Reliable Aggregation Method for Vector Regression Tasks in Crowdsourcing

Joonyoung Kim, Donghyeon Lee, and Kyomin Jung(✉)

Seoul National University, Seoul, Republic of Korea
{kimjymcl,donghyeon,kjung}@snu.ac.kr

Abstract. Crowdsourcing platforms are widely used for collecting large amount of labeled data. Due to low-paid workers and inherent noise, the quality of acquired data could be easily degraded. To solve this, most previous studies have sought to infer the true answer from noisy labels in discrete multiple-choice tasks that ask workers to select one of several answer candidates. However, recent crowdsourcing tasks have become more complicated and usually consist of real-valued vectors. In this paper, we propose a novel inference algorithm for vector regression tasks which ask workers to provide accurate vectors such as image object localization and human posture estimation. Our algorithm can estimate the true answer of each task and a reliability of each worker by updating two types of messages iteratively. We also prove its performance bound which depends on the number of queries per task and the average quality of workers. Under a certain condition, we prove that its average performance becomes close to an oracle estimator which knows the reliability of every worker. Through extensive experiments with both real-world and synthetic datasets, we verify that our algorithm are superior to other state-of-the-art algorithms.

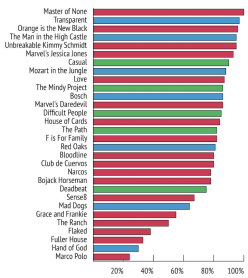
Keywords: Crowdsourcing · Vector regression · Algorithm

1 Introduction

The problem of collecting large amounts of labeled data is of practical importance, particularly in the artificial intelligence field [15], since the amount of data is a dominant factor in determining whether a model is well-trained. Recently, it has become common to collect labeled data through web-based crowdsourcing platforms such as Amazon Mechanical Turk.

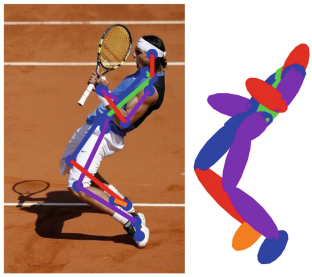
Although a crowdsourcing paradigm is widespread, it has fatal weaknesses: human workers' decisions may vary significantly due to misconceptions of task instructions, the lack of responsibility, and inherent noise [5,14,21]. One simple way to solve this problem is to aggregate multiple responses for each task from different workers. Such aggregation can helps us elicit the wisdom of crowds instead of relying on a single low-paid worker [12].

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12085, pp. 261–273, 2020.
https://doi.org/10.1007/978-3-030-47436-2_20



(a) Movie rating

(b) Object localization



(c) Pose estimation

Fig. 1. Applications of the regression tasks in crowdsourcing. (a) Movie rating: to score movies from 0 to 100. (b) Image object localization: to draw a tight bounding box capturing the target object. (c) Pose estimation: to find the proper positions of the skeleton's joints.

Over the years, several papers have proposed aggregation methods and verified theoretical bounds for binary-choice tasks [1,3,9] and discrete multiple-choice tasks [2,7,19]. However, most of recent crowdsourcing tasks ask workers to solve a problem with vectors. Actually, in web-based crowdsourcing platforms such as Amazon Mechanical Turk and CrowdFlower, a considerable number of requesters ask workers to solve vector regression tasks. (ex Monthly statistics for June 2019, about 22%) As described in Fig. 1, the examples of vector regression tasks are as follow: (1) Rating movies or items, (2) Finding the location of an object in an image, and (3) Estimating a human posture in an image.

There have been studies to devise an inference algorithm for regression tasks. [12] extended their binary classification model to learn a simple linear regressor. As for Expectation Maximization (EM) methods, [18] and [13] proposed a probabilistic graphical model for image object localization. However, those models have a difficulty in learning parameters with relatively small number of responses.

