Pairwise HITS: Quality Estimation from Pairwise Comparisons in Creator-Evaluator Crowdsourcing Process

Takeru Sunahase, Yukino Baba, Hisashi Kashima
Department of Intelligence Science and Technology, Kyoto University
stakeru@ml.ist.i.kyoto-u.ac.jp, {baba, kashima}@i.kyoto-u.ac.jp

Abstract

A common technique for improving the quality of crowdsourcing results is to assign a same task to multiple workers redundantly, and then to aggregate the results to obtain a higher-quality result; however, this technique is not applicable to complex tasks such as article writing since there is no obvious way to aggregate the results. Instead, we can use a two-stage procedure consisting of a creation stage and an evaluation stage, where we first ask workers to create artifacts, and then ask other workers to evaluate the artifacts to estimate their quality. In this study, we propose a novel quality estimation method for the two-stage procedure where pairwise comparison results for pairs of artifacts are collected at the evaluation stage. Our method is based on an extension of Kleinberg's HITS algorithm to pairwise comparison, which takes into account the ability of evaluators as well as the ability of creators. Experiments using actual crowdsourcing tasks show that our methods outperform baseline methods especially when the number of evaluators per artifact is small.

1 Introduction

With the recent growth of crowdsourcing platforms such as Amazon Mechanical Turk, crowdsourcing has become a popular approach for accomplishing a wide variety of tasks, including audio transcription, article writing, and graphic designing. Crowdsourcing has been successfully used in various areas of computer science research such as natural language processing (Snow et al. 2008), computer vision (Sorokin and Forsyth 2008), and human–computer interaction (Bernstein et al. 2015; Bigham et al. 2010).

Quality of results is one of the critical issues with crowdsourcing. Since crowdsourcing workers have different levels of expertise and diligence, there is no guarantee that all workers complete the offered tasks with a satisfactory level of quality. A convenient quality control approach is to assign the same task to multiple workers and aggregate their results to obtain more reliable outputs. Majority voting and averaging are examples of simple aggregation methods, and several statistical methods considering the ability of each worker or the difficulty of each task have been proposed (Dawid and Skene 1979; Whitehill et al. 2009; Welinder et al. 2010; Lin, Mausam, and Weld 2012; Demartini, Difallah, and Cudré-Mauroux 2013).

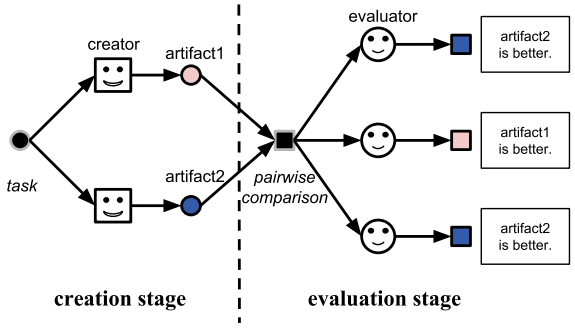


Figure 1: Two-stage procedure with pairwise comparison

These aggregation methods have been widely applied to simple crowdsourcing tasks such as multiple-choice questions; however, these methods are not applicable for general crowdsourcing tasks such as article writing or graphic designing, in which agreement between the results may not be possible. Many tasks fall into this category, and a study showed that the top requesters on Amazon Mechanical Turk (based on the total rewards posted) typically request such general tasks, including content generation and website feedback (Ipeirotis 2010). A natural strategy for controlling the work quality on general crowdsourcing tasks is to introduce a two-stage procedure, whereby workers (called *creators*) first create artifacts, and then another set of workers (called *evaluators*) evaluate the results. This procedure allows us to estimate the quality level of each artifact based on the evaluations from the workers.

In this paper, we focus on a two-stage procedure with pairwise comparison (Figure 1), wherein evaluators are asked to compare a pair of artifacts and vote for one of them in the evaluation stage. Pairwise comparison has several advantages over rating single artifacts; it can capture a small difference in quality between artifacts, and evaluators do not need to calibrate their standards over time.

