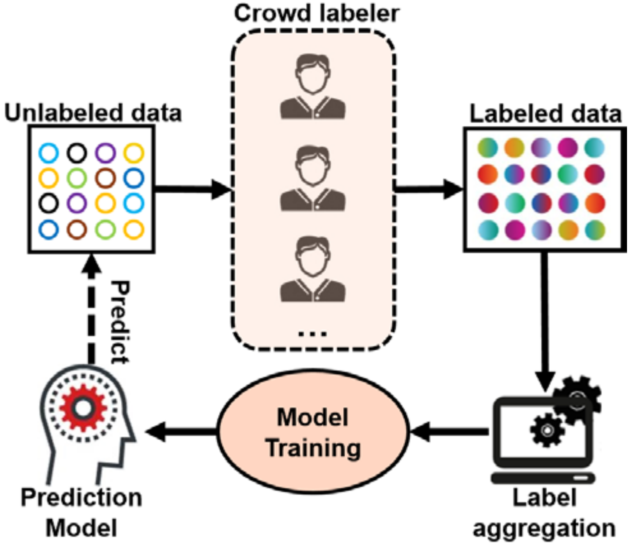
Crowdsourcing and Machine learning

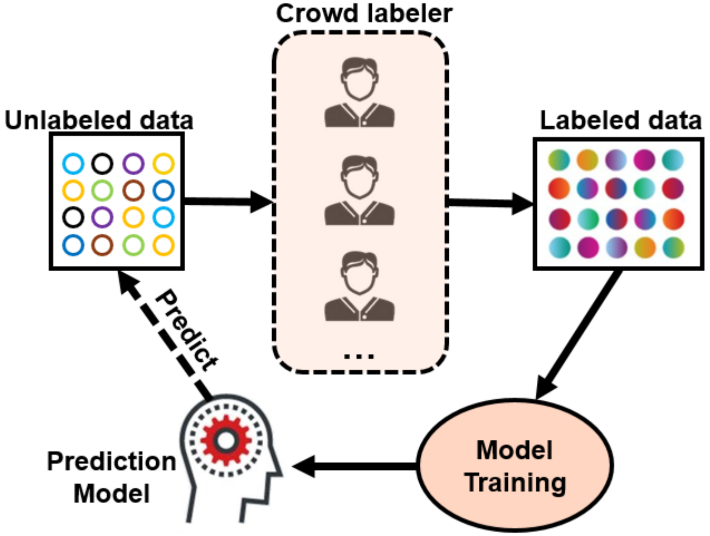
- A popular paradigm of outsourcing work to individuals in the form of an *open call*
- A tool for cost-effectively and efficiently collecting labels for training data in the ML community



Prior work

Two Stage	One Stage（LFC）
Label aggregation	Simultaneous
Model training	Mutual reinforcement

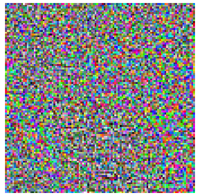




Machine learning with crowdsourcing: A brief summary of the past research and future directions. AAAI-2019

Problem

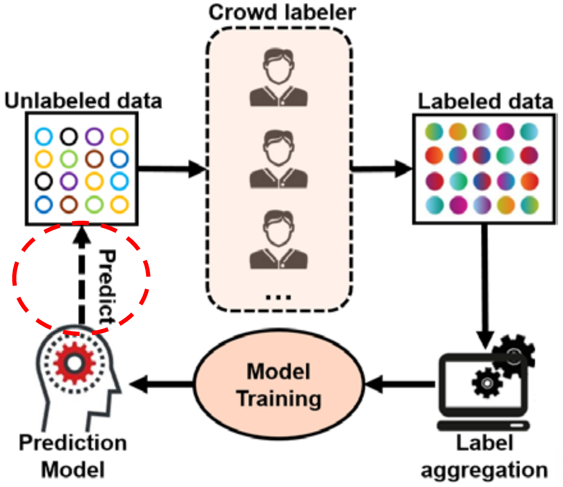
FGSM



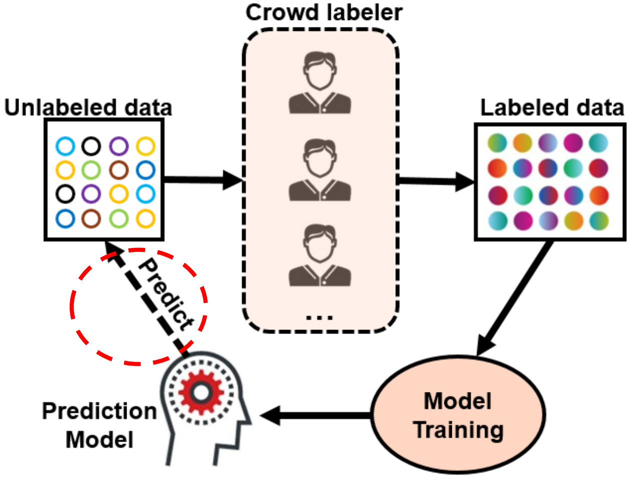
$\mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