In this paper, we propose an iterative algorithm for inferring true answers from noisy responses in vector regression tasks. As in many previous works [3,13,18,19], we also consider the "*reliability*" of a worker represented by a parameter indicating the worker's expertise level and ability. Our algorithm computes two types of messages alternately. First, the worker message estimates the reliability of each worker, and the task message computes the weighted averages of their responses using those reliabilities as weights. These processes contribute to infer more accurate answers by sorting the order of responses by importance. Then we prove the error bound of our algorithm's average performance based on a probabilistic crowd model. This result shows our algorithm achieves better performance than other existing algorithms with a small number of queries and comparatively low average quality of the crowd. Furthermore, we provide that under a certain condition, the ℓ_2 error performance of ours is close to that of an oracle estimator which knows the reliability of every worker. Through extensive

experiments, we empirically verify that our algorithm outperforms other existing algorithms for both real world datasets crowd-sourced from *CrowdFlower*, and synthetic datasets (Table 1).

Table 1. Comparisons of the types of tasks covered by well-known crowdsourcing algorithms

Source	Binary	Multi-class	Regression
Dawid and Skene [2]	✓	✓	
Whitehill et al. [19]	✓		
Welinder et al. [18]	✓		✓
Raykar et al. [12]	✓	✓	✓
Karger et al. [3]	✓		
Liu et al. [9]	✓		
Dalvi et al. [1]	✓		
Salek et al. [13]			✓
Karger et al. [4]	✓	✓	
Zhang et al. [20]	✓	✓	
Lee et al. [7]	✓	✓	

Related Work. For aggregation methods, majority voting is a widely used for its simplicity and intuitiveness. [6] shows majority voting can effectively reduce the error in the attribute-based setting. However, it regards every worker as equally reliable and gives an identical weight to all responses. Therefore, the performance of majority voting suffers even with a small number of erroneous responses [14]. To overcome this limitation, there have been several approaches for improving the inference performance from unreliable responses. [2,18,19] adopt Expectation and Maximization (EM) to evaluate the implicit characteristics of tasks and workers. Also, [20] improves this EM approach using a spectral method with performance guarantees. However, in practice, there is a difficulty in parameter estimation since these EM approaches are aimed at estimating a huge confusion matrix from relatively few responses.

[3,9] proposed Belief Propagation (BP)-based iterative algorithms and proved that their error performances are bounded by worker quality and the number of queries in binary-choice tasks. Furthermore, there are several researches for crowdsourcing systems with multiple-choice tasks. [4] focused on multi-class labeling using a spectral method with low rank approximation, [22] proposed an aggregating method with minimax conditional entropy and [17] suggested an aggregation method using a decoding algorithm of coding theory. In addition, [7] exploits a inner product method ($\mathcal{IP}$) for evaluating similarity measures between an answer from a worker and the group consensus.

There have been studies to target vector regression tasks: [16] and the DALE model in [13], which focus on finding the location of a bounding box in an image. The former suggests a simple serial task assignment method for a quality-controlled crowdsourcing system with no theoretical guarantee. The latter proposes a probabilistic graphical model for image object localization and inference method with expectation propagation. However, the worker model assumption in these papers has two limitations; it strictly divides the workers' expertise level and ignores the order of selection when a crowd divides a length into multiple segments. Also, the latter graphical model has too many parameters to learn from relatively small number of responses.

On the other hand, there are *outlier rejection* methods that can be used to filter unreliable responses without a graphical model. For non-parametric setting, *mean shift* and *top-k selection* are typically used as classical methods. *mean shift* is the technique for locating the maxima of a density function and *top-k selection* picks k most reliable responses based on distances between the mean vector and each response itself. For parametric setting, *RANSAC* (random sample consensus) is widely used. it is an iterative method to estimate parameters of a mathematical model from a set of responses that contains outliers, when they are to be accorded no influence on the values of the estimates.

While most of the papers mentioned above assume random regular task assignments, [1,10] proposed inference methods in irregular task assignments. Also, [4,7,11] suggested the adaptive task assignment which gives more tasks to more reliable workers in order to infer more accurate answers given a limited budget.

