

Adversarial machine learning - Truth inference and data poisoning in crowdsourcing

Farnaz Tahmasebian

Crowdsourcing



Applicable as traffic and road condition, transcriptions, translations of language, or image annotations.

Crowdsourcing

Reputation and the rise of the 'rating' society

Tomas Chamorro-Premuzic

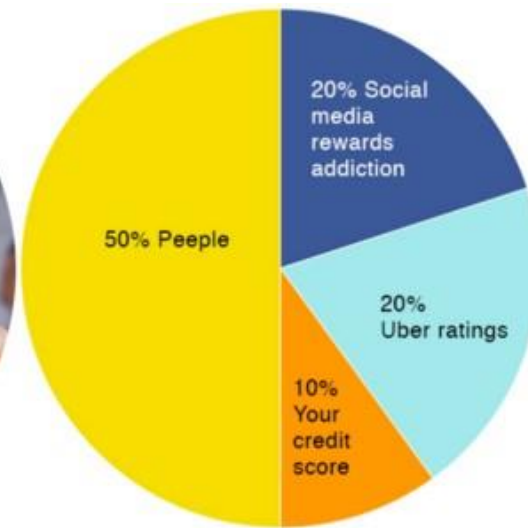
From Uber to Airbnb, ratings rule our world, but the system is far from perfect



▲ 'Ratings are usually skewed, particularly when raters are not motivated to be entirely honest.' Photograph: Noel Hendrickson/Getty Images

*The researchers said their findings showed that “despite crowdsourcing **being a more efficient way** of accomplishing many tasks, it’s also **a less secure approach.**”*

BlackMirror TV Series



Malicious Crowdsourcing

Malicious crowdsourcing = crowdturfing

Hiring a large army of **real users** for malicious attacks

Fake customer reviews, rumors, targeted spam

Most existing defenses fail against real users (CAPTCHA)

Connectivity

Fake Followers for Hire, and How to Spot Them

It's possible to buy a good reputation on the Internet for a modest price, but some are trying to put an end to that.

Algorithms can write fake reviews that humans rate as "helpful"



Sign in

News

Sport

Weather

Shop

Reel

Travel

Technology

Keeping crowdsourcing honest: can we trust the reviews?

By Orin Gordon
BBC Business reporter

Over the last week, Trump supporters and others accusing CNN of threatening a Reddit user took justice into their own hands by leaving [thousands of 1-star reviews](#) on CNN's mobile app. [The digital lynch mob left CNN with an average rating of just one star in the Apple App Store.](#)

CNN suffered a similar attack in Google's App Store but was able to more easily withstand the assault thanks to hundreds of thousands of preexisting positive reviews.

Customer Ratings

Current Version All Versions

Average Rating: ★★★★★ 5,527 Ratings

Click to rate: ★ ★ ★ ★ ★



APPLE APP STORE

CNN currently has a one-star average in the Apple App Store.

Fake content was widespread during the presidential campaign. Facebook has estimated that 126 million of its platform users saw articles and posts promulgated by Russian sources. Twitter has found 2,752 accounts established by Russian groups that tweeted 1.4 million times in 2016.^[11] The widespread nature of these disinformation efforts led Columbia Law School Professor Tim Wu to ask: "Did Twitter kill the First Amendment?"^[12]

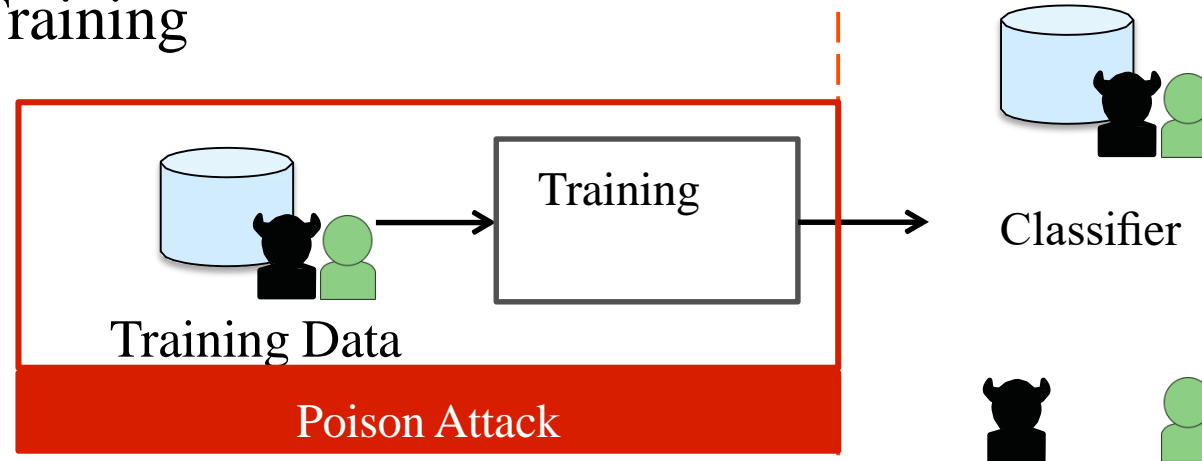
When [fake news] activities move from sporadic and haphazard to organized and systematic efforts, they become disinformation campaigns with the potential to disrupt campaigns and governance in entire countries.

Recall: Data poisoning attack in ML

- Data poisoning attacks:

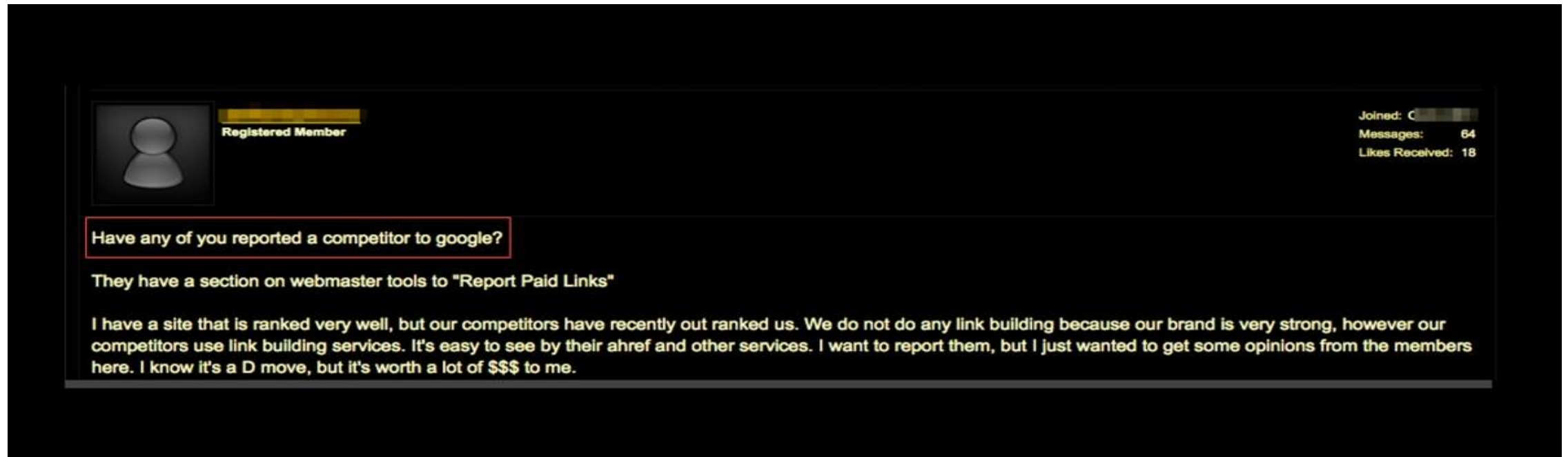
- Involve feeding adversarial training data to the classifier.
- Attacker attempts to pollute training data in such a way that the boundary between what the classifier categorizes as good data, and what the classifier categorizes as bad, shifts in his favor.

