Full Characterization of Adaptively Strong Majority Voting in Crowdsourcing

Margarita Boyarskaya
mb6599@stern.nyu.edu
New York University

Panos Ipeirotis
panos@stern.nyu.edu
New York University

ABSTRACT

In crowdsourcing, quality control is commonly achieved by having workers examine items and vote on their correctness. To minimize the impact of unreliable worker responses, a δ-margin voting process is utilized, where additional votes are solicited until a predetermined threshold δ for agreement between workers is exceeded. The process is widely adopted but only as a heuristic. Our research presents a modeling approach using absorbing Markov chains to analyze the characteristics of this voting process that matter in crowdsourced processes. We provide closed-form equations for the quality of resulting consensus vote, the expected number of votes required for consensus, the variance of vote requirements, and other distribution moments. Our findings demonstrate how the threshold δ can be adjusted to achieve quality equivalence across voting processes that employ workers with varying accuracy levels. We also provide efficiency-equalizing payment rates for voting processes with different expected response accuracy levels. Additionally, our model considers items with varying degrees of difficulty and uncertainty about the difficulty of each example. Our simulations, using real-world crowdsourced vote data, validate the effectiveness of our theoretical model in characterizing the consensus aggregation process. The results of our study can be effectively employed in practical crowdsourcing applications.

Keywords: crowdsourcing, labeling aggregation, majority voting, data quality control, fair remuneration, Markov random walk.

ACM Reference Format:
Margarita Boyarskaya and Panos Ipeirotis. 2024. Full Characterization of Adaptively Strong Majority Voting in Crowdsourcing. In *Collective Intelligence Conference (CI '24), June 27–28, 2024, Boston, MA, USA.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3643562.3673895

1 INTRODUCTION

The rise of machine learning solutions in the business world has resulted in companies needing to handle the noisy output that often accompanies these processes. While some cases may allow for noisy output, there are many high-stakes situations where human intervention is necessary to ensure accuracy. Some examples of these cases include:

- Security systems flag user accounts with unusual activity as potentially problematic, and humans must verify whether the account should be suspended.
- Databases entries (e.g., products in a database, CRM entries for customers, knowledge graph entities) are marked as potential duplicates by an algorithm, and humans verify whether they correspond to the same person to proceed with the merging.
- Financial activity is marked as money laundering, and human investigators must decide whether to launch a detailed investigation.
- Creating evaluation data and benchmarks for machine learning algorithms, where it is critical to minimize label noise.
- Labeling examples where no machine learning product is directly feasible (e.g., citizen science projects such as Zooniverse, disaster relief efforts, etc.)

The challenge in this context is that human decision-makers can also introduce noise in the verification process. It is common to rely on aggregating multiple votes to ensure high-quality outcomes.[1] There is a plethora of literature proposing various vote aggregation schemes to ensure quality control for noisy labels (e.g., (Chilton et al. 2013, Dai et al. 2013, Mortensen et al. 2013)). However, simple aggregation schemes like majority voting or variations are often used in practice due to their simplicity. One reason for this choice is that most aggregation schemes require a long record of votes from individual workers to assess their accuracy. Nonetheless, worker participation typically follows a power law distribution (Alonso and Baeza-Yates 2011), with only a few workers having extensive histories. In contrast, most workers churn quickly, making it challenging to assess individual workers' quality accurately.

A commonly encountered variation of majority voting asks for multiple votes until the positive votes are δ votes more than the negatives (or vice versa). This process is called *δ-margin majority voting*. Compared to the usual majority voting scheme, the δ-margin voting scheme imposes additional scrutiny on the 'strength' of the agreement between the voters. Even though the δ-margin voting consensus scheme is salient in practice, the design of crowdsourced tasks that use this scheme tends to be heuristic and adhoc, lacking a solid theoretical foundation that describes the properties of the voting process.

This work shows that a δ-margin voting process can be easily modeled with well-established mathematical tools (namely, Markov chains with absorbing states). Our research contribution is simple:

[1] In early use cases, crowdsourcing was used for generating *training data* for machine learning algorithms, e.g., (Russakovsky et al. 2015). However, recent advancements indicate that it is possible to attain high levels of machine learning performance without reliance on *multiple* labeling of each training data point (Lin et al. 2016), even with noisy data.

CI '24, June 27–28, 2024, Boston, MA, USA

ACM ISBN 979-8-4007-0554-0/24/06
https://doi.org/10.1145/3643562.3673895

we provide a thorough theoretical description of key properties of δ-margin voting, allowing decision-makers to give ex-ante answers to some of the more pervasive questions in crowdsourced experiment design:

- How to structure a crowdsourced voting process to achieve a given label accuracy?
- What is the cost of running such a process?
- Can a pool of lower-accuracy workers achieve the same accuracy?

We provide closed-form theoretical results that quantify the quality and cost of the results. We also show that the results have a high degree of generalizability, making minimal assumptions about the quality of the workers or the difficulty of the labeled items. We show that we only need some prior expectation on how well a *worker pool* will work on an item.[2] Then, we can estimate the process's quality and speed as the labeling occurs.

We believe such "human in the loop" systems will become more prevalent, where humans play a crucial role in overseeing and guiding AI systems. These systems are vital for ensuring AI technologies' responsible and ethical deployment, particularly in high-stakes decision-making scenarios. *One crucial aspect of these systems is the ability to provide theoretical performance guarantees for the guardrails.* Establishing such guarantees is essential for meeting regulatory requirements and building trust in hybrid AI systems. By offering rigorous theoretical analyses of the performance of human-in-the-loop components, we can assure these systems' reliability, effectiveness, and safety.

Regulatory tests often require transparency and accountability in AI decision-making processes. Theoretical performance guarantees for the human-in-the-loop components can prove the system's robustness and ability to ensure compliance with legal, ethical, and societal norms. This can be especially important in sensitive domains such as healthcare, finance, or autonomous vehicles, where the impact of AI decisions can have significant consequences. As the adoption of hybrid AI systems continues to grow, it becomes increasingly necessary to develop theoretical frameworks and methodologies to assess and ensure the performance of human-in-the-loop components. These frameworks should consider various factors, such as human annotators' quality and expertise, their decisions' impact on the overall system, and the dynamic interplay between human and AI decision-making.

By addressing these challenges and providing theoretical guarantees, we can pave the way for the responsible integration of AI systems with human oversight, fostering transparency, fairness, and accountability in decision-making processes.

The organization of this paper is as follows. Section 2 introduces the Markov chain formalism for the δ-margin voting process. Section 3 overviews existing work on consensus aggregation design and quality assurance, highlighting both the novelty and relevance of our theoretical presentations. Section 4 derives the theoretical equations for key characteristics of the δ-margin voting process:

[2]In this work, we are not trying to characterize *individual* labelers' accuracy for each item they work on. Following a power-law argument for the distribution of workers in a voting process, estimating the quality characteristics of the (numerous) infrequent participants is hard. Even in the corporate setting of Business Process Outsourcing (BPO), decision-makers prefer not to identify or track individual employees but rather allow workforce fungibility and provide access to *pools* of workers.

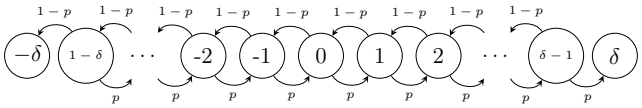


Figure 1: A Markov chain diagram illustrating state transitions for δ-margin majority voting on a single item, for which the average worker pool accuracy is p. Node labels indicate the difference between the numbers of correct (n_1) and incorrect (n_0) votes, $n_1 - n_0$. Consensus is reached in absorbing states δ and $-\delta$, the former resulting in a correct consensus vote and the latter in an incorrect one.

quality of the results, expected value and variance of the number of votes required to reach consensus, and the overall distribution for time to completion. Section 5 examines how to operate when there is uncertainty about the quality of workers when annotating the items. Section 6 expands these results to demonstrate that two or more δ-margin processes that involve different levels of worker response accuracy can have an equivalent quality of results, and we further elaborate on this equivalence to propose a quality-sensitive payment scheme that connects result quality with item-level worker accuracy. Section 7 presents experimental validation with real data comparing the theoretical predictions against results from real processes. Section 8 includes a discussion about the usage of δ-margin voting in a real setting, the problems faced by the managers, and how the prescriptive nature of our work helped them understand better and improve their business. Finally, Section 9 discusses the immediate and the more involved research developments that can be built upon this work.

2 MARKOV CHAIN MODEL OF δ-MARGIN MAJORITY VOTING

In the context of labeling items on a crowdsourcing platform using δ-margin voting, we seek votes from a *pool of workers* for a specific item i that requires a binary label. We assume that each worker has an (unobserved) accuracy drawn from a distribution $\mathcal{A}_i$. We make no assumptions about the distribution $\mathcal{A}_i$ other than it has an expected value $E[\mathcal{A}_i] = p_i$, meaning that workers are expected to be correct for item i with probability p_i and incorrect with probability $1 - p_i$; we only have this expectation across the pool, and not for individual workers. We continue to ask workers to assign labels to an item until the absolute difference between the number of votes for the two classes exceeds a predefined consensus threshold δ. We allow different items to have varying difficulty levels (i.e., the distribution of worker accuracies $\mathcal{A}$ may differ across items); however, we assume that the mean of the distribution $E[\mathcal{A}_i]$ remains stable while labeling the same item i.

Definition 1. A *δ-margin voting* consensus scheme is a process of sequentially collecting binary votes that terminates when the number of votes collected for one class exceeds the number of votes collected for the other class by a fixed integer threshold δ. The class with the larger number of votes is then the consensus label.

Example. Set $\delta = 2$. Then, a δ-margin voting process might stop when the vote-count tuples $\langle n_{correct}, n_{incorrect} \rangle$ attain one of the following values: $\langle 2, 0 \rangle$, $\langle 0, 2 \rangle$, $\langle 3, 1 \rangle$, $\langle 1, 3 \rangle$, $\langle 4, 2 \rangle$, $\langle 2, 4 \rangle$, ..., and so on. A consensus vote obtained in one of the states $\{\langle 2, 0 \rangle, \langle 3, 1 \rangle, \langle 4, 2 \rangle, \ldots\}$ is correct, while a consensus vote obtained in one of the states $\{\langle 0, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 4 \rangle, \ldots\}$ is incorrect.

Since the only desideratum for consensus is the difference between the numbers of the two types of votes, the process can be modeled as a *time-homogeneous*[3] *Markov random walk, with two absorbing states*. We define the current state as the difference between the number of correct and incorrect votes. If the difference is δ the process terminates, resulting in a correct consensus label, and if the difference is $-\delta$, the process terminates with an incorrect label. In all other states, we procure an additional vote, which will be correct with probability p, which corresponds to a transition from state k to state $k + 1$; or it can be an incorrect vote with probability $1 - p$, which makes for a transition from state k to state $k - 1$.

This process has a state diagram illustrated in Figure 1 and is also known as *Gambler's Ruin* model.[4] The model is a common introductory example for random walks and describes the probability of a gambler winning a certain amount in a game of chance vs. the probability of losing her entire gaming budget.[5] Surprisingly, this model has not been used to describe the process for label aggregation in crowdsourcing, and we are unaware of anyone providing results about label aggregation quality and the number of votes required for this process in crowdsourcing.

Note that the model makes minimal assumptions about the process. The transition probability p of the Markov process corresponds to $E[\mathcal{A}_i]$, i.e., *the mean of the distribution of the workers' accuracies when labeling a single item.* In other words, we may have an arbitrary distribution of worker accuracies and only need to know the mean for the process to be accurately modeled by the Markov chain. (In Section 4, we show the results assuming p is known, and in Section 5, alleviate this restriction and show how to work when p is unknown.) Furthermore, we do not need to assume that the mean accuracy p is the same across items: in Section 5, we will explicitly assume that p varies across items and only remains stable within the context of labeling a single item.

Before providing the analytical results for several key characteristics of δ-margin voting, we provide next, in Section 3, an overview of the related literature.

3 LITERATURE REVIEW

In this section, we offer a concise summary of relevant research in label aggregation in crowdsourcing (Section 3.1), followed by references to related work on δ-margin voting (Section 3.2). To our knowledge, no research has yet focused on providing theoretical outcomes for ex-ante cost and quality estimation for δ-margin voting.

[3]In this context, time-homogeneous means that p remains stable within the context of labeling a single item.

[4]Not to be confused with *Gambler's Fallacy* (Ayton and Fischer 2004), which is the behavioral phenomenon that captures human belief that an event that has occurred more frequently than expected has a higher chance of re-occurring in the future (e.g., a "hot streak". This is distinct from the statistical problem of Gambler's Ruin which is our focus.

[5]See (Feller 1968, page 344) for details.

3.1 Literature on quality control in label aggregation

A substantial amount of literature on crowdsourcing proposes various quality controls for aggregating worker expertise. However, most of this work is experimental (Hansen et al. 2013, Kazai et al. 2011, Yin et al. 2014), and the proposed quality control mechanisms are often ad-hoc, lacking theoretical guarantees (Dai et al. 2013, De Boer and Bernstein 2017, Kucherbaev et al. 2016). Moreover, many studies rely on accurate priors of essential process or workforce parameters, (Abassi and Boukhris 2017, Dalvi et al. 2013, Heer and Bostock 2010, Jung and Lease 2011, Laureti et al. 2006, Rutchick et al. 2020, Tao et al. 2018), which are costly to obtain (Bonald and Combes 2016, De Boer 2017). Prominent examples of crowdsourced datasets (Krishna et al. 2017, Russakovsky et al. 2015, Zhou et al. 2019) tacitly support the thesis that majority voting is the de-facto method of choice used to aggregate labels and ensure quality. Much of this work also acknowledges the many iterations of experimenting with crowdsourced process design that are required to perfect the data collection and choose (without guarantees) best-performing parameters for the voting procedure. The absence of a good prior for what accuracy one is to expect given a voting method may be one of the drivers that propels most practitioners to opt for the simple majority voting –often in conjunction with rapid judgments techniques (Krishna et al. 2016). In constructing ImageNet (Russakovsky et al. 2015), the authors acknowledged the challenge inherent in the fact that different item categories require different levels of consensus among users. They address this by dynamically determining the number of agreeing votes needed for a given category of images using an initial sample of items, and then requiring the chosen confidence of agreement for the remaining items. Using δ–margin voting eliminates the need for this expensive preliminary discovery step.

