Instance-Privacy Preserving Crowdsourcing

Hiroshi Kajino*
The University of Tokyo

Yukino Baba
National Institute of Informatics
JST, ERATO, Kawarabayashi Large Graph Project

Hisashi Kashima
Kyoto University

Abstract

Crowdsourcing is a technique to outsource tasks to a number of workers. Although crowdsourcing has many advantages, it gives rise to the risk that sensitive information may be leaked, which has limited the spread of its popularity. Task instances (data workers receive to process tasks) often contain sensitive information, which can be extracted by workers. For example, in an audio transcription task, an audio file corresponds to an instance, and the content of the audio (*e.g.*, the abstract of a meeting) can be sensitive information. In this paper, we propose a quantitative analysis framework for the instance privacy problem. The proposed framework supplies us performance measures of instance privacy preserving protocols. As a case study, we apply the proposed framework to an instance clipping protocol and analyze the properties of the protocol. The protocol preserves privacy by clipping instances to limit the amount of information workers obtain. The results show that the protocol can balance task performance and instance privacy preservation. They also show that the proposed measure is consistent with standard measures, which validates the proposed measure.

Introduction

Crowdsourcing is a web-based approach to outsource tasks to a number of unspecified workers. Because it provides an easy way to access abundant human resources at very low cost, it has become a popular method for executing a large amount of tasks that require human intelligence. However, crowdsourcing entails the problem of *instance privacy*[1]: sensitive information contained in task instances can easily leak. A worker who processes tasks must access task instances and may extract sensitive information contained in them. For example, consider a task to transcribe audio recordings of business meetings, where an audio recording corresponds to an instance. The content of such recordings includes confidential information. There are also many other tasks whose instances contain sensitive information, such as a task to digitize analog texts or to detect objects in images. Therefore, there is a strong demand on realizing *instance-privacy preserving crowdsourcing*, where tasks can be processed with high accuracy and low privacy leakage.

To the best of our knowledge, three research groups handled this issue. Varshney (2012) proposed a random perturbation approach: adding perturbations to instances to preserve instance privacy. This was a theoretical study aiming at establishing a mathematical model of the random perturbation approach. It was not verified in a real experiment, and it lost some generality because its analysis specialized in the random perturbation approach. Little and Sun (2011) and Chen et al. (2012) dealt with a human OCR task whose objective was to digitize handwritten forms. They assumed that each form could be decomposed into items perfectly using a template, and therefore, the methods lost some generality. In addition, they did not evaluate the privacy aspect of their method. To summarize the existing work, there exists no work evaluating the privacy aspect quantitatively in a real crowdsourcing environment.

For research on instance-privacy preserving crowdsourcing, it is crucial to evaluate performance of instance privacy preservation protocols. Therefore, we propose a quantitative analysis framework of instance-privacy preserving (IPP) protocols. In general, an IPP protocol preserves instance privacy in exchange for a task performance. For example, adding perturbations to instances or decomposing a form may preserve privacy while they may degrade the quality of task results. Our framework evaluates the trade-off quantitatively using a real crowdsourcing platform, given definitions of a task and privacy and an IPP protocol. Such a framework allows us to check whether an IPP protocol works well for the pair of a task and privacy or to choose a better protocol among multiple ones.

As a case study, we investigate the properties of an *instance clipping (IC) protocol*. The IC protocol preserves instance privacy by clipping instances. It is a generalization of the protocols proposed by Little and Sun (2011) and Chen et al. (2012). The difference from the existing ones is that the IC protocol does not utilize a template of an instance; it clips instances by a window of a fixed size. From a qualitative analysis, the protocol will work effectively for a pair of a task and privacy such that the task can be processed using local parts of instances, and the privacy cannot be invaded through local parts of instances. To investigate the proper-

*JSPS Research Fellow

[1] *Instance privacy* is defined as a state that sensitive information contained in an instance cannot be revealed.

ties quantitatively, we analyze the IC protocol by the proposed framework using a real crowdsourcing platform. The target task is a task to detect the area containing a head given an image as a task instance, which we call a head detection task. The target privacy is defined as contextual information of a person in an image, especially his/her activities. The experimental results show that the IC protocol can balance task performance and instance privacy preservation as expected by a qualitative analysis. We also compare the proposed performance measure with standard measures to show that the proposed measure is consistent with the standard measures, which validates the proposed measure.

In summary, this paper makes two contributions. First, we propose a framework to evaluate an IPP protocol quantitatively. Second, we propose an IC protocol by extending methods by Little and Sun (2011) and Chen et al. (2012) and evaluate its performance empirically using the proposed framework.

Terminology and Settings

There are a *requester* and a *worker* in crowdsourcing. A requester has *instance* $I \in \mathcal{I}$ and wants to obtain *result* $R \in \mathcal{R}$ of performing a *task* on instance I, where $\mathcal{I}$ and $\mathcal{R}$ are sets of possible instances and results, respectively. For example, when a task is to "give yes if an image contains a face, and no otherwise," the image corresponds to an instance, and the label {yes/no} corresponds to a result. A worker performs tasks to obtain a reward on the completion of the tasks.