Model Training



Data Poisoning Attack in CrowdSourcing

Attack to legitimate users feedbacks, rating



<https://elie.net/blog/ai/attacks-against-machine-learning-an-overview>

Data poisoning attacks in
ML

vs

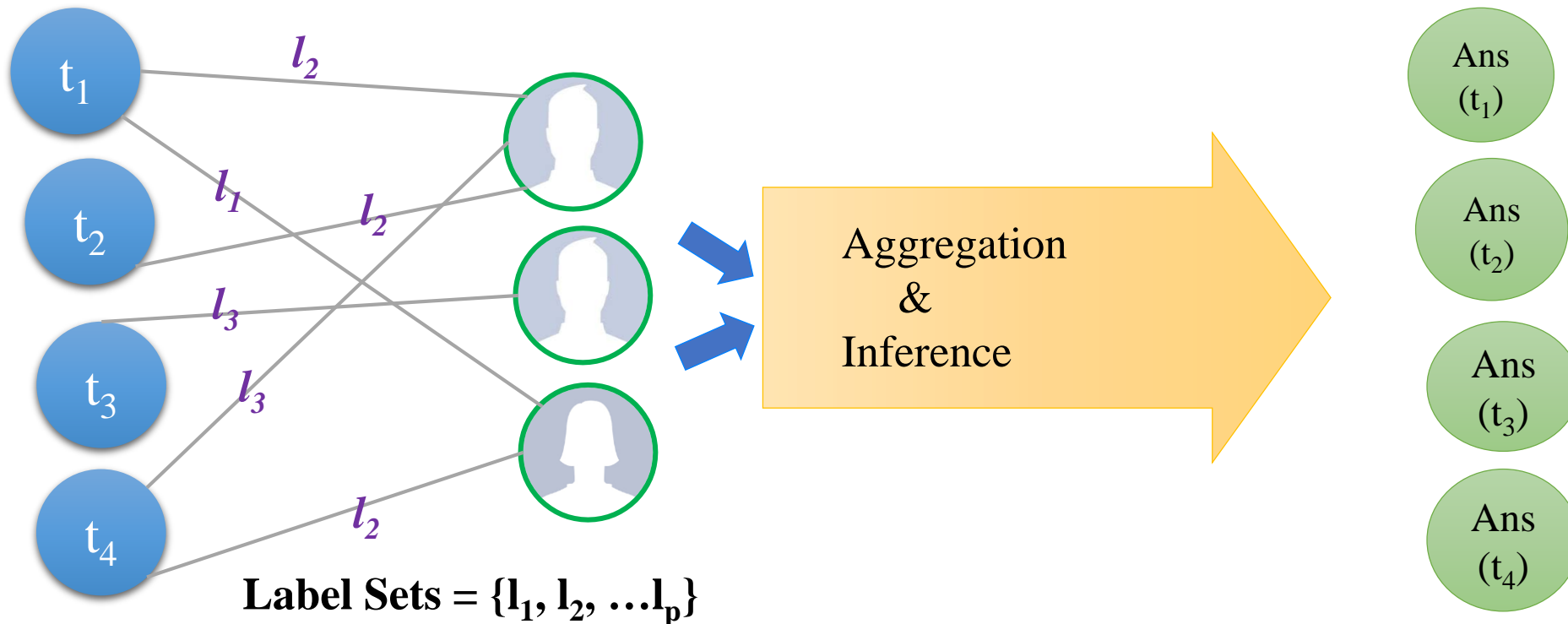
Data poisoning attacks in
Crowdsourcing

- Classification problem
- Labeled data
- Features are available

- Unsupervised problem
- Non-labeled data
- No feature available (Just Labels)

Truth inference

Given answer matrix for m task which are collected from n workers, the target is to infer the truth label of each task.