2 Preliminaries

In this section, we describe a problem setup with variables and notations. First, we assume that there are m tasks in total and each task i is assigned to distinct l_i workers. Similarly, there are n workers in total and each worker j solves different r_j tasks. Here and after, we use $[N]$ to denote the set of first N integers. If we regard tasks and workers as set of vertices and connect the edge $(i, j) \in E$ when the task i is assigned to the worker j, our system can be described as a bipartite graph $G = \{[m], [n], E\}$ in Fig. 2.

Our crowdsourcing system considers a specific type of task whose answer space spans a finite continuous domain. If a task asks D number of real values, a response $\tilde{\boldsymbol{A}}$ is a D-dimensional vector. On one task node i, given all of responses $\{\tilde{\boldsymbol{A}}_{ij} | (i, j) \in E\}$, we transform them to $\boldsymbol{A}$ subject to $\|\boldsymbol{A}_{ij}\|_1 = 1$ by the min-max normalization since each task can have a different domain length.

For a simple example, in an image object localization regression task, a response is a bounding box to capture the target object. Considering the x axis only for brevity, the box coordinate is $\tilde{\boldsymbol{A}} = [x_{tl}, x_{br}]$, where x_{tl} and x_{br} stand for the top-left and bottom-right coordinates. Then it can be transformed as

$$\boldsymbol{A} = \big(x_{tl}, x_{br} - x_{tl}, x_{max} - x_{br}\big)/x_{max},$$

where x_{max} represents the width of the image. Since images have different size of width and height, all responses are transformed to have the same domain length.

In summary, when the worker j solves the task i, the response is denoted as $\tilde{\boldsymbol{A}}_{ij} \in \mathbb{R}^D$ and transformed to $\boldsymbol{A}_{ij} \in \mathbb{R}^{D+1}$ with respect to $\|\boldsymbol{A}_{ij}\|_1 = 1$. For convenience, δ_i and δ_j denotes the group of workers who give responses to the task i and the group of tasks which are assigned to worker j respectively.

Majority Voting ($\mathcal{MV}$). The simplest method in response aggregation is majority voting, well-known sub-optimal estimator, which computes the centroid of responses. However, its performance can be easily degraded whether there exist a few adversarial workers or spammers who give wrong answers intentionally or random answers respectively (Fig. 3).

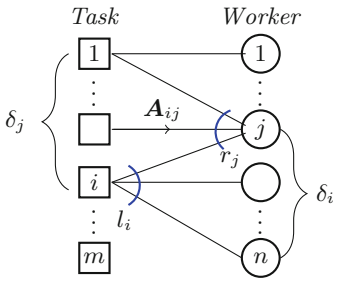


Fig. 2. System model for task-worker assignments.

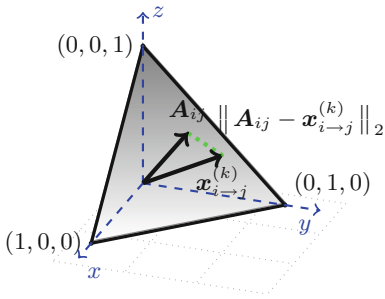


Fig. 3. Distance between answer $\boldsymbol{A}_{ij}$ and x message $\boldsymbol{x}_{i\to j}$ in the standard 2-simplex space when $D_i = 2$.

Majority voting method gives the identical weight to every worker who annotates the task for fixed task i.

$$\hat{\boldsymbol{t}}_i^{(\mathcal{MV})} = \sum_{j \in \delta_i} \frac{1}{l_i} \boldsymbol{A}_{ij}. \tag{1}$$

3 Inference Algorithm

In this section, we propose a message-passing algorithm for vector regression tasks. Our iterative algorithm alternatively estimates two types of messages: (1) task messages $\boldsymbol{x}_{i\to j}$, and worker messages $\boldsymbol{y}_{j\to i}$. This updating process estimates the ground truth of each task and the reliability of each worker respectively. From now on, $\hat{l_i}$ and $\hat{r_j}$ denote $(l_i - 1)$ and $(r_j - 1)$ respectively for brevity.

