Lightweight Privacy-Preserving Task Assignment in Skill-Aware Crowdsourcing

Louis Béziaud, Tristan Allard, David Gross-Amblard

Lightweight Privacy-Preserving Task Assignment in Skill-Aware Crowdsourcing

Louis Béziaud[1,3], Tristan Allard[1,2], and David Gross-Amblard[1,2]

[1] Univ. Rennes 1, France
[2] IRISA, France, first.last@irisa.fr
[3] ENS Rennes, France, first.last@ens-rennes.fr

Abstract. Crowdsourcing platforms dedicated to work are used by a growing number of individuals and organizations, for tasks that are more and more diverse, complex, and that require very specific skills. These highly detailed worker profiles enable high-quality task assignments but may disclose a large amount of personal information to the central platform (*e.g.,* personal preferences, availabilities, wealth, occupations), jeopardizing the privacy of workers. In this paper, we propose a lightweight approach to protect workers privacy against the platform along the current crowdsourcing task assignment process. Our approach (1) satisfies differential privacy by letting each worker perturb locally her profile before sending it to the platform, and (2) copes with the resulting perturbation by leveraging a taxonomy defined on workers profiles. We overview this approach below, explaining the lightweight upgrades to be brought to the participants. We have also shown (full version of this paper [1]) formally that our approach satisfies differential privacy, and empirically, through experiments performed on various synthetic datasets, that it is a promising research track for coping with realistic cost and quality requirements.

Keywords: Crowdsourcing; Task assignment; Differential privacy; Randomized response

1 Introduction

Crowdsourcing platforms are disrupting traditional work marketplaces. Their ability to compute high-quality matchings between tasks and workers, instantly and worldwide, for paid or voluntary work, has made them unavoidable actors of the 21^{st} century economy. Early crowdsourcing platforms did not (and still do not) require strong and specific skills; they include for example Amazon Mechanical Turk[4] (for online micro-tasks), Uber[5] (for car-driving tasks), or TaskRabbit[6] (for simple home-related tasks—*e.g.,* cleaning, repairing). Today's crowdsourcing platforms now go one step further by addressing skill-intensive contexts (*e.g.,*

[4] https://www.mturk.com/
[5] https://www.uber.com/
[6] https://www.taskrabbit.com/

general team building[7], collaborative engineering[8]) through the collection and use of fine-grained worker profiles. Such platforms carry the promise to facilitate, fasten, and spread innovation at an unprecedented scale.

However abusive behaviors from crowdsourcing platforms against workers are frequently reported in the news or on dedicated websites, whether performed willingly or not (see, *e.g.,* the privacy scandals due to illegitimate accesses to the geolocation data of a well-known drivers-riders company[9], or the large-scale exposure of workers' identifying and sensitive information—*e.g.,* real name, book reviews, or wish-list—through Amazon Mechanical Turk IDs [8]). The problem is even more pregnant with skill-intensive crowdsourcing platforms since they collect detailed workers' profiles for computing highly accurate matchings (*e.g.,* demographics, encompassive set of skills, detailed past experiences, personal preferences, daily availabilities, tools possessed). We advocate thus for a sound protection of workers' profiles against illegitimate uses: in addition to the necessary compliance with fundamental rights to privacy, it is a precondition for a wide adoption of crowdsourcing platforms by individuals.

Computing the assignment of tasks to workers is the foundamental role of the platform (or at least facilitating it). This paper considers precisely the problem of computing a high-quality matching between skill-intensive tasks and workers while preserving workers' privacy. To the best of our knowledge, this problem has only been addressed by a single recent work [6]. However, this work is based on costly homomorphic encryption primitives which strongly hamper its performances and prevent it to reason about skills within the assignment algorithm (*e.g.,* no use of semantic proximity).

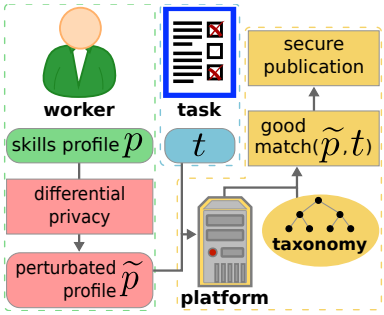

Fig. 1: Our Approach to Privacy-Preserving Task Assignment

We propose an approach (see Fig. 1) that addresses these issues by making the following contributions:

