Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Statistical Quality Control for Human Computation and Crowdsourcing

Yukino Baba
University of Tsukuba
baba@cs.tsukuba.ac.jp

Abstract

Human computation is a method for solving difficult problems by combining humans and computers. Quality control is a critical issue in human computation because it relies on a large number of participants (i.e., crowds) and there is an uncertainty about their reliability. A solution for this issue is to leverage the power of the "wisdom of crowds"; for example, we can aggregate the outputs of multiple participants or ask a participant to check the output of another participant to improve its quality. In this paper, we review several statistical approaches for controlling the quality of outputs from crowds.

1 Introduction

Human computation is a method for solving difficult problems by combining humans and computers. By querying human intelligence from computer systems, human computation solves problems that cannot be solved by either humans or computers individually. An example application of human computation is reCAPTCHA, which is designed to digitize a large number of books by involving humans in character recognition [von Ahn *et al.*, 2008]. Another example is VizWiz, a mobile application for assisting blind people [Bigham *et al.*, 2010]. When a user takes a picture and asks a question, this application returns answers provided by humans.

To achieve a high throughput, human computation requires the participation of a large number of people. Crowdsourcing marketplaces such as Amazon Mechanical Turk or CrowdFlower provide an access to human intellect; by using crowdsourcing, one can therefore easily outsource tasks to crowds of people through the Internet. The expansion of crowdsourcing marketplaces has thus played an important role for evolving research on human computation.

One major challenge in human computation is how to handle uncertainty as well as the variety of crowds. Quality control is a particularly critical issue because all people make mistakes and have different levels of reliability. A solution to this issue is to leverage the power of the "wisdom of crowds"; for example, we can aggregate the outputs from multiple workers (i.e., participants in human computation) or ask a worker to check the output of another worker to improve its quality. In this paper, we review several statistical approaches for controlling the quality of outputs from crowds.

2 Quality Control for Simple Tasks

2.1 Multiple-choice Questions

A multiple-choice question is a typical format for querying workers. Reliable answers can be obtained by asking the same question to multiple workers and applying majority voting. In addition to this simple aggregation method, various sophisticated statistical methods have been proposed [Dawid and Skene, 1979; Whitehill *et al.*, 2009; Welinder *et al.*, 2010; Bachrach *et al.*, 2012; Venanzi *et al.*, 2014]. These methods estimate the reliability of each worker and aggregate the results. Specifically, they assume that workers who provide majority answers are reliable workers (i.e., the opinions of the majority are strengthened). Consequently, most such approaches often fail when the majority of workers provide wrong answers. These cases occur when tasks require highly specialized knowledge and the reliability of a large majority of the workers is inadequate.

Targeting Experts with Hyper Questions

In [Li *et al.*, 2017], we proposed a method for targeting reliable workers who are overwhelmed by the large number of unreliable workers. Our key idea is to focus on sets of questions (which we call *hyper questions*) instead of individual questions. Let us assume that two types of workers exist, experts and non-experts; experts are likely to provide correct answers and non-experts are likely to guess randomly. When we consider multiple questions, non-experts are thus less likely to reach a consensus than experts. This fact enables us to let experts win in taking the majority voting on multiple questions.

Our method, called *Hyper-MV*, takes the majority voting on hyper questions. A hyper question consists of a subset of the original single questions. We specifically consider a k-hyper question, which contains k single questions. For example, when we have a set of four questions, $\{1, 2, 3, 4\}$, $\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}$, and $\{2, 3, 4\}$ are 3-hyper questions. Given k, our method first enumerates all possible k-hyper questions and then transforms answers to single questions into answers to the hyper questions. Each answer to a hyper question is represented by the concatenation of the

answers to the single questions included in the hyper question. For example, when a worker answers questions 1, 2, and 3 with "A," "B," and "C," respectively, the answer to the hyper question $\{1, 2, 3\}$ is "ABC." Hyper-MV then performs the majority voting on each hyper question, which results in an answer to the hyper question. The aggregated results of the hyper questions are next decoded into answers to the single questions. Finally, another round of majority voting is carried out for each single question to obtain the final aggregated answer to each one. It is worth noticing that the first round of majority voting on hyper questions can be replaced with other aggregation methods, such as GLAD [Whitehill *et al.*, 2009] or DARE [Bachrach *et al.*, 2012].

