OpenCrowd: A Human-AI Collaborative Approach for Finding Social Influencers via Open-Ended Answers Aggregation

Inès Arous, Jie Yang, Mourad Khayati, Philippe Cudré-Mauroux

22 April 2020

UNI FR
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

eXascale Infolab

Contents

Importance of Finding Social Influencers

Finding influencers helps to reach a new audience:

- **Brand marketing**: 40% of Twitter users have made a purchase as a direct result of a tweet from an influencer[1].
- **Fundraising**: The followers of *Chiara Ferragni* raised over 2M Euros in fight against COVID-19 in half a day after her post on Instagram.
- **Analysing presidential elections**: Finding social influencers helps US presidential candidates reach an audience that they might not otherwise be able to connect to. (Forbes 2020)

[1] https://blog.twitter.com/en_us/a/2016/new-research-the-value-of-influencers-on-twitter.html

Categories of Social Influencers

Macro-Influencers

Chiara Ferragni
426.1k **Followers** 80 **Following** 45k **Tweets**

#LancomexChiaraFerragni now available

- Broad audience (>10k followers).
- Post new content regularly.
- Partnership with luxurious brands.

Micro-Influencers

Monroe Steele
3033 **Followers** 1749 **Following** 30.9k **Tweets**

Loving this vegan leather blazer. It is on sale!

- Niche audience (<5k followers).
- Post new content regularly.
- Communicate with her followers through comments.

Identifying Social Influencers

- Network properties: # active neighbors, authority score, Pagerank
  - Network properties can help detect influencers[2].
  - The structure of the network is **hard to obtain**.
- Social Features: # Followers, #Followings, # and content Tweets
  - Macro-influencers can be detected using a combination of social features[3].
  - Such a combination is, however, **not discriminative** for micro-influencers.
  - When used together with ML, we need a large number of expert labels.

[2] Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K. and Tang, J., *Deepinf: Social influence prediction with deep learning*. KDD 2018

[3] Wei, W., Cong, G., Miao, C., Zhu, F. and Li, G.,. *Learning to find topic experts in twitter via different relations*. TKDE 2016

Crowdsourcing for Influencer Finding

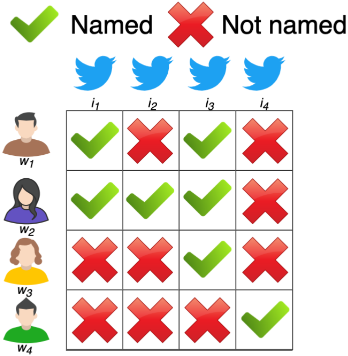
Could crowdsourcing help us to identify influencers?

- Yes, by asking the crowd to name influencers in a predefined domain.
  - + A cost-effective way to find a **large** number of social influencers.
  - + Exploit the broad knowledge of crowds through the **open-ended** task.
  - + Suitable to collect micro-influencers.

Challenges

- Workers have different reliability $\Rightarrow$ Low quality results.
- All answers are deemed as relevant by workers $\Rightarrow$ Lack of negative examples for ML model training.
- The names are given freely $\Rightarrow$ Infinite pool of answers.

Truth Inference for Open-ended Tasks



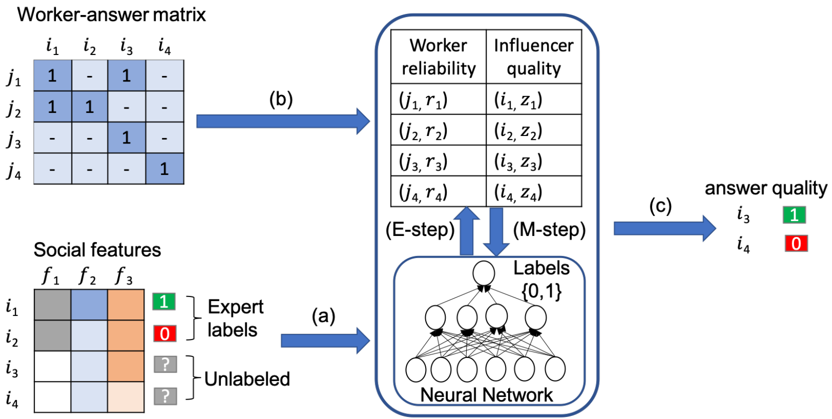
- Are answers given by more workers more likely to be correct?
- How to model worker's reliability when all answers are deemed as positive?
- Can we quantify our confidence in worker's reliability when they have different number of answers?