1. A simple skills model for a worker's profile: a bit vector and a taxonomy.
2. An algorithm run independently by each worker for perturbing her profile locally before sending it to the platform. By building on the proven *randomized response* mechanism [3,11], this algorithm is privacy-preserving (provides sound *differential privacy* guarantees [5]), and lightweight (no cryptography, no distributed computation, only bitwise operations).
3. A suite of weight functions to be plugged in a traditional assignment algorithm run by the platform and dedicated to increase the quality of matchings

[7] `https://tara.ai/`
[8] `https://makake.co/`
[9] `https://tinyurl.com/wp-priv`

performed over perturbed profiles. Our weight functions reduce the impact of the perturbation by leveraging the skills taxonomy, vertically and horizontally, averaging the skills according to their semantic proximity in order to reduce the variance of the differentially-private perturbation. The variance reduction is mathematically sound and does not jeopardize privacy.

4. An experimental study (see the full version [1]), over a synthetic taxonomy and various synthetic datasets, that shows promising preliminary results about the practical adequacy of our approach from the sides of performance and quality.

For space reasons, we give in this paper an overview of our approach. We refer the interested reader to the full version of our work [1] that describes our approach in details, presents its experimental results, and positions it with respect to related work.

The rest of the paper is organized as follows. Section 2 introduces the notions used in our approach and defines more precisely the problem we tackle. We overview our algorithms in Section 3 and conclude in Section 4 outlining interesting future works.

2 Problem Definition

Skills and Participants The set of skills that can be possessed by a worker (*resp.* requested by a task) is denoted $\mathcal{S}$. A worker's profile $p_i \in \mathcal{P}$ (*resp.* a task $t_i \in \mathcal{T}$) is represented by a bit vector, *i.e.,* $p_i = \{0,1\}^{|\mathcal{S}|}$, where each bit corresponds to a skill $s_j \in \mathcal{S}$ and is set to 1 if the given worker has the given skill (*resp.* the given task $t_i = \{0,1\}^{|\mathcal{S}|}$ requests the given skill). Without loss of generality, we consider that each requester has a single task and that the number of workers and requesters is the same (*i.e.,* $|\mathcal{P}| = |\mathcal{T}|$). Furthermore, we assume that a skills taxonomy $\mathcal{S}_T$ exists[10][9], structuring the skills according to their semantic proximity, and is such that the skills in $\mathcal{S}$ are the leaves of $\mathcal{S}_T$ (*i.e.,* no non-leaf node can be possessed nor requested). The non-leaf nodes of the taxonomy are called *super-skills*.

The platform is essentially in charge of intermediating between workers and requesters. The workers' profiles are considered private while the requesters' tasks are not. The platform holds the set of workers' profiles, *perturbed* to satisfy differential privacy (defined below) and denoted $\widetilde{\mathcal{P}}$, as well as the exact set of requesters' tasks $\mathcal{T}$. All participants, *i.e.,* workers, requesters, and the platform, are considered to be *honest-but-curious*. This means that they participate in the protocol without deviating from its execution sequence (*e.g.,* no message tampering, no data forging) but they will try to infer anything that is computationally-feasible to infer about private data (*i.e.,* the set of workers' non-perturbed profiles $\mathcal{P}$).

[10] In practice, skills taxonomies concerning numerous real-life contexts exist today (see, *e.g.,* the Skill-Project `http://en.skill-project.org/skills/`, or Wand's taxonomies `http://www.wandinc.com/wand-skills-taxonomy.aspx`).