Entities in Crowdsourcing



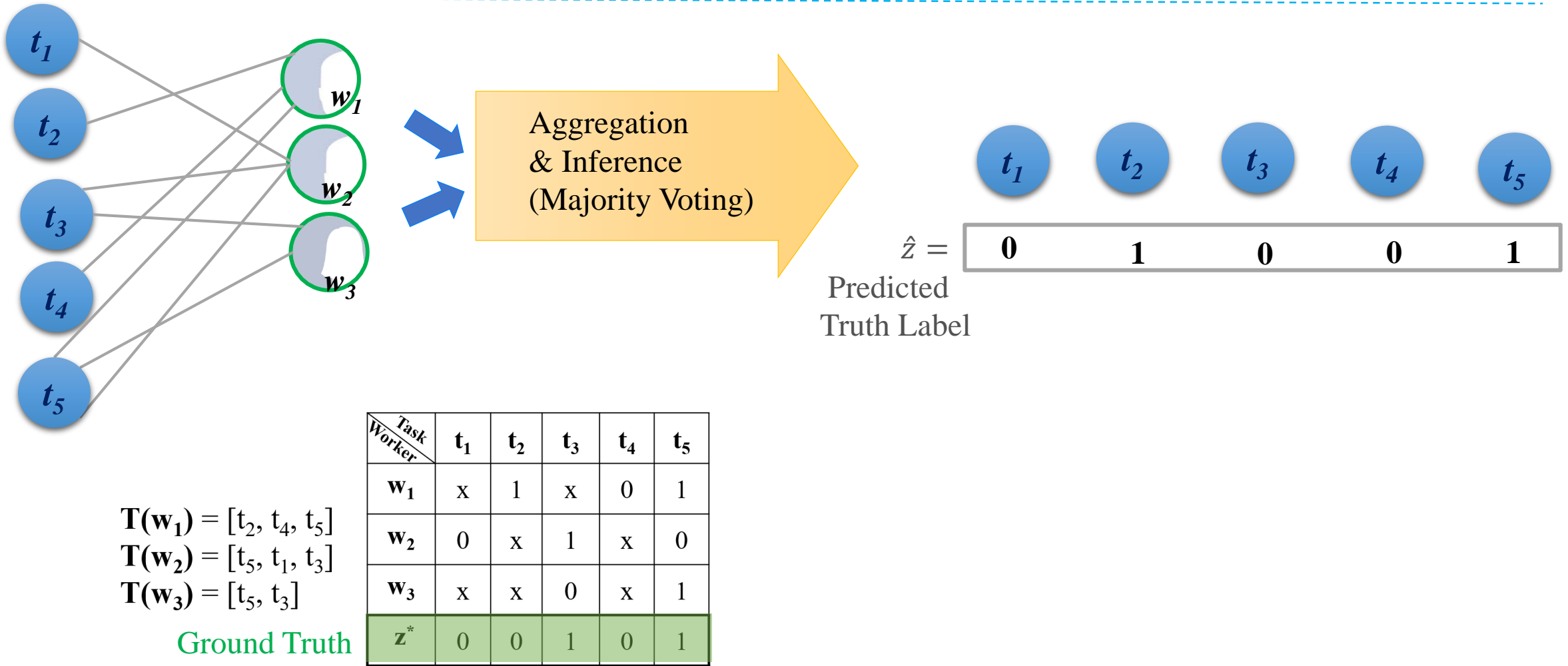
Tasks

- **Different Task Types**
 - What type of tasks, such as multi label, Decision making (Yes/No) and Numeric tasks
- **Different Task Models**
 - How they model each tasks? E.g. task difficulty

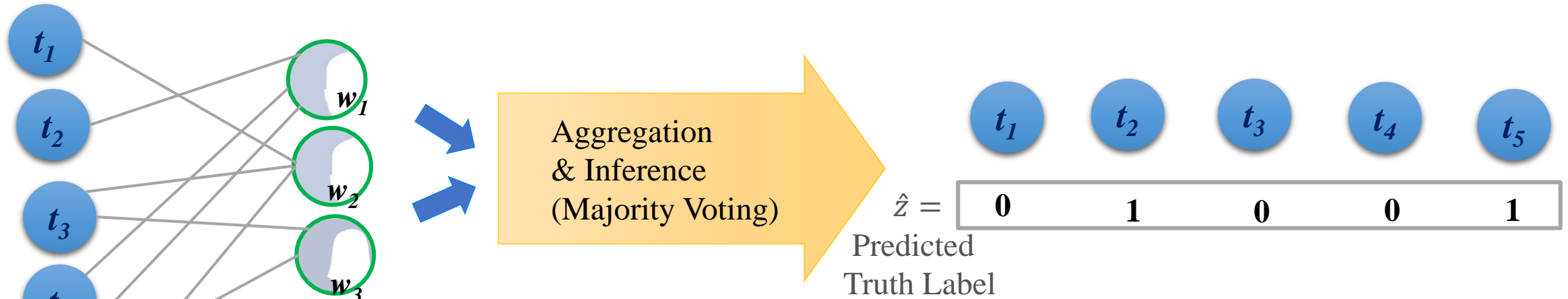
Workers

- **Different Worker Models**
 - How they model each worker? E.g. worker probability, Confusion Matrix
-

Example of Truth inference



Example of Truth inference



$$T(w_1) = [t_2, t_4, t_5]$$

$$T(w_2) = [t_5, t_1, t_3]$$

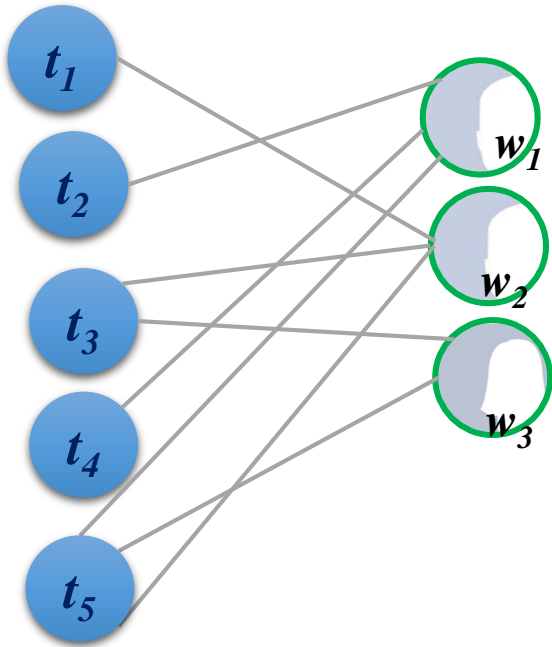
$$T(w_3) = [t_5, t_3]$$

Ground Truth

Task Worker	t_1	t_2	t_3	t_4	t_5
w_1	x	1	x	0	1
w_2	0	x	1	x	0
w_3	x	x	0	x	1
z^*	0	0	1	0	1

How reliable are workers?

Example of Reliability of Workers



$T(w_1) = [t_2, t_4, t_5]$
 $T(w_2) = [t_5, t_1, t_3]$
 $T(w_3) = [t_5, t_3]$

Ground Truth

Task \ Worker	t_1	t_2	t_3	t_4	t_5
w_1	x	1	x	0	1
w_2	0	x	1	x	0
w_3	x	x	0	x	1
z^*	0	0	1	0	1


Consider w_2 :