Targeting Experts by Confidence Reports

Another solution for strengthening the answers provided by experts is to ask each worker for his/her level of confidence in his/her responses. This confidence information is useful for finding reliable workers from crowds; however, some workers can be overconfident and report a high level of confidence even though they provide incorrect answers. Additionally, some workers are underconfident and report a low level of confidence even when their answers are correct. To leverage this confidence information for quality control purposes, we proposed a statistical method for handling the characteristics of the confidence report of each worker [Oyama *et al.*, 2013].

Our method is based on two generative models, the answer model and the confidence model. For simplicity, we consider binary questions and we denote the true answer to the i-th question by $t_i \in \{0, 1\}$ and the answer of the j-th worker to the i-th question by $y_{ij} \in \{0, 1\}$. Similar to the Dawid–Skene model [Dawid and Skene, 1979], we assume that each worker has a reliability parameter $\boldsymbol{\alpha}^{(j)} = \{\alpha_0^{(j)}, \alpha_1^{(j)}\}$, where $\alpha_0^{(j)}$ and $\alpha_1^{(j)}$ are the probabilities that the j-th worker answers $y_{ij} = 1$ when the true answer is $t_i = 0$ and $t_i = 1$, respectively. The answer model is formalized as follows:

$$\Pr[y_{ij} \mid t_i = 1] = \left(\alpha_1^{(j)}\right)^{y_{ij}} \left(1 - \alpha_1^{(j)}\right)^{1-y_{ij}} \quad (1)$$

$$\Pr[y_{ij} \mid t_i = 1] = \left(\alpha_0^{(j)}\right)^{y_{ij}} \left(1 - \alpha_0^{(j)}\right)^{1-y_{ij}}. \quad (2)$$

A worker chooses his/her confidence level (i.e., high or low) depending on his/her answer and the true answer, and the confidence level is denoted by $c_{ij} \in \{0, 1\}$ where $c_{ij} = 1$ indicates that the j-th worker reports a high confidence in the i-th question. We introduce a confidence parameter for each worker $\boldsymbol{\beta}^{(j)} = \{\beta_{00}^{(j)}, \beta_{01}^{(j)}, \beta_{10}^{(j)}, \beta_{11}^{(j)}\}$. For example, $\beta_{01}^{(j)}$ is the probability that the j-th worker reports high confidence (i.e., $c_{ij} = 1$) when the true answer is $t_i = 0$ and the answer from the worker is $y_{ij} = 1$. Our confidence model is given as follows:

$$\Pr[c_{ij} \mid t_i = 0, y_{ij} = 1] = \left(\beta_{01}^{(j)}\right)^{c_{ij}} \left(1 - \beta_{01}^{(j)}\right)^{1-c_{ij}}. \quad (3)$$

The probabilities $\Pr[c_{ij} \mid t_i = 0, y_{ij} = 0]$, $\Pr[c_{ij} \mid t_i = 1, y_{ij} = 0]$, and $\Pr[c_{ij} \mid t_i = 1, y_{ij} = 1]$ are given in the same manner. This confidence model enables us to capture worker characteristics in the confidence report and thus strengthen reliable workers in the aggregated answers.

2.2 Other Types of Questions

In addition to multiple-choice questions, several question formats can be used to query workers, and statistical quality control methods have been proposed for them, such as pairwise comparison tasks [Chen *et al.*, 2013], numerical response tasks [Lin *et al.*, 2012], ordinal response tasks [Raykar and Yu, 2011], and sequence labeling tasks [Wu *et al.*, 2012].