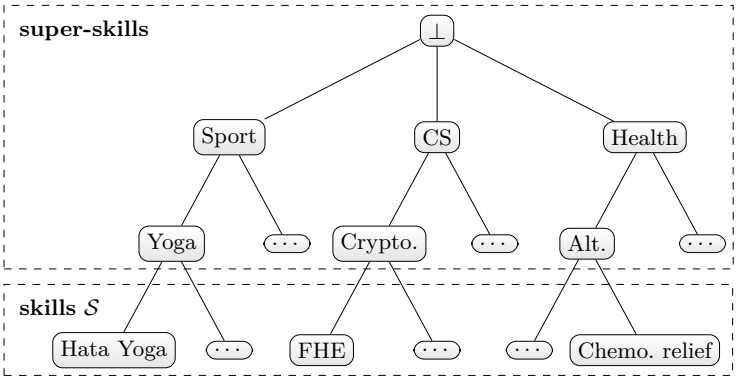

Fig. 2: Example of a skill taxonomy $\mathcal{S}_T$

The Traditional Tasks-to-Workers Assignment Problem In a traditional context, where workers' profiles are not considered private, the objective of the crowdsourcing platform is to assign a worker to each task such that the overall expected quality is maximized. This well-known combinatorial optimization problem is referred as the assignment problem and can be expressed as a standard linear problem [7] (assuming $|\mathcal{T}| = |\mathcal{P}|$). Assignment algorithms rely on a *weight function* $\mathtt{C}\colon \mathcal{T} \times \mathcal{P} \to \mathbb{R}$ in charge of defining the cost of each assignment, *i.e.,* the divergence between the requirements vector of a task and the skills vector of a worker. Common weight functions include the usual distance metrics (*e.g.,* Hamming distance) or disimilarities (*e.g.,* cosine distance). Since our approach is independent from the algorithm, we simply use the HUNGARIAN method [7], a standard academic choice.

Security We say that our approach is secure against honest-but-curious participants if and only if no participant learns information about the set of non-perturbed profiles $\mathcal{P}$ that has not been perturbed by a differentially-private mechanism, where differential privacy [4] - the current *de facto* standard model for disclosing personal information while satisfying sound privacy guarantees - is defined below. Differential privacy is self-composable [10] and secure under post-processing [5].

Definition 1 (Differential Privacy [4]). *A randomized mechanism* $\mathtt{M}$ *satisfies* ϵ*-differential privacy with* $\epsilon > 0$ *if for any possible set of workers' profiles* $\mathcal{P}$ *and* $\mathcal{P}'$ *such that* $\mathcal{P}'$ *is* $\mathcal{P}$ *with one additional profile (or one profile less), and any possible set of output* $O \subseteq \mathit{Range}(\mathtt{M})$*,*

$$\Pr[\mathtt{M}(\mathcal{P}) \in O] \leq e^{\epsilon} \times \Pr[\mathtt{M}(\mathcal{P}') \in O]. \tag{1}$$

Quality The inherent information loss due to the differentially-private perturbation impacts the quality of the worker-to-task assignment. This is the price to pay to satisfy a sound privacy model. We quantify this impact by measuring the relative increase of the assignment cost as well as the fraction of profiles that have *all* the skills required by the task to which they are assigned (see the full version for formal definitions [1]).

3 A Flip-based Approach

This section overviews our approach. We first focus on the workers' side: we describe the algorithm that we propose for perturbing each worker's profile, show its adequacy to our context, and demonstrate that it complies with our security model. Second, we shift to the platform's side. We explain how to reduce the impact of the differentially private perturbation (while still satisfying differential privacy) and we describe the assignment algorithm based on perturbed profiles. Finally, we overview technical means for letting workers fetch their assignment in a secure way in order to complete it. We refer the interested reader to the full version [1] for more details (including the formal proofs).

3.1 At a Worker's Side: Local Perturbation

Building Block: Randomized Response Randomized response [11] is a simple though powerful perturbation mechanism shown to satisfy differential privacy (see below). Basically, it inputs a single bit (*e.g.,* the answer of an individual to a sensitive boolean question) and flips it randomly according to a well-chosen distribution probability. We describe below the variant called *innocuous question* that we use in this paper and show that it satisfies differential privacy[11]. Let $x \in \{0, 1\}$ be a private value. The randomized response mechanism simply outputs the perturbed value of x, denoted $\widetilde{x}$, as follows:

$$\widetilde{x} = \begin{cases} x & \text{with probability } 1 - \Pr_{flip} \\ 1 & \text{with probability } \Pr_{flip} \times \Pr_{inno} \\ 0 & \text{with probability } \Pr_{flip} \times (1 - \Pr_{inno}) \end{cases}$$

where Pr_{flip} depends on ϵ (see below) and $Pr_{inno} \in [0, 1]$. We use $\Pr_{inno} = 0.5$ in the rest of the paper, since it minimizes the variation of the estimated value of x after perturbation [3].

Claim. For a given differential privacy parameter $\epsilon > 0$, and a worker's profile made of a single bit to be flipped, the innocuous question randomized response scheme satisfies ϵ-differential privacy if $\Pr_{flip} = \frac{2}{1+e^{\epsilon}}$ (see [1] for the proof).

Flip Mechanism Our Flip mechanism (Alg. 1) essentially consists in applying the randomized response mechanism to each binary skill of a worker's profile before sending it to the platform and inherits thus its high efficiency. The self-composability properties of differential privacy allow that by distributing ϵ over the bits of the skills vector.