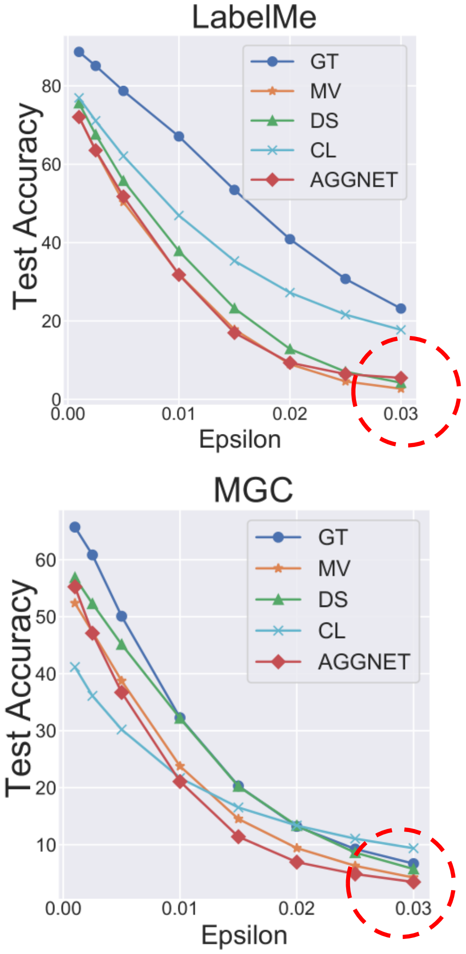
ICLR-2015

Two Stage



One Stage





There is an urgent need to investigate adversarial attacks and defense for the LFC family.

Problem formulation

- We formalize the problem of adversarial learning from crowds as a bilevel min-max problem

Outer subproblem: $\min_{\Theta} -\alpha \log p(\mathbf{Y} \mid \mathcal{X}, \Theta) - (1-\alpha) \log p(\mathbf{Y} \mid \mathcal{X}', \Theta)$

Inner subproblem: $s.t.\ \ \mathcal{X}' = \operatorname*{argmax}_{\mathcal{X}'} - \log p(\mathbf{Y} \mid \mathcal{X}', \Theta),$

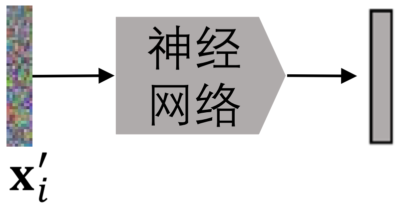
$and\ \ \mathcal{X}' = \{\mathbf{x}'_i \mid \|\mathbf{x}'_i - \mathbf{x}_i\|_p \leq \epsilon\},$

Method (solving the *inner problem*)

- We solve the inner problem to generate the adversarial examples.
  - The **Loss** of **inner problem** is reformulated as the cross entropy between the posterior probability distribution of the true label and output of the neural network

$$-\log p(\mathbf{Y} \mid \mathcal{X}', \Theta) = -\sum_i \mathbb{E}_{\rho(t_i)} \log[p(t_i \mid \mathbf{x}'_i; \boldsymbol{\theta})]$$

  - Calculating the gradient of Loss with respect to $\mathbf{x}'_i$
  - Using Projected Gradient Descent (PGD) to find $\mathbf{x}'_i$



Method (solving the *outer problem*)

- We solve the *inner problem* by expectation-maximization (EM) algorithm
  - In E-step, it targets to infer the posterior probability distribution of the true label

$$\rho\left(t_i = k\right) \propto \prod\nolimits_j p\left(y_{ij} \mid t_i = k; \mathbf{\Pi}^{(1)}, \ldots \mathbf{\Pi}^{(N)}\right)$$
$$\cdot \left(\alpha p\left(t_i = k \mid \mathbf{x}_i; \boldsymbol{\theta}\right) + (1-\alpha) p\left(t_i = k \mid \mathbf{x}'_i; \boldsymbol{\theta}\right)\right)$$

Method (solving the *outer problem*)