3.1 Task Message

We first describe a task message that estimates the current candidate of a ground truth. It simply computes the centroid of weighted responses from the workers assigned to the task. Thus, it can be viewed as a simple estimator of weighted voting in that those weights are computed according to how workers are reliable. Note that a task message $\boldsymbol{x}_{i\to j}$ averages weighted responses from workers assigned to a task i except for the response from worker j. This helps to block any correlation between the task message and the responses from worker j.

$$\boldsymbol{x}_{i\to j}^{(k)} = \sum_{j'\in\delta_i\setminus j} \left(\frac{y_{j'\to i}^{(k-1)}}{y_{\delta_i\setminus j}^{(k-1)}} \right) \boldsymbol{A}_{ij'}, \tag{2}$$

where $y_{\delta_i\setminus j}^{(k-1)} = \sum_{j'\in\delta_i\setminus j} y_{j'\to i}^{(k-1)}$.

Algorithm 1. Inference Algorithm

Input: $G = \{[m], [n], E\}$, $\{\boldsymbol{A}_{ij}\}_{(i,j)\in E}$, k_{max}
Output: Estimated truths $\hat{\boldsymbol{t}}_i$, $^{\forall} i \in [m]$

Initialization
for $^{\forall}(i,j) \in E$ **do**
  $y_{j\to i}^{(0)} \leftarrow \mathcal{N}(0,1)$
Iteration Step
for $k = 1$ **to** k_{max} **do**
  for $^{\forall}(i,j) \in E$ **do**
    Update task message, $\boldsymbol{x}_{i\to j}^{(k)}$ using Eq. 2
  for $^{\forall}(i,j) \in E$ **do**
    Update worker message, $y_{j\to i}^{(k)}$ using Eq. 3
end for
Final Estimation
  for $^{\forall} j \in [n]$ **do**
    $y_j \leftarrow \left(\frac{1}{r_j} \sum_{i\in\delta_j} (\|\boldsymbol{A}_{ij} - \boldsymbol{x}_{i\to j}^{(k_{max})}\|_2)^2 \right)^{-1}$
  for $^{\forall} i \in [m]$ **do**
    $\boldsymbol{x}_i \leftarrow \sum_{j\in\delta_i} \left(\frac{y_j}{y_{\delta_i}} \right) \boldsymbol{A}_{ij}$
return $\hat{\boldsymbol{t}}_i^{(\mathcal{ALG})} \leftarrow \boldsymbol{x}_i$, $^{\forall} i \in [m]$

3.2 Worker Message

The next step is to compute worker messages $y_{j\to i}$ which represents the importance of response $\boldsymbol{A}_{ij}$. These worker messages are used as weights in the weighted voting process in task messages update. Since it is desirable to give a higher weight to more reliable workers, each worker's reliability should be evaluated as

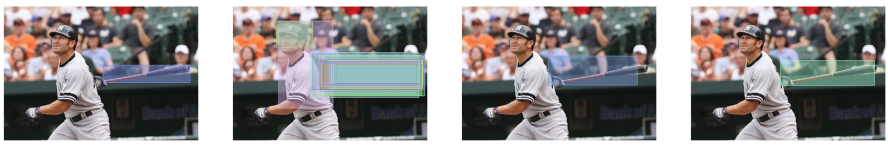
the similarity between his response and the task message which indicates the consensus of other workers' responses. In our algorithms, it takes advantage of the reciprocal of the summation of the euclidean distance between the response and the task message as a similarity measure. In analysis section, our analysis verify that this measure is proper to estimate weights of workers' responses. Note that a worker message $y_{j\to i}$ represents the average of similarities between worker j's responses and the average response of other workers' responses in the same task.

$$y_{j\to i}^{(k)} = \left(\frac{1}{\hat{r}_j} \sum_{i' \in \delta_j \setminus i} \left(\| \boldsymbol{A}_{i'j} - \boldsymbol{x}_{i'\to j}^{(k)} \|_2 \right)^2 \right)^{-1}. \tag{3}$$

In the worker message update (3), we adopt the reciprocal of ℓ_2 norm in the vector space as a similarity measure. However, our algorithm can be generalized with any metric induced by other norm and similarity function which is continuous and monotonically decreasing.