We consider two processes in crowdsourcing. One is a task execution process defined as follows. First, a requester submits a task with instance I. Second, a worker performs the task on instance I to generate result R. Third, the requester receives result R and rewards the worker. The other is a privacy invasion process. While a worker performs a task, the worker, at the same time, extracts sensitive information from the instance. We denote a set of possible sensitive values as $\mathcal{S}$ and a sensitive value of instance I as $S \in \mathcal{S}$. Then, the privacy invasion process is defined as follows. First, a requester submits a task with instance I. Second, a worker extracts sensitive information S from instance I. Note that the definition of the sensitive information is assumed to be known, *i.e.*, we know which information a worker tries to extract.

Given the definitions of a task and privacy, the problem setting of instance-privacy preserving crowdsourcing is to propose a protocol in which the task can be processed while the privacy cannot be invaded. We aim at proposing a framework to evaluate such an instance-privacy preserving protocol quantitatively.

Quantitative Analysis Framework

The objective of this section is to introduce a framework to evaluate an instance-privacy preserving (IPP) protocol given a pair of a task and privacy. First, we give probabilistic models for the task execution process and the privacy invasion process. Then, we formalize the two processes using the models and define the IPP protocol. Finally, we propose performance measures of it.

Models

We give models of the task execution and the privacy invasion, which share the same modeling approach.

Task Execution Model. A task can be represented as a conditional probability distribution over the set of results given an instance, which we call a *task execution model* (Def. 1). Then, the task execution process can be modeled as a sampling from this model. We assume that only a human can sample from the model, which is a basic assumption in human computation.

Definition 1 (Task execution model). *Let instance I and result R be random variables whose ranges are $\mathcal{I}$ and $\mathcal{R}$. A task execution model is a conditional probability distribution $p_t(R \mid I)$. The execution of the task given instance I is modeled as a sampling from $p_t(R \mid I)$.*

Privacy Invasion Model. Privacy invasion is represented in the same way as the task execution model by regarding a sensitive value as a result, which we call a *privacy invasion model* (Def. 2).

Definition 2 (Privacy invasion model). *Let instance I and sensitive value S be random variables whose ranges are $\mathcal{I}$ and $\mathcal{S}$. A privacy invasion model is a conditional probability distribution $p_p(S \mid I)$. The privacy invasion given instance I is modeled as a sampling from $p_p(S \mid I)$.*

Validity of the Models. Modeling the process as a sampling from a probability distribution can be justified considering that the quality of the results depends on the abilities of workers and the difficulty levels of instances, as often stated (Whitehill et al. 2009). We introduced a probability distribution to capture these diversities. In fact, our model is more general than standard models, *e.g.*, that proposed by Dawid and Skene (1979), in that the details of a process such as the ability of a worker are not explicitly modeled. The standard models can be regarded as a special case of our model if we additionally assume a probability distribution of selecting workers. We would like to stress that the generality has to be ensured in order to keep the applicability of our performance measures introduced later, because the performance measures are built on our models.

Protocols

Based on these models, the processes of crowdsourcing are summarized as Protocol 1, which we call a *Non-Privacy-Preserving (NPP) protocol*. The NPP protocol allows a requester to obtain a sample from $p_t(R \mid I)$ and a worker to obtain a sample from $p_p(S \mid I)$. It does not preserve instance privacy because a worker receives a raw instance.

An *instance-privacy preserving (IPP) protocol* derived from the NPP protocol is defined as a protocol that allows a requester to obtain a sample from $p'_t(R \mid I)$ and a worker to obtain a sample from $p'_p(S \mid I)$. The IPP and NPP protocols share the random variables R, S, and I, but they have different models because of a privacy preservation mechanism.

Protocol 1 Non-Privacy-Preserving (NPP) Protocol

Inputs: instance I, task execution model $p_t(R \mid I)$, and privacy invasion model $p_p(S \mid I)$.
Output of a requester: result R.
Output of a worker: sensitive information S.

1: The requester submits a task with instance I.
2: A worker samples result R from $p_t(R \mid I)$.
3: The worker returns result R to the requester.
4: The worker extracts sensitive information from $p_p(S \mid I)$.

Performance Measures

We propose generally-applicable performance measures of an IPP protocol. The performance should be measured from two perspectives. The task execution model of the IPP protocol should be close to that of the NPP protocol. Also, the privacy invasion model of the IPP protocol should give little about the sensitive information. We introduce two performance measures to capture these two aspects.

Task Information Loss. A *task information loss* is introduced in Def. 3. It captures the information loss incurred by using the IPP protocol instead of the NPP protocol. A small task information loss indicates that the task performance is preserved as compared to the NPP protocol.