- We solve the *inner problem* by expectation-maximization (EM) algorithm
  - In M-step, using back propagation, it learns the neural network parameters with the following Loss function

$$-\alpha \log p(\mathbf{Y} \mid \mathcal{X}, \Theta) - (1-\alpha) \log p(\mathbf{Y} \mid \mathcal{X}', \Theta)$$

  - In M-step, it update the worker confusion matrix as follows

$$\pi_{kk'}^{(j)} = \frac{\sum_i \rho\left(t_i = k\right) \mathbb{I}\left(y_{ij} = k'\right)}{\sum_i \rho\left(t_i = k\right) \mathbb{I}\left(y_{ij} \neq \perp\right)}$$

Method: A-LFC

- We summarize the proposed method
  - Step 1: solving the **inner** subproblem via using Projected Gradient Descent (PGD)
  - Step 2: solving the **outer** subproblem Expectation-Maximization (EM) algorithm
    - In **E** Step, we infer the posterior probability distribution of the true label
    - In **M** Step, we learn the parameters of neural network using back propagation and the confusion matrices of workers

Empirical results

- Datasets
  - LabelMe: image classification dataset
  - MGC: classification of music genres
  - Sentiment: sentiment polarity of movie reviews
- Baselines
  - Two stage method: MV+NN and DS+NN
  - One stage method: AggNet and CL
- The adversarial attack
  - FGSM, PGD, CW, and MIM

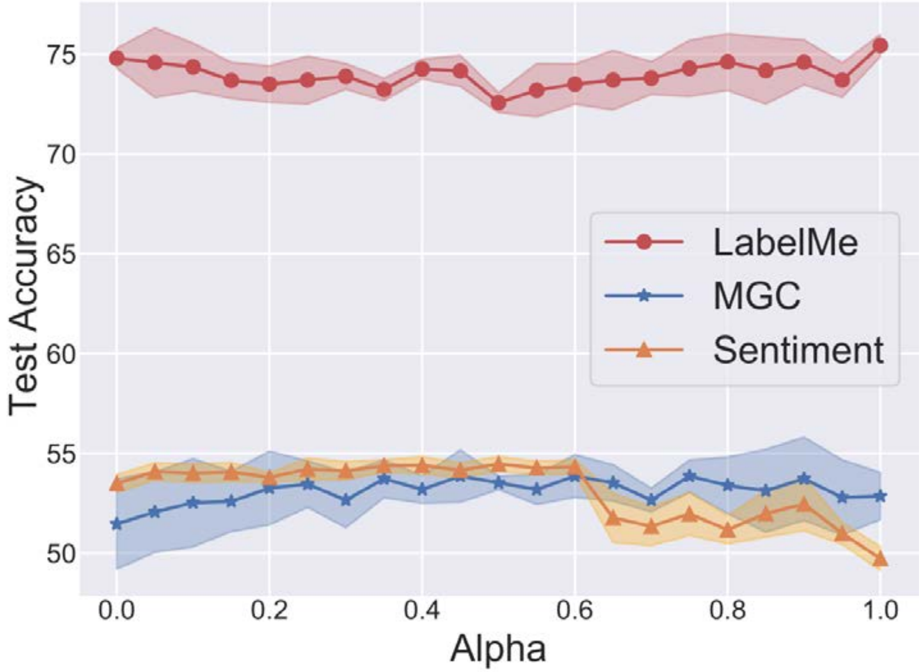
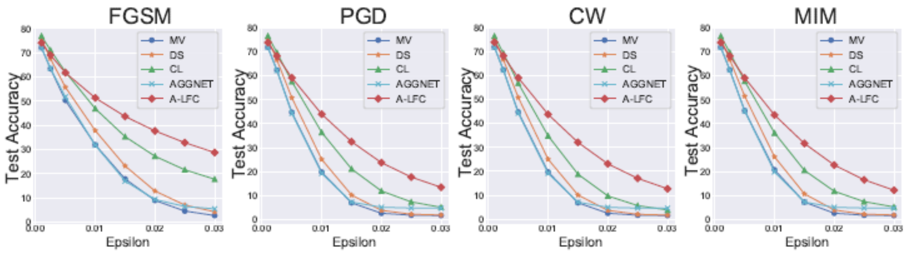
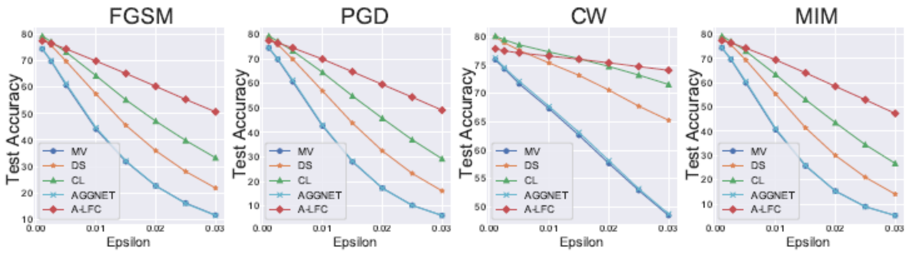