4 Experimental Results

In our experiments, we have evaluated the performance of our algorithm with two popular benchmarks, MSCOCO [8] and the Leeds Sports Pose Extended Training (LSPET) datasets. We compare our algorithm with baselines algorithms which are majority voting ($\mathcal{MV}$) and weighted voting ($\mathcal{WV}$) whose weights are externally given by web-based crowdsourcing platform. We also implemented several state-of-the art which are inner-product method ($\mathcal{IP}$) [7], Welinder's EM model [18], DALE model [13], and outlier rejection methods which are *Mean shift* and *Top-K* selection (Fig. 4).



(a) Ground truth (b) Responses (c) MV (d) Ours

Fig. 4. Drawing a bounding box task on the 'bat'. (a) the ground truth (b) bounding boxes drawn by 25 workers. (c) Estimated answer of majority voting. (d) Estimated answer of our algorithm.

4.1 Real Crowdsourcing Data

We crowdsourced two types of tasks in CrowdFlower. One is for image object localization in which the task is to draw a bounding box on the specified object as tightly as possible. The other one is for human pose estimation, where the task is to construct a skeleton-like structure of a human in a given image.

Bounding Box on MSCOCO Dataset. In this task, we randomly chose 2,000 arbitrary images from MSCOCO dataset, and each image was distributed to 25 distinct workers, so there were 50,000 tasks to be solved in total. Total 618 workers were employed, and each worker solved 10 (min) to 100 (max) tasks. We exclude some invalid responses (no box, box over out of bounds [0, image size]). Note that a general bipartite graph is created with different node degrees l_i and r_j, which is not a regular bipartite graph. We measured algorithms' performances by the average error in the ℓ_2 norm and the Intersection over Union (IoU), which is another standard measure for object localization computed by a ratio of intersection area to union area of two bounding boxes. In this experiment, DALE model does not converge due to its complex graphical model raising an out of memory error.

To measure the performance of DALE model in smaller data, we collected a dedicated dataset of 100 images each of which was assigned to 20 distinct workers. Results are listed in Table 2 with two evaluation metric Euclidean distance(ℓ_2) and Intersection over Union(IoU). Our algorithm significantly outperforms others and, even with small number of iterations, can reduce errors rapidly. Empirically, our algorithm converges in less than 20 iterations as plotted in Fig. 6.

Table 2. An error table of experimental results on real crowdsourced data where the tasks are (1st column) an object detection on MSCOCO dataset, (2nd column) same task with Intersection of Union measure (3rd column) a human joints estimation and (4th column) an angle segmentation by neck and adjacent human joints on LSPET dataset. For Top-K selection, we choose K as a half of the task degree l.

Dataset	MSCOCO		LSPET	
Type	Box(ℓ_2)	Box(IoU)	Joints	Angles
$\mathcal{WV}$	0.22227	0.89593	0.15877	0.10524
$\mathcal{MV}$	0.22090	0.89666	0.15858	0.10462
$\mathcal{IP}$	0.22026	0.89712	0.15483	0.10462
Welinder	0.21886	0.89821	N/A	N/A
DALE	0.21834	0.89914	N/A	N/A
Top-K	0.18869	0.91250	0.12222	0.10051
MeanShift	0.18034	0.92150	0.11812	0.09962
Ours	**0.14837**	**0.93445**	**0.09308**	**0.09941**

Varying Degree on MSCOCO Dataset. Here we show how the performances of different algorithms vary with task degree l. We made a number of task-worker bipartite graphs by randomly dropping some edges to make degree l for each task. As expected, the average error of each algorithm decreases as the task degree l increases. Even when the degree value falls until 5, ours can still keep the large gap among other algorithms. In other words, our algorithm needs less budget to get same error rate. The results are listed in Fig. 5.