Contributions

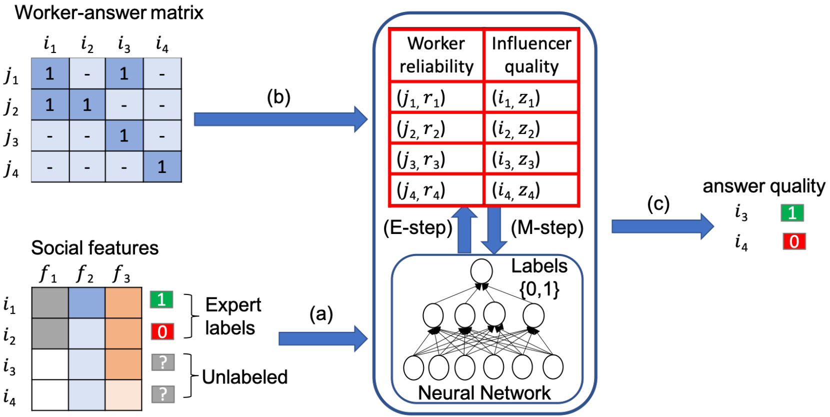
- We propose a human-AI collaborative approach, OpenCrowd, a Bayesian framework for open-ended answers aggregation.
- We derive an efficient learning algorithm based on Variational Inference to estimate answer quality and worker reliability.
- We demonstrate on two domains that OpenCrowd improves SOA by **11.5%** AUC.

Contents

OpenCrowd Framework



OpenCrowd Framework



Generative Model

- We denote the worker reliability as r_j and define it as a continuous distribution: $r_j \sim Beta(A, B)$
- We denote the quality of an answer as z_i and define it as a binary distribution: $z_i \sim Ber(\theta_i)$
- Given a worker-answer matrix $\mathbf{A}$, a reliable worker has a higher likelihood of naming a real influencer, i.e.,

$$p(\mathbf{A}_{i,j}|z_i, r_j) = {r_j}^{\mathbb{1}[z_i = \mathbf{A}_{i,j}]}(1 - r_j)^{\mathbb{1}[z_i \neq \mathbf{A}_{i,j}]}$$

Algorithm

Algorithm 1 Coordinate Ascent Variational Inference

Input : **A**, social features, expert labels

Output : answer quality dist. $q(z_i)$, worker reliability dist. $q(r_j)$

repeat
- E-step:
  - **for** *all answers i* **do**
    - update $q(z_i)$ using answer quality inference rule;
  - **for** *all workers j* **do**
    - update $q(r_j)$ using worker reliability inference rule;
- M-step:
  - **for** *all answers i* **do**
    - Update weights of social features via standard gradient descent;

until *convergence*;

Answer Quality Inference

- α_j and β_j: worker's reliability parameters
- θ_i: output of the answer quality model

Answer Quality

The true label distribution $q(z_i = 1)$ can be incrementally computed using θ_i and r_j's parameters.

$$q(z_i = 1) = \overset{\text{Answer Quality}}{\boxed{\theta_i}} \times \underbrace{\prod_{j \in \mathcal{J}_i} \exp\left\{\Psi(\beta_j) - \Psi(\alpha_j + \beta_j)\right\}}_{\text{Geo. Mean of the reliability}}$$

Worker Reliability Inference

- α_j and β_j: worker's reliability parameters.
- θ': Label of answers given by the worker.

Worker Reliability

The reliability of workers can be computed by weighting the number of **correct** and **wrong** answers with **the reliability parameters**.

$$q(r_j) = Beta(\alpha_j + \overbrace{\boxed{\sum_{i \in \mathcal{I}_j} \theta'}}^{\#\text{ of correct answers}}, \beta_j + \underbrace{\boxed{\sum_{i \in \mathcal{I}_j} (1 - \theta')}}_{\#\text{ of wrong answers}})$$

Contents

- Task: we ask workers to give Twitter usernames of social influencers in two domains Fashion and Information Technology.