Our proposed methods are extensions of the HITS algorithm (Kleinberg 1999) to pairwise comparison, which take into account the ability of evaluators as well as the ability

of creators. Analogous to the *hubs* and *authorities* in the HITS algorithm, we assume that a good evaluator votes for many good artifacts and that a good artifact is voted for by many good evaluators. We modify the HITS algorithm so that it is applicable to pairwise comparison, which we call the *Pairwise HITS* algorithm (Section 3). Moreover, with the assumption that the ability of creators affect the quality of their artifacts, we propose the *Two-Stage Pairwise HITS* algorithm (Section 4), which estimates the ability of the creators in addition to the quality of the artifacts and the ability of the evaluators.

We conducted experiments using image description, logo designing, and article language translation tasks on a commercial crowdsourcing platform (Section 5). Our methods outperformed the Bradley–Terry model (Bradley and Terry 1952) and Crowd-BT (Chen et al. 2013). In addition, we discover that the Two-Stage Pairwise HITS algorithm consistently showed better performance than the Pairwise HITS algorithm; this result demonstrates the benefit of modeling the creators' ability levels.

Contributions of this paper are summarized as follows:

- We address the quality estimation problem for the two-stage crowdsourcing process with pairwise comparison.
- We focus on the relationship between a web page ranking problem and the quality estimation problem and build the Pairwise HITS algorithm by adapting the HITS algorithm for pairwise comparison.
- We further modify the HITS algorithm to incorporate the creators' ability levels, and propose the Two-Stage Pairwise HITS algorithm.

2 Problem Setting

We address the quality estimation problem in a two-stage crowdsourcing procedure from pairwise comparison data (Figure 2). We assume that there are m crowdsourcing tasks. We denote a task by $t \in \{1, \dots, m\}$.

In the creation stage, n_t different creators create artifacts for each task t. We have p creators and n artifacts in total, and we denote a creator by $k \in \{1, \dots, p\}$ and an artifact by $j \in \{1, \dots, n\}$. We denote the set of creation information by $W = \{(j, k)\}_{j=1}^{n}$, where (j, k) indicates that creator k creates artifact j.

In the evaluation stage, evaluators compare each pair of artifacts created for the same task, and vote for one of them. We have l pairs in total. There are o evaluators, and we denote an evaluator by $i \in \{1, \dots, o\}$. The notation $j \succ_i j'$ indicates that evaluator i prefers artifact j over artifact j', and the tuple (j, j', i) indicates the result that $j \succ_i j'$. We denote the set of observed evaluation information by $V = \{(j, j', i) \mid j \succ_i j'\}$.

Given the observed information W and V, our goal is to estimate the quality level q_j of each artifact j. Sorting the quality levels results in the ranking of artifacts for each task.

3 Pairwise HITS

The HITS algorithm (Kleinberg 1999) is a ranking method for web pages based on link structure among them. Authorities and hubs are central concepts of HITS; the authorities are web pages with highly reliable and informative information, and the hubs are web pages that are catalogs of these good information sources. HITS assumes that an authority is linked to by many hubs while a hub links to many authorities. The idea is applicable to the quality estimation problem of crowdsourcing artifacts (Kajimura et al. 2015); a good evaluator votes for many good artifacts, while a good artifact collects votes from many good evaluators. The quality level q_j of artifact j and the ability level r_i of evaluator i are respectively given as

$$q_j = \sum_{i \in I_j} r_i, \quad r_i = \sum_{j \in O_i} q_j, \tag{1}$$

where I_j is the set of evaluators voting for artifact j, and O_i is the set of artifacts voted for by evaluator i.

The HITS-based quality estimation method is not directly applicable to pairwise comparisons because a pairwise comparison does not vote for a single artifact but votes on which artifact is the better of two given artifacts. Analogous to the HITS-based quality estimation where the artifact quality is given as the sum of evaluators' ability levels voting for it, we give the difference between the quality levels of two artifacts j and j' as the difference between the sum of evaluators' ability levels voting for them in their pairwise comparison:

$$q_j - q_{j'} = \sum_{i \in V_{j,j'}} r_i - \sum_{i \in V_{j',j}} r_i, \tag{2}$$

where $V_{j,j'} = \{i \mid j \succ_i j'\}$ is the set of evaluators who prefer artifact j to artifact j'.