In [Matsui *et al.*, 2014], we proposed an aggregation method for ranking tasks, in which workers are asked to arrange given objects in the correct order. We assume that there is the (latent) true order $\boldsymbol{\pi}_i$ for the i-th task and that each worker has a reliability parameter λ_j. We model the generative process of the worker's answer as follows:

$$\Pr[\boldsymbol{\pi}_{ij} \mid \boldsymbol{\pi}_i] = \frac{1}{Z(\lambda_j)} \exp\left(-\lambda_j d(\boldsymbol{\pi}_{ij}, \boldsymbol{\pi}_i)\right), \quad (4)$$

where $\boldsymbol{\pi}_{ij}$ is the answer of the j-th worker for the i-th task, $d(\cdot, \cdot)$ denotes the distance between the two rank vectors, and $Z(\lambda)$ is a normalizing constant given as $Z(\lambda) = \sum_{\boldsymbol{\pi}'} \exp\left(-\lambda d(\boldsymbol{\pi}', \boldsymbol{\pi})\right)$. Specifically, we employ the Euclidean distance (also referred to as the Spearman distance in the ranking model literature). Given the answers from workers, we estimate the true order of each task. We also developed statistical quality control methods for other types of tasks such as hierarchical labeling [Otani *et al.*, 2015] and collecting points of interest [Kajimura *et al.*, 2015].

3 Quality Control for Complex Tasks

3.1 General Tasks

Statistical quality control approaches for simple response tasks aggregate workers' answers. Hence, such aggregation approaches are unsuitable for complex response tasks, such as providing translations and designing logos. An alternative approach is to choose the highest quality output among those given by multiple workers for the same tasks.

Two-stage Workflow with Grading

To estimate the quality of each output, we proposed a two-stage workflow [Baba and Kashima, 2013]. In the first stage, each worker (called the *author*) creates an output for the task. Then, authors' outputs are processed in the second stage, where each output is graded by another set of workers (called *reviewers*); each reviewer chooses a grade from the given set (e.g., a five-point scale). Although it is inevitably difficult to directly estimate the quality of the output, this two-stage workflow enables us to indirectly estimate the quality from the grades and distinguish high quality outputs from the others.

Because there is no guarantee that all reviewers have sufficient reliability, we introduce statistical models for estimating the characteristics of both reviewers and authors. We assume that an author with higher ability creates higher quality outputs on average; hence, each author has the ability μ_j. We also assume that a reliable author shows stable performance for different tasks. The reliability of the j-th author is represented by λ_j. Our generative model for the first stage formalizes the generative process of the quality of an output as follows:

$$q_{ij} \sim \mathcal{N}\left(q_{ij} \mid \mu_j, \lambda_j^{-1}\right), \quad (5)$$

where q_{ij} is the (latent) true quality of the output created by the j-th author for the i-th task.

In the review stage, we assume that each reviewer has a bias $\eta_k \in \mathbb{R}$, assuming that a reviewer with a lower (higher) bias tends to give lower (higher) grades to the given output. We also incorporate the reliability of the k-th reviewer, τ_k. If reviewers have low reliability, their grades are likely to contain noise. In our model, when a reviewer grades an output, the reviewer first estimates the (latent) quality score of the output, $s_{ijk} \in \mathbb{R}$. The generative model for the quality score is represented as follows:

$$s_{ijk} \sim \mathcal{N}\left(s_{ijk} \mid q_{ij} + \eta_k, \tau_k^{-1}\right). \tag{6}$$

Then the reviewer returns the final grade label $g_{ijk} \in \{1, 2, \ldots, n\}$. Since the final grade label is a discrete value depending on the quality score, we apply $\Pr\left[g_{ijk} = d \mid s_{ijk}\right]$ which is the conditional probability of selecting d given the quality score s_{ijk}. For modeling this probability, we employ the graded response model [Samejima, 1969]. In this model, the conditional probability of a graded response is decomposed by using $n-1$ binary response models, namely,

$$\begin{aligned}&\Pr\left[g_{ijk} = d \mid s_{ijk}\right] \\ &\quad = \Pr[g_{ijk} > d-1 \mid s_{ijk}] - \Pr[g_{ijk} > d \mid s_{ijk}],\end{aligned} \tag{7}$$

where $\Pr[g_{ijk} > 0 \mid s_{ijk}] = 1$ and $\Pr[g_{ijk} > n \mid s_{ijk}] = 0$. The Rasch model [Rasch, 1960] is applied as a binary response model as follows:

$$\Pr[g_{ijk} > d \mid s_{ijk}] = \frac{1}{1 + \exp\left(-(s_{ijk} - b_d)\right)}, \tag{8}$$

where $\{b_d\}_d$ are threshold parameters. For simplicity, we set the thresholds $(b_1, b_2, \cdots, b_{n-1}) = (1, 2, \cdots, n-1)$. Given the grades from the reviewers, we can estimate the true quality of each output as well as the other model parameters.