Dataset	#Cand. Infl.	#Workers	#Answers	Sparsity
Fashion	890	250	1416	99.36%
InfoTech	1057	200	1643	99.22%

- Ground Truth: we follow the guidelines given by companies that connect brands to social influencers and label 40% of the cand. infls.

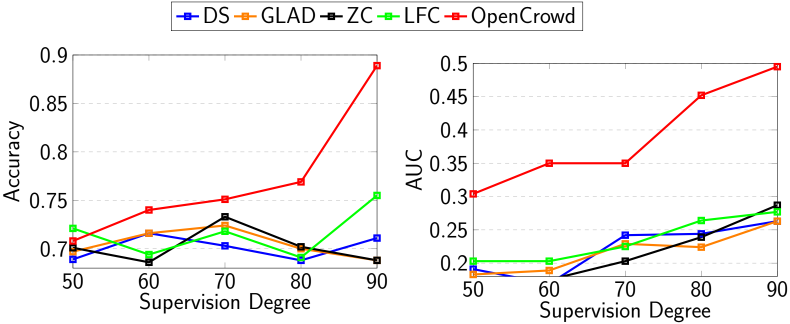
- Metrics: we use accuracy and area under the precision-recall curve (AUC)[4].

[4] *Saito, T. and Rehmsmeier, M.*, The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PloS one 2015

Comparison with Boolean Aggregation Methods

- We compare against Boolean aggregation methods that take into account:
  - the worker reliability: Dawid Skene, ZenCrowd (WWW'12)
  - the worker reliability with priors: LFC (JMLR'10)
  - the task difficulty: GLAD (NIPS'09)
- Supervision Degree: The percentage of expert labels used.
- All methods are used in a semi-supervised setting.

Comparison with Boolean Aggregation on Fashion

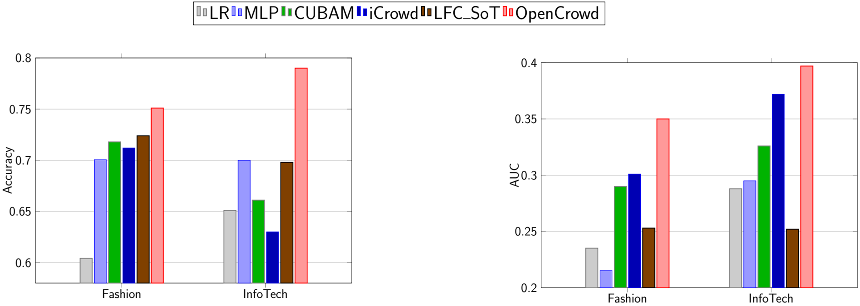


- OpenCrowd improves the SOA by **7%** accuracy and **62%** AUC.
- The ML model in OpenCrowd leverages the similarity between the candidate influencers to propagate correct labels.
- **87.5%** of the real fashion influencers are discovered through OpenCrowd.

Comparison with Feature-based Methods

- Feature-based aggregation methods that take into account in addition to worker reliability:
  - the task **clarity**: LFC_SoT (KDD'12)
  - the task **domain**: CUBAM (NIPS'10)
  - the tasks **topical similarity**: iCrowd (SIGMOD'15)
- Social Influencer Finding methods (feature-based): Logistic Regression (LR), Multi-Layer Perceptron (MLP)

Comparison with Feature-based Methods



- OpenCrowd outperforms the second best method by **8.44%** accuracy and **11.5%** AUC on average.
- It's easier for OpenCrowd to find IT influencers because workers know more IT than fashion influencers.

Contents

- We introduced OpenCrowd a human-AI collaborative approach that combines social features with worker reliability to accurately identify social influencers.
- OpenCrowd is easily generalizable to solve any open-ended answers aggregation problem.
- OpenCrowd substantially improves the state of the art by 11.5% AUC.

Thanks for your attention!
Any questions?

OpenCrowd

https://bit.ly/2wvuh4c

eXascale Infolab

https://exascale.info/