Definition 3 (Task information loss). *Given a task execution model of the NPP protocol $p_t(R \mid I)$ and that of the IPP protocol $p'_t(R \mid I)$, the task information loss is defined as*

$$L_t(p'_t, p_t) := \mathbb{E}_{p(I)}[\mathrm{KL}(p_t(R \mid I) \parallel p'_t(R \mid I))],$$

where $\mathrm{KL}(p \parallel q)$ is the KL divergence of q from p, and $p(I)$ is a probability distribution over $\mathcal{I}$.

Privacy Information Gain. A *privacy information gain* is introduced in Def. 4. It captures the information leakage about the sensitive information from an instance when the IPP protocol is used. Therefore, it captures the uninformativeness of an instance. A small privacy information gain shows that an instance and a sensitive value are almost independent, and therefore, the sensitive information cannot be obtained from the instance.

Definition 4 (Privacy information gain). *Given a privacy invasion model of the IPP protocol $p'_p(S \mid I)$, the privacy information gain is defined as*

$$L_p(p'_p) := \mathbb{E}_{p(I)}[\mathrm{KL}(p'_p(S \mid I) \parallel p'_p(S))],$$

which is the mutual information of S and I.

This measure is closely related to the uninformative principle (Machanavajjhala et al. 2007), especially the t-closeness principle (Li, Li, and Venkatasubramanian 2007), which were introduced in the research area of privacy preserving data publishing. The uninformative principle roughly states that published data should give little additional information beyond the background knowledge. The privacy information gain penalizes even if a worker extracts "wrong" sensitive information. This property is necessary to evaluate privacy leakage because the malicious worker can harm others using even the wrong information. The difference from privacy preserving data publishing is that, in our setting, we cannot know the amount of privacy leakage from published data I without human powers while in privacy preserving data publishing it can be computed by machines.

Empirical Estimation. These performance measures can be estimated empirically. We apply the plug-in estimation of the performance measures using the empirical estimation of distributions. We repeatedly execute a protocol M ($\in \mathbb{N}$) times to obtain M samples $\{R^{(m)}\}_{m \in \mathbb{Z}_M}$ from $p_t(R \mid I)$ and calculate an empirical probability distribution of $p_t(R \mid I)$ using an additive smoothing as

$$\hat{p}_t(R = r \mid I) \propto |\{m \in \mathbb{Z}_M \mid R^{(m)} = r\}| + \tau,$$

for each $r \in \mathcal{R}$, where τ (> 0) is a smoothing parameter, and we denote $\mathbb{Z}_M := \{0, 1, \ldots, M-1\}$. Other distributions can also be empirically calculated in the same way.

Discussions. We discuss the advantages of both the task information loss and the privacy information gain, respectively. The task information loss has two main advantages over standard measures such as precision/recall and accuracy scores. First, the task information loss can be applied to various types of task results with little modification. A task result can be a multi-class label, an integer, or a real number depending on a task definition. The task information loss can be calculated for them simply by changing the probability distributions. On contrary, a standard measure is basically task-specific, and therefore, it has little general applicability. Second, it can even be applied to a survey task, where a requester wants to collect subjective opinions of people. On contrary, standard measures cannot be applied to a subjective task because the ground truths for subjective opinions cannot be defined.

The privacy information gain is the first criterion to evaluate the amount of privacy leakage in a crowdsourcing setting, to the best of our knowledge. Standard measures are not suitable for privacy evaluation. They judge that privacy is preserved even when a worker extracts a false sensitive value from an instance. Considering that even a false sensitive value can be harmful, such criteria are not appropriate for privacy evaluation.

Case Study: Instance Clipping Protocol

As a case study, we examine the properties of an *instance clipping (IC) protocol*, which is a generalization of the methods by Little and Sun (2011) and Chen et al. (2012). Intuitively, the IC protocol clips an instance by a fixed-size window. Limiting the information a worker obtains preserves privacy. Figure 1 illustrates the IC protocol along with the terminology introduced in this section. Below, we give formal explanations of the protocol as well as the qualitative properties of the protocol.

Assumptions of the Protocol

We make assumptions on a task and workers. We assume that an instance consists of a D-dimensional array A, and