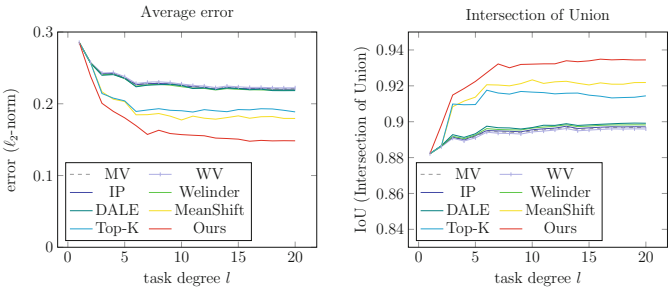


Fig. 5. Comparisons of error and IoU between different algorithms with varying task degree l.

Robustness. Since it is well known that message-passing algorithms suffers from the initialization issue in general, we tested robustness of our algorithm by initializing workers' weights to be sampled from proper distributions with moderate hyperparameters. Here we used *Beta* distribution with (α, β), and *Gaussian* distribution with (μ, σ^2) sampled from uniform distribution $\mathcal{U}$. The result is shown by error bar plots in Fig. 6 which represents the deviation reduces rapidly. This result shows that our algorithm is robust to the initialization of workers' weights.

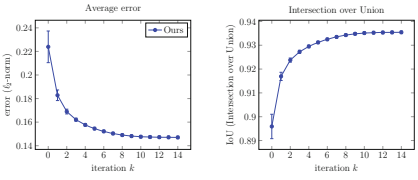


Fig. 6. Error bar plots of our algorithm for the initialization issue on 2k-edge bounding box task.

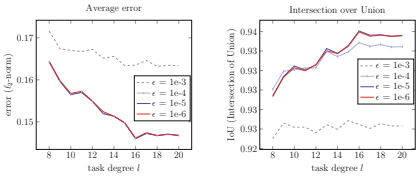


Fig. 7. The influence of ϵ on error and IoU when computing y-messages with varying task degree l.

When the number of edges are not sufficient to estimate worker message, our algorithm can diverge as iteration progresses since worker message is computed by the reciprocal of the summation between the response and the task message. It can be resolved by adding a very small positive constant ϵ on the summation before computing the reciprocal.

$$y_{j\to i}^{(k)} = \left(\frac{1}{\hat{r}} \sum_{i' \in \delta_{j \setminus i}} \left(\| \boldsymbol{A}_{ij} - \boldsymbol{x}_{i\to j}^{(k)} \|_2 \right)^2 \right) + \epsilon \Bigg)^{-1}. \tag{4}$$

We investigate the influence of ϵ in Fig. 7. This result shows our algorithm works well when $\epsilon \leq 10^{-5}$.

Human Pose Estimation. We collected the human pose estimation data of 1,000 images chosen from LSPET dataset using CrowdFlower platform. Each image was distributed to ten distinct workers who were asked to mark dots on the 14 human joints (head, neck, left/right shoulders, elbows, wrists, hips, knees, and ankles). In this experiment, we aggregated their answers to estimate the point of each human joint. Moreover, we estimated angles from the neck and adjacent joints (head, shoulders, hips) as another task which is also important in pose estimation. Estimating angles can be viewed as dividing angle task whose domain is $[0, 2\pi]$. As shown in Table 2, our algorithm outperforms others on both joint and angle estimation tasks.

5 Performance Analysis

In this section, we analyze the average performance of our algorithm using a probabilistic crowd model called "Dirichlet" crowd model (in Appendix Sect. 6).