Claim. The Flip mechanism satisfies ϵ-differential privacy (see [1] for the proof).

[11] Any other variant could have been used, provided that it satisfies differential privacy.

Algorithm 1: FLIP (run by each Worker)

Input: The original profile $p = \langle p[1], \ldots, p[l] \rangle$, the differential privacy budget $\epsilon > 0$.

Let $\texttt{Pr}_{flip}$ be the flipping probability: $\Pr_{flip} \leftarrow \frac{2}{1+e^{\epsilon/l}}$.

Initiate the perturbed profile: $\widetilde{p} \leftarrow \langle \widetilde{p}[1] = 0, \ldots, \widetilde{p}[l] = 0 \rangle$.

for $1 \leq i \leq l$ **do**

  $\widetilde{p}[i] \leftarrow \texttt{RandomizedResponse}(p[i], \Pr_{flip})$.

Return *The perturbed profile* $\widetilde{p}$

3.2 At the Platform's Side: Task Assignment

Efficient traditional assignment algorithms do not need any modification to work with perturbed profiles, which are bit vectors, exactly as non-perturbed profiles are. The main question is the impact on quality due to our perturbation, and this impact is naturally related to the weight function $\texttt{C}\colon \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}$ on which assignment algorithms rely. As there is no clear consensus on what is a good weight function for task assignment, in the sequel we recall several reasonable functions, ignoring or using the skill taxonomy. We also propose new weight functions and explain how they could cope with the diffentially-private perturbation.

Existing Weight Functions Numerous weight functions have been proposed as metrics over skills. The *Hamming distance* is a common choice to compute dissimilarity between two vectors of bits but it does not capture the semantics needed for crowdsourcing (*e.g.*, a worker possessing all the skills has a high Hamming distance from a task requiring only one skill, although he is perfectly able to perform it). The weight function proposed in [9] adresses this problem based on a taxonomy. We slightly adapt it and call it the `Ancestors` weight function (`AWF` for short).

Definition 2 (`Ancestors` Weight Function (adapted from [9])). *Let d_{max} be the maximum depth of the taxonomy $\mathcal{S}^T$. Let $\text{lca}(s, s') \in \mathcal{S}^T$ be the lowest common ancestor of skills s and s' in the taxonomy.*

$$\texttt{AWF}(t, \widetilde{p}) = \sum_{s_i \in \widetilde{p}} \min_{s_j \in t} \left(\frac{d_{max} - \text{depth}(\text{lca}(s_i, s_j))}{d_{max}} \right) \tag{2}$$

Naive Skill-Level Weight Functions The `Missing` weight function (`MWF` for short) between a worker and a task revisits the Hamming distance. It settles a task-to-worker assignment cost that is intuitive in a crowdsourcing context: it is defined as the fraction of skills required by the task that the worker does not have (see Definition 3).

Definition 3 (`Missing` Weight Function (`MWF`)). $\texttt{MWF}\colon \mathcal{T} \times \widetilde{\mathcal{P}} \rightarrow \mathbb{R}$ *is defined as follows:*

$$\texttt{MWF}(t, \widetilde{p}) = \sum_{\forall i} t[i] \wedge \neg \widetilde{p}[i] \tag{3}$$

Leveraging the Taxonomy In realistic contexts, the differentially private perturbation may overwhelm the information contained in the original profiles and make the perturbed profiles be close to uniformly random bit vectors. We cope with this challenging issue by building on the taxonomy $\mathcal{S}_{\mathtt{T}}$. Indeed, the taxonomy allows to group large numbers of skills according to their semantic proximity and to reduce the variance of the perturbation by using group averages [2].

Climbing Weight Function The `Climbing` weight function (`CWF` for short) leverages *the vertical relationship* given by the taxonomy by averaging, for each profile, the skills along the root-to-leaf paths. In other words, before performing the assignment, the platform converts each perturbed profile into a tree, *i.e.,* the same as the taxonomy S_T, and for each node n of the tree, it computes the mean of the skills that appear below n (interpreting the boolean values 1 and 0 as integers). For a given node, this mean is actually a rough estimator of the fraction of descendant skills possessed. We call it *score* below. During an assignment, given a task and a perturbed profile, the `Climbing` weight function consists essentially in computing the distance between the scores vector of the task and the scores vector of the profile at each level. Definition 4 formalizes the `Climbing` weight function.