Two-stage Workflow with Pairwise Comparison

We proposed another method for the two-stage workflow in which a reviewer compares a pair of outputs and votes for one of them [Sunahase *et al.*, 2017]. Pairwise comparison has several advantages over rating single outputs; for example. it can capture small differences in quality between outputs, and reviewers do not need to calibrate their standards over time. Our proposed method is an extension of the HITS algorithm [Kleinberg, 1999], and thus we call it *two-stage pairwise HITS*. Analogous to the hubs and authorities in the HITS algorithm, we assume that a good reviewer votes for many good outputs and that a good output is voted for by many good reviewers. We modify the HITS algorithm so that it is applicable to pairwise comparison.

Two-stage pairwise HITS aims to estimate the true quality of each output, q_i. We assume that each reviewer has a reliability parameter r_k, and the difference between the quality of the two outputs is calculated as the difference between the sum of reviewers' reliability voting for them in their pairwise comparison:

$$q_i - q_{i'} = \sum_{j \in V_{i,i'}} r_k - \sum_{j \in V_{i',i}} r_k, \tag{9}$$

where $V_{i,i'} = \{k \mid i \succ_k i'\}$ is the set of reviewers who prefer the i-th output to the i'-th output.

Meanwhile, we define the reliability of the k-th reviewer, r_k, as the proportion of his/her correct decisions out of all his/her comparisons:

$$r_k = \frac{\mid \{(i, i') \in V_k \mid q_i > q_{i'}\} \mid}{|V_k|}, \tag{10}$$

where $V_k = \{(i, i') \mid i \succ_k j'\}$ is the set of pairwise comparisons that the k-th reviewer makes.

In addition, we incorporate the author's ability, c_j, into the output quality as follows:

$$q_i = (1 - \lambda) q_i^* + \lambda c_{j_i}, \tag{11}$$

where q_i^* is the quality of the i-th output that is the solution of Eq. (9), j_i is the index of the author of the i-th output, and λ is the tradeoff parameter, where $0 < \lambda < 1$. We define the ability of an author as the average of the quality of his/her outputs; that is, the ability c_k of the j-th author is given as

$$c_j = \frac{1}{|W_j|} \sum_{i \in W_j} q_i, \tag{12}$$

where W_j is the set of outputs created by the j-th author. Our two-stage pairwise HITS algorithm iteratively calculates q_i, c_j, and r_k to estimate the true quality of each output.

3.2 Translation Tasks

We also studied a quality control method for language translation tasks in [Ehara *et al.*, 2016]. A typical method of estimating translation ability involves translation tests, in which a translator is asked to translate given sentences and professional reviewers grade the translation results. By contrast, our convenient method uses vocabulary tests to assess translation ability, meaning that we do not need to rely on professional reviewers. In the tests we designed, translators both translate several sentences as well as tell if they know the meaning of each word in the source language sentences. We then use the results of the tests to estimate the translator's ability.

We are given the contents of the test. In other words, we have each sentence in the translation test, $s = \{w_{s1}, w_{s2}, \ldots\}$, where w_{si} is a word in the sentence; each sentence s and word w are associated with a feature vector $\boldsymbol{s}$ and $\boldsymbol{w}$, respectively. We are also given the results of the vocabulary test $\{y_{kw}\}$; $y_{kw} = 1$ if translator k answers that he/she knows the meaning of the word and $y_{kw} = 0$ otherwise. In addition, some of the translation results can be evaluated by the professional reviewers: $z_{ks} = 1$ if the translation by translator k for sentence s is graded as acceptable and $z_{ks} = 0$ otherwise. We use the information to estimate the translator's ability.