Number of labels (0) answered correctly: 1

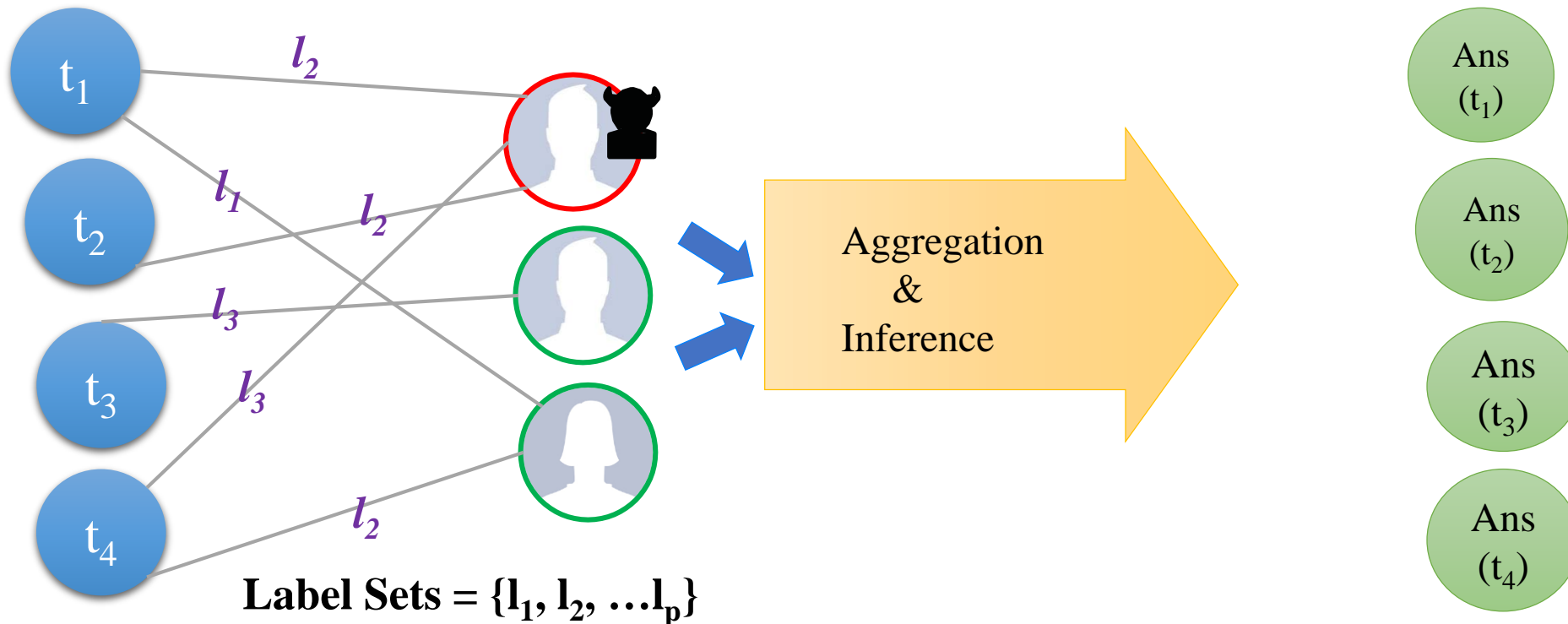
Number of labels (1) answered correctly: 1

$$\pi = \begin{matrix} & \text{truth} \\ & 0 & 1 \\ \text{observed} & \begin{bmatrix} \beta = 0.5 & 0.5 \\ 0.0 & \alpha = 1.0 \end{bmatrix} \end{matrix}$$

Truth inference in presence of **attackers**

Given answer matrix for m task which are collected from n workers, the target is to infer the truth label of each task. 

The answers can be potentially poisoned by **malicious workers**.



Challenge

- Wide range of workers' behavior
- Skewed data
- Various quality of workers
- Potentially malicious behavior

Aggregation & Inference algorithm

- Majority Voting (MV)
 - MV
 - Penalty Based + MV
- Probabilistic Graphical Model (PGM):
 - EM (PGM, Confusion matrix)
 - BCC (PGM, Confusion matrix)
 - KOS (PGM, probability)
 - GLAD (PGM, probability, consider task's difficulty level)
- Neural Network
 - Variational AutoEncoder

Aggregation & Inference algorithm

- Majority Voting (MV)
 - MV
 - Penalty Based + MV
- Probabilistic Graphical Model (PGM):
 - EM (PGM, Confusion matrix)
 - BCC (PGM, Confusion matrix)
 - KOS (PGM, probability)
 - GLAD (PGM, probability, consider task's difficulty level)
- Neural Network
 - Variational AutoEncoder

Majority Voting

The Naïve approach is Majority Voting (MV)

Takes the answer given by majority workers as the truth.

The biggest limitation of MV:

- It treats all workers as equal.
- Workers may have different levels of qualities:
 - Workers carefully answers tasks;
 - Workers may randomly answer tasks
 - Malicious worker may even intentionally give wrong answers.

Penalty Based + MV

- Adversaries' labeling patterns differ from those of honest workers.
- Adversary labeling patterns should be statistical outliers.
- Identifies outliers by penalizing the workers for the number of 'conflicts' they are involved in.
- How much of the penalty budget to allocate to each worker for each task
- How to aggregate the penalties allocated to each worker to arrive at the final reputation score.

Penalty Based + MV

Algorithm 1 SOFT PENALTY

- 1: **Input:** W, \mathcal{T} and \mathcal{L}
- 2: For every task $t_j \in \mathcal{T}_{cs}$, allocate penalty s_{ij} to each worker $w_i \in W_j$ as follows:

$$s_{ij} = \begin{cases} \frac{1}{d_j^+}, & \text{if } \mathcal{L}_{ij} = +1 \\ \frac{1}{d_j^-}, & \text{if } \mathcal{L}_{ij} = -1 \end{cases}$$

- 3: **Output:** Net penalty of worker w_i

$$\text{pen}(w_i) = \frac{\sum_{t_j \in \mathcal{T}_i \cap \mathcal{T}_{cs}} s_{ij}}{|\mathcal{T}_i \cap \mathcal{T}_{cs}|}$$

Algorithm 2 HARD PENALTY

- 1: **Input:** W, \mathcal{T} and \mathcal{L}
- 2: Create a bipartite graph \mathcal{B}^{cs} as follows:
(i) Each worker $w_i \in W$ is represented by a node on the left (ii) Each task $t_j \in \mathcal{T}_{cs}$ is represented by two nodes on the right t_j^+ and t_j^- (iii) Add the edge (w_i, t_j^+) if $\mathcal{L}_{ij} = +1$ or edge (w_i, t_j^-) if $\mathcal{L}_{ij} = -1$.
- 3: Compute an optimal semi-matching \mathcal{M} on \mathcal{B}^{cs}
- 4: **Output:** Net penalty of worker w_i , $\text{pen}(w_i) = \deg_{\mathcal{M}}(w_i)$

Aggregation & Inference algorithm

- Majority Voting (MV)
 - MV
 - Penalty Based MV
 - Probabilistic Graphical Model (PGM):
 - EM (PGM, Confusion matrix)
 - BCC (PGM, Confusion matrix)
 - KOS (PGM, probability)
 - GLAD (PGM, probability, consider task's difficulty level)
 - Neural Network
 - Variational AutoEncoder
 - Complete
-

Problem Setting

- Set of M items and set of K workers
- Assume that each item has only two possible labels: 0 and 1
- Each worker u_k provides label for item o_m

$$\Theta = \{p, \{\alpha_k, \beta_k\}_{k=1}^K\}$$

$$\alpha_k = \Pr(x_m^k = 1 | x_m^* = 1)$$

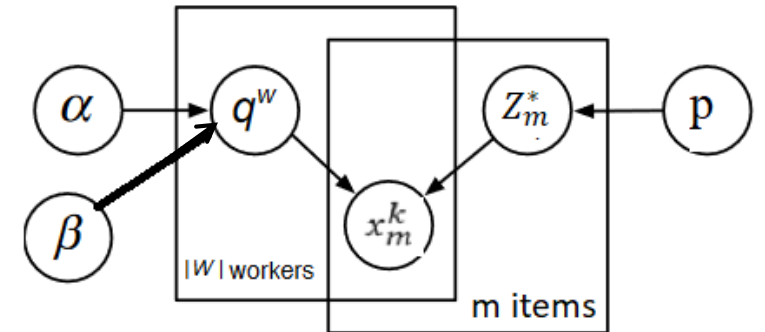
$$\beta_k = \Pr(x_m^k = 0 | x_m^* = 0),$$