Another strand of literature examines non-computational approaches to data quality assurance in crowdsourcing. A prominent direction here is the study of incentives (usually in form of monetary reward) for workers (Daniel et al. 2018, Heer and Bostock 2010, Hossfeld et al. 2014, Mason and Watts 2009, Singer and Mittal 2013). Here too, our work offers a novel contribution: the formulas for expected accuracy given the chosen decision threshold and the accuracy of responses give rise to a ratio of payments for two worker pools, given the same expected result quality. This offers a principled way to set the incentives for various tasks in alignment with the resulting label quality. Some notable work in the study of compensation incentives (Mason and Watts 2009) suggests that increased financial reward increases the quantity, but not the quality, of crowdsourced work. Other research (Kazai 2011), however, presents evidence to the contrary, showing that higher pay encourages better work, and especially so among qualified workers (Kazai et al. 2013). Findings such as in Mason and Watts (2009) do not threaten the results we present in Section 6, where we propose a way to relate the relative payments to two queues of workers depending on the expected accuracy of the outcome. Here we do not assume that by paying certain workers more we will incentivize them to perform better – we assume that the performance will remain stable for the pool of workers labeling a given item. Our objective is to relate another voting process to a given one. In

particular, we show how to use the expected level of performance and a given payment rate for one voting process an 'anchor' to set a different level of payment for another voting process on a different item (given that we expect the same quality of results, which might require fewer or more votes to be cast for that second item).

In the broader literature on the quality and costs of crowdsourcing, a subset of work provides theoretical guarantees. For instance, Berend and Kontorovich (2014) establish accuracy bounds for the weighted majority voting scheme. Meanwhile, Khetan and Oh (2016) formulate a theoretical trade-off between budget and accuracy in voting processes with adaptive task assignment, where tasks are assigned based on data collected until the point of assignment (Barowy et al. 2012). The authors present an adaptive assignment scheme that achieves the fundamental limit.

Another example of theoretical work in this area is provided by Manino et al. (2018), who explore the adaptive assignment of the next worker to an item. They formulate an accuracy gap between the uniform allocation of workers (Karger et al. 2014), adaptive allocation, and an assignment that maximizes information gain (Simpson and Roberts 2015). Their work derives tight but not exact bounds on the accuracy of this assignment policy. Additionally, Livshits and Mytkowicz (2014) is a theoretical study that examines the costs of completing voting tasks. The authors use power analysis to obtain ex-ante estimates for the number of votes needed to resolve each item with a specific level of statistical significance.

While worker wages are not often addressed in the literature, a notable exception is the work of Singer and Mittal (2013). In their paper, the authors present mechanisms compatible with incentives, maximizing the number of tasks within a budget constraint and minimizing worker payments given a fixed number of tasks.

In contrast to the previous papers, our work focuses on the δ-margin consensus rule and provides exact expressions for the probability of error, expected time until consensus, and its variance. This allows for a more detailed consideration of the requester's utility, including the workers' wages.

3.2 Literature on δ-margin voting

The literature on group decision-making and voting mechanisms (Larruelle and Valenciano 2011) distinguishes between two types of majority consensus vote: *simple* majority and *absolute* majority. A simple majority is achieved when the number of votes for an option A exceeds the number of votes for an alternative option B ($s_A > s_B$), while an absolute majority requires that the number of votes for an option A is greater than half of all votes ($s_A > \frac{n}{2}$).[6] In the case of binary (or *dichotomic*) voting, the distinction between simple and absolute majorities only exists when some voters abstain or cast neutral votes.

In (Dietrich and List 2007), majority voting is extended to a broader category of rules known as "quota rules" (also referred to as "k-unanimity" or "k-majority" in cybernetics and discrete mathematics (Alon et al. 2006, Scheidler et al. 2015)). Under quota rules, an item is assigned a particular label if the number of workers voting for that label exceeds a threshold value of k. In situations where all workers vote simultaneously within a fixed workforce, the quota rule is similar to δ-margin voting, with two notable differences: (a) the margin may not be satisfied at the end of the voting process, and (b) the process may accumulate more votes than needed, which can be a disadvantage when each vote incurs a cost.

The δ-margin voting aggregation method is widely used by practitioners but is typically only applied in empirical or experimental settings in crowdsourcing literature, which means there is a missed opportunity for theoretical evaluation of its benefits. Notably, even outside of the crowdsourcing literature field, some influential papers refer to δ-margin majority voting as "the forgotten decision rule" (García-Lapresta and Llamazares 2001, Llamazares 2006), and the earliest acknowledgments of the δ-margin voting scheme can be found in Fishburn (2015) and Saari (1990), though these are brief. According to (De Boer 2017), in experimental comparisons of typical crowdsourcing tasks, the δ-margin method, called "Beat-By-K" by Goschin (2014), provides very accurate results for relatively high values of δ, but it is expensive to run in settings that prioritize utility without any budget constraints. Other consensus aggregation methods in the literature capture similar ideas to δ-margin voting but require a stronger, more confident agreement among workers. For instance, in the "Automan" scheme (Barowy et al. 2012), the requester keeps sampling votes until the voting process reaches a given statistical confidence value.

4 THEORETICAL CHARACTERISTICS OF δ-MARGIN VOTING PROCESSES

This section presents closed-form theoretical formulations of essential features of the δ-margin voting process. We utilize established results from Markov Chain theory to characterize the following aspects:

- The quality of the resulting labels obtained from the δ-margin voting process.
- The expected number of votes necessary to reach a consensus, i.e., the expected time until consensus.
- The variance of the votes needed to reach a consensus.
- The probability density of the votes required to reach a consensus.

We begin by assuming that the mean p of the distribution of worker response accuracies for a fixed item is known. In Section 5, we will relax this assumption by allowing p to be a random variable, starting with some prior beliefs about the quality of the worker pool, which are updated during the voting process.

4.1 Background on Markov Chains with absorbing states

We demonstrated that a δ-margin voting process can be modeled as a Markov chain with two absorbing states and $2(\delta-1)+1$ transient states. This section presents the necessary background (Grinstead and Snell 1997) for working with such Markov chains to obtain relevant results. To begin, we define the *canonical form* $\mathbf{M}$ of the transition matrix for this process:

$$\mathbf{M} = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & I_2 \end{pmatrix}$$

[6] Alternatively, the strength of the majority condition can be defined as a multiplier k on the required inequality requirements, which are modified to $s_A > k \cdot s_B$ for $k > 1$ and $s_A > k \cdot n$ for $\frac{1}{2} \leq k \leq 1$.

The $\mathbf{Q}$ is a matrix of size $(2\delta-1)\times(2\delta-1)$ containing the probabilities of transitioning from a transient state i into a transient state j, and $\mathbf{R}$ is a $(2\delta-1)\times 2$ matrix of transition probabilities from transient into absorbing states. Formally:

$$\mathbf{Q} = \begin{pmatrix} 0 & p & 0 & 0 & \cdots & 0 \\ 1-p & 0 & p & 0 & \cdots & 0 \\ 0 & 1-p & 0 & p & \cdots & 0 \\ 0 & 0 & 1-p & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1-p & 0 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1-p & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & p \end{pmatrix}^{\mathsf{T}}$$

Finally, $\mathbf{I}_2$ is a 2×2 identity matrix, and $\mathbf{0}$ is a $2\times((2\delta-1)+1)$ null matrix. Using the above, the definition of the *fundamental matrix* $\mathbf{N}$ of this absorbing matrix chain is:

$$\mathbf{N} := \sum_{k=0}^{\infty} (\mathbf{Q})^k$$

which can be written (Grinstead and Snell 1997) as:

$$\mathbf{N} = (\mathbf{I}_{2\delta-1} - \mathbf{Q})^{-1}$$

The elements $\mathbf{N}_{ij}$ correspond to the expected number of visits to the state j, given that the initial state was i. We can now define the essential features of interest by utilizing the fundamental matrix.

4.2 Quality of the consensus vote

The first quantity of interest is the quality of the final consensus votes. To calculate this value, we want to estimate the probabilities of reaching the absorbing states δ and $-\delta$. Following the notation introduced in the previous section, we define the matrix $\mathbf{B}$:

$$\mathbf{B} := \mathbf{N}\cdot\mathbf{R}$$

The matrix $\mathbf{B}$ is a row-stochastic matrix, with $2\delta-1$ rows and two columns. The matrix's (i,j)-entry contains the probability of eventually reaching absorbing state j when starting from transient state i. The process typically starts at state 0 (no votes), so we are interested in the row's contents corresponding to the state 0.[7] Through standard algebraic manipulations, we get that the probability of reaching states δ and $-\delta$, when starting from state 0 are:

$$\mathbf{B}_{0,\delta} = \frac{\varphi^\delta}{1+\varphi^\delta}$$

$$\mathbf{B}_{0,-\delta} = \frac{1}{1+\varphi^\delta}$$

where $\varphi = \frac{p}{1-p}$ are the *odds* of the average worker's vote on the item being correct.[8] Using these quantities, we can easily estimate the expected labeling quality of an item:

[7] In particular cases, we may want to start at a different state. For example, we may require $\delta = 3$ to flag an account for money laundering while dismissing such an alert with $\delta = 1$. In such a case, we may start the process from state -1, setting $\delta = 2$. Deriving the results, in this case, is reasonably simple, but for brevity, we do not provide the details in the current paper.

[8] The proof is also readily available in many introductory texts on Markov Chains, e.g., in (Feller 1968, page 344), without using the approach that relies on the canonical transition matrix $\mathbf{M}$ and the fundamental matrix $\mathbf{N}$.

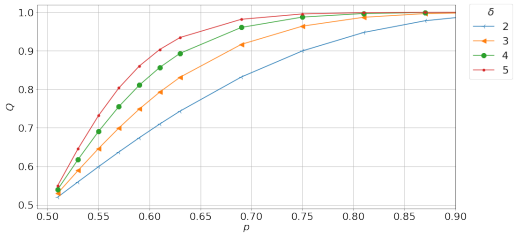


Figure 2: Theoretical values of quality Q of consensus vote (Theorem 1) as a function of the probability of a correct answer p, for a fixed consensus threshold δ.

Theorem 1. *For a δ-margin voting process on a single item, with mean accuracy p of the worker responses, and a consensus threshold δ, the probability $Q(\varphi,\delta)$ of obtaining a correct consensus vote C is:*

$$Q(\varphi,\delta) := \frac{\varphi^\delta}{1+\varphi^\delta} \tag{1}$$

where $\varphi = \frac{p}{1-p}$ are the odds *of the average worker's vote on the item being correct.*

Discussion: Figure 2 plots the dependence of consensus vote quality Q on item difficulty p for $\delta \in \{2,3,4,5\}$. Note that the parameter δ plays a significant role in the quality of the consensus vote: Based on Equation 1, the *odds* of the *consensus* vote being correct, are φ^δ. Therefore by increasing δ, we exponentially increase the odds that the consensus vote is correct. For example, consider a pool of workers with expected response accuracy $p = 0.75$ (i.e., $\varphi = \frac{0.75}{1-0.75} = 3$) for a given item. If we set $\delta = 2$, the expected quality of the overall classification will be $Q(3,2) = 0.9$ (i.e., odds 9 to 1 being correct). If we increase δ to $\delta = 3$, $Q(3,3) = 0.964$ (i.e., odds 27 to 1), and if we increase to $\delta = 4$, then $Q(3,4) = 0.9878$ (i.e., odds 81 to 1). Next, we discuss how the cost of the process changes when we change δ, and we show that we achieve exponential improvements in quality with a mostly linear increase in cost.

4.3 Time until consensus

The next set of properties we want to explore is the number of votes required until reaching a consensus. We start by examining the expected number of votes until the process terminates. While the expected number of votes to completion is useful for characterizing the voting process, it describes just the average across runs. In addition to the expectation, we also want to know the robustness of a process and how reliably it will finish within the expected time frame. For this reason, it is important to know the *variance* of the process in terms of the number of votes required to complete and, more generally, the distribution's overall *pdf*.

4.3.1 Expected Time until consensus: What is the *expected number of steps* until consensus is reached? Following the notation, we define the vector $\mathbf{t}$, which contains in position i the expected number of steps before consensus when starting in transient state i.

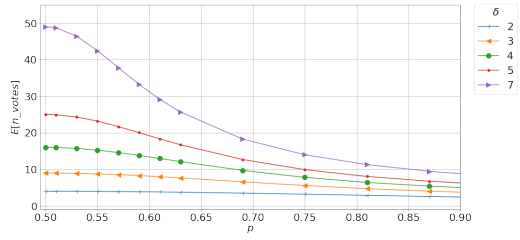


Figure 3: Expected time to reach consensus as a function of the probability of correct answer p, for a fixed consensus threshold δ.

$$\mathbf{t} = \mathbf{N} \cdot \mathbf{1}$$

where $\mathbf{1}$ is vector of ones of length $2\delta - 1$. Through algebraic simplification, we get the following:

Theorem 2. *The expected number of votes n_{votes} it takes to reach a (correct or incorrect) consensus when classifying an item using a δ-margin voting scheme with item-level expected worker accuracy $p \neq \frac{1}{2}$ and consensus threshold δ is:*

$$\mathbb{E}[n_{votes}|\varphi, \delta] = \delta \cdot \frac{\varphi + 1}{\varphi - 1} \cdot \frac{\varphi^{\delta} - 1}{\varphi^{\delta} + 1} \tag{2}$$

where $\varphi = \frac{p}{1-p}$ is the odds *that the average worker vote on the item is correct. When $p = 0.5$, we have $\mathbb{E}[n_{votes}|\varphi, \delta] = \delta^2$.*

***Discussion**:* Note that the expected time to termination increases *mostly* linearly with δ. When φ gets close to 1 (i.e., mostly random worker responses), the expected time to termination peaks at δ^2. Figure 3 illustrates the time until voting termination $\mathbb{E}(n_{votes})$ on expected item-level worker accuracy p, for a given consensus threshold δ.