Figure: Sensitivity to imitation parameter

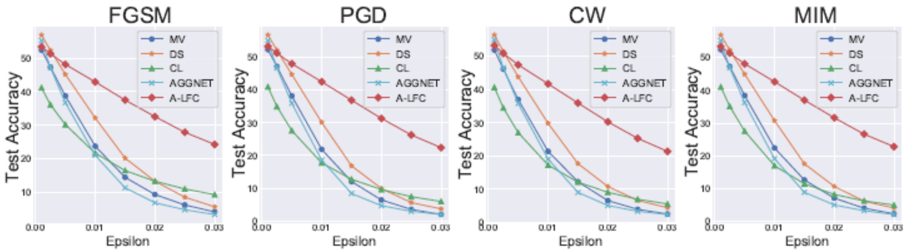
Empirical results



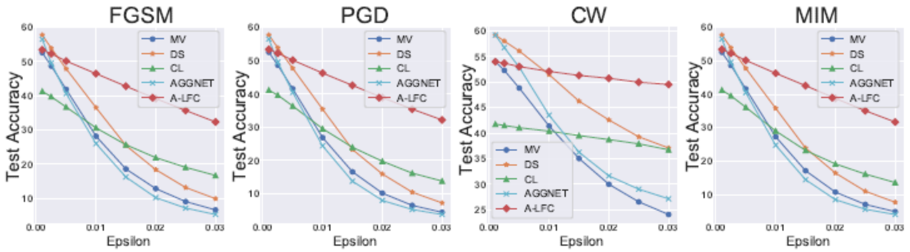
(a) Dataset LabelMe.



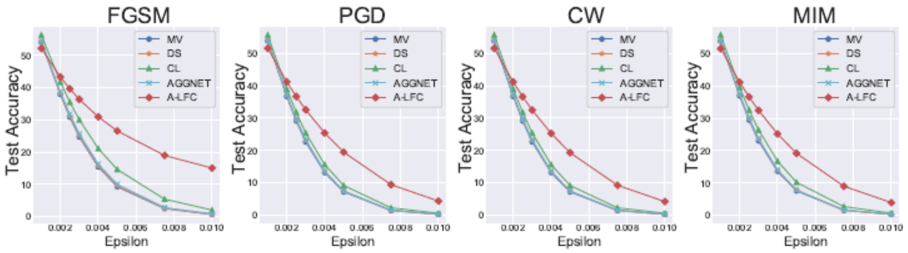
(a) Dataset LabelMe.



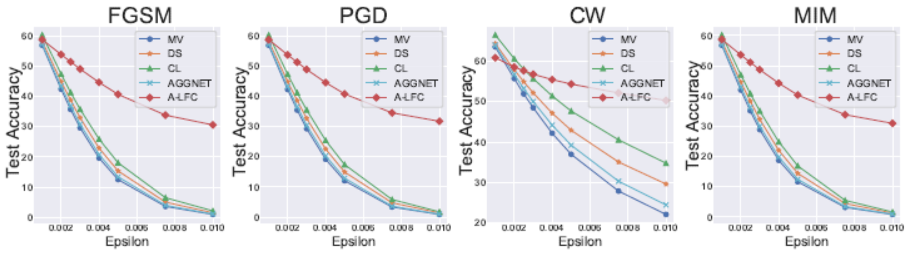
(b) Dataset MGC.



(b) Dataset MGC.



(c) Dataset Sentiment.



(c) Dataset Sentiment.

Figure: White- and black-box robustness (test accuracy (%) of the classifier under white- and black-box attacks