The constraints (2) are summarized as a system of linear equations $P\boldsymbol{q} = \boldsymbol{b}$, where P is the $l \times n$ matrix each of whose elements takes one of $\{+1, -1, 0\}$, $\boldsymbol{q} = (q_1, \dots, q_n)^\top$, and $\boldsymbol{b}$ is the l-dimensional vector that stacks the right-hand sides of the constraints (2). Since the number of constraints is

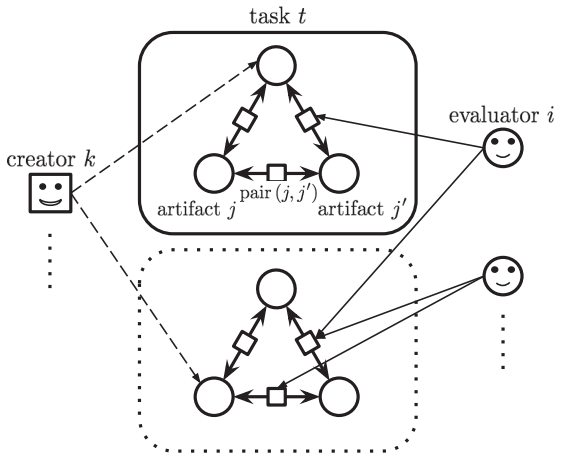


Figure 2: For each crowdsourcing task t, several artifacts are created by different creators. Evaluators compare each pair of artifacts. Any creator or evaluator can participate in multiple creation or evaluation tasks, respectively.

Algorithm 1: Pairwise HITS

Data: Evaluation information $V = \{(j, j', i)\}$
Result: Artifact qualities $\boldsymbol{q} = (q_1, \ldots, q_n)$; Evaluator abilities $\boldsymbol{r} = (r_1, \ldots, r_o)$
Initialization: $r_i = 0\ \forall i \in \{1, \ldots, o\}$
repeat
  /* quality update */
  update $\boldsymbol{q}$ by using Eq. (3)
  /* ability update */
  foreach $i \in \{1, \ldots, o\}$ **do**
    update r_i by using Eq. (4)
  end
  Normalize $\boldsymbol{r}$ so that $\|\boldsymbol{r}\|_2^2 = 1$
until *convergence or the maximum number of iterations is reached*
return $\boldsymbol{q}$; $\boldsymbol{r}$

larger than the number of variables, i.e., $l > n$, it is not possible to meet all of them exactly. Instead, we solve them approximately using the Moore–Penrose pseudo-inverse matrix $P^\dagger$ of P, that is,

$$\boldsymbol{q} = P^\dagger \boldsymbol{b}, \tag{3}$$

which gives an approximate solution in terms of the squared errors.

Meanwhile, we define the ability level r_i of evaluator i as the proportion of the number of correct decisions out of all of the comparisons the evaluator makes:

$$r_i = \frac{|\ \{(j, j') \in V_i \mid q_j > q_{j'}\}\ |}{|V_i|}, \tag{4}$$

where $V_i = \{(j, j') \mid j \succ_i j'\}$ is the set of pairwise comparisons that evaluator i makes, and $|\cdot|$ denotes the number of elements in a set.

Algorithm 1 illustrates our Pairwise HITS algorithm, which iteratively uses Eqs. (3) and (4) to update the artifact quality scores and evaluator ability scores. Note that although solving Eq. (3) seems costly, P and $P^\dagger$ are block-diagonal and can be decomposed into a set of small equations each of which corresponds to an artifact. The size of matrix in the decomposed equation is equal to the number of evaluators assigned to the corresponding artifact (usually around 20 at most).

4 Two-Stage Pairwise HITS

The Pairwise HITS algorithm considers only the ability levels of the evaluators. We further extend it so that the ability of *creators* affects the quality of artifacts as well. Since the extension of Pairwise HITS explicitly considers the information of both the creation and evaluation stages, we call it Two-Stage Pairwise HITS.