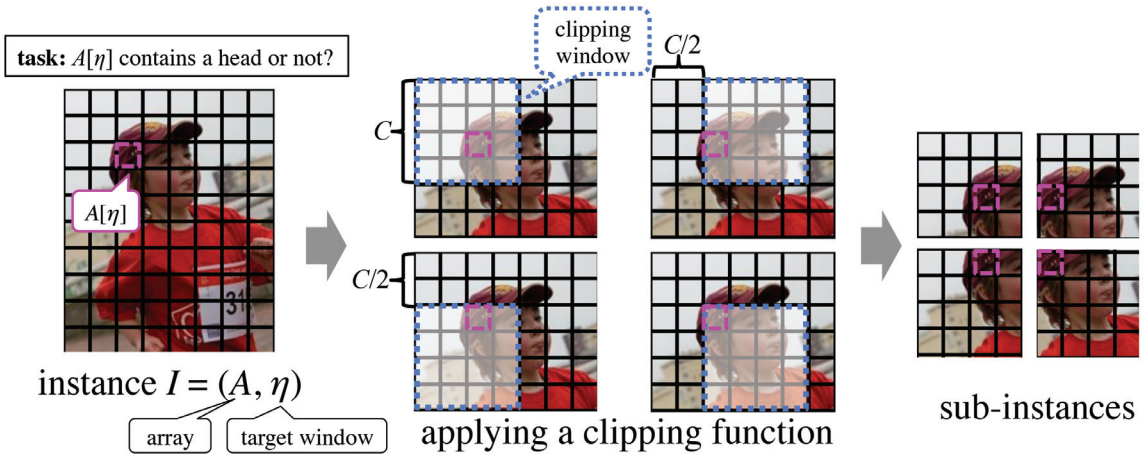

Figure 1: Illustration of the IC protocol. **(Left)** An instance is a pair of array A and target window η, and a worker is asked to check whether $A[\eta]$ (the rectangle framed by magenta broken lines) contains a head or not. **(Middle)** Given instance $I = (A, \eta)$, a clipping window of size C (a rectangle framed by blue dotted lines) is moved in steps of $C/2$ so long as the clipping window contains the target window. A clipping function clips the instance with the clipping window. **(Right)** All the sub-instances obtained by clipping are used to submit a task.

a set of indices of the array η. An instance is denoted by $I = (A, \eta)$. We call A an *array* and η a *target window*. A sub-array indexed by η is denoted by $A[\eta]$. In addition, we assume that a result of a task is a label on the part of the array indexed by the target window η. In other words, result R is defined as a label on $A[\eta]$. Figure 1 **(left)** illustrates the terminology using a head detection task. An array corresponds to an image, a target window to a specific region of the image (the rectangle framed by magenta broken lines), and a result to a label indicating whether $A[\eta]$ contains a head or not. Note that annotations on the whole image can be obtained by running the target window over the entire image, and therefore, these assumptions are not restrictive.

We also assume that workers do not collude for simplicity, *i.e.*, workers do not communicate with each other outside the protocol. This assumption ensures the amount of information each worker obtains throughout the protocol. It can be relaxed to some extent, which will be discussed later.

Protocol

The IC protocol clips an instance using a *clipping function*. It restricts the information a worker obtains from the instance. We will introduce the formal definitions of the clipping function and the IC protocol below.

Given instance $I = (A, \eta)$, a clipping function generates a *sub-instance* by clipping an instance with a *clipping window* of size C. The clipping window is required to include the target window η. Definition 5 formalizes the notion of the clipping function.

Definition 5 (Clipping function). *Given instance $I = (A, \eta)$, for index set θ such that $\theta \supseteq \eta$, a clipping function $\phi(I; \theta)$ is defined as*

$$\phi(I; \theta) := (A[\theta], \eta) \ (=: I[\theta]).$$

We call θ a clipping window, $A[\theta]$ a sub-array, and $I[\theta] = (A[\theta], \eta)$ a sub-instance.

Protocol 2 Instance Clipping (IC) Protocol

Inputs: instance I, task execution model $p_t(R \mid I)$, privacy invasion model $p_p(S \mid I)$, and size of a clipping window C.
Output of a requester: a set of results.
Output of a worker: sensitive information.

1: $\Theta_C \leftarrow \emptyset$.
2: **for** clipping window θ defined in steps of $C/2$ **do**
3: **if** $\theta \supseteq \eta$ **then**
4: $\Theta_C \leftarrow \Theta_C \cup \{\theta\}$.
5: **end if**
6: **end for**
7: The requester submits $|\Theta_C|$ tasks with $\{\phi(I; \theta)\}_{\theta \in \Theta_C}$.
8: **for** $\theta \in \Theta_C$ **do**
9: A worker is randomly selected.
10: The worker samples result $R^{(\theta)}$ from $p_t(R \mid \phi(I; \theta))$.
11: The worker returns result $R^{(\theta)}$ to the requester.
12: The worker extracts sensitive information from $p_p(S \mid \phi(I; \theta))$.
13: **end for**
14: The requester regards $\{R^{(\theta)}\}_{\theta \in \Theta_C}$ as samples of R.

Then, the IC protocol can be described as follows. At first, a requester has instance $I = (A, \eta)$ and fixes window size C. Then, the requester moves a clipping window of size C over array A in steps of $C/2$, and applies the clipping function when clipping window θ contains target window η. Finally, the requester submits a task using all of the sub-instances and regards the results as a sample from $p'_t(R \mid I)$. The formal description of the IC protocol is given in Protocol 2.

The reason why the clipping window is moved in steps of $C/2$ is to ensure that the resultant sub-instances have overlapped areas. Figure 1 **(right)** shows that all the sub-instances have overlaps around the target window. The over-

laps are necessary because, without them, a target object can be divided into two sub-instances, which can degrade the task performance severely. For example, in a head detection task, a head can be divided into two without overlaps, while that does not happen so often with overlaps.