Empirical results

- The learned confusion matrices

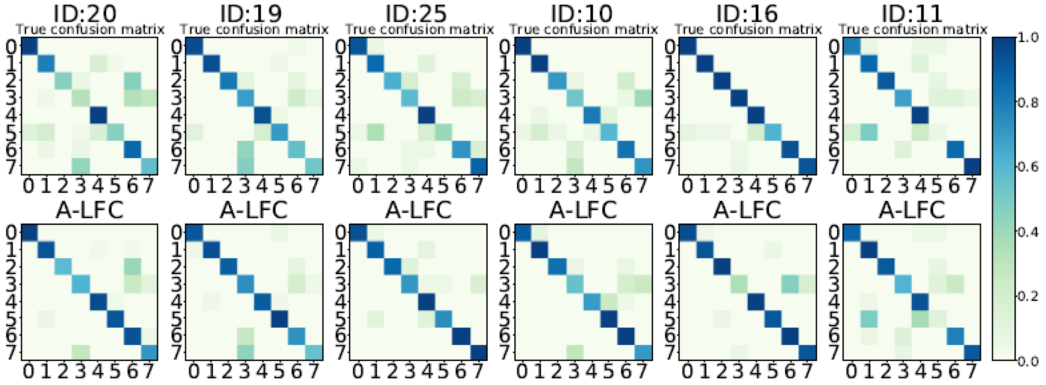


Figure: Confusion matrices of workers on LabelMe

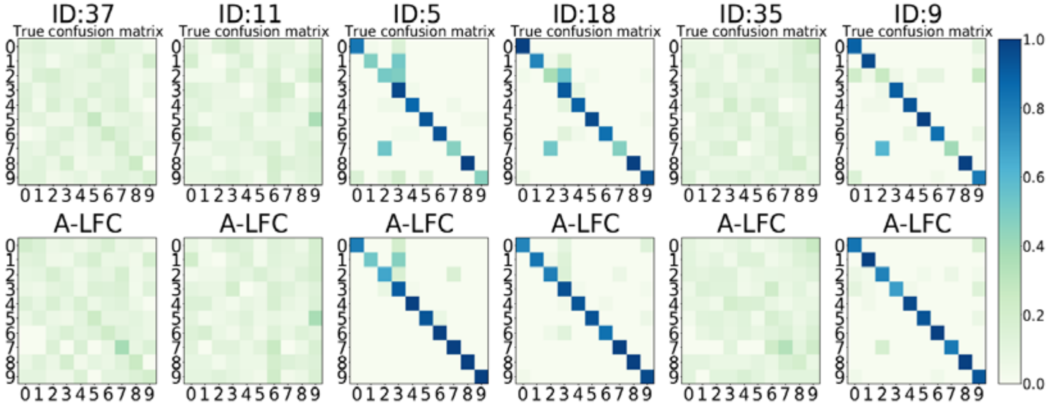


Figure: Confusion matrices of workers on MGC

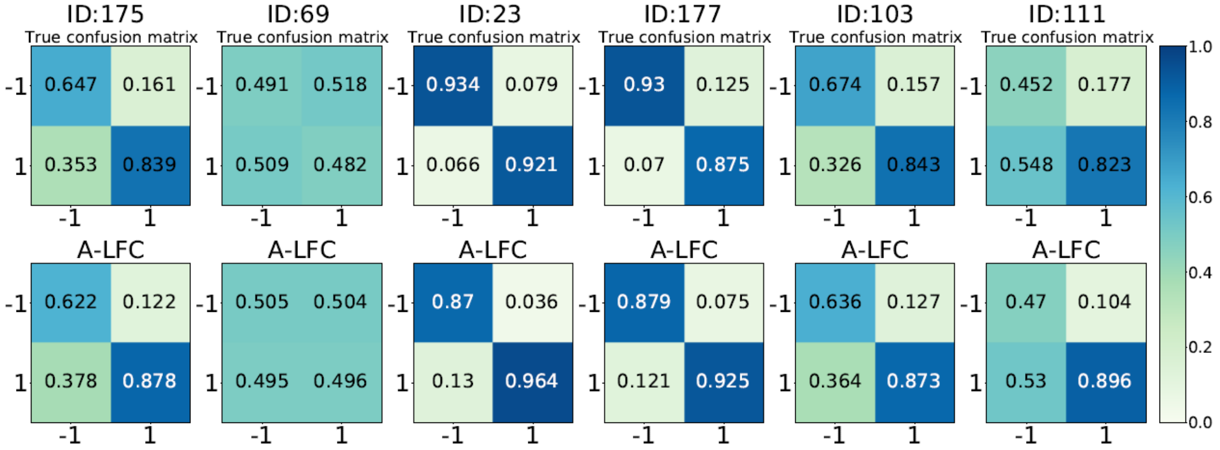


Figure: Confusion matrices of workers on Sentiment

Conclusion

- In this work, we move one step further and explore how to learn an LFC model robust to the adversarial examples.
  - We investigate the influence of adversarial examples on the performance of representative LFC models.
  - We formulate the problem of LFC in the adversarial environment as a bilevel min-max problem
  - We propose a novel LFC framework robust to the adversarial examples.、
- In future work, we plan to investigate approaches to defending against other types of adversarial attacks such as data poisoning.