We introduce the new assumption that good creators are likely to make good artifacts and good artifacts are likely to be made by good creators. This assumption gives the artifact quality as

$$q_j = (1 - \lambda) q_j^* + \lambda c_{k_j}, \tag{5}$$

Algorithm 2: Two-Stage Pairwise HITS

Data: Creation information $W = \{(j, k)\}$; Evaluation information $V = \{(j, j', i)\}$
Result: Artifact qualities $\boldsymbol{q} = (q_1, \ldots, q_n)$; Creator abilities $\boldsymbol{c} = (c_1, \ldots, c_p)$; Evaluator abilities $\boldsymbol{r} = (r_1, \ldots, r_o)$
Parameter: hyperparameter λ
Initialization:
  $c_k = 0\ \forall k \in \{1, \ldots, p\}$; $r_i = 0\ \forall i \in \{1, \ldots, o\}$
repeat
  /* quality update */
  update $\boldsymbol{q}$ by using Eq. (5)
  /* ability update */
  foreach $k \in \{1, \ldots, p\}$ **do**
    update c_k by using Eq. (6)
  end
  foreach $i \in \{1, \ldots, o\}$ **do**
    update r_i by using Eq. (4)
  end
  Normalize $\boldsymbol{c}$ and $\boldsymbol{r}$ so that $\|\boldsymbol{c}\|_2^2 = 1$ and $\|\boldsymbol{r}\|_2^2 = 1$
until *convergence or the maximum number of iterations is reached*
return $\boldsymbol{q}$; $\boldsymbol{c}$; $\boldsymbol{r}$

where q_j^* is the quality of artifact j given by Pairwise HITS (i.e. the solution of Eq. (3)), c_{k_j} is the ability level of the creator of artifact j, and λ is the hyperparameter where $0 < \lambda < 1$. This model indicates that the quality of an artifact depends on both the quality estimated from pairwise comparison results and the ability of its creator.

Meanwhile, we define the ability level of a creator as the average of the quality levels of the artifacts that the creator made; that is, the ability c_k of creator k is given as

$$c_k = \frac{1}{|W_k|} \sum_{j \in W_k} q_j, \tag{6}$$

where W_k is the set of all artifacts made by creator k. The ability of evaluators is the same as the one given as Eq. (4).

In summary, our Two-Stage Pairwise HITS algorithm is given in Algorithm 2.

5 Experiments

To evaluate the efficacy of our two proposed algorithms, we posted three kinds of tasks on a commercial crowdsourcing platform and created real datasets from their actual results. Based on the artifacts obtained in the creation stage, we posted evaluation tasks for each pair of artifacts. We compared the precisions of the quality estimation by our two algorithms with those of two existing methods.

5.1 Datasets

We use three kinds of tasks, that are, *image description*, *logo designing*, and *language translation* for our experiments.[1]

[1]We provide these datasets on http://www.ml.ist.i.kyoto-u.ac.jp/en/en-research/sunahase2017aaai .

Table 1: Statistics of the datasets of the creation information

	# tasks	# unique creators	Avg. # creators per task	Avg. # tasks per creator	Total # artifacts
Image description	20	20	10.0	10.0	200
Logo designing	16	47	18.4	6.3	295
Language translation	20	17	9.5	11.2	190

Table 2: Statistics of the datasets of the evaluation information

	# evaluated artifacts	# compared pairs of artifacts	# unique evaluators	Avg. # evaluators per pair	Avg. # comparisons per evaluator	Total # obtained comparisons
Image description	200	940	114	16.5	136.2	15526
Logo designing	295	2797	125	14.6	325.7	40717
Language translation	190	825	105	10.2	79.8	8376

We first prepared datasets of the creation information (denoted by W) by using Lancers[2], a crowdsourcing marketplace. Table 1 gives general statistics of the datasets. We then collected the datasets of the evaluation information (denoted by V). We asked workers on Lancers to compare pairs of artifacts from each of the three tasks. Table 2 gives general statistics of the datasets of the evaluation information. In this experiment, we did not allow evaluators to evaluate the same pair of artifacts more than once. We had the evaluators evaluate all the possible pairs for each task.