Discussions

We discuss five topics of the IC protocol.

Qualitative Property. We analyze the performance of the IC protocol qualitatively. First, its task performance preservation capability depends on the *locality* of a task, *i.e.*, the number of elements on which a result probabilistically depends. In an extreme case, if result R (a label on $A[\eta]$) depends only on sub-array $A[\eta]$, the task performance is not degraded in the IC protocol, because clipping with a clipping window θ ($\supseteq \eta$) does not reduce the information to complete the task. In another extreme case, if result R depends heavily on the entire array A, the task performance will be severely degraded because clipping removes the information that is necessary to determine a result. Thus, the IC protocol will be effective for local tasks. The same logic can be applied to the instance privacy preservation capability. To prevent privacy invasion by the IC protocol, the privacy invasion should not have locality. In summary, the IC protocol is suitable to a pair of a *local* task and a *global* privacy definition.

Examples. We give three examples for which the IC protocol is suitable according to the previous observation. The first example is a task to detect heads in images. Sensitive information can be defined as the associations between persons in images and their contexts, *i.e.*, their location, activities, companions, etc. The task often has the locality because the area of a head is usually small. The privacy invasion will not have the locality because the association of persons and their contexts often requires a large part of an image to infer. This example is used in the experiment.

The second example is a task to transcribe an audio recording of a meeting, and the third is a task to digitize a handwritten document. Sensitive information is the abstract of the meeting or the document, which cannot easily be inferred from a local part of the recording or the scanned image. By contrast, the task can be processed even with clipped arrays if a clipped array (a segment of an audio file or a document) contains a few words. Note that, in practice, it is necessary to align the results obtained in the protocol (texts consisting of a few words) to create a sentence. The latter two examples are left for future research.

Selecting Clipping Window Size C. The IC protocol has parameter C that controls the trade-off between task performance and privacy preservation capabilities. To select the parameter, it is necessary to prepare a test dataset to evaluate the performances. By setting the maximum tolerable task information loss, one can select clipping window size C so that the privacy information gain is minimized.

Non-Collusion Assumption. We assumed that workers do not collude in the IC protocol. By choosing workers appropriately, the assumption ensures that each worker obtains no more than one sub-instance of $C \times C$ for every instance. In fact, the assumption can be relaxed to some extent by evaluating the risk that a collusion by a group of workers succeeds. We consider that a collusion by a group succeeds if a group obtains at least two sub-instances generated from the same instance. We also assume that when a requester submits tasks with n sub-instances generated from one instance, the requester chooses n workers from N workers randomly. Then, the probability that a collusion by a group of J workers succeeds can be calculated as $1 - \binom{N-J}{n} / \binom{N}{n} - \binom{N-J}{n-1} \cdot \binom{J}{1} / \binom{N}{n}$, which is negligible if N is large enough as compared to J.

Extensions. The IC protocol can be further extended in several directions. It is possible to replace the clipping function with any *instance transformation function*, which transforms an instance to preserve privacy. For example, a function to add noise on an instance or that to blur an image instance will be suitable. Another direction is to select the clipping window size adaptively. For example, a protocol starts with a small clipping window and expands as necessary until workers can perform tasks. We leave these extensions as future research.

Experiments

We conduct experiments to show that the IC protocol works well on a suitable pair of a task and privacy. First, we introduce a dataset, a task definition, and a privacy definition. Then, we show the evaluation results of the task performance preservation capability using both the proposed measure and the standard measures, and those of the privacy preservation capability using the proposed measure.

Task and Privacy Definitions and Dataset

We chose a task, a privacy definition, and a dataset to which the IC protocol was expected to be suitable according to the qualitative analysis provided in the previous section.

Dataset. We used the Stanford 40 Action Dataset (Yao et al. 2011), which contains images of humans performing actions belonging to forty classes. We selected ten classes: cooking, fishing, running, throwing frisby, watching TV, feeding a horse, playing guitar, texting message, using a computer, and writing on a book. For each class, we selected fifty images in which all the humans were engaged in the action. Therefore, we used 500 pairs of images and action labels. All the images were resized to fit in 500×500 pixels.

Task Definition. We employed a head detection task where an instance was an image, and a task was to detect areas containing human heads in images. We converted the task as a set of labeling tasks as follows. First, we divided an image into blocks of $S \times S$ pixels as shown in Fig. 1, where blocks are illustrated by lattices of black lines.[2] Then, we created a set of labeling tasks by regarding each block as a target window. The labeling task is, given an image and a target window (one block of $S \times S$ pixels), to judge whether the target window contains human heads ($R = 1$)

[2]We set $S = 25$ pixels in this study.