4.3.2 Variance of time until consensus. The variance in the number of steps required to reach consensus when starting in a state i is the i-th entry of vector $\bar{v}$, defined as:

$$\bar{v} = (2\mathbf{N} - \mathbf{I}_t)\mathbf{t} - \mathbf{t}_{\text{sq}}$$

where $\mathbf{t}_{\text{sq}}$ is the vector of squared elements of $\mathbf{t}$. Applying this result to our model for δ-margin voting, and starting with the state with no votes, we have the following expression for the variance of time until consensus:

Theorem 3. *For $p \neq \frac{1}{2}$, the variance of the number of votes it takes to reach consensus using the δ-margin voting process is:*

$$Var[n_{votes}|\varphi, \delta] = 4\delta\varphi \left(\frac{\varphi+1}{\varphi^{\delta}+1}\right)^2 \cdot$$
$$\cdot \left[h(\delta) \cdot \varphi^{\delta-2} + \sum_{i=1}^{\delta-2} \left(h(\delta - i)(\varphi^{\delta+i-2} + \varphi^{\delta-i-2})\right)\right]$$

δ	$\mathbf{Var(n_{votes})}$
2	$8\varphi \left(\frac{\varphi+1}{\varphi^2+1}\right)^2$
3	$12\varphi \left(\frac{\varphi+1}{\varphi^3+1}\right)^2 (\varphi^2 + 2\varphi + 1)$
4	$16\varphi \left(\frac{\varphi+1}{\varphi^4+1}\right)^2 (\varphi^4 + 2\varphi^3 + 4\varphi^2 + 2\varphi + 1)$
5	$20\varphi \left(\frac{\varphi+1}{\varphi^5+1}\right)^2 (\varphi^6 + 2\varphi^5 + 4\varphi^4 + 6\varphi^3 + 4\varphi^2 + 2\varphi + 1)$
6	$24\varphi \left(\frac{\varphi+1}{\varphi^6+1}\right)^2 (\varphi^8 + 2\varphi^7 + 4\varphi^6 + 6\varphi^5 + 9\varphi^4 + 6\varphi^3 + 4\varphi^2 + 2\varphi + 1)$
7	$28\varphi \left(\frac{\varphi+1}{\varphi^7+1}\right)^2 (\varphi^{10} + 2\varphi^9 + 4\varphi^8 + 6\varphi^7 + 9\varphi^6 + 12\varphi^5 + 9\varphi^4 + 6\varphi^3 + 4\varphi^2 + 2\varphi + 1)$

Table 1: Particular values of variance on the number of votes until consensus

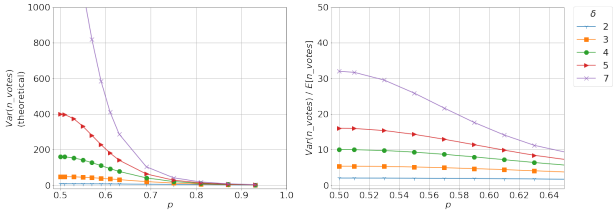


Figure 4: Left: $Var(n_{votes})$ (left) and $Var(n_{votes})/\mathbb{E}(n_{votes})$ (right) as a function of probability of correct answer p, for a fixed consensus threshold δ (d). (Note the differing scales.)

where coefficients h are defined using the floor function:

$$h(z) := \left\lfloor \frac{z^2}{4} \right\rfloor.$$

Stated differently,

$$h(z) = \begin{cases} \frac{z^2}{4}, & \text{if } z \text{ is even} \\ \frac{z-1}{2} \cdot \frac{z+1}{2}, & \text{if } z \text{ is odd.} \end{cases}$$

The sequence of coefficients defined by $h(z)$ is also known as the quarter squares *sequence and can be defined as the interleaving of square numbers and pronic numbers (Losanitsch 1897, Sloane 2019) (the latter defined as the product of two consecutive integers).*

Equation 3 simplifies the formula proposed by (Anděl and Hudecová 2012) while agreeing with it numerically. Moreover, once the polynomial in square brackets is formulated for a given value of δ, one can observe a simple and intuitive 'pyramidal' pattern that governs coefficient generation. To provide the reader with intuition on the look of the polynomial in Equation 3, we list explicit formulas for $Var(n_{votes})$ for the first few values of δ in Table 1.

The formulation in Equation 3 allows us to elaborate upon the plots in Figure 3 by adding bands of the size $2\sqrt{Var(n_{votes})}$ to each trajectory of $\mathbb{E}[n_{votes}]$. Detailed plots for selected values of δ are shown in Figure 5.

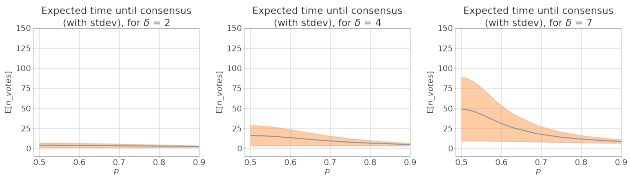


Figure 5: Expected time until reaching consensus, with standard deviation bounds, for selected values of δ.

4.3.3 Distribution of time to consensus: Using the notation used in the previous subsections and known results about discrete phase-type distributions from queuing theory (Latouche and Ramaswami 1999, Neuts 1994), we can describe the probability density function of time until consensus. The event that the voting process will terminate after casting exactly m votes is characterized by the following probability density function:

$$T_m = \mathbf{z} \cdot \mathbf{Q}^{m-1} \cdot \mathbf{R} \tag{3}$$

where $\mathbf{z}$ is a vector of length $2\delta - 1$ that encodes the initial state of the process. The expression T_m is a 1×2 vector, specifying the probability densities of the voting process terminating in exactly m steps, in each of the two absorbing states – i.e., reaching a consensus with a correct label and an incorrect one. Using matrix $\mathbf{B}$, which contains the probabilities of reaching each absorbing state, we derive the *pdf* of the time to completion.

Theorem 4. *The probability $pdf(m)$ that a δ-margin majority voting process will terminate after exactly m votes when starting from zero votes, is characterized by the following probability density function:*

$$pdf(m) = \mathbf{z} \cdot \mathbf{Q}^{m-1} \cdot \mathbf{R} \cdot \mathbf{B}^\intercal \cdot \mathbf{z}^\intercal \tag{4}$$

where $\mathbf{z}$ is a vector of length $2\delta - 1$ with $\mathbf{z} := \langle z_{-\delta+1}, \ldots, z_{\delta-1} \rangle$ with $z_0 = 1$ and $z_i = 0$ for $i \neq 0$.

Figure 6 visualizes a result of this computation for $\delta = 4$ and for various values of φ.

5 MODELING UNCERTAINTY IN p

In the prior section, we presented results that assumed that the mean labeling quality p is exogenously known. This section alleviates this assumption by modeling the mean worker pool accuracy p as a random variable. In our case, the quantity p corresponds to the average accuracy when labeling an item: we assume that the average accuracy is constant when labeling a single item, but we do not know its exact value. Instead, we have some prior expectations about p, and based on the votes that come in, we update our beliefs. Note that we only need the assumption of "constant p" within the context of a single item. We explicitly allow p to be different across different items.

We split our discussion into two parts. First, Section 5.1 presents the results when the prior belief of p is Beta distribution. Then, in Section 5.2, we show how we can incorporate priors of arbitrary complexity using a *mixture of Betas* approach for the priors.

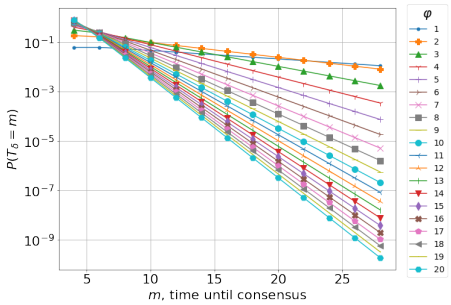


Figure 6: Probability density of reaching consensus in exactly m steps (i.e., any consensus, correct or incorrect) $pdf(m)$ for $\delta = 4$, as a function of the number of votes m, for varying item-level odds of expected worker accuracy φ.

5.1 Working with a simple prior for p

Without loss of generality, we can assume that we have a time-homogeneous Markov process, which means that the mean of the workers' accuracy distribution remains stable while labeling the item. (In other words, the next worker labeling a given item is as good, on average, as the previous one.)

The discussion in the previous section relied on the assumption that we know the value of p. Now, we are making a more realistic assumption that p is a quantity we estimate during voting. Since p is a value between 0 and 1, we can use the standard Bayesian estimation approach, modeling p as the binomial distribution parameter and using the $Beta(\alpha, \beta)$ distribution as a conjugate prior for p. In this case, the likelihood of collecting $n_1 + n_2$ votes and observing a combination of n_1 positive and n_2 negative votes is:

$$P(p|\alpha, \beta, n_1, n_2) \sim Beta(\alpha + n_1, \beta + n_2)$$

Note that, since we do not know which of the two classes is correct during voting, our priors should be symmetric, i.e., we always start with $\alpha = \beta$. (This limits the shape of priors we can use, and we alleviate this concern in Section 5.2.) Nevertheless, even operating with a simple prior $Beta(\alpha, \alpha)$ can be remarkably effective, as we illustrate in the experimental evaluation in Section 7.

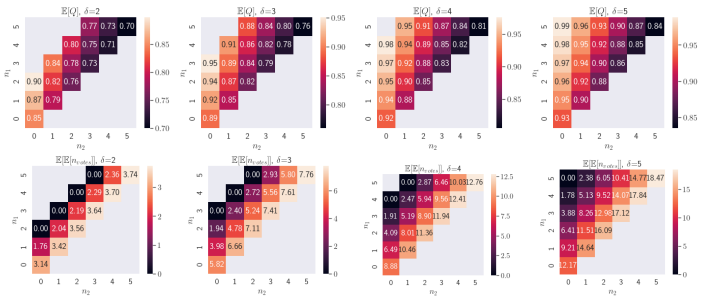
Given the above likelihood for p, we can now leverage the results from Section 4 and get estimates of quality and time till completion based on our prior beliefs and the votes received. For example, Equation 1 (expected quality Q) can be now written as:

$$\begin{aligned} &Q(\delta, \alpha, \beta, n_1, n_2) = \\ &= \int_0^1 Q(\varphi(p), \delta, n_1, n_2) \cdot P(p|\alpha, \beta, n_1, n_2) dp = \\ &= \int_0^1 \frac{1 - \varphi(p)^{\delta+n_1-n_2}}{1 + \varphi(p)^{2\delta}} \cdot \frac{p^{\alpha-1+n_1} \cdot (1-p)^{\beta-1+n_2}}{B(\alpha + n_1, \beta + n_2)} dp \end{aligned} \tag{5}$$

Similarly, we can rewrite all the other equations of interest by integrating over all possible values of p.[9] While these equations lack the "closed form simplicity" of the equations in Section 4, they

[9] Alternatively, we can assume that $p > 0.5$ and integrate partially over the domain of p, but in that case, we need to add a normalizing coefficient, equal to the CDF of the Beta distribution over the domain.

Table 2: Expected quality of the consensus vote and expected remaining number of steps to completion, for various values of δ and collected votes n_1 and n_2, when we assume with a prior distribution *Beta*$(1, 1)$ for quantity p and assuming $p > 0.5$.



are suitable for estimating the quantities of interest at any point of the voting process. Given that the combinations of n_1 and n_2 that someone would be interested in calculating are reasonably small, we can use a sampling approach (see Section 5.3) to estimate the quantities of interest.[10]

For instance, Table 2 shows the expected quality of the final consensus and the number of steps till completion, when we start with a *Beta*$(1, 1)$ prior, and for different combinations of the collected votes.[11] Notice that as we update our beliefs about p while running the voting process, the probability of labeling an item correctly will be different for different combinations of outcomes. For example, for $\delta = 2$, a voting process that terminates with a combination $(2, 0)$ of two correct and zero incorrect votes will have a different estimate than a process that terminates with a combination $(3, 1)$ of three correct votes and one incorrect vote – and different form $(5, 7)$. This intuitively makes more sense than the results we would have obtained assuming a p that is fixed.

5.2 Working with mixture priors for p

The prior section demonstrates the basic principle of estimating p given a prior estimate of the worker pool quality and the information from the votes for the item being adjudicated. The approach works for any prior of the form *Beta*(α, β), but this is often insufficient to express our prior knowledge of worker quality. For example, considering a prior distribution with two peaks at the low and high ends of the accuracy may be more realistic—a mixture of two symmetrically skewed *Beta* distributions. For example, a more realistic prior may have peaks at 0.1 and 0.9, as in Figure 7), with a mixture of random votes, reflecting a prior belief that the workers are often 95% accurate, but there are still items that are confusing for the workers.

[10]Note also that we can estimate the quality of the consensus label in the case where we start not at a "neutral" state of zero difference between the two kinds of votes, but from an arbitrary state (n_1, n_2) of votes collected so far. To do so, one needs to substitute the generalized expression in 1 for the modified expression in Section 4.

[11]For the complete set of Monte-Carlo estimated expectations of quality, number of remaining votes to consensus, and the variance thereof for various values of δ and a *Beta*$(1, 1)$ prior for p, please refer to Appendix A, Tables 6,7,8,9.