5.1 Error Bound

Theorem 1. *For fixed $l > 1$, $r > 1$ and dimension $D \geqslant 1$, assume that m tasks are assigned to n workers according to a random (l, r)-regular bipartite graph. If the average quality satisfies $q > (1 + (D+1)/\hat{l}\hat{r})$, then when $k \to \infty$ the average error of the our algorithm achieves*

$$\mathrm{E}_{\mathcal{ALG}} \leqslant \left(\frac{(1 + 1/\hat{l}\hat{r})^2}{(\sqrt{2}+1)q\hat{r}} \right) \cdot \frac{1}{\hat{l}m} \sum_{i \in [m]} T_i. \tag{5}$$

This result implies that we can control the error performance by adjusting the average quality of workers and the number of queries assigned to each task. As q and lr increase, the upper bound of our algorithm becomes lower.

Proof Sketch. We consider any worker distribution with the average quality q. Under this worker distribution, our strategy is to inspect the average behavior of worker messages, $\mathbb{E}\big[y_{j \to i}^{(\infty)}\big]$ as $k_{max} \to \infty$.

$$\left\{ \mathbb{W} | q^{-1} = \mathbb{E}_{\mathbb{W}} \Big[\frac{1}{\mathrm{w}+1} \Big] \right\}. \tag{6}$$

According to task and worker messages update processes, we compute the '*average message*' passed through edges of graph G. Then we look into the Probabilistic accuracy of the message.

Detailed proof of Theorem 1 will be omitted here but the whole process of the proof is provided in the Appendix.

Corollary 1. *Under the hypotheses of Theorem 1, if the distribution of the reliability satisfies*

$$\mathbb{P}\Big((\mathrm{w}+1) \geqslant 2\mu_w\Big) \leqslant \frac{(\sqrt{2}+1)\hat{l}\hat{r}}{l(1+1/(\hat{l}\hat{r}))^2},$$

and symmetrical, then the upper bound of $\mathrm{E}_{\mathcal{ALG}}$ *is close to the oracle estimator's average performance.*

$$\mathrm{U}_{\mathcal{ALG}} \to \mathrm{E}_{\mathcal{OC}}. \tag{7}$$

5.2 Verification of Theorem with Synthetic Data

In order to empirically verify the correctness of the analysis, experiments were performed with synthetic dataset. Assuming hypothetical 2000 workers and 2000 tasks with two dimensions ($D = 2, 5$), task assignment follows regular bipartite graph. The performance of the oracle estimator is presented as a theoretical lower bound. Also, each result is averaged of 20 experiments by changing the initial value.

Spammer/Hammer Ratio. In this experiment, we assume the *Spammer/Hammer* scenario which means that each worker is randomly sampled from a Spammer ($w_s = 0.5$) or a Hammer ($w_h = 5$); the response of a Hammer is much closer to the ground truth than that of a Spammer. The ratio γ denotes the Hammer proportion of all workers. Figure 8 (left) shows that our algorithm can distinguish Hammer from Spammer much better than others.

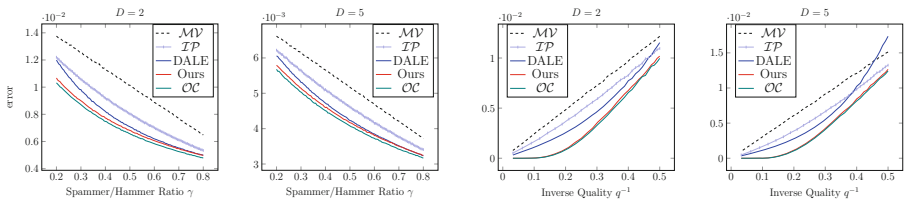


Fig. 8. Comparison of average errors between different algorithms with $D = (2, 5)$: (*top*) varying l with fixed q, (*mid*) varying γ ($w_s = 0.5$, $w_h = 5$), (*bottom*) varying q.

Quality. According to the definition of (6), the reliability of each worker was drawn from *Beta* distribution (i.e., $(1+w)^{-1} \sim Beta(\alpha, \beta)$). In Fig. 8 (right), our algorithm shows a large performance gap when the quality is sufficiently high. The average errors of the five algorithms are indistinguishable when the quality is low, but our algorithm is better at estimating the workers' reliabilities if the quality is sufficiently high. Since our algorithm regards the average response of other workers as approximated true answers, high quality promotes its performance.