5.2 Methods

We compared the Pairwise HITS algorithm and the Two-Stage Pairwise HITS algorithm with the following two ranking aggregation methods: the Bradley-Terry model (Bradley and Terry 1952) and Crowd-BT (Chen et al. 2013). The Bradley-Terry model provides a generative model of pairwise comparisons wherein the true quality levels of artifacts affect the results of comparisons, and the true quality levels are estimated by using maximum-likelihood estimation. Crowd-BT incorporates the evaluator ability levels into the Bradley-Terry model; therefore, we can put weight to each result according to the skill level of the corresponding evaluator to estimate the quality levels of the artifacts.

The Bradley-Terry model can be considered the most basic approach because it does not take into account the information about who gives each evaluation or who creates each article. Crowd-BT and the Pairwise HITS algorithm use only the evaluation information V, and the Two-Stage Pairwise HITS algorithm uses both the creation information W and the evaluation information V.

Crowd-BT was originally designed for aggregating the results of pairwise comparisons for a single task, whereas we aimed to aggregate the results of multiple tasks. We combined the results for all the tasks and then applied Crowd-BT, and after obtaining the quality level estimates, we generated the ranking of artifacts for each task.

We set initial ability levels to zero when we used the Pairwise-HITS algorithm and the Two-Stage Pairwise HITS algorithm.[3] We fixed $\lambda = 0.1$ in the Two-Stage Pairwise HITS algorithm for all the three tasks. In our experiments, we considered the estimation of the Pairwise HITS algorithm and the Two-Stage Pairwise HITS algorithm to have converged when the norm of the difference between parameters of the current and the previous iteration was less than 1.0×10^{-5}. Ten to twenty iterations are usually sufficient for convergence of both the PairwiseHITS algorithm and the Two-Stage Pairwise HITS algorithm.

5.3 Evaluation methodology

We used Spearman's rank correlation coefficient between the estimated ranking and the ground truth ranking of all the artifacts for the evaluation measure. Spearman's rank correlation evaluates how similar two rankings are. We investigated the effect on estimation accuracy by the number of evaluators assigned to each pair. We varied the number of evaluators per pair, sampled 100 subsets of the evaluation data for each number of evaluators, and performed the Wilcoxon signed-rank test.

Because we did not have the ground truth quality levels, we used the results of the Bradley-Terry model with all the evaluation data; this simulation approach was applied by Baba and Kashima (2013), which was supported by a report that the accuracy of a majority vote with ten or more non-experts was comparable to that of experts in various natural language processing tasks (Snow et al. 2008), and the Bradley-Terry model can be considered as the majority vote of pairwise comparisons in terms of assigning equal weights to all workers.

5.4 Results

Table 3 shows the rank correlations between the estimated artifact rankings and the ground truth rankings for each number of evaluators per artifact pair. In all the three tasks, the Two-Stage Pairwise HITS algorithm achieved statistically significant higher performance over the other methods. In

[2]http://www.lancers.jp/

[3]The initial ability levels can be set to pre-estimated worker ability levels if we are able to prepare gold standard datasets for estimating the ability levels.

Table 3: Results of each task: averages and standard deviations of Spearman correlations between estimated quality levels and ground truth by the number of evaluators per pair. Statistically significant ($p < 0.05$) winners by the Wilcoxon signed-rank test are boldfaced. Statistically significant better results than the Bradley-Terry model are marked with †. Two-stage Pairwise HITS achieved statistically significant higher performance over the other methods.