Definition 4 (`Climbing` Weight Function (`CWF`)). *Let v_i (*resp. *u_i) denote the scores vector at level i in the tree corresponding to the profile $\widetilde{p}$ (*resp. *to the task t), and $\mathtt{d}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a classical distance function on real-valued vectors (*e.g., *`Cosine`). Then, $\mathtt{CWF}\colon \mathcal{T} \times \widetilde{\mathcal{P}} \to \mathbb{R}$ is defined as follows:*

$$\mathtt{CWF}(t, \widetilde{p}) = \sum_{\forall i} i \times \mathtt{d}(u_i, v_i) \tag{4}$$

Touring Weight Function The `Touring` weight function (`TWF` for short) leverages *the horizontal relationship* given by the taxonomy, *i.e.,* the neighbouring proximity degree between skills. As described in Definition 5, it returns the average path-length—according to the taxonomy $\mathcal{S}_T$—between the skills required by the task and the skills of the worker's profile. The expected variance reduction comes from the average path-length of the full cartesian product between the skills required by a task and the skills set to 1 in a perturbed profile. The reduction depends on the taxonomy (similarly to `Climbing`) and on the number of skills averaged.

Definition 5 (`Touring` Weight Function (`TWF`)). *Let $\rightsquigarrow$ denote the path-length operator between two skills in the taxonomy $\mathcal{S}_T$. Then, $\mathtt{TWF}\colon \mathcal{T} \times \widetilde{\mathcal{P}} \to \mathbb{R}$ is defined as follows:*

$$\mathtt{TWF}(t, \widetilde{p}) = \frac{\sum_{\forall i} \sum_{\forall j} (t[i] \wedge \widetilde{p}[j]) \times (s_i \rightsquigarrow s_j)}{\sum_{\forall i} \widetilde{p}[i] \times \sum_{\forall i} t[i]} \tag{5}$$

3.3 Post-Assignment Phase

Workers need a secure way to fetch their own assignment. This can be solved easily by well-known technical means. For example, the platform could post the

assignments on the Web (*e.g.,* to a dedicated webpage for each perturbed profile) so that each worker would then access it through a secure web browser (*e.g.,* TOR[12]).

4 Conclusion

We have overviewed in this paper a lightweight privacy-preserving approach to the problem of assigning tasks to workers. Our approach allows each worker to perturb her skill profile locally in order to satisfy the stringent differential privacy model without any need for additional communication or computation cost. We have proposed novel weight functions that can be easily plugged in traditional centralized assignment algorithms, and that are able to cope with the differentially private perturbation by leveraging the presence of a skill taxonomy. Additionally, promising preliminary results of experiments performed over a synthetic taxonomy and synthetic datasets are presented in the full version of the paper [1]. Future works include consolidating experiments (*e.g.,* more profiles and tasks, alternative quality measures), use the taxonomy during the Flip mechanism, collecting a large-scale skill dataset, and continue exploring the performance/quality tradeoff by designing other profile perturbation strategies (*e.g.,* collaborative perturbation protocols designed to minimize the perturbation).

References

1. L. Béziaud, T. Allard, and D. Gross-Amblard. Lightweight Privacy-Preserving Task Assignment in Skill-Aware Crowdsourcing [Full Version]. working paper or preprint, 2017.
2. I.-J. Bienaymé. *Considérations à l'appui de la découverte de Laplace sur la loi de probabilité dans la méthode des moindres carrés.* Imprim. Mallet-Bachelier, 1853.
3. G. Blair, K. Imai, and Y.-Y. Zhou. Design and analysis of the randomized response technique. *Journal of the American Stat. Assoc.*, 110(511):1304–1319, 2015.
4. C. Dwork. Differential privacy. In *Proc. of ICALP '06*, pages 1–12, 2006.
5. C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.
6. H. Kajino. *Privacy-Preserving Crowdsourcing.* PhD thesis, Univ. Tokyo, 2016.
7. H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
8. M. Lease, J. Hullman, J. P. Bigham, M. S. Bernstein, J. Kim, W. Lasecki, S. Bakhshi, T. Mitra, and R. C. Miller. Mechanical turk is not anonymous. *SSRN Electronic Journal*, 2013.
9. P. Mavridis, D. Gross-Amblard, and Z. Miklós. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *Proc. of WWW '16*, pages 843–853, 2016.
10. F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proc. of ACM SIGMOD '09*, pages 19–30, 2009.
11. S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Stat. Assoc.*, 60(309):63–69, 1965.

[12] `https://www.torproject.org/`