When the prior distribution of p comprises a blend of Beta distributions, then we are working with a blend of *conjugate* priors. In this case, updating the parameters of the individual Beta distributions as well as the mixture weights can be done concurrently, simplifying the process. Specifically, if we assume that the prior distribution $P(\theta)$ is a mixture of K Beta distributions with mixing proportions $w = \{w_1, w_2, ..., w_k\}$ and corresponding parameters $\{(\alpha_1, \beta_1), (\alpha_2, \beta_2), ..., (\alpha_k, \beta_k)\}$, then the prior distribution can be written as:

$$P(\theta) = \sum_{i=1}^{k} w_i \cdot Beta(\theta; \alpha_i, \beta_i)$$

where θ is the probability of success in a Bernoulli trial. To jointly update both the parameters of the individual Beta distributions and the mixture weights, we can use the following procedure:

(1) **Update the parameters of each Beta distribution**: For the i-th component of the mixture, we have: $\alpha_i^{post} = \alpha_i + n_1$ and $\beta_i^{post} = \beta_i + n_2$, where n_1 is the number of successes in the observed data, $n_1 + n_2$ is the sample size, and (α_i, β_i) are the prior parameters for the i-th component of the mixture.

(2) **Update the mixture weights**: For the i-th component of the mixture, we have:

$$w_i^{post} = \frac{w_i \cdot B(\alpha_i + n_1, \beta_i + n_2)}{\sum_{j=1}^{k} w_j \cdot B(\alpha_j + n_1, \beta_j + n_2)}$$

(3) **Specify the posterior distribution**: Once the updated mixture weights and Beta parameters have been obtained, the posterior distribution can be calculated as:

$$P(\theta | n_1, n_2) = \sum_{i=1}^{k} w_i^{post} \cdot Beta(\theta; \alpha_i^{post}, \beta_i^{post})$$

This procedure is analogous to updating a single Beta prior but includes an additional step of updating the mixture weights. The updated mixture weights reflect the extent to which each individual Beta distribution contributes to the posterior distribution based on the observed data.

5.3 Monte Carlo estimates for the key quantities

We use the Monte Carlo estimator to compute the expectation for a given quantity of interest (i.e., quality, time to completion, etc.). Each time a new vote is collected on an item, we draw N samples from the posterior distribution of p given the current tally $\langle n_1, n_2 \rangle$ of votes. We then compute the average value of the target function across the sampled values $p_i, i \in 1, ..., N$ to obtain the expected value of the target. For example, to estimate the expected value of quality $Q(\delta, \alpha, \beta, n_1, n_2)$, given votes $\langle n_1, n_2 \rangle$, a threshold δ and prior distribution *Beta*(α, β), we compute:

$$\mathbb{E}[Q | \delta, \alpha, \beta, n_1, n_2] \approx \frac{1}{N} \sum_{i=1}^{N} Q(\delta, \varphi(p_i))$$

where p_i are i.i.d. drawn from *Beta*$(\alpha + n_1, \beta + n_2)$.

Here, the meaning of the quantity on the left is *the quality of consensus vote that can be expected given the available data.* Similarly, applying the Monte-Carlo estimator to Equation 2, we can obtain *the expected time to a consensus that can be expected (with respect to*

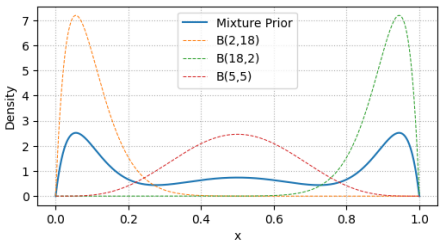


Figure 7: An example of a mixture prior distribution, mixing three Beta distributions: 0.35*B(2,18) + 0.35*B(18,2) + 0.3*B(5,5)

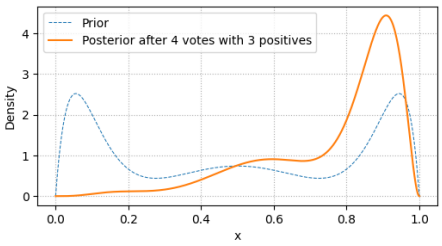


Figure 8: The posterior distribution, starting with prior from Figure 7 and updating after receiving 3 positive out of 4 votes.

randomness in p) given the available data, and, for Equation 3, we will obtain *the variance of time to a consensus that can be expected given the available data.*

6 EQUIVALENCE CONDITIONS FOR TWO δ-MARGIN VOTING PROCESSES

In this section, we build on the results presented so far to explore the following questions:

- How should we design the voting process to generate similar overall quality when we have two sets of workers with different and known accuracies (denoted as p_1 and p_2) (Section 6.1)?
- How should we adjust worker payment to achieve the same level of quality while keeping the overall cost constant when we have two sets of workers with different known accuracies (p_1 and p_2) (Section 6.2)?
- How can we apply these findings when we don't know the exact accuracy of the workers but have prior expectations about their accuracy at the item level?

As discussed in Section 3, the evidence on the effect of monetary reward on the accuracy of work in crowdsourcing is mixed (Kazai 2011, Kazai et al. 2013, Mason and Watts 2009). In this section, we assume that setting the payments differently for a group of workers will have minimal effects on the level of effort (and hence, accuracy) of worker performance.[12] Following the presentation pattern from earlier sections, we first assume that the parameters p_1 and p_2 for the two queues of workers are exogenously given and known. Then we show how to alleviate this assumption.

6.1 Quality equivalence for processes with different values of δ

Suppose we have two sets of workers, one with accuracy p_1 and another with accuracy p_2. The first set of workers operates a δ-margin majority voting scheme with threshold parameter δ_1. Assuming, for now, that the employer only cares about the quality of the resulting work, what is the value of δ_2 so that the set of workers with accuracy p_2 generate the same quality of the results?

Corollary 4.1. *If workers classify an item with accuracy $p_1 \neq \frac{1}{2}$ and threshold δ_1, we can achieve the same quality of results by a set of workers with accuracy $p_2 \neq \frac{1}{2}$ if we set the threshold δ_2 to be:*

$$\delta_2 = \delta_1 \cdot \frac{\ln \varphi_1}{\ln \varphi_2} \tag{6}$$

where $\varphi_1 = \frac{p_1}{1-p_1}$ and $\varphi_2 = \frac{p_2}{1-p_2}$ are the odds *of a single worker classifying the item correctly. The result is obtained by setting $Q(\varphi_1, \delta_1) = Q(\varphi_2, \delta_2)$ (from Equation 1) and solving for δ_2.*

6.2 Worker Pay Equivalence

Assume that workers with accuracy φ are paid $pay(\varphi)$ per vote. Then a reasonable notion of the cost associated with labeling a single example is:

$$Cost(\varphi, \delta) = pay(\varphi) \cdot n_{votes}(\varphi, \delta)$$

where $pay(\varphi)$ is constrained to take only non-negative values. Then, using Equation 2, the expected cost of classifying an item is:

$$\mathbb{E}[Cost|\varphi, \delta] = pay(\varphi) \cdot \delta \cdot \frac{\varphi + 1}{\varphi - 1} \cdot \frac{\varphi^{\delta} - 1}{\varphi^{\delta} + 1} \tag{7}$$

For now, we assume that an employer is *risk-neutral* and cares only about the expected quality of the result and the expected cost.[13] A risk-neutral employer would like to fairly pay teams of workers with different accuracies p_1 and p_2. As long as the teams can generate results of equal utility to the requester, they should be paid the same amount.

From Section 6.1, we can set δ_2 to adjust for the different worker accuracy p_2, thereby assuring that the quality of results is the same: $Q(\varphi_1, \delta_1) = Q(\varphi_2, \delta_2)$. Of course, as shown in Section 4.3.1, a different consensus threshold δ_2 also changes the expected number of votes required to reach consensus. When $\delta_2 = \delta_1 \cdot \frac{\ln \varphi_1}{\ln \varphi_2}$, we have results of equal quality; we can ensure equal costs by setting:

$$\mathbb{E}[Cost|\varphi_1, \delta_1] = \mathbb{E}\left[Cost|\varphi_2, \delta_1 \cdot \frac{\ln \varphi_1}{\ln \varphi_2}\right]$$

With a few simple algebraic manipulations, and knowing that $\varphi_2^{\delta_1 \cdot \frac{\ln \varphi_1}{\ln \varphi_2}} = \varphi_1^{\delta_1}$, we get:

[12] In Appendix B, we provide some additional, related discussion: If workers can vary their effort, and therefore the time invested and the resulting quality, what type of effort-quality curves result in unique payment-maximizing points, and what type of effort-quality curves make workers indifferent to the quality-sensitive payment scheme?

[13] It is reasonably easy to extend the results to the case of risk-averse requesters by using our results for the variance of time until consensus (which is a proxy for the cost of the process).

Corollary 4.2. *If workers with response accuracy φ_1 are paid $pay(\varphi_1)$ per vote, then workers with accuracy φ_2 will generate results of the same quality and the same cost, if the ratio of the payments is:*

$$\frac{pay(\varphi_1)}{pay(\varphi_2)} = \frac{\ln \varphi_1}{\ln \varphi_2} \cdot \frac{\varphi_2 + 1}{\varphi_1 + 1} \cdot \frac{\varphi_1 - 1}{\varphi_2 - 1} \tag{8}$$

where $\varphi_i = \frac{p_i}{1-p_i}$ are the odds that a worker in pool i classifies an item correctly with $p_i \neq \frac{1}{2}$. Based on the above, we can infer that

$$pay(\varphi) \propto \ln \varphi \cdot \frac{\varphi - 1}{\varphi + 1} \tag{9}$$

and if expressed as a function of p:

$$pay(p) \propto (\log(p) - \log(1-p)) \cdot (p - (1-p)) \tag{10}$$

which can also be interpreted as a measure of the information gain provided by the workers while adjudicating the task.

6.3 The case of unknown p

In this subsection, we relax the assumption of knowing apriori the accuracy values of the worker pool. We follow the same pattern as in Section 5: we start with a prior belief about the quality of the worker pool, and we update the belief as we receive votes. We examine first the case where the worker pool has a single, but unknown, accuracy. Then we examine the case where the accuracies of the worker pools are dependent and vary across items.

6.3.1 Single but unknown accuracy for the worker pool. If we want to estimate the payment for a worker pool with an accuracy value p, we can use Equation 10 and assume that p is a random variable following a $Beta(a, b)$ distribution. In that scenario, the value $\frac{p}{1-p}$ follows a Beta Prime distribution, and the log of a Beta Prime ($\log \frac{p}{1-p}$) is known to follow the logistic-beta distribution. In this scenario, if we treat each worker pool independently and integrate over Equation 10 with $p \sim Beta(a, b)$ (Archer et al. 2014), then the expected payment is:

$$pay(a, b) \propto \frac{a-b}{a+b} \cdot (\psi(a) - \psi(b)) \tag{11}$$

where $\psi(\cdot)$ is the digamma function.

Figure 9 illustrates the function values corresponding to various values of a and b. It's important to emphasize that these function values are primarily useful for comparative purposes rather than as absolute values. For instance, consider a scenario where a pool adjudicates an item with a $\delta = 3$, a voting result of 3-0, and a prior of $Beta(1, 1)$. In this case, the payment level corresponds to cell 4-1 in the figure, resulting in a payment level of 1.1 per vote. Alternatively, if another pool adjudicates the same item with $\delta = 4$ and a voting result of 5-1 (corresponding to cell 6-2 in the figure), the payment per vote decreases to 0.64. Despite the different voting results and payment levels, the expected quality of both outcomes remains the same, with $E[Q|3, 0] = E[Q|5, 1] = 0.951$, as per Equation 5. (Detailed values for $E[Q]$ can be found in Tables 6,7,8,9 in Appendix A.) In contrast, an adjudication with a voting score of 5-2 results in a lower payment of 0.26 per vote and a reduced expected quality of $E[Q|5, 2] = 0.88$.

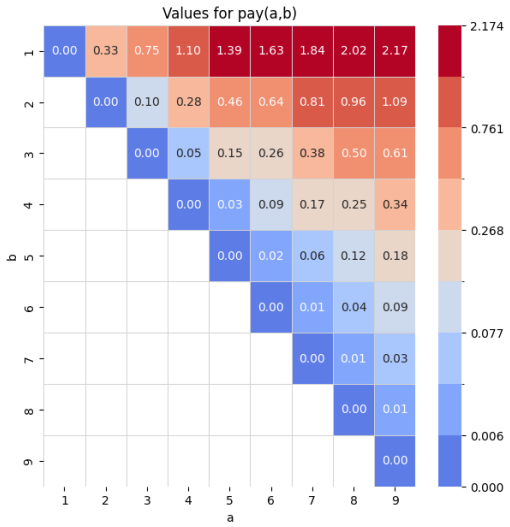


Figure 9: The values of the payment function $pay(p|a, b)$ from Equation 11 for different values of a and b.

6.3.2 Payment ratios under a joint distribution of worker accuracies. Suppose we aim to determine the optimal balance of payments across two distinct worker pools with unknown accuracies, such as a pool of "novice" and "experienced" workers. We must comprehend how each pool would adjudicate the same items to achieve this. The process we follow is outlined below:

(1) We start with two worker pools, each with our prior beliefs about their accuracy, as discussed in Section 5. Let's denote the priors for the two pools as $Beta(a_1, b_1)$ and $Beta(a_2, b_2)$, respectively.
(2) For each item i, we proceed as follows:
   (a) Item i is independently adjudicated by both worker pools.
   (b) From the first pool, we receive y_{i1} positive votes and $n_{i1} - y_{i1}$ negative votes.
   (c) Similarly, for the second pool, we receive y_{i2} positive votes and $n_{i2} - y_{i2}$ negative votes.
   (d) We then use Equation 11 to estimate the payments for each pool.
   (e) The payment for the first pool, pay_{i1}, is calculated as $pay(a_1 + y_{i1}, b_1 + n_{i1} - y_{i1})$.
   (f) The payment for the second pool, pay_{i2}, is calculated as $pay(a_2 + y_{i2}, b_2 + n_{i2} - y_{i2})$.