Our method is based on two generative models: the translation ability model and vocabulary ability model. To build the former, we assume that translation quality varies according to the translation ability of the translator as well as the translation difficulty of the source language sentence. We model the translation ability of the translator k as ψ_k and the translation difficulty of sentence s as ν_s. By using the Rasch

model [Rasch, 1960], the translation ability model thus represents the probability that translator k acceptably translates sentence s as follows:

$$\Pr[z_{ks} = 1] = \frac{1}{1 + \exp(-(\psi_k - \nu_s))}. \quad (13)$$

We further model sentence difficulty ν_s as the inner product of the sentence feature vector $\boldsymbol{s}$ and a weight parameter $\boldsymbol{\tau}$, that is, $\nu_s = \boldsymbol{\tau}^\top \boldsymbol{s}$. Incorporating sentence feature vectors allows our model to estimate the probability of a translator providing an acceptable translation for a new sentence.

We next present the vocabulary ability model. We assume that the probability of a translator knowing the meaning of a word is dependent on his/her vocabulary ability and the difficulty of the word. We model the vocabulary ability of translator k as θ_k and the difficulty of word w as μ_w. Because we apply the self-report vocabulary test, there may be concerns that overconfident or unreliable translators are likely to answer that they know the meaning of most words. Thus, we introduce a parameter ϕ_k to model the unreliability of translator k. We assume that if translator k provides an acceptable translation of sentence s, the probability of translator k answering that he/she knows the meaning of word $w \in s$ depends on vocabulary ability θ_k. Otherwise, we model the probability by using ϕ_k. We apply the Rasch model and represent the probabilities, as follows:

$$\Pr[y_{kw} = 1 \mid z_{ks} = 1] = \frac{1}{1+\exp(-(\theta_k - \mu_w))} \quad (14)$$

$$\Pr[y_{kw} = 1 \mid z_{ks} = 0] = \frac{1}{1+\exp(-(\phi_k - \mu_w))}. \quad (15)$$

We also incorporate the word feature vectors into the vocabulary ability model by modeling the difficulty of word w as $\mu_w = \boldsymbol{\pi}^\top \boldsymbol{w}$, where $\boldsymbol{\pi}$ denotes a weight parameter. We further introduce a prior for $\psi_k \sim \mathcal{N}\left(\psi_k \mid \theta_k, \lambda^{-1}\right)$, which enforces the assumption that translation ability correlates with vocabulary ability.

4 Conclusion and Future Directions

In this paper, we reviewed statistical quality control methods for human computation and crowdsourcing. The presented methods implicitly assume that reliable workers agree on the correct answer; however, the correct answer may not be easily determined in practical situations such as when discussions between experts are required. In addition, there may be multiple correct answers owing to the variety of criteria. The presented quality control methods are not designed for such cases.

A promising solution is to ask workers to decide the correct answer(s) for themselves. For example, an approach has been proposed that workers could inspect the answers of others to make it easy for them to agree [Lorenz *et al.*, 2011]. Organizing the answers based on different criteria is also beneficial to support group decision making. For the two-stage workflow (see Section 3.1), we propose a method for visualizing the clusters of the outputs and the suitability of each output [Li *et al.*, 2018]. This method asks workers in the second stage (i.e., reviewers) to judge the similarity of a pair of outputs as well as the preference, and uses these two types of responses for the visualization.

Machine learning is a promising domain for applying human computation approaches.[1] In machine learning projects, human computation is typically used only for collecting training labels; however, it can contribute in various phases. For example, we applied human computation to feature engineering [Takahama *et al.*, 2018]; this method adaptively chooses pairs of positive and negative instances and asks workers to define features that are informative for classifying them. The features generated by humans are naturally interpretable, and we observed that they improve the classification accuracy in some tasks. Additionally, human computation is applicable for creating machine learning models. It can be used for model investigation [Ribeiro *et al.*, 2016], or for modeling itself by aggregating the models created by the crowd of data scientists [Baba *et al.*, 2014]. Human computation approaches can thus contribute towards improving the productivity of machine learning engineers and advancing new fields of machine learning applications.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 18K18105.

References

[Baba and Kashima, 2013] Yukino Baba and Hisashi Kashima. Statistical quality estimation for general crowdsourcing tasks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 554–562, 2013.