X^* the set of the items' true labels: $X^* = \{x_m^*\}_{m=1}^M$

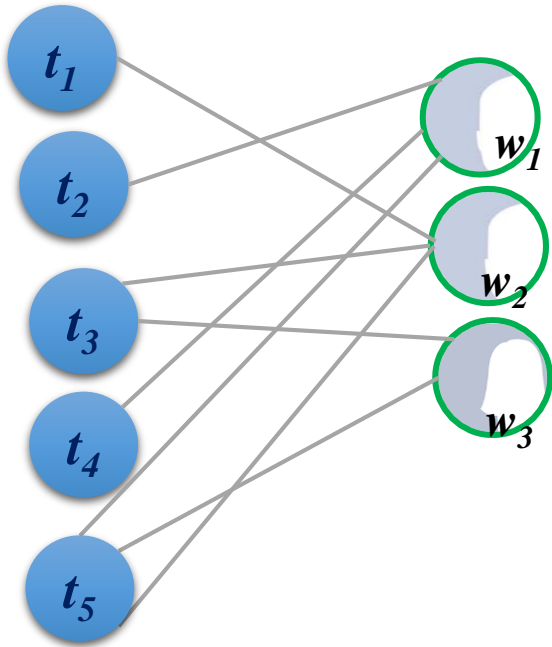
x_m^k the label provided by the normal worker u_k for the item o_m

x_m^* the true label of the item o_m

X_m the set of labels for the item o_m



Example of Reliability of Workers



$T(w_1) = [t_2, t_4, t_5]$
 $T(w_2) = [t_5, t_1, t_3]$
 $T(w_3) = [t_5, t_3]$

Ground Truth

Task Worker	t_1	t_2	t_3	t_4	t_5
w_1	x	1	x	0	1
w_2	0	x	1	x	0
w_3	x	x	0	x	1
z^*	0	0	1	0	1

Consider w_2 :

Number of labels (0) answered correctly: 1

Number of labels (1) answered correctly: 1

$$\pi = \begin{matrix} & \text{truth} \\ & 0 & 1 \\ \text{observed} & \begin{bmatrix} \beta = 0.5 & 0.5 \\ 0.0 & \alpha = 1.0 \end{bmatrix} \end{matrix}$$

Basic EM Framework

Algorithm 1 The basic EM framework of Dawid and Skene (1979).

Input: Sets of worker-generated labels for each instance

Initialize each instance's label based on a simple majority vote

repeat

for all Workers w **do**

 Calculate w 's quality parameter(s), treating each instance's current label as ground truth

end for

for all Instances i **do**

 Calculate the most likely label for i , treating each worker's approximated quality parameter(s) as ground truth

end for

until Label assignments have converged

Output: The current label assignments for each instance

E-Step

- Model parameters Θ are fixed.
- For each item calculate $\omega_m = \Pr\{x_m^* = 1|X\}$

Inferred probability that item m belongs to class 1

$$\begin{aligned}\omega_m &= \Pr\{x_m^* = 1|X; \Theta\} \\ &= \frac{\Pr\{X|x_m^* = 1\} \cdot p}{\Pr\{X|x_m^* = 1\} \cdot p + \Pr\{X|x_m^* = 0\} \cdot (1-p)} \\ &= \frac{A_{m1}}{A_{m1} + A_{m0}}\end{aligned}$$

Maximizing likelihood function

where

$$A_{m1} = \prod_{k \in U_m} \alpha_k^{x_m^k} (1 - \alpha_k)^{1-x_m^k} \cdot p$$

$$A_{m0} = \prod_{k \in U_m} \beta_k^{1-x_m^k} (1 - \beta_k)^{x_m^k} \cdot (1-p).$$

$$\begin{aligned}Q(\Theta) &= E[\log L(\Theta; X, X^*)] = E[\log \prod_{m=1}^M L(\Theta; X_m, x_m^*)] \\ &= \sum_{m=1}^M \{ \omega_m \log[\prod_{k \in U_m} \alpha_k^{x_m^k} (1 - \alpha_k)^{1-x_m^k} \cdot p] \\ &\quad + (1 - \omega_m) \log[\prod_{k \in U_m} \beta_k^{1-x_m^k} (1 - \beta_k)^{x_m^k} \cdot (1-p)] \},\end{aligned}$$

M-Step

- The posterior probabilities are fixed.

p: probability of tasks whose truth is 1

$$p = \frac{\sum_{m=1}^M \omega_m}{M},$$

Reliability of worker k associated
with task whose truth is 1

$$\alpha_k = \frac{\sum_{m \in O_k} \omega_m \cdot x_m^k}{\sum_{m \in O_k} \omega_m},$$



Reliability of worker k associated
with task whose truth is 0

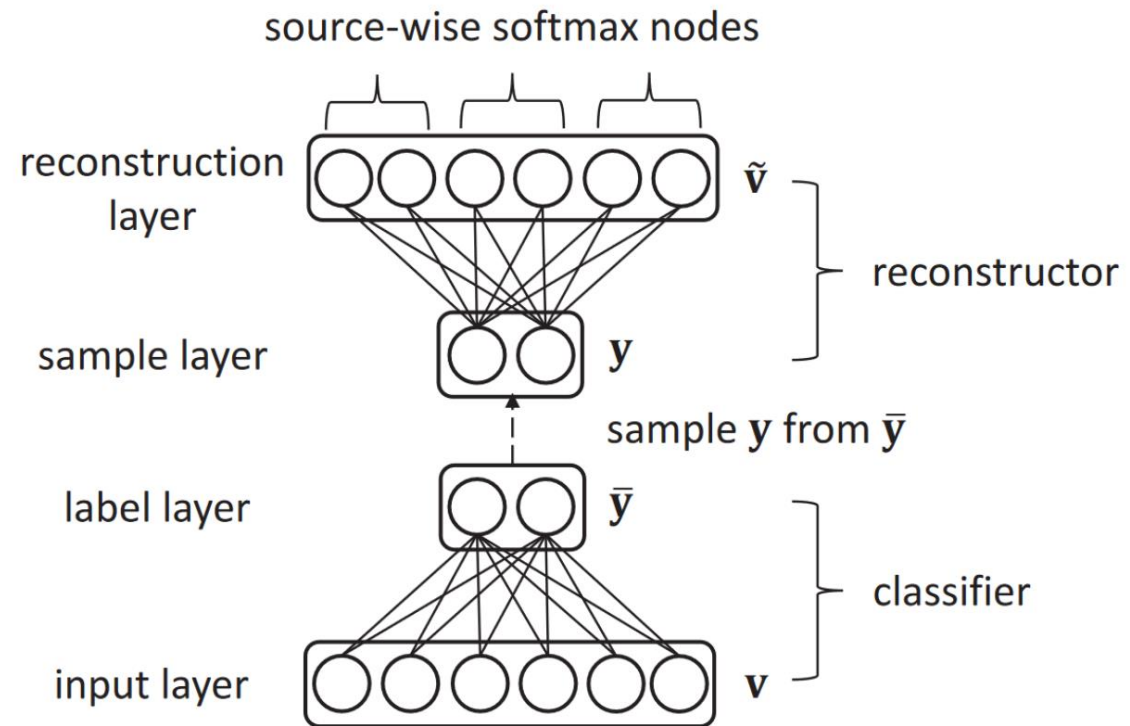
$$\beta_k = \frac{\sum_{m \in O_k} (1 - \omega_m) \cdot (1 - x_m^k)}{\sum_{m \in O_k} (1 - \omega_m)},$$