The payment ratio between two pools can be calculated using either micro- or macro-aggregation across items. Macro-aggregation first calculates the overall payment for each pool and then computes the ratio: $\frac{\sum_i pay_{i1}}{\sum_i pay_{i2}}$. On the other hand, micro-aggregation first calculates the ratios $\frac{pay_{i1}}{pay_{i2}}$ for each item and then aggregates these ratios. In the case of micro-aggregation, the geometric mean of the ratios is typically more stable than the arithmetic mean.

At the end of this process, we obtain a payment ratio that ensures a "fair" balance of payments between the two worker pools. When

the pools produce similar quality outcomes, they receive the same payment per item.

7 EXPERIMENTAL EVALUATION

While theoretically sound and asymptotically accurate, the above results naturally invite the question: How effectively do they mirror the outcomes of a genuine crowdsourced voting process? Our models have various simplifying assumptions and, according to the well-known adage, "all models are wrong, but some are useful." Consequently, we aim to determine whether our model can provide useful guidance in structuring a crowdsourcing process to achieve specified quality standards within defined cost parameters. In this section, we use data from an actual crowdsourcing process. We compare the theoretical values of quality, completion time, and variance thereof with (simulated) outcomes that use real-world data.

7.1 Data set description

We use a publicly available dataset of real MTurk annotator votes for our evaluation – the *Bluebirds* dataset (Welinder et al. 2010). This dataset contains 39 binary labels from different workers for each of the 108 unique images. It also includes the ground truth labels for each item (with a 44:56 size ratio of the *True* and *False* classes).[14] The histogram in Figure 10 shows the empirical distribution of individual accuracy levels among workers, computed as the frequency of labeling an item correctly. The mean worker accuracy stands at 0.636, with significant heterogeneity in the labeling quality for each item. The distribution of computed per-item rates of correct response suggests that a third of items are "misleading" for the average worker (i.e., the frequency of correct responses across all workers for an item is below 0.5).

One of the advantages of the *Bluebirds* dataset is a relatively large number of labels per item. While no realistic process will ever collect 39 labels for an item, by having such a large number of labels per item, we can perform multiple simulations by drawing a small(er) number of votes and examining the outcomes of the voting process. This allows multiple runs of the δ-margin voting process to be conducted, allowing us to understand whether the model predictions are accurate. Figure 11 shows the total distribution of all votes in the `Bluebirds` dataset, relative to ground truth (also given in the dataset).

7.2 Simulation setup

We first want to know how well the quality estimate $Q(\phi, \delta)$ in Equation 1 and the time estimate $\mathbb{E}[n_{votes}|\varphi, \delta]$ in Equation 2 match the actual results of a crowdsourcing process when people vote in a crowdsourcing labeling task. We compare the theoretical values to estimations with the results of our simulations that use real data. We simulate multiple voting runs using random draws from our dataset. We then compare the results of the simulated runs with our quality estimate to examine the accuracy of our prediction.

Specifically, for a given $\delta \in [1, 2, .., 11]$, the the process we used for simulating δ-margin voting is the following:

[14]For more information on the dataset, please see (Welinder et al. 2010). The data set is available at https://github.com/welinder/cubam.

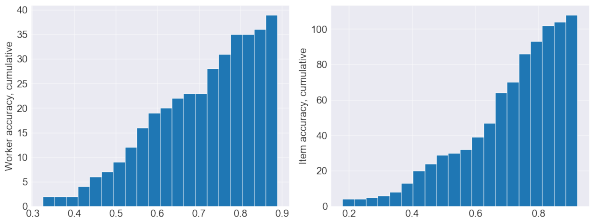


Figure 10: Cumulative histograms describing the *Bluebirds* dataset. Left: worker accuracy (calculated as proportion of correct answers among the votes on all items for a given worker). Right: item difficulty (average accuracy of all worker responses for a given item).

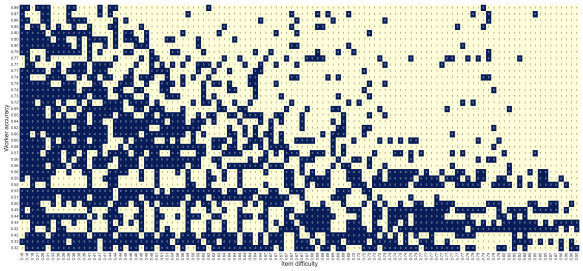


Figure 11: All votes in the `Bluebirds` dataset, relative to ground truth; light color means correct answer; dark color means incorrect answer. The x-axis shows each item in the dataset; the items are ordered based on response correctness across workers in ascending order ('item difficulty'), with the easiest items being on the right. The y axis shows the individual workers; the workers are ordered based on response correctness across items in ascending order ('worker accuracy'), with the most accurate workers at the top.

- We run $r = 100$ simulations of the δ-margin voting process for each item.
- For a given item, we iteratively draw votes (with replacement) from the *Bluebirds* dataset labels (which are real-life observational votes given by real workers on this item) until we attain the agreement threshold δ.
- We perform sampling *with replacement*. This prevents termination of the voting process due to exhausting the total set of available votes[15] for high values of δ. The results are similar to sampling without replacement for smaller values of δ (e.g., up to 5), which captures most real-life settings.
- We compute the consensus vote quality (correct or not) and the number of votes to completion for the 100 simulations for each item.

Once we have the results of the simulation, we can then compare the obtained results with our theoretical predictions and see how accurate our model is.

[15]Total number of available votes is 39 per item.

		Q Empirical					Q Theoretical					n_{votes} Empirical					$\mathbb{E}[n_{votes}]$ Theoretical				
	δ	2	3	4	5	6	2	3	4	5	6	2	3	4	5	6	2	3	4	5	6
Item ID	p																				
11573	0.69	0.83	0.94	0.96	0.98	1.00	0.84	0.92	0.96	0.98	0.99	3.40	5.92	9.54	12.42	14.74	3.48	6.54	9.62	12.56	15.36
11574	0.49	0.47	0.48	0.41	0.45	0.41	0.47	0.46	0.45	0.44	0.42	3.88	10.22	18.54	26.78	36.94	4.00	8.98	15.95	24.87	35.73
11575	0.67	0.70	0.88	0.95	0.98	0.98	0.80	0.89	0.94	0.97	0.98	3.60	6.96	9.28	14.58	17.46	3.60	7.00	10.59	14.09	17.45
11577	0.36	0.26	0.17	0.12	0.04	0.01	0.24	0.15	0.09	0.05	0.03	3.78	6.78	11.84	17.36	16.70	3.71	7.46	11.64	15.88	20.00
11578	0.44	0.40	0.33	0.23	0.21	0.16	0.37	0.32	0.26	0.22	0.18	4.30	9.06	13.68	18.46	32.86	3.94	8.62	14.80	22.15	30.37
11579	0.72	0.86	0.94	0.96	1.00	1.00	0.87	0.94	0.98	0.99	1.00	3.26	6.14	9.30	12.04	13.88	3.36	6.10	8.75	11.26	13.66
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
36959	0.74	0.90	0.97	0.98	1.00	1.00	0.89	0.96	0.99	1.00	1.00	3.48	5.74	7.78	9.82	13.10	3.23	5.67	7.98	10.16	12.27
36960	0.79	0.94	0.98	1.00	1.00	1.00	0.94	0.98	1.00	1.00	1.00	2.98	4.86	6.90	8.52	10.96	2.97	4.92	6.72	8.46	10.17
36961	0.77	0.94	0.97	0.98	1.00	1.00	0.92	0.97	0.99	1.00	1.00	2.98	5.18	7.66	8.90	11.84	3.10	5.28	7.31	9.24	11.13
36962	0.77	0.95	0.99	0.99	0.99	1.00	0.92	0.97	0.99	1.00	1.00	2.98	5.10	6.76	9.56	10.90	3.10	5.28	7.31	9.24	11.13
36963	0.90	0.97	0.99	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	2.40	3.70	5.28	5.94	7.76	2.45	3.76	5.03	6.29	7.55
36964	0.85	0.96	0.98	0.99	1.00	1.00	0.97	0.99	1.00	1.00	1.00	2.52	4.42	5.46	7.00	8.86	2.70	4.28	5.77	7.22	8.67

Table 3: Results of empirical vs. theoretical computation of consensus label accuracy $Q(\varphi, \delta)$ and time (number of votes used) until voting completion n_{votes}. The empirical results are averaged across 100 experiments per item of the *Bluebirds* dataset. A sample of 12 items is shown.

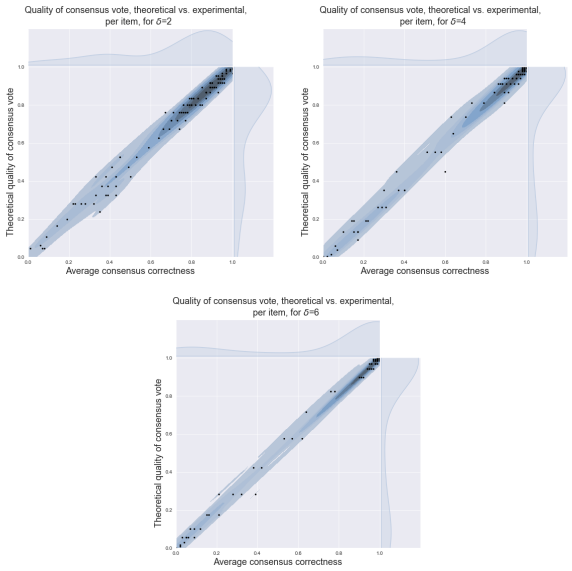


Figure 12: Comparison between theoretical and empirical values of the overall resulting label accuracy for various values of consensus threshold δ. Each scatterplot point represents an item, where the x-coordinate is computed as average correctness for 100 experiments using the *Bluebirds* data. The color indicates the distribution of the estimated density of the joint distribution of estimated vs. theoretical results.

7.3 Evaluating the model accuracy for the quality of the consensus label: the case of known p

First, we want to evaluate how well the estimate $Q(\phi, \delta)$ (Equation 1) predicts the actual outcome of the voting process. For us to calculate $Q(\phi, \delta)$, we first assume that p is known for the item: this is the *expected* worker quality when labeling the item. We take the item difficulty p to be equal to the average accuracy of all 39 workers' responses for this item in the *Bluebirds* dataset (regardless of whether a vote was solicited from all 39 workers during the actual voting). Using these values of p and δ, we compute the **theoretical value of consensus vote quality** from Equation 1.

For example, for the first item in the *Bluebird* data, the empirically-obtained item-level correctness of crowdsourced responses across all 39 workers is 0.692, and $\varphi \approx \frac{0.692}{1-0.692} \approx 2.247$. Hence for a voting process with $\delta = 3$, we get $Q_{th}(\varphi, \delta) \approx \frac{2.247^3}{1+2.247^3} \approx 0.918$. We then compare this value against the average correctness of the consensus vote obtained across the 100 simulated runs.

Figure 12 demonstrates the results of these experiments. Empirical consensus label quality deviates minimally from the theoretical values, with the variance of the difference becoming smaller for stronger consensus requirements (greater δ). The gap is also smaller for more 'certain' items, which are frequently classified either correctly or incorrectly (the discrepancy between theoretical and empirical values appears to be concentrated around 'divisive' items close to the $(0.5, 0.5)$ region of the plot.

Table 3 contains a sample of the detailed results, showing the theoretical and empirical values of the consensus label quality $Q(\varphi, \delta)$. The deviation from the theoretical values is small for every value of δ.

7.4 Theoretical vs. experimental time until completion. (The case of known p)

This section compares the theoretical and empirical values of the votes required to reach a consensus. To estimate the theoretical value $\mathbb{E}[n_{votes}|\varphi, \delta]$ (Equation 2) for each item, we again estimate p for each item using the average accuracy of the workers that labeled it. We then compare the theoretical estimate with the actual number of votes required to each consensus, averaged over the 100 runs.

Figure 13 illustrates the results. For the "easy" items, any sample of workers is likely to agree. Hence, a series of δ votes is likely enough to reach a δ majority – these are the cases in which the discrepancy between the theoretically expected number of votes and the experimental results is the smallest. Considering the change in scale between the left and right panels in Figure 13, one can observe conclude that for higher threshold values δ, a few items may occasionally take a greater number of votes to reach consensus. Still, even such "outlier" cases only drive up the discrepancy between the mean value of N_{votes} and the theoretically expected value by at most 15%. We should also note that the variance (see Equation 3) also increases with δ, so observing such "outliers" is expected.

Table 3 also includes a presentation of the results of our experiments simulating the $delta$-margin voting on the *Bluebirds* items and observational votes, side-by-side with the theoretical quantities

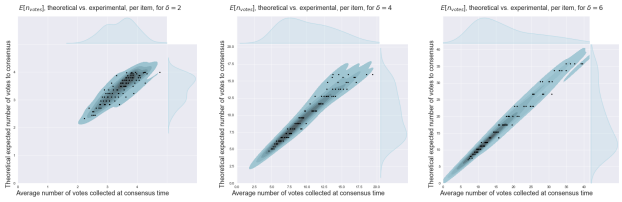


Figure 13: Comparison between theoretical and empirical values of time until completion (i.e., number of votes to reach consensus), for various values of consensus threshold δ. Each scatterplot point represents an item, where the x-coordinate is computed as the average number of votes taken to reach consensus across 100 experiments using the *Bluebirds* data. The color indicates density. (Note the differing scales.)

for the corresponding item. The discrepancy between the theoretical and empirical values of time (i.e., number of votes used) until voting completion n_{votes} is small for every value of δ.

7.5 Monte-Carlo estimates using simulated sequential voting process with *Bluebirds* data. (The case of unknown p)

We now revisit the relaxation of modeling assumptions discussed in Section 5. Recall that we no longer assume that p is known but a random variable estimated using Bayesian updating for a given prior. Below we describe the experiments we ran using Monte Carlo estimation for a $Beta(1,1)$ prior on p and a Bernoulli likelihood, where the sequential voting process is simulated using real-life observational data on the votes from the *Bluebirds* dataset.