[Baba *et al.*, 2014] Yukino Baba, Nozomi Nori, Shigeru Saito, and Hisashi Kashima. Crowdsourced data analytics: a case study of a predictive modeling competition. In *Proceedings of the 2014 International Conference on Data Science and Advanced Analytics (DSAA)*, pages 284–289, 2014.

[Bachrach *et al.*, 2012] Yoram Bachrach, Tom Minka, John Guive, and Thore Graepel. How to grade a test without knowing the answers - a bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[Bigham *et al.*, 2010] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology (UIST)*, pages 333–342, 2010.

[Chen *et al.*, 2013] Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 193–202, 2013.

[1] The combination of human computation and machine learning is also known as *human-in-the-loop machine learning* or *interactive machine learning*.

[Dawid and Skene, 1979] Alexander P. Dawid and Allan M. Skene. 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, 1979.

[Ehara *et al.*, 2016] Yo Ehara, Yukino Baba, Masao Utiyama, and Eiichiro Sumita. Assessing translation ability through vocabulary ability assessment. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3712–3718, 2016.

[Kajimura *et al.*, 2015] Shunsuke Kajimura, Yukino Baba, Hiroshi Kajino, and Hisashi Kashima. Quality control for crowdsourced POI collection. In *Proceedings of the 19th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 255–267, 2015.

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

[Li *et al.*, 2017] Jiyi Li, Yukino Baba, and Hisashi Kashima. Hyper questions: Unsupervised targeting of a few experts in crowdsourcing. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*, pages 1069–1078, 2017.

[Li *et al.*, 2018] Jiyi Li, Yukino Baba, and Hisashi Kashima. Simultaneous clustering and ranking from pairwise comparisons. In *Proceedings of 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence (IJCAI-ECAI)*, 2018.

[Lin *et al.*, 2012] Christopher H. Lin, Mausam, and Daniel S. Weld. Crowdsourcing control: moving beyond multiple choice. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 491–500, 2012.

[Lorenz *et al.*, 2011] Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences*, 108(22):9020–9025, 2011.

[Matsui *et al.*, 2014] Toshiko Matsui, Yukino Baba, Toshihiro Kamishima, and Hisashi Kashima. Crowdordering. In *Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 336–347, 2014.

[Otani *et al.*, 2015] Naoki Otani, Yukino Baba, and Hisashi Kashima. Quality control for crowdsourced hierarchical classification. In *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM)*, pages 937–942, 2015.

[Oyama *et al.*, 2013] Satoshi Oyama, Yukino Baba, Yuko Sakurai, and Hisashi Kashima. Accurate integration of crowdsourced labels using workers' self-reported confidence scores. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2554–2560, 2013.

[Rasch, 1960] G. Rasch. *Probabilistic models for some intelligence and attainment tests*. Denmarks Paedagogiske, 1960.

[Raykar and Yu, 2011] Vikas C Raykar and Shipeng Yu. Ranking annotators for crowdsourced labeling tasks. In *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 1809–1817. 2011.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144, 2016.

[Samejima, 1969] Fumi Samejima. Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement*, 1969.

[Sunahase *et al.*, 2017] Takeru Sunahase, Yukino Baba, and Hisashi Kashima. Pairwise HITS: Quality estimation from pairwise comparisons in creator-evaluator crowdsourcing process. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 977–983, 2017.

[Takahama *et al.*, 2018] Ryusuke Takahama, Yukino Baba, Nobuyuki Shimizu, Sumio Fujita, and Hisashi Kashima. AdaFlock: Adaptive feature discovery for human-in-the-loop predictive modeling. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 1619–1626, 2018.

[Venanzi *et al.*, 2014] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web (WWW)*, pages 155–164, 2014.

[von Ahn *et al.*, 2008] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. re-CAPTCHA: human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[Welinder *et al.*, 2010] Peter Welinder, Steve Branson, Pietro Perona, and Serge J. Belongie. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 2424–2432. 2010.

[Whitehill *et al.*, 2009] Jacob Whitehill, Ting fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 2035–2043. 2009.

[Wu *et al.*, 2012] Xian Wu, Wei Fan, and Yong Yu. Sembler: Ensembling crowd sequential labeling for improved quality. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1713–1719, 2012.