Aggregation & Inference algorithm

- Majority Voting (MV)
 - MV
 - Penalty Based MV
- Probabilistic Graphical Model (PGM):
 - EM (PGM, Confusion matrix)
 - BCC (PGM, Confusion matrix)
 - KOS (PGM, probability)
 - GLAD (PGM, probability, consider task's difficulty level)
- Neural Network
 - Variational Autoencoder

Label-Aware Autoencoder

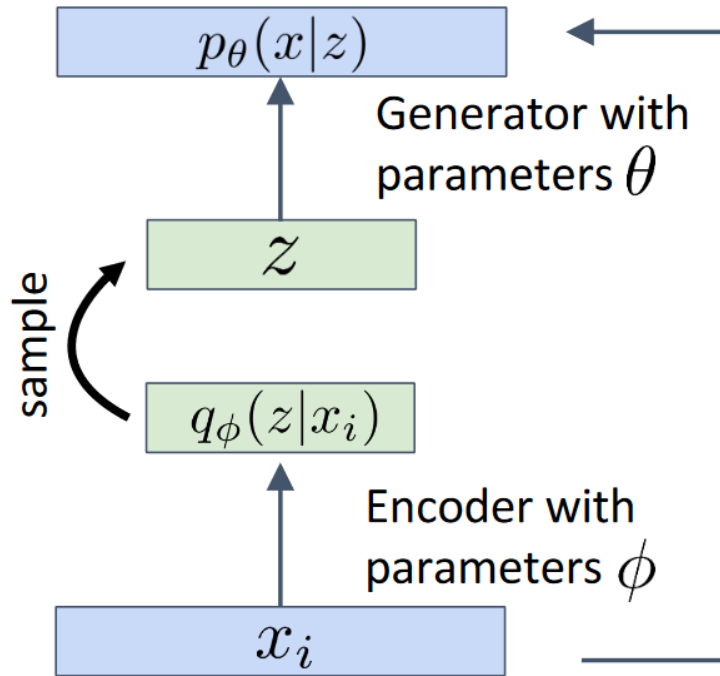
- Classifier as an encoder $q_{\theta}(y|v)$ 
- Reconstruct as a decoder $p_{\phi}(v|y)$ 
- Inferred labels as latent features of input
- Goal: maximizing the lower bound of the log-likelihood of input.



Label-Aware Autoencoder

Loss function:

$$-\log p_{\theta}(x_i|z) + KL(q_{\phi}(z|x_i) \parallel p(z))$$



- Can we still easily sample a new z ?
- Need to make sure $q_{\phi}(z|x_i)$ approximates $p(z)$
- Regularize with **KL-divergence**
- Negative loss can be shown to be a lower bound for the likelihood, and equivalent if $q_{\phi}(z|x) = p_{\theta}(z|x)$

Label-Aware Autoencoder

$$\log p(v) = \sum_{y=1}^K q_{\theta}(y|v) \log \frac{p(y, v)}{q_{\theta}(y|v)} + D_{KL}(q_{\theta}(y|v) || p(y|v))$$

$$\geq \sum_{y=1}^K q_{\theta}(y|v) \log \frac{p_{\phi}(v|y)p(y)}{q_{\theta}(y|v)}$$

$$= \underbrace{\mathbb{E}_{q_{\theta}(y|v)} \log p_{\phi}(v|y)}_{\text{Measures the expectation of reconstruction quality}} - \underbrace{D_{KL}(q_{\theta}(y|v) || p(y))}_{\text{Measures the expectation of reconstruction quality}}.$$

Distribution of inferred label

Prior distribution

Act as regularization

Which model works better under poisoning
attack?

Chosen Aggregation & Inference algorithm

- Majority Voting (MV)
 - MV
 - Penalty Based MV
- Probabilistic Graphical Model (PGM):
 - EM (PGM, Confusion matrix)
 - BCC (PGM, Confusion matrix)
 - KOS (PGM, probability)
- Neural Network
 - Variational Autoencoder

Attack Framework

Goal of the attack:

Targeted: only focus on a specific task or small set of tasks and flip their labels

Untargeted: Goal is to decrease the accuracy of system

Strategy of attack:

Heuristic behavior based on confusion matrix and disguise

Optimized behavior formulate the attack as an optimization problem

Type of attack:

Modification: Modify the provided labels by normal workers

Injection: Inject the new labels to crowdsourcing system

Power of attack:

Percentage of malicious workers

Perception of ground truth by attackers

Percentage of available normal users rating

Attack Scenarios

- Heuristics based:
 - **Goal** of the attack: Untargeted, attack on availability of system
 - **Type** of attack: Injection data
 - **Strategy** of attack: Heuristically set the behavior of attackers on confusion matrix and disguise parameters.
 - Confusion matrix: Indicates the reliability of attackers
 - Disguise: Percentage of times attackers behave normally to hide their malicious intent.
 - **Power** of attack:
 - Attacker does not have knowledge of other reports
- Optimization based
 - Attacker does have knowledge of other reports

Experiments

- Experiments on two real dataset:
 - Product
 - PosSent
- Effect of these parameters on accuracy:
 - Percentage of malicious workers
 - Level of disguise