The experimental process is as follows: For a given value of δ, $(2 \le \delta \le 5)$ and each of the 108 items in the dataset, run 100 experiments. In each experiment, simulate the sequential voting process by sampling votes randomly with replacement from the worker responses (until reaching a consensus). With each new vote collected, we compute the expectation of the quality $Q(\delta, \alpha, \beta, y, n-y)$ using Monte-Carlo estimation (see Equation 5), given the current state $(y, n-y)$. Similarly, we estimate the expected remaining number of votes until consensus.[16]

Table 4 compares the Monte-Carlo estimated expectation of the quality of the consensus vote, using a $Beta(1,1)$ prior, versus the ground truth quality of the consensus vote. The results are very close, even with a relatively uninformed quality prior. The estimation accuracy can be further improved when the prior distribution for the quality is closer to the actual quality distribution of the labeling process. We should note that we have a relatively small number of cases ending with a 5-3 vote, so there is a higher variance of the empirical estimates compared to the 4-2, 3-1, and 2-0 empirical

[16]Note that the results in Section 2 are formulated for estimating the quality, time until consensus, and variance when starting from the initial voting state – zero votes. However, using the matrix formulation in Section 4.3.1, obtaining the results for the number of steps remaining is trivial, given the current state of the voting $(y, n-y)$, which corresponds to the transient state $2y-n$.

Terminal vote counts	Ground truth Q, averaged	$Q(\delta, \alpha, \beta, y, n-y)$, Monte-Carlo
2-0	0.843	0.848
3-1	0.829	0.840
4-2	0.796	0.802
5-3	0.809	0.774

Table 4: Summary of results of experiments for estimating quality when p is unknown. Ground truth quality is recorded for a δ-margin voting process with $\delta = 2$. Given the observed combination of votes at the terminal stage, the expected quality is computed using Monte-Carlo estimation for $Beta(1,1)$ prior. The resulting estimates for each voting process were averaged across all voting processes that terminated with the same final combination of votes.

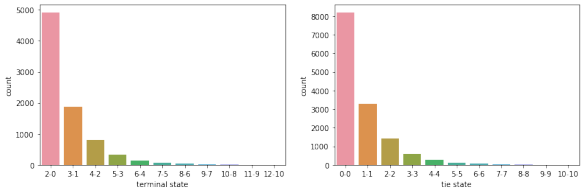


Figure 14: Left: Frequency of occurrence for all possible terminal vote pairs in δ-margin voting with $\delta = 2$, for all 108 items, for 100 experiments per item. Right: Frequency of occurrence for the (initial or intermediate) tied-states vote pairs in δ-margin voting with $\delta = 2$, for all 108 items, for 100 experiments per item.

estimates (see Figure 14 for the frequency distribution of terminal states).

Table 5 compares the Monte-Carlo theoretical estimates and the empirical number of votes to completion. In contrast to the results on quality demonstrated above, we show the results for various "intermediate states" for the time to completion, showing that our estimates can provide a good estimate of how much longer the voting process is expected to take. The results are similar for other intermediate states and values of δ.

8 MANAGERIAL EXAMPLE

We have had a chance observe the usage of the δ-margin majority voting within an organization that provides services for augmenting machine learning workflows. In the first case, alerts about money laundering were generated by a machine learning system, and humans had to examine the available evidence and decide whether the case required further escalation or whether it was a false alert; in the second case, the machine learning algorithm was identifying database entries as potential duplicates (customers, businesses, products) and the human labelers had to examine the available evidence and decide whether to merge the two entries or leave them as-is.

(Quasi-) initial vote counts	Ground truth remaining n_{votes}, averaged	$\mathbb{E}[n_{votes}]$, Monte-Carlo
0-0	3.409	3.141
1-1	3.521	3.419
2-2	3.552	3.560
3-3	3.633	3.642
4-4	3.712	3.697
5-5	3.675	3.740

Table 5: Comparison of ground truth vs. Monte-Carlo expectation of key quantities, for $\delta = 2$. We show the remaining time to termination vs. the Monte-Carlo estimates of $\mathbb{E}[n_{votes}]$ across all the processes that reached the state in the first column.

The clients of the firm, who were the owners of the machine learning processes, wanted to have guarantees of quality. The adopted solution for measuring quality was to generate some cases of "gold" examples and measure how often the workers were labeling the cases correctly. After generating thousands of such "gold" examples, which was a costly process, the measurements indicated that the delivered performance was well below the agreed quality ("99% accuracy") in the service level agreement.

It was unclear to the firm what the appropriate course of action should be to fix the quality issue: while the quality measurement using gold examples could estimate past performance, it was not usable as a prescriptive tool. There is a trade-off between cost and quality, but the firm did not have any principled way to choose, and it remained unclear if the firm could meet the quality expectations within the agreed budget. The only solution was to deploy various schemes and see empirically what works.

One idea was to increase the value of δ, but they did not know the effect on the deliverables' quality or the expected cost increase. Alternatively, they could further invest in training the annotators or migrate the process to a team of experienced employees with better performance and a significantly higher compensation cost. For the money laundering case, the labeling quality was about 78% average worker accuracy (i.e., $\varphi \approx 3.5$); the firm used $\delta = 2$, which resulted in a delivered quality of $Q(\varphi, \delta) = 0.92$ and an expected cost of $\mathbb{E}[n_{votes}|\varphi, \delta] \approx 3$ votes per example.

Based on our analysis, reaching the target quality of 99% requires setting $\delta = 4$, which also results in an expected $\mathbb{E}[n_{votes}|\varphi, \delta] \approx 7$, which would increase the cost by a factor of 132%, compared to the existing setting. The alternative is to switch the workload to the pool of more experienced workers, who had an accuracy of about 91% ($\varphi \approx 10$) and could meet the quality requirement with $\delta = 2$ and $\mathbb{E}[n_{votes}|\varphi, \delta] \approx 2.4$. However, the cost of the senior workers was almost double that of the junior ones, and, more importantly, the senior workers' supply was insufficient to meet the required annotation volume.

The adopted solution was to provide a bonus incentive of 10% to junior workers when they meet an individual quality target of 82% average accuracy ($\varphi \approx 4.5$). Under this setting, they could still run the process with $\delta = 3$, and $Q(\varphi, \delta) = 0.99$ with $\mathbb{E}[n_{votes}|\varphi, \delta] \approx 4.6$. This still resulted in an increased cost of 68% compared to the prior setting that did not meet the quality targets, which was then negotiated with the client.

While not yet implemented by the time of this manuscript, we also expect to see deployments that use our analysis in Section 5 and allow the implementation of solutions that dynamically move items across queues based on the dynamically estimated quality and cost for labeling an item.

9 CONCLUSIONS, MANAGERIAL IMPLICATIONS, AND FUTURE WORK

9.1 Summary

Due to its convenience and ease of understanding, δ-margin majority voting is widely used for aggregating worker votes on crowdsourcing platforms. However, it is commonly regarded as a heuristic approach without a comprehensive theoretical foundation. Our research aims to address this gap by proposing a model that represents the process as a Markov chain with absorbing states. By analyzing this model, we can derive valuable insights into various properties such as expected quality and time to consensus.

One significant advantage of our approach is its applicability to diverse crowdsourcing settings. We accommodate heterogeneous worker qualities as long as the mean worker quality is known. Moreover, even without prior knowledge about worker quality, we demonstrate that the accumulated votes can be utilized to estimate worker quality. By leveraging our analytical results, we can operate effectively under uncertainty.

We conducted experimental evaluations using real voting data to validate our theoretical findings. The results indicate that our estimates align well with practical outcomes in scenarios where worker quality is known and when it needs to be estimated dynamically. Our study contributes valuable insights into the understanding and practical implementation of delta-margin majority voting on crowdsourcing platforms.

9.2 Managerial Implications

Our findings have significant implications for effectively managing worker pools in crowdsourcing settings. One key advantage of the δ-margin majority voting method is its familiarity, as it does not require the implementation of complex worker allocation schemes. Leveraging our results, both before the voting process and in real-time, can offer valuable insights into the expected quality of deliverables and the anticipated completion time.

In critical decision-making scenarios where maintaining high levels of quality is crucial, our analysis can serve as a guiding tool during the voting process. For instance, if our estimates predict a quality level below the acceptable threshold, the process owner can make informed decisions, such as increasing the delta parameter or potentially transitioning the annotation process to a different worker pool with higher expected accuracy.

By utilizing our theoretical framework, crowdsourcing platforms can enhance their decision-making capabilities, optimize resource allocation, and ultimately improve the overall quality of outcomes. Our study empowers platform owners to make data-driven choices and take proactive measures to ensure desired levels of quality and efficiency in crowdsourcing endeavors.

Furthermore, our research includes guidelines on establishing payment schemes that align with worker performance. By considering the performance of individual workers, we can design a compensation structure that rewards their contributions accordingly. This approach ensures that workers who deliver high-quality results receive appropriate compensation.

When organizing workers in pools, based roughly on performance, we can offer differentiated payment schemes and even design schemes that balance required accuracy and cost. For example, easy tasks can be directed to low-accuracy and low-cost worker pools. When detecting that the item requires higher accuracy, we can escalate to a higher-quality worker pool.

Our comprehensive analysis, encompassing the aggregation of worker votes and the appropriate reward mechanisms, provides a holistic approach to optimize crowdsourcing outcomes, ultimately leading to improved results and possibly higher worker satisfaction.

9.3 Future Work

One limitation of our current work is its focus on binary classification tasks. To expand the scope of our research, future work should aim to derive corresponding results applicable to non-binary classification tasks. One possible approach is to represent multi-class tasks as a series of binary ones by combining all but the correct class labels. This would allow us to extend our current theoretical framework to handle multi-class scenarios.

Additionally, we can explore the application of our analysis to continuous tasks, such as regression problems. By providing theoretical estimates of quality and cost for such tasks, we can further enhance the understanding and optimization of crowdsourcing processes in a broader range of applications.

Looking ahead, we plan to extend our analysis to other types of crowdsourcing processes, specifically investigating the interplay between quality, time to completion, and payment costs. Our goal is to develop a modular set of analyses that enable practitioners to input the structure of a crowdsourcing task or process and obtain ex-ante predictions of its behavior. This would provide valuable insights without the need to execute the process itself. For example, we can consider iterative tasks, where workers try to improve upon each other's work: Can we establish a framework that estimates the quality and the number of iterations until convergence? By investigating these aspects in future work, we aim to allow practitioners to infer cost-optimal parameters for multistage workflows in advance.

A MONTE CARLO ESTIMATES OF THE KEY TARGETS

Tables 6,7,8,9 list the results of Monte-Carlo simulations for the key quantities examined in this paper. In particular, the expectation is estimated for the following quantities: the probability Q of the correctness of the consensus vote, the expected number of votes $\mathbb{E}[n_{votes}]$ needed to reach consensus from a given state, the variance of the latter $Var(n_{votes})$, and the payment for one worker pool (equation 10). In 6,7,8,9, the columns n_1 and n_2 define the current state of the voting process – namely, the number of votes for class 1 and class 2 respectively. Note that in the fourth and fifth columns, n_{votes} refers to the number of votes *remaining* until the termination of the process, given the current state (n_1, n_2). Accordingly, for 'correct' terminal states the expected quality is always equal to one. Instead of Monte-Carlo estimates, the last column in all tables contains direct computations of the payment rate for a single worker pool from equation 11 given a prior $Beta(1, 1)$. This quantity is independent of δ. We compute the payment value for the terminal states only.

$\delta = 2$					
n_2	n_1	$\mathbb{E}[Q]$	$\mathbb{E}[\mathbb{E}[\mathbf{n_{votes}}]]$	$\mathbb{E}[\mathbf{Var(n_{votes})}]$	**pay***
0	0	0.847	3.138	3.998	
1	0	0.954	1.758	2.862	
1	1	0.791	3.423	5.128	
2	**0**	1.000	0.000	0.000	0.750
2	1	0.930	2.036	4.013	
2	2	0.756	3.561	5.754	
3	**1**	1.000	0.000	0.000	0.278
3	2	0.913	2.183	4.695	
3	3	0.732	3.640	6.129	
4	**2**	1.000	0.000	0.000	0.146
4	3	0.901	2.282	5.140	
4	4	0.716	3.697	6.401	
5	**3**	1.000	0.000	0.000	0.090
5	4	0.891	2.365	5.491	
5	5	0.700	3.737	6.589	

Table 6: Results of Monte Carlo estimation of the key quantities for various values of δ, when starting with a prior $Beta(1, 1)$ for the item difficulty p. For $\delta = 2$.

$\delta = 3$					
n_2	n_1	$\mathbb{E}[Q]$	$\mathbb{E}[\mathbb{E}[\mathbf{n_{votes}}]]$	$\mathbb{E}[\mathbf{Var(n_{votes})}]$	**pay***
0	0	0.893	5.836	17.582	
1	0	0.955	3.978	12.988	
1	1	0.846	6.663	23.886	
2	0	0.986	1.935	6.917	
2	1	0.930	4.755	19.062	
2	2	0.817	7.119	27.623	
3	**0**	1.000	0.000	0.000	1.100
3	1	0.977	2.409	11.096	
3	2	0.912	5.236	22.972	
3	3	0.794	7.408	30.275	
4	**1**	1.000	0.000	0.000	0.464
4	2	0.969	2.719	14.168	
4	3	0.897	5.559	25.806	
4	4	0.776	7.606	32.138	
5	**2**	1.000	0.000	0.000	0.261
5	3	0.962	2.933	16.482	
5	4	0.885	5.799	27.956	
5	5	0.762	7.771	33.694	
6	**3**	1.000	0.000	0.000	0.168
6	4	0.957	3.106	18.239	
6	5	0.875	5.989	29.810	
6	6	0.750	7.897	34.921	
7	**4**	1.000	0.000	0.000	0.118
7	5	0.951	3.241	19.700	
7	6	0.867	6.137	31.006	
7	7	0.739	7.988	35.976	

Table 7: Results of Monte Carlo estimation of the key quantities for various values of δ, when starting with a prior $Beta(1, 1)$ for the item difficulty p. For $\delta = 3$.