	Spearman correlation				
# evaluators per pair	1	2	3	4	5
Image description					
Bradley-Terry model	0.782 ± 0.031	0.868 ± 0.018	0.904 ± 0.014	0.925 ± 0.011	0.939 ± 0.011
Crowd-BT	$\dagger 0.796 \pm 0.029$	0.866 ± 0.019	0.889 ± 0.014	0.896 ± 0.015	0.901 ± 0.012
Pairwise HITS	$\dagger 0.800 \pm 0.029$	$\dagger 0.879 \pm 0.018$	$\dagger 0.912 \pm 0.012$	$\dagger 0.931 \pm 0.010$	$\dagger 0.943 \pm 0.009$
Two-Stage Pairwise HITS	$\dagger \mathbf{0.857} \pm 0.020$	$\dagger \mathbf{0.899} \pm 0.016$	$\dagger \mathbf{0.923} \pm 0.011$	$\dagger \mathbf{0.936} \pm 0.010$	$\dagger \mathbf{0.947} \pm 0.008$
Logo designing					
Bradley-Terry model	0.721 ± 0.030	0.827 ± 0.022	0.875 ± 0.016	0.902 ± 0.013	0.924 ± 0.009
Crowd-BT	0.710 ± 0.030	0.792 ± 0.025	0.819 ± 0.019	0.828 ± 0.017	0.838 ± 0.019
Pairwise HITS	$\dagger 0.735 \pm 0.028$	$\dagger 0.836 \pm 0.021$	$\dagger 0.882 \pm 0.015$	$\dagger 0.907 \pm 0.013$	$\dagger 0.927 \pm 0.009$
Two-Stage Pairwise HITS	$\dagger \mathbf{0.766} \pm 0.023$	$\dagger \mathbf{0.849} \pm 0.019$	$\dagger \mathbf{0.888} \pm 0.013$	$\dagger \mathbf{0.910} \pm 0.013$	$\dagger 0.928 \pm 0.008$
Language translation					
Bradley-Terry model	0.580 ± 0.049	0.716 ± 0.038	0.788 ± 0.031	0.839 ± 0.024	0.872 ± 0.020
Crowd-BT	0.580 ± 0.057	0.720 ± 0.039	0.785 ± 0.037	0.819 ± 0.023	0.837 ± 0.021
Pairwise HITS	$\dagger 0.604 \pm 0.047$	$\dagger 0.737 \pm 0.037$	$\dagger 0.808 \pm 0.029$	$\dagger \mathbf{0.856} \pm 0.022$	$\dagger \mathbf{0.885} \pm 0.019$
Two-Stage Pairwise HITS	$\dagger \mathbf{0.661} \pm 0.035$	$\dagger \mathbf{0.758} \pm 0.031$	$\dagger 0.811 \pm 0.025$	$\dagger 0.852 \pm 0.020$	$\dagger 0.877 \pm 0.017$

particular, when the number of evaluators was small, the Pairwise HITS and the Two-Stage Pairwise HITS algorithms showed great improvement. The comparison of the performance of the Pairwise HITS algorithm with that of the Two-Stage Pairwise HITS algorithm shows that the introduction of creator ability levels led to high performance. Therefore, we have shown that creator ability levels provide improvement in the accuracy of quality estimation in the two stage procedure with pairwise comparison.

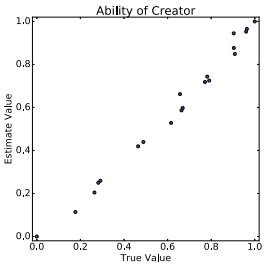
We then investigate the accuracies of estimated worker ability levels. Figures 3 and 4 show the relations between the true values and the estimated values of creator ability level and evaluator ability level, respectively. The true ability level of each creator and that of each evaluator were calculated by using Eq. (4) and Eq. (6) with the ground truth quality levels, respectively. The estimated creator ability levels and the estimated evaluator ability levels were obtained from a subset of the results with randomly selected five evaluators per pair. It can been seen that the Two-Stage Pairwise HITS algorithm precisely estimated the creator ability levels; this explains the performance improvement by incorporating the creator abilities for the artifact quality estimation. The evaluator ability levels were accurately estimated as well. Especially, the low-ability evaluators were correctly identified by the Two-Stage Pairwise HITS algorithm.

In summary, we verified the effectiveness of Two-Stage Pairwise HITS and Pairwise HITS, especially with a small number of evaluators.

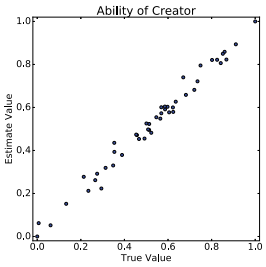
6 Related Work

Quality estimation methods for crowdsourced artifacts have been the subject of recent study. The two-stage procedure with the creation and the evaluation stages was introduced by Baba and Kashima (2013) and they proposed a statistical quality estimation method for this procedure. Their probabilistic model incorporates the ability and task-dependent performance of each creator and the bias and contextual preference of each evaluator. Whereas they focused on a rating-based two-stage procedure in which evaluators use a rating scale to evaluate each single artifact, we target a two-stage procedure with pairwise comparison. A statistical method for aggregating pairwise rankings has also been proposed (Chen et al. 2013). This method, called Crowd-BT, models the ability of an evaluator as the probability of the evaluators' providing the correct order for a given pair. Whereas Crowd-BT considers only the ability levels of evaluators, our Two-Stage Pairwise HITS algorithm incorporates the ability levels of both evaluators and creators.