Dataset

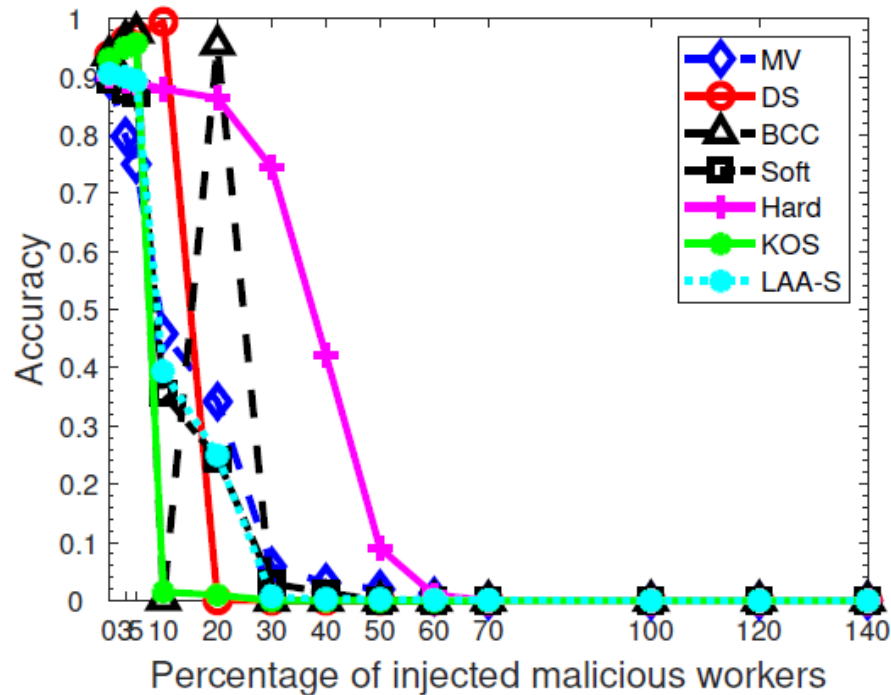
Dataset	#tasks (n)	#truth	$ V $	$ V /n$	$ W $
<i>Datasets for Decision-Making Tasks</i>					
D_Product	8,315	8,315	24,945	3	176
D_PosSent	1,000	1,000	20,000	20	85

Dataset	Balanced/Unbalanced (B/U)	Degree of Sparsity
D_product	U	3
D_posSent	B	20
Synthetic	B	3
Synthetic	B	10

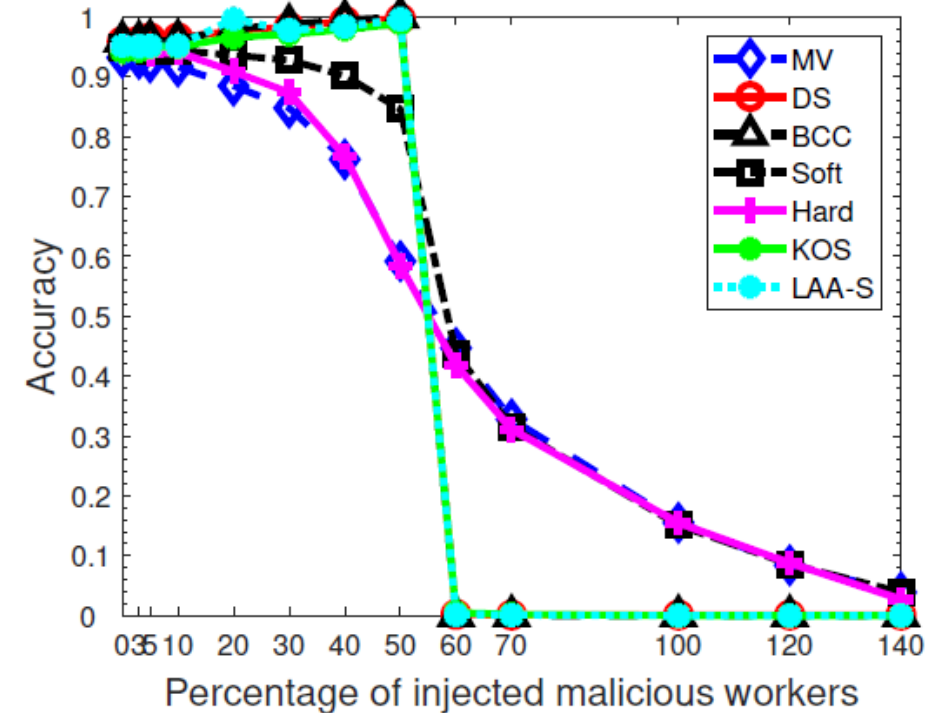
Dataset	Description
D_Proudect	Each task contains two products and two choices (T, F) Sony Camera CarryingLCSMX100 and Sony LCS-MX100 Camcorder are the same?
D_PosSent	Each task contains a tweet related to a company and two choices (Y, N) “The recent products of Apple is amazing!”