$\delta = 4$					
n_2	n_1	$\mathbb{E}[\mathbf{Q}]$	$\mathbb{E}[\mathbb{E}[\mathbf{n_{votes}}]]$	$\mathbb{E}[\mathbf{Var(n_{votes})}]$	$\mathbf{pay}^*$
0	0	0.917	8.915	45.960	
1	0	0.961	6.501	33.656	
1	1	0.880	10.455	64.657	
2	0	0.984	4.078	21.512	
2	1	0.938	7.979	51.128	
2	2	0.854	11.383	76.509	
3	0	0.995	1.903	9.857	
3	1	0.972	5.199	35.158	
3	2	0.920	8.910	62.978	
3	3	0.835	11.948	84.664	
4	**0**	1.000	0.000	0.000	1.389
4	1	0.991	2.469	17.644	
4	2	0.962	5.932	45.913	
4	3	0.906	9.555	71.839	
4	4	0.818	12.395	90.737	
5	**1**	1.000	0.000	0.000	0.642
5	2	0.987	2.871	24.364	
5	3	0.954	6.474	53.891	
5	4	0.895	10.012	78.803	
5	5	0.806	12.727	96.691	

$\delta = 4$					
n_2	n_1	$\mathbb{E}[\mathbf{Q}]$	$\mathbb{E}[\mathbb{E}[\mathbf{n_{votes}}]]$	$\mathbb{E}[\mathbf{Var(n_{votes})}]$	$\mathbf{pay}^*$
6	**2**	1.000	0.000	0.000	0.380
6	3	0.983	3.198	29.625	
6	4	0.946	6.885	60.942	
6	5	0.884	10.382	84.901	
6	6	0.793	13.031	100.967	
7	**3**	1.000	0.000	0.000	0.253
7	4	0.980	3.442	34.271	
7	5	0.940	7.253	66.604	
7	6	0.876	10.736	89.723	
7	7	0.781	13.261	104.608	
8	**4**	1.000	0.000	-0.000	0.181
8	5	0.977	3.656	37.907	
8	6	0.934	7.518	71.345	
8	7	0.868	10.998	94.118	
8	8	0.774	13.463	107.939	
9	**5**	1.000	0.000	0.000	0.136
9	6	0.974	3.831	41.596	
9	7	0.929	7.765	75.524	
9	8	0.862	11.237	97.239	
9	9	0.765	13.602	110.709	

Table 8: Results of Monte Carlo estimation of the key quantities for various values of δ, when starting with a prior *Beta*$(1, 1)$ for the item difficulty p. For $\delta = 4$.

$\delta = 5$					
n_2	n_1	$\mathbb{E}[\mathbf{Q}]$	$\mathbb{E}[\mathbb{E}[\mathbf{n_{votes}}]]$	$\mathbb{E}[\mathbf{Var(n_{votes})}]$	$\mathbf{pay}^*$
0	0	0.933	12.145	94.191	
1	0	0.966	9.187	68.850	
1	1	0.902	14.659	135.330	
2	0	0.984	6.393	45.822	
2	1	0.945	11.570	105.911	
2	2	0.879	16.055	161.141	
3	0	0.994	3.909	26.198	
3	1	0.972	8.267	77.027	
3	2	0.929	12.966	132.362	
3	3	0.862	17.097	181.086	
4	0	0.998	1.783	10.863	
4	1	0.987	5.146	48.012	
4	2	0.962	9.500	101.607	
4	3	0.917	14.044	152.959	
4	4	0.848	17.821	197.148	
5	**0**	1.000	0.000	0.000	1.631
5	1	0.996	2.381	21.910	
5	2	0.982	6.080	66.724	
5	3	0.953	10.485	120.725	
5	4	0.907	14.823	169.516	
5	5	0.836	18.422	209.422	
6	**1**	1.000	0.000	0.000	0.806
6	2	0.994	2.842	31.572	
6	3	0.977	6.760	82.731	
6	4	0.946	11.203	136.651	
6	5	0.897	15.509	182.890	
6	6	0.826	18.978	221.973	
7	**2**	1.000	0.000	-0.000	0.497
7	3	0.992	3.238	40.730	
7	4	0.972	7.350	95.839	
7	5	0.939	11.855	151.603	
7	6	0.888	16.051	195.441	
7	7	0.815	19.336	230.199	
8	**3**	1.000	0.000	-0.000	0.340
8	4	0.990	3.556	49.011	
8	5	0.969	7.842	106.722	
8	6	0.934	12.360	162.937	
8	7	0.879	16.518	205.669	
8	8	0.808	19.708	239.370	
9	**4**	1.000	0.000	0.000	0.249
9	5	0.988	3.789	55.714	
9	6	0.965	8.251	118.670	
9	7	0.928	12.771	172.994	
9	8	0.872	16.854	215.285	
9	9	0.799	20.008	247.119	
10	**5**	1.000	0.000	0.000	0.190
10	6	0.986	4.022	62.404	
10	7	0.961	8.588	126.756	
10	8	0.924	13.141	181.694	
10	9	0.867	17.251	223.283	
10	10	0.791	20.295	253.224	
11	**6**	1.000	0.000	0.000	0.150
11	7	0.985	4.239	68.460	
11	8	0.958	8.894	134.953	
11	9	0.918	13.505	190.358	
11	10	0.861	17.536	231.317	
11	11	0.785	20.540	258.980	

Table 9: Results of Monte Carlo estimation of the key quantities for various values of δ, when starting with a prior *Beta*$(1, 1)$ for the item difficulty p. For $\delta = 5$.

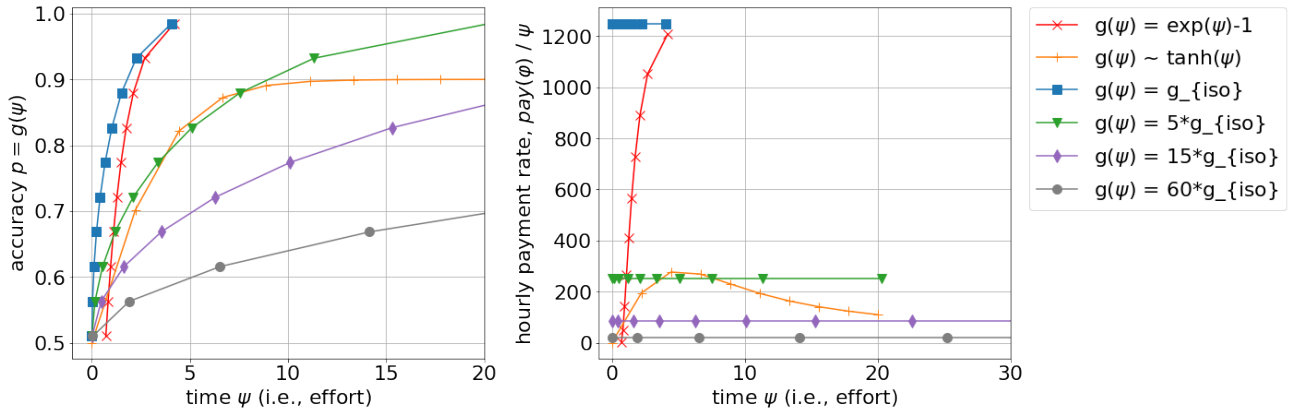


Figure 15: Properties of various accuracy-effort curves, plotted for a range of accuracy values $0.51 \leq p \leq 0.98$. Left: sample iso-payment accuracy curves, contrasted with two accuracy functions chosen without iso-payment considerations: $p = e^{\psi} - 1$ and $p \sim tanh(\psi)$. Right: visual demonstration of the iso-payment property.

B ISO-PAYMENT EFFORT FUNCTION

Using the results of corollary 3, we can further investigate the relationship between payment per item and the per-item *effort* made by a worker. Consider that for a given worker, an unknown function $p = g(\psi)$ governs the relationship between the effort ψ expended by the worker on an item and the accuracy p of the response given. (One possible measurable proxy for the level of effort is the amount of time spent working on an item.) Given our understanding of the relative amounts of payment, given in (8), we can connect payment to effort for a given function $g(\psi)$.

For example, assume that the relationship between a worker's effort expended and accuracy of the answers is $p = e^{\psi} - 1$. This particular accuracy-effort function is represented with the x-marked line on the left panel of Figure 15. Since we are interested in relative adjustments to payment rate, set $pay(\varphi_2)$ in equation (8) to be equal to 1. We then get $pay(\varphi) = c_0 \cdot ln(\varphi) \cdot \frac{\varphi-1}{\varphi+1}$, where c_0 is a constant, and φ is equal to $\frac{p}{1-p}$ as before. We can now connect worker effort to the level of payment per unit of effort $pay(\varphi)/\psi$ (if we continue our example of effort measured as time spent per item, this last quantity would represent *hourly wage*). The x-marked line on the right panel of Figure 15 represents the relationship for our particular chosen function $p = g(\psi) = e^{\psi} - 1$.

We now ask: what relationship between a worker's effort and the accuracy of her responses would make her payment invariant to effort expended? In other words, we are looking for such $g(\psi)$ that the 'hourly wage' $pay(\varphi)/\psi$ is independent of ψ. It is immediate that all such functions can be defined by

$$\psi = c \cdot ln\varphi \cdot \frac{\varphi - 1}{\varphi + 1} =: g_{iso}^{-1}(\varphi), \tag{12}$$

where c is a constant. Figure 15 illustrates a sample of several functions $p = g_i(\psi)$, different only in the chosen values of the constant factor c.

A more realistic hypothesis about a relationship between worker accuracy φ and her effort ψ would incorporate an assumption of diminishing returns. Figure 15 also includes a plot of an accuracy function $p = \frac{1}{2} + \frac{2}{5}\tanh(\frac{\psi}{4})$, which conforms to this assumption[17]. Since a worker's natural objective is to maximize pay, the best strategy for a worker with this kind of a relationship between effort and accuracy would be to choose the global maximum of their hourly wage curve, if the global maximum is unique, and otherwise choose a global maximum corresponding to the lowest time expenditure.

C UTILITY-EQUALIZING PAYMENTS

We offered a way to set the pay rate for workers in each pool relative to the other pool, subject to *risk-neutral utility constraints*. A risk-neutral approach would entail a utility function that would only involve the expected cost of getting a consensus vote. However, a more nuanced notion of utility should consider not only the expected quality and expected number of votes required to reach consensus, but also the variance of each. Thus, a *risk-averse* procedure for establishing an appropriate equivalence for payment for a single item is as follows:

$$Q_1(\varphi_1, \delta_1) = Q_2(\varphi_2, x) \implies \delta_2 := x$$

$$U_i := \mathbb{E}[Q(\varphi_i, \delta_i)] - \text{pay}_i(\varphi_i)\mathbb{E}[n_{votes,i}|\varphi_i, \delta_i] - \lambda \cdot \left(\sqrt{Var(Q(\varphi_i, \delta_i))} + \text{pay}_i(\varphi_i)\sqrt{Var(n_{votes,i})}\right)$$

[17]The shape of a hyperbolic tangent is convenient for describing the natural assumption on the diminishing returns to effort. The constants are chosen to scale tanh with accordance to our context.

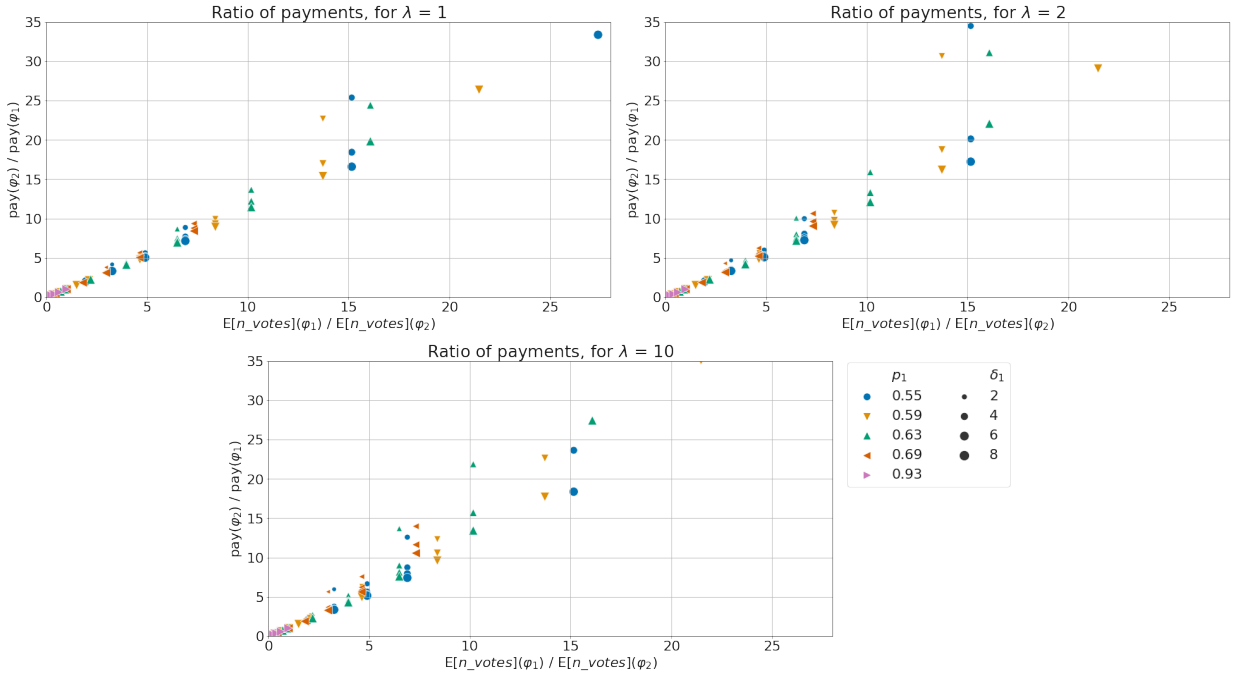


Figure 16: Utility-equalizing ratio of payment rates $pay(\varphi_2)/pay(\varphi_1)$ for workers on two voting processes (13), where δ_2 is derived in a way that satisfies the equal quality constraint. Top left: $\lambda = 1$. Top right: $\lambda = 2$. Bottom: $\lambda = 10$.

$$\begin{cases} U_1 = U_2 \\ Q_1 = Q_2 \end{cases} \iff$$

$$\text{pay}_2 := \text{pay}_1 \frac{\mathbb{E}[n_{votes,1}|\varphi_1, \delta_1] + \lambda\sqrt{Var(n_{votes,1})}}{\mathbb{E}[n_{votes,2}|\varphi_2, \delta_2] + \lambda\sqrt{Var(n_{votes,2})}} \tag{13}$$

A remark on notation: strictly, Q is not a random variable but a first moment of $\mathbb{1}_C$. We use the notation $Var(Q)$ to mean, more strictly, $Var(\mathbb{1}_C)$. Although computing the term $Var(Q(\varphi_i, \delta_i))$ is not necessary for using the payment equivalence condition in Equation 13, it is worth mentioning that the variance of the indicator of reaching a correct consensus is the variance of a Bernoulli random variable: $Var(\mathbb{1}_C) = \mathbb{E}[\mathbb{1}_C^2] - (\mathbb{E}[\mathbb{1}_C])^2 = P(\mathbb{1}_C = 1)(1 - P(\mathbb{1}_C = 1)) = Q(1 - Q)$.