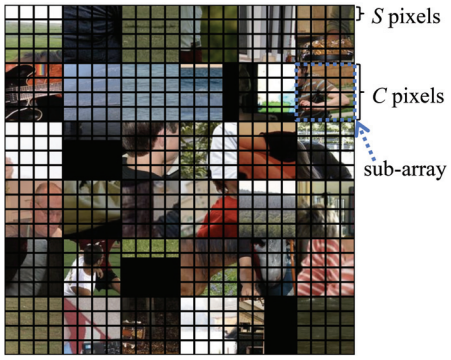

Figure 2: Combination of sub-arrays of $C \times C$ pixels used to execute the IC protocol at low cost. A worker is asked to annotate all the blocks of $S \times S$ pixels by clicking blocks that contain a part of heads.

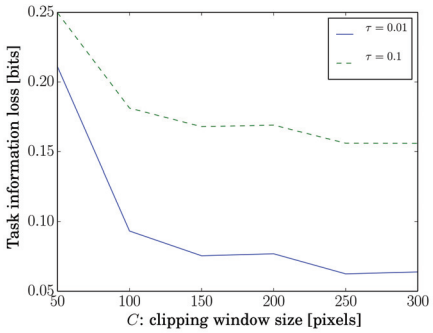

Figure 3: Task information losses for different clipping window sizes. The x-axis corresponds to the clipping window size, and the y-axis to the task information loss.

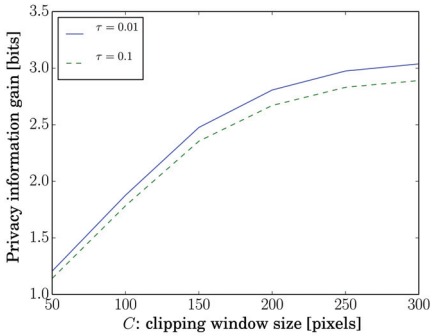

Figure 4: Privacy information gains for different clipping window sizes. The x-axis corresponds to the clipping window size, and the y-axis to the privacy information gain.

or not ($R = 0$). This conversion allows us to apply the IC protocol.

Privacy Definition. We defined the sensitive information as the associations between humans and their actions. By assuming that humans are identifiable by their heads, the instance privacy is preserved when a worker cannot infer the action context from a sub-array that contains heads. For example, if the worker cannot infer the action context *running* from a clipped image containing a head of a woman who is actually running, then the instance privacy is preserved.

Task Performance Preservation Capability

We first introduce the experimental setting for executing the IC protocol. Then, we evaluate the task performance preservation capability through the task information loss and standard measures.

Setting. We aim at investigating the relationship between the task performance preservation capability and clipping window size C. We repeatedly executed the IC protocol, varying C from 50 pixels to 300 pixels in steps of 50 pixels, and for each C, we obtained annotations of all the images and calculated the performance measure using all the annotations. In the implementation, we applied two ideas to reduce costs. First, we combined sub-arrays (clipped images) generated from different images as shown in Fig. 2 and submitted a task using the combined image. We combined the sub-arrays so that the size of a combined image was roughly 500×500 pixels. Second, we asked workers to annotate all the blocks of $S \times S$ pixels in a combined image. We allocated one worker per combined image, and therefore, the large parts of images were labeled by four workers.[3] A reward of 0.5 cents was given for the completion on a task with a combined image.

We also executed the NPP protocol using all the instances. In the implementation, we applied the second idea, *i.e.*, one worker gave annotations for all the blocks of $S \times S$ pixels of an image. We assigned one worker per image and gave a reward of one cent per image.

Experiment Using Proposed Measure. We first evaluated the task performance preservation capability by the task information loss. The loss was estimated empirically using two smoothing parameters, $\tau = 0.1$ and 0.01. For each C, we calculated the task information loss using all the results. The experimental results are shown in Fig. 3. Although the values of the task information losses were different on different smoothing parameters, they showed the same trend. The results show that the task information loss was almost the same when C was larger than 100 pixels. Therefore, we conclude that the IC protocol can mostly preserve task performance. This result matches the intuition that the head detection task is a local task.

Experiment Using Standard Measures. We also evaluated the task performance preservation capability using standard measures: precision, recall, and the F-measure. By comparing the results of the proposed measure with those of the standard measures, we investigated the validity of its use. We repeatedly stress that, although we can apply the standard measures for a binary labeling task, we have to devise other measures for other tasks, or sometimes, it is almost impossible to devise other measures, *e.g.*, for a subjective task.

In order to apply the standard measures, it was necessary to aggregate multiple labels given to each instance. We used three popular label aggregation methods.

OR method: Multiple labels are aggregated by taking their logical disjunction. The aggregated label is 1 if there is a label 1 in the multiple labels and 0 if all the labels are 0.

Majority Voting (MV) method: Multiple labels are aggregated by majority vote. Ties are broken uniformly randomly.

Latent Class (LC) method: This is a standard method for aggregating multiple labels given by workers with different abilities. We used the method proposed by Dawid and Skene (1979), where the labeling processes of workers are modeled using their ability parameters. The aggregated la-

[3] As shown in Fig. 1 **(right)**, each instance generates four sub-instances in general.

bels are obtained by inferring the true labels in the model.