6 Conclusion

In this paper, we have proposed an iterative algorithm for vector regression tasks. We observed the considerable gains with both real and synthetic datasets through various experiments. In the theoretical analysis, we proved that the error bound depends on the average worker quality and the number of queries batch achieving near-optimal performance in the probabilistic worker model. Our work can be easily generalized to many image processing tasks such as 3D image processing and multiple object detection. Also, it can be exploited for estimating the precise level of workers in an adaptive manner.

References

1. Dalvi, N., Dasgupta, A., Kumar, R., Rastogi, V.: Aggregating crowdsourced binary ratings. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 285–294. ACM (2013)
2. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. Appl. Stat. **28**, 20–28 (1979)
3. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: Advances in Neural Information Processing Systems, pp. 1953–1961 (2011)
4. Karger, D.R., Oh, S., Shah, D.: Efficient crowdsourcing for multi-class labeling. In: Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems, pp. 81–92. ACM (2013)
5. Kazai, G., Kamps, J., Milic-Frayling, N.: An analysis of human factors and label accuracy in crowdsourcing relevance judgments. Inf. Retrieval **16**(2), 138–178 (2013). https://doi.org/10.1007/s10791-012-9205-0
6. Khan, A.R., Garcia-Molina, H.: Attribute-based crowd entity resolution. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 549–558. ACM (2016)
7. Lee, D., Kim, J., Lee, H., Jung, K.: Reliable multiple-choice iterative algorithm for crowdsourcing systems. In: Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 205–216. ACM (2015)
8. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
9. Liu, Q., Peng, J., Ihler, A.: Variational inference for crowdsourcing. In: Advances in Neural Information Processing Systems, pp. 692–700 (2012)
10. Ma, Y., Olshevsky, A., Saligrama, V., Szepesvari, C.: Crowdsourcing with sparsely interacting workers. arXiv preprint arXiv:1706.06660 (2017)
11. Qiu, C., Squicciarini, A.C., Carminati, B., Caverlee, J., Khare, D.R.: CrowdSelect: increasing accuracy of crowdsourcing tasks through behavior prediction and user selection. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 539–548. ACM (2016)
12. Raykar, V.C., et al.: Learning from crowds. J. Mach. Learn. Res. **11**, 1297–1322 (2010)
13. Salek, M., Bachrach, Y.: Hotspotting-a probabilistic graphical model for image object localization through crowdsourcing. In: AAAI (2013)

14. Sheng, V.S., Provost, F., Ipeirotis, P.G.: 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, pp. 614–622. ACM (2008)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. Su, H., Deng, J., Fei-Fei, L.: Crowdsourcing annotations for visual object detection. In: Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence, vol. 1 (2012)
17. Vempaty, A., Varshney, L., Varshney, P.: Reliable crowdsourcing for multi-class labeling using coding theory. IEEE J. Sel. Top. Sign. Process. **8**(4), 667–679 (2014)
18. Welinder, P., Perona, P.: Online crowdsourcing: rating annotators and obtaining cost-effective labels. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 25–32. IEEE (2010)
19. Whitehill, J., Wu, T.f., Bergsma, J., Movellan, J.R., Ruvolo, P.L.: Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In: Advances in Neural Information Processing Systems, pp. 2035–2043 (2009)
20. Zhang, Y., Chen, X., Zhou, D., Jordan, M.I.: Spectral methods meet EM: a provably optimal algorithm for crowdsourcing. In: Advances in Neural Information Processing Systems, pp. 1260–1268 (2014)
21. Zheng, Y., Scott, S., Deng, K.: Active learning from multiple noisy labelers with varied costs. In: 2010 IEEE 10th International Conference on Data Mining (ICDM), pp. 639–648. IEEE (2010)
22. Zhou, D., Liu, Q., Platt, J.C., Meek, C.: Aggregating ordinal labels from crowds by minimax conditional entropy. In: ICML, vol. 14, pp. 262–270 (2014)