Impact of the percentage of the malicious workers

Product Dataset



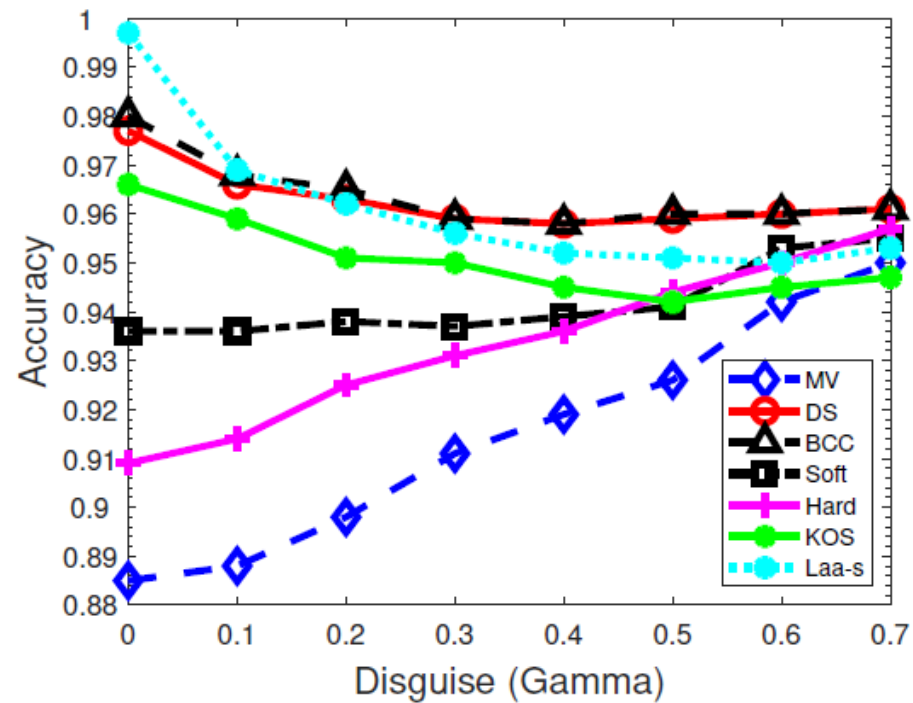
PosSent Dataset



Untargeted Attack, **Injection Data**, Heuristics based

Impact of the disguise parameter

PosSent Dataset



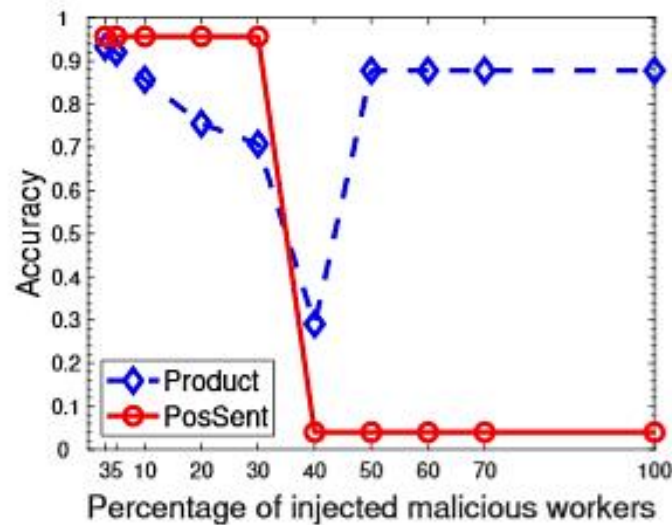
Untargeted Attack, **Injection Data**, Heuristics based

Optimization based Attack

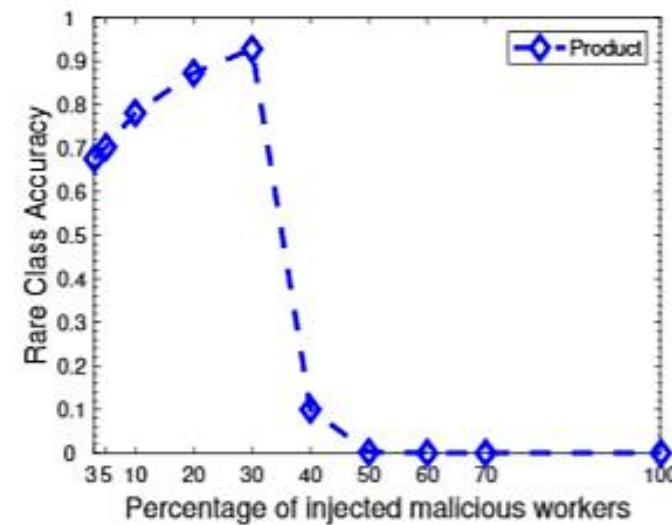
The attacker conducts the data poisoning attacks for the purpose of **maximizing the error** of the final aggregated results, and at the same time tries to **disguise his attack** behaviors as much as possible.

$$\begin{aligned} \max_{\tilde{X}} \quad & \sum_{m=1}^M \mathbb{1}(x_m^{*a} \neq x_m^{*b}) + \lambda \sum_{k'=1}^{K'} (\tilde{\alpha}_{k'} + \tilde{\beta}_{k'}) \\ \text{s.t.} \quad & \{X^{*a}, \tilde{\Theta}\} = \arg \max_{X^{*a}, \tilde{\Theta}} \log L(\tilde{\Theta}; \hat{X}, X^{*a}) \\ & \{\tilde{x}_m^{k'}\}_{m, k'=1}^{M, K'} \in \{0, 1\} \end{aligned}$$

Impact of the percentage of the malicious workers



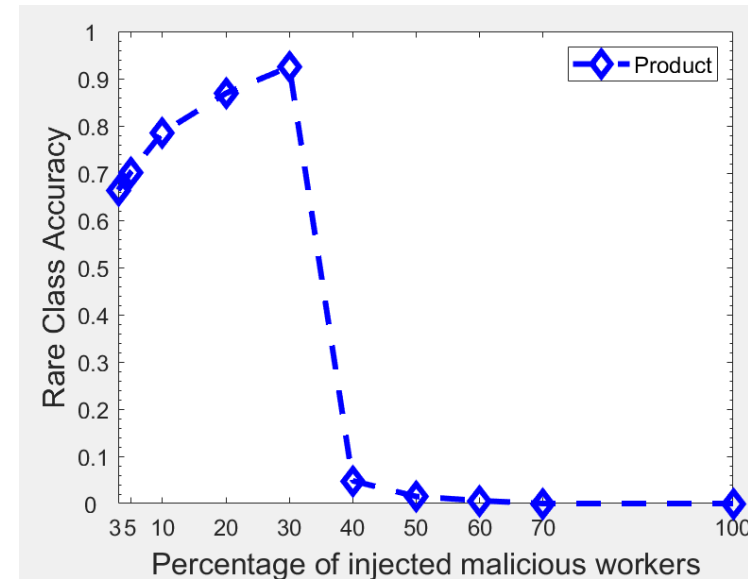
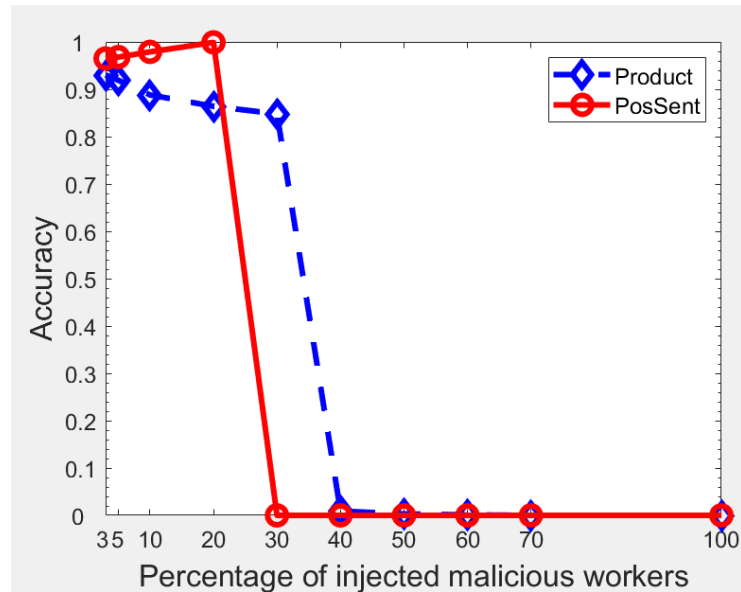
(a)



(b)

Untargeted Attack, Injection Data, Optimized based, Inference Algorithm : **EM**, having others reports

Impact of the percentage of the malicious workers



Untargeted Attack, Injection Data, Optimized based, Inference Algorithm : **BCC**, having others reports

Discussion

- We still need to find a robust inference method
- We found that MV-Hard penalty works very well in the case of high number of malicious users.
- In presence of few adversaries, there is not much difference in terms of performance across different methods.
- Combining MV-Hard and one of the EM based algorithms will yield the highest accuracy for inferring the truth.

?

Thanks