For each C, we calculated the standard measures on the aggregated labels of all the instances by regarding the results of the NPP protocol as the ground truths.

Results. The results are shown in Fig. 5. First, the trends of all the scores were almost the same as those of the proposed measure shown in Fig. 3, including the outlier value at $C = 200$ pixels. Therefore, we conclude that the proposed measure is consistent with the standard measures, which supports the validity of its use.

Then, we investigated the precision score failure at $C = 200$ pixels when we applied the OR and MV methods. This can be explained by the workers' abilities. Figure 6 shows the estimated abilities of workers in the LC method. The top frame in Fig. 6 shows the probability of each worker assigning label 1 when the estimated true label is 1, which is related to the recall scores. We denote the probability of worker j by α_j. The middle frame shows the probability of each worker assigning label 0 when the estimated true label is 0, which is related to the precision scores. We denote the probability of worker j by β_j. In the middle frame, while most of the workers had high abilities, the ability of worker 19 was quite low at $C = 200$ pixels. Considering that the number of labels 0 is much bigger than the number of labels 1 in this task, worker 19 gave a significant amount of labels 1 to sub-instances that should have been labeled 0. Therefore, we conclude that the failure was caused by the low precision of worker 19. The LC method could handle the low quality labels by taking the estimated workers' abilities into account, while the other methods could not handle them, which led to the results.

Instance Privacy Preservation Capability

We evaluate the instance privacy preservation capability using the proposed measure. We could not compare the proposed measure with other measures because ours is the first performance measure to the best of our knowledge.

Setting. We investigated the relationship between the instance privacy preservation capability and clipping window size C. We repeated the following procedure, varying C from 50 pixels to 300 pixels in steps of 50 pixels.

We used ten-choice questions to simulate the privacy invasion. A worker was given a sub-array and ten choices of action labels and was asked to assign an appropriate label to the sub-array. For each action label, we randomly chose twenty-five sub-arrays from the sub-arrays that were judged to contain heads in the previous experiments. Thus, we had 250 sub-instances in total. We assigned fifty workers to each question. A reward of 0.2 cents was given to a worker for answering one question. We calculated the privacy information gain using two smoothing parameters, $\tau = 0.1$ and 0.01.

Results. The results are shown in Fig. 4. The scores of the privacy information gain on different smoothing parameters showed the same trend. The privacy information gain increased monotonically as C increased. This matches intuition: a privacy invasion is easy for large sub-instances. The remarkable result is the speed of the incline as compared to the task information loss. As we increased C, we found that the task information loss saturated more quickly than the privacy information gain. This result indicates that the privacy invasion is a global task as compared to the head detection task. As a result, we conclude that the IC protocol is effective in this setting.

Related Work

Human computation is a research area to solve problems harnessing human intelligence (Law and von Ahn 2011). Despite the widespread use of many human computation-based services, including reCAPTCHA (von Ahn et al. 2008) and Foldit (Cooper et al. 2010), many problems have arisen. One of the most well-known problems is the quality control problem. The quality of computation performed by humans is variable because of the difficulty of the computation and the ability of the human. The basic strategy for addressing the problem is to obtain multiple computation results and aggregate them to estimate a reliable result using majority voting (Sheng, Provost, and Ipeirotis 2008) or probabilistic models (Dawid and Skene 1979; Whitehill et al. 2009; Welinder et al. 2010).

In contrast, research on the privacy preservation problem in crowdsourcing has begun only recently. Little and Sun (2011) and Chen et al. (2012) proposed human OCR systems that preserved the sensitive information in documents. These systems decompose, for example, medical charts into items using a template, which prevents workers from extracting the sensitive information. The proposed protocol is more general than these systems because it does not require a template. Varshney (2012) proposed a mathematical model of the random perturbation approach. However, the model was not evaluated in a real crowdsourcing platform. Lasecki, Teevan, and Kamar (2014) raised the problem of the information extraction by workers. However, they did not propose a concrete protocol to address the problem. In addition, Harris (2011) discussed several unethical uses of crowdsourcing, including information extraction by workers, which supports our claim that our problem setting is quite important.

Conclusion

This paper proposed the framework of evaluating instance privacy preserving protocols and studied properties of a generalization of existing protocols as a case study. We introduced models of task execution and privacy invasion and proposed two performance measures of task performance preservation and instance privacy preservation capabilities based on the models. We also extended existing methods to derive an instance clipping protocol and investigated its properties by experiments in a real environment. The results showed that the instance clipping protocol was effective for a pair of a locally-executable task and a globally-dependent instance privacy.