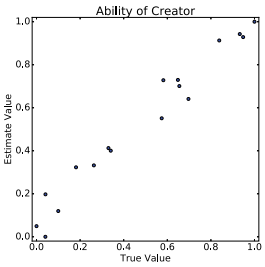
The HITS algorithm was originally proposed for ranking the web pages, and we have applied it to the problem of quality estimation of crowdsourced results. A few studies utilized the HITS algorithm for other applications. Fujimura and Tanimoto employed the HITS algorithm to assess the reliability of user-generated content, such as answers on problem-solving web services and reviews on product review forums (Fujimura and Tanimoto 2005). Wu, Zubair, and Maly applied the HITS algorithm to discover reliable content in social tagging systems by using the relationships between content, tags, and users (Wu, Zubair, and Maly 2006). There was a study that applied the HITS algorithm to the quality control problem in crowdsourcing (Kajimura et al. 2015), which focused on point-of-interest (POI) collection tasks, wherein workers were given a query (e.g., "Good steak houses in NYC") and asked to list appropriate POIs. The authors assumed that a reliable worker provides a reliable POI and applied the HITS algorithm for estimating the



(a) Image description

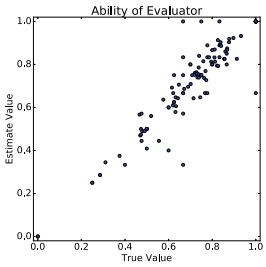


(b) Logo designing

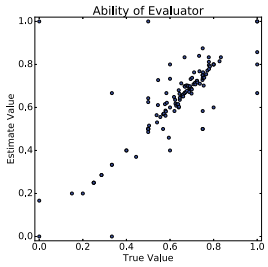


(c) Language translation

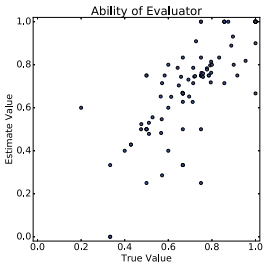
Figure 3: Accuracy evaluation of estimated creator ability levels. Relations between the true creator ability level and the estimated creator ability level are shown. Strong correlations over 0.98 are confirmed for all the three tasks.



(a) Image description



(b) Logo designing



(c) Language translation

Figure 4: Accuracy evaluation of estimated evaluator ability levels. Relations between the true evaluator ability level and the estimated evaluator ability level are shown. High correlations (0.92, 0.75, and 0.79 for the image description tasks, the logo designing tasks, and the language translation tasks, respectively) are confirmed.

quality of collected POIs. They did not target pairwise comparison nor the two-stage procedure.

In the educational data mining area, several statistical methods for peer assessment have been proposed to estimate the quality of student submissions. Piech et al. focused on the fact that a student can be both an author of a submission and an evaluator of submissions from other students, and proposed a model wherein the evaluation ability depends on the ability to create a good submission (Piech et al. 2013). PeerRank is a method that is based on the assumption that a student uses the same ability when she creates and evaluates a submission (Walsh 2014). In the PeerRank model, the ability level of a student is determined by a weighted sum of the grades given by other students, where the weights are the ability levels of the students. The author of PeerRank applied the concept from PageRank that the score of a web page depends on the scores of the web pages linking to the page. Although PeerRank and our methods both apply the web link analysis methods for estimating the quality of artifacts, they are different in two respects: our methods focus on pairwise comparison while PeerRank assumes that a grade is given as a numerical value to each submission, and we model the ability levels of creators and evaluators separately because it is not very frequent on crowdsourcing platforms that a worker acts as both a creator and an evaluator.