The plots in Figure 16 below visualize the following process: for every fixed value of utility parameter λ and every triplet (p_1, δ_1, p_2) (where $0.5 < p_1 < 1, 0.5 < p_2 < 1$), we compute δ_2 that guarantees $Q_1 = Q_2$. We then derive the ratio of payments $\text{pay}_2/\text{pay}_1$ for a given triplet, and we visually organize the results by the values of corresponding ratios of expected times to completion, $\mathbb{E}[n_{votes}(p_1, \delta_1)]/\mathbb{E}[n_{votes}(p_2, \delta_2)]$. The plots display the results for selected values of the utility parameter: $\lambda = 1, \lambda = 2$, and $\lambda = 10$.

D ISO-PAYMENT ACCURACY CURVES

Let us further examine the equation 12. Figure 17 below displays the relationship between the odds φ of classifying an item correctly, and effort ψ expended per item. While the left panel may at first suggest that the relationship for all iso-curves is exponential in character, the change of scale on the right panel of Figure 17 reveals the existence of two function branches adjoined at $\psi = 0$ for each iso-curve.

Note that the setting relevant for crowdsourcing is the one where $p > 0.5$, and hence $\varphi > 1$. In this region, iso-payment curves are well-defined, and their shape is asymptotically exponential.

Let us further note the close match between the first iso-payment plot and our chosen non-iso accuracy curve on the left panel of Figure 17. It is easy to attest to both visually and analytically that the iso-payment accuracy curve with $c = 1$ converges to our initially chosen exponential curve: $\lim_{\varphi \to \infty} \ln(1 + \varphi) - \ln \varphi \frac{\varphi - 1}{\varphi + 1} = 0$. This means that as her accuracy pushes closer to being perfect, a worker whose accuracy and effort are connected as $p = e^{\psi} - 1$ tends to become indifferent to time spent per unit under this payment scheme.

REFERENCES

Lina Abassi and Imen Boukhris. 2017. An Adaptive Approach of Label Aggregation Using a Belief Function Framework. In *International Conference on Digital Economy*. Springer, 198–207.

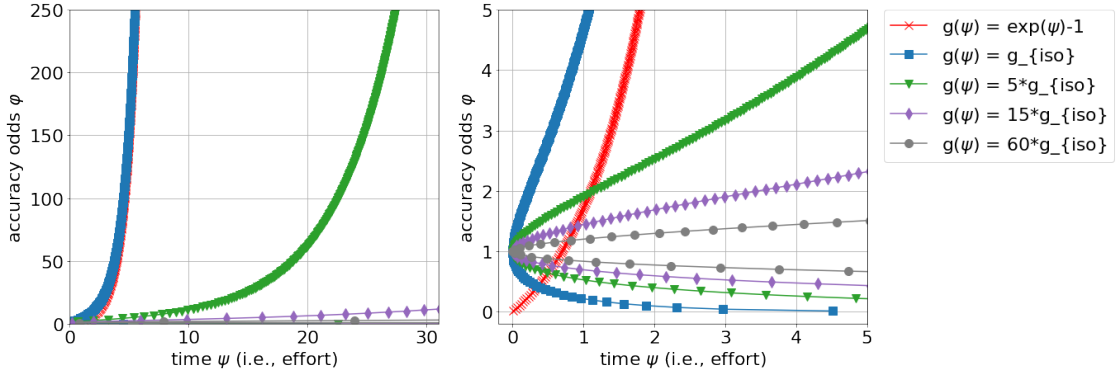


Figure 17: A close-up plot of several iso-payment accuracy curves, revealing multiple branches jointly defining $p = g(\psi)$, where $p = \frac{\varphi}{\varphi+1}$ by definition of φ.

Noga Alon, Graham Brightwell, Hal A Kierstead, Alexandr V Kostochka, and Peter Winkler. 2006. Dominating sets in k-majority tournaments. *Journal of Combinatorial Theory, Series B* 96, 3 (2006), 374–387.

Omar Alonso and Ricardo Baeza-Yates. 2011. Design and implementation of relevance assessments using crowdsourcing. In *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings 33*. Springer, 153–164.

Jiří Anděl and Šárka Hudecová. 2012. Variance of the game duration in the gambler's ruin problem. *Statistics & Probability Letters* 82, 9 (2012), 1750–1754.

Evan Archer, Il Memming Park, and Jonathan W. Pillow. 2014. Bayesian Entropy Estimation for Countable Discrete Distributions. *Journal of Machine Learning Research* 15, 81 (2014), 2833–2868. http://jmlr.org/papers/v15/archer14a.html

Peter Ayton and Ilan Fischer. 2004. The hot hand fallacy and the gambler's fallacy: Two faces of subjective randomness? *Memory & cognition* 32 (2004), 1369–1378.

Daniel W Barowy, Charlie Curtsinger, Emery D Berger, and Andrew McGregor. 2012. Automan: A platform for integrating human-based and digital computation. In *Proceedings of the ACM international conference on Object oriented programming systems languages and applications*. 639–654.

Daniel Berend and Aryeh Kontorovich. 2014. Consistency of weighted majority votes. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*. 3446–3454.

Thomas Bonald and Richard Combes. 2016. A minimax optimal algorithm for crowdsourcing. *arXiv preprint arXiv:1606.00226* (2016).

Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. 2013. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1999–2008.

Peng Dai, Christopher H Lin, Daniel S Weld, et al. 2013. POMDP-based control of workflows for crowdsourcing. *Artificial Intelligence* 202 (2013), 52–85.

Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi. 2013. Aggregating crowdsourced binary ratings. In *Proceedings of the 22nd international conference on World Wide Web*. 285–294.

Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 1–40.

Patrick De Boer. 2017. *Crowd process design: how to coordinate crowds to solve complex problems*. Ph. D. Dissertation. University of Zurich.

Patrick M De Boer and Abraham Bernstein. 2017. Efficiently identifying a well-performing crowd process for a given problem. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 1688–1699.

Franz Dietrich and Christian List. 2007. Judgment aggregation by quota rules: Majority voting generalized. *Journal of Theoretical Politics* 19, 4 (2007), 391–424.

William Feller. 1968. *An Introduction to Probability Theory and Its Applications, Vol. 1* (3rd edition ed.). Wiley.

Peter C Fishburn. 2015. *The theory of social choice*. Princeton University Press.

José Luis García-Lapresta and Bonifacio Llamazares. 2001. Majority decisions based on difference of votes. *Journal of Mathematical Economics* 35, 3 (2001), 463–481.

Sergiu Goschin. 2014. *Stochastic dilemmas: foundations and applications*. Rutgers The State University of New Jersey-New Brunswick.

C Grinstead and L Snell. 1997. Ch. 11: Markov Chains. *Introduction to Probability. American Mathematical Society* (1997).

Derek L Hansen, Patrick J Schone, Douglas Corey, Matthew Reid, and Jake Gehring. 2013. Quality control mechanisms for crowdsourcing: peer review, arbitration, & expertise at familysearch indexing. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 649–660.

Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 203–212.

Tobias Hossfeld, Christian Keimel, and Christian Timmerer. 2014. Crowdsourcing quality-of-experience assessments. *Computer* 47, 9 (2014), 98–102.

Hyun Joon Jung and Matthew Lease. 2011. Improving consensus accuracy via z-score and weighted voting. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

David R Karger, Sewoong Oh, and Devavrat Shah. 2014. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research* 62, 1 (2014), 1–24.

Gabriella Kazai. 2011. In search of quality in crowdsourcing for search engine evaluation. In *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings 33*. Springer, 165–176.

Gabriella Kazai, Jaap Kamps, Marijn Koolen, and Natasa Milic-Frayling. 2011. Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 205–214.

Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. 2013. An analysis of human factors and label accuracy in crowdsourcing relevance judgments. *Information retrieval* 16 (2013), 138–178.

Ashish Khetan and Sewoong Oh. 2016. Achieving budget-optimality with adaptive schemes in crowdsourcing. *Advances in Neural Information Processing Systems* 29 (2016), 4844–4852.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision* 123 (2017), 32–73.

Ranjay A Krishna, Kenji Hata, Stephanie Chen, Joshua Kravitz, David A Shamma, Li Fei-Fei, and Michael S Bernstein. 2016. Embracing error to enable rapid crowdsourcing. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 3167–3179.

Pavel Kucherbaev, Florian Daniel, Stefano Tranquillini, and Maurizio Marchese. 2016. Relauncher: crowdsourcing micro-tasks runtime controller. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. 1609–1614.

Annick Laruelle and Federico Valenciano. 2011. Majorities with a quorum. *Journal of Theoretical Politics* 23, 2 (2011), 241–259.

Guy Latouche and Vaidyanathan Ramaswami. 1999. *Introduction to matrix analytic methods in stochastic modeling*. SIAM.

Paolo Laureti, Lionel Moret, Y-C Zhang, and Y-K Yu. 2006. Information filtering via iterative refinement. *EPL (Europhysics Letters)* 75, 6 (2006), 1006.

Christopher H Lin, M Mausam, and Daniel S Weld. 2016. Re-active learning: Active learning with relabeling. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Benjamin Livshits and Todd Mytkowicz. 2014. Saving money while polling with interpoll using power analysis. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Bonifacio Llamazares. 2006. The forgotten decision rules: Majority rules based on difference of votes. *Mathematical Social Sciences* 51, 3 (2006), 311–326.

SM Losanitsch. 1897. Die Isomerie-Arten bei den Homologen der Paraffin-Reihe. *Berichte der deutschen chemischen Gesellschaft* 30, 2 (1897), 1917–1926.

Edoardo Manino, Long Tran-Thanh, and Nicholas Jennings. 2018. On the efficiency of data collection for crowdsourced classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*.

Winter Mason and Duncan J Watts. 2009. Financial incentives and the" performance of crowds". In *Proceedings of the ACM SIGKDD workshop on human computation*. 77–85.

Jonathan M Mortensen, Mark A Musen, and Natalya F Noy. 2013. Crowdsourcing the verification of relationships in biomedical ontologies. In *AMIA Annual symposium proceedings*, Vol. 2013. American Medical Informatics Association, 1020.

Marcel F Neuts. 1994. *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Courier Corporation.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet: Large scale visual recognition challenge. *International journal of computer vision* 115 (2015), 211–252.

Abraham M Rutchick, Bryan J Ross, Dustin P Calvillo, and Catherine C Mesick. 2020. Does the "surprisingly popular" method yield accurate crowdsourced predictions? *Cognitive research: principles and implications* 5, 1 (2020), 1–10.

Donald G Saari. 1990. Consistency of decision processes. *Annals of Operations Research* 23, 1 (1990), 103–137.

Alexander Scheidler, Arne Brutschy, Eliseo Ferrante, and Marco Dorigo. 2015. The {k}-Unanimity Rule for Self-Organized Decision-Making in Swarms of Robots. *IEEE Transactions on Cybernetics* 46, 5 (2015), 1175–1188.

Edwin Simpson and Stephen Roberts. 2015. Bayesian methods for intelligent task assignment in crowdsourcing systems. In *Decision Making: Uncertainty, Imperfection, Deliberation and Scalability*. Springer, 1–32.

Yaron Singer and Manas Mittal. 2013. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd international conference on World Wide Web*. 1157–1166.

Neil JA Sloane. 2019. *The on-line encyclopedia of integer sequences*. Princeton University Press. http://oeis.org/A002620

Dapeng Tao, Jun Cheng, Zhengtao Yu, Kun Yue, and Lizhen Wang. 2018. Domain-weighted majority voting for crowdsourcing. *IEEE transactions on neural networks and learning systems* 30, 1 (2018), 163–174.

Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. 2010. The Multidimensional Wisdom of Crowds. In *NIPS*.

Ming Yin, Yiling Chen, and Yu-An Sun. 2014. Monetary interventions in crowdsourcing task switching. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Li F Fei-Fei, and Michael Bernstein. 2019. Hype: A benchmark for human eye perceptual evaluation of generative models. *Advances in neural information processing systems* 32 (2019).