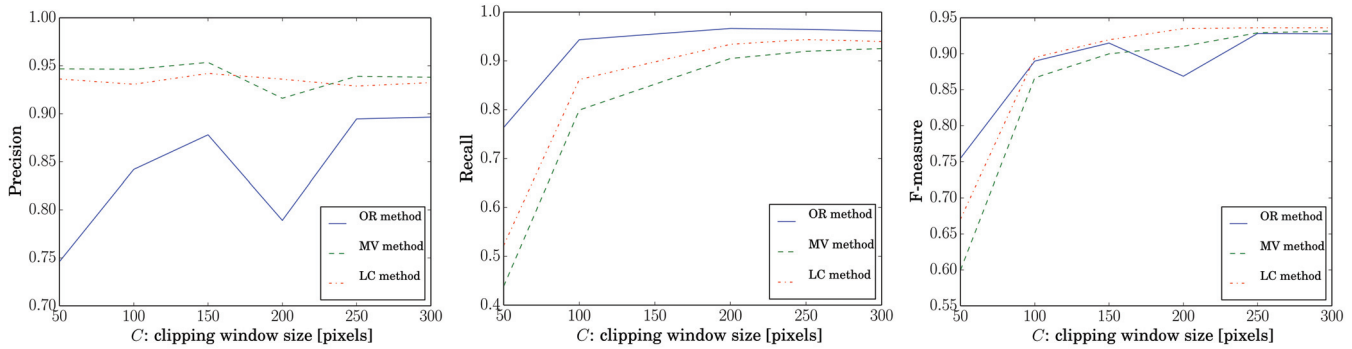

Figure 5: Precision (left), recall (middle), and the F-measure scores (right) for different window sizes.

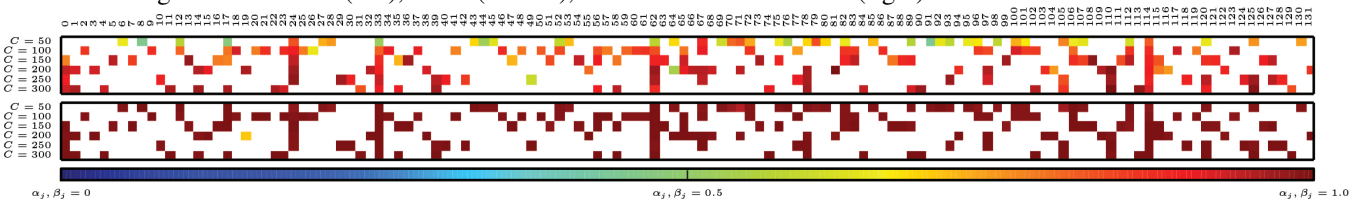

Figure 6: Estimated abilities of workers. The top and mid frames show the abilities of workers α_j and β_j. The x-axis corresponds to worker IDs j, the y-axis corresponds to tasks with specific window sizes, and each element corresponds to the value of α_j or β_j at each C. The bottom frame shows a colormap. The white elements indicate that the worker did not perform tasks.

References

Chen, K.; Kannan, A.; Yano, Y.; Hellerstein, J. M.; and Parikh, T. S. 2012. Shreddr: pipelined paper digitization for low-resource organizations. In *Proceedings of the 2nd ACM Symposium on Computing for Development*.

Cooper, S.; Khatib, F.; Treuille, A.; Barbero, J.; Lee, J.; Beenen, M.; Leaver-Fay, A.; Baker, D.; Popović, Z.; and Players, F. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466(7307):756–760.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28(1):20–28.

Harris, C. G. 2011. Dirty deeds done dirty cheap: A darker side to crowdsourcing. In *Proceedings of 2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing*, 1314–1317.

Lasecki, W. S.; Teevan, J.; and Kamar, E. 2014. Information extraction and manipulation threats in crowd-powered systems. In *Proceedings of the 2014 ACM Conference on Computer Supported Cooperative Work*.

Law, E., and von Ahn, L. 2011. *Human computation*. Morgan & Claypool Publishers.

Li, N.; Li, T.; and Venkatasubramanian, S. 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of 2007 IEEE 23rd International Conference on Data Engineering*, 106–115.

Little, G., and Sun, Y.-A. 2011. Human OCR: Insights from a complex human computation process. In *Proceedings of CHI 2011 Workshop on Crowdsourcing and Human Computation*, 8–11.

Machanavajjhala, A.; Kifer, D.; Gehrke, J.; and Venkitasubramaniam, M. 2007. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data* 1(1).

Sheng, V. S.; Provost, F.; and Ipeirotis, P. G. 2008. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 614–622.

Varshney, L. R. 2012. Privacy and reliability in crowdsourcing service delivery. In *Proceedings of the 2012 Annual SRII Global Conference*, 55–60.

von Ahn, L.; Maurer, B.; McMillen, C.; Abraham, D.; and Blum, M. 2008. reCAPTCHA: human-based character recognition via Web security measures. *Science* 321(5895):1465–1468.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems 23*, 2424–2432.

Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. 2009. Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22*, 2035–2043.

Yao, B.; Jiang, X.; Khosla, A.; Lin, A. L.; Guibas, L.; and Fei-Fei, L. 2011. Human action recognition by learning bases of action attributes and parts. In *Proceedings of 2011 IEEE International Conference on Computer Vision*, 1331–1338.