7 Conclusion

We have addressed the quality estimation of crowdsourced artifacts for general crowdsourcing tasks whose results are hardly to aggregate, and we have presented two unsupervised algorithms using pairwise comparisons. We proposed two quality estimation methods; one was the Pairwise HITS algorithm, which was adapted the HITS algorithm to estimate the evaluator ability levels and the artifact quality levels. The other one was the Two-Stage Pairwise HITS algorithm, which was also an extension of HITS but it incorporated the creator ability levels. Based on experiments comparing our methods with baseline methods, results showed that Two-Stage Pairwise HITS outperformed all other methods. The introduction of the ability levels of creators led to improvement in the precision of quality estimation for not only in the rating based two-stage procedure but also in the two-stage procedure with pairwise comparison.

Finally, we mention some possible future work. We have introduced a new pairwise quality estimation method based on the HITS algorithm and extended it for the two-stage procedure of creation and evaluation. On the other hand, an existing pairwise quality estimation method, Crowd-BT, has been developed, and its extension for application to the two-stage procedure can be valuable. Active selection of tasks and workers (Sheng, Provost, and Ipeirotis 2008; Donmez, Carbonell, and Schneider 2009; Yan et al. 2011) is also an important future direction.

Acknowledgments

We would like to thank Xi Chen for sharing his Crowd-BT code with us. This work was supported by JSPS KAKENHI Grant Number 15H01704.

References

Baba, Y., and Kashima, H. 2013. Statistical quality estimation for general crowdsourcing tasks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 554–562.

Bernstein, M. S.; Little, G.; Miller, R. C.; Hartmann, B.; Ackerman, M. S.; Karger, D. R.; Crowell, D.; and Panovich, K. 2015. Soylent: a word processor with a crowd inside. *Communications of the ACM* 58(8):85–94.

Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B.; White, S.; et al. 2010. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, 333–342.

Bradley, R. A., and Terry, M. 1952. The rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika* 39(3):324–345.

Chen, X.; Bennett, P. N.; Collins-Thompson, K.; and Horvitz, E. 2013. Pairwise ranking aggregation in a crowd-sourced setting. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, 193–202.

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

Demartini, G.; Difallah, D. E.; and Cudré-Mauroux, P. 2013. Large-scale linked data integration using probabilistic reasoning and crowdsourcing. *The VLDB Journal* 22(5):665–687.

Donmez, P.; Carbonell, J. G.; and Schneider, J. 2009. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 259–268.

Fujimura, K., and Tanimoto, N. 2005. The EigenRumor algorithm for calculating contributions in cyberspace communities. In *Trusting Agents for Trusting Electronic Societies*. Springer. 59–74.

Ipeirotis, P. G. 2010. Analyzing the Amazon Mechanical Turk marketplace. *ACM XRDS* 4(2):16–21.

Kajimura, S.; Baba, Y.; Kajino, H.; and Kashima, H. 2015. Quality control for crowdsourced POI collection. In *Proceedings of the 19th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 255–267.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46(5):604–632.

Lin, C. H.; Mausam, M.; and Weld, D. 2012. Crowdsourcing control: moving beyond multiple choice. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, 491–500.

Piech, C.; Huang, J.; Chen, Z.; Do, C.; Ng, A.; and Koller, D. 2013. Tuned models of peer assessment in MOOCs. In *Proceedings of the 6th International Conference on Educational Data Mining*, 153–160.

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.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 254–263.

Sorokin, A., and Forsyth, D. 2008. Utility data annotation with Amazon Mechanical Turk. In *Proceedings of the 1st IEEE Workshop on Internet Vision*.

Walsh, T. 2014. The PeerRank method for peer assessment. In *Proceedings of the 21st European Conference on Artificial Intelligence*, 909–914.

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.; fan Wu, T.; Bergsma, J.; Movellan, J. R.; and Ruvolo, P. L. 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.

Wu, H.; Zubair, M.; and Maly, K. 2006. Harvesting social knowledge from folksonomies. In *Proceedings of the 17th Conference on Hypertext and Hypermedia*, 111–114.

Yan, Y.; Fung, G. M.; Rosales, R.; and Dy, J. G. 2011. Active learning from crowds. In *Proceedings of the 28th International Conference on Machine Learning*, 1161–1168.