Fair Payments in Adaptive Voting

Margarita Boyarskaya, Panos Ipeirotis
New York University Leonard N. Stern School of Business
{mb6599@, panos@stern.}nyu.edu

Abstract

Adaptive voting is a commonly used scheme for aggregating consensus in crowdsourced binary labeling tasks. Workers assign labels to an item until the votes for one class outnumber the votes for the other class by a given integer threshold. Modeling the process as a Markov random walk, we offer results on how workers with different accuracy levels should be paid comparatively to each other under an adaptive voting scheme. We calculate the expected accuracy of the final consensus vote and estimate the number of votes necessary for the process to finish. We show how to derive a *fair* payment policy for two groups of workers with different accuracy rates in a way that guarantees that the task is associated with the same cost and generates results of the same quality when assigned to either group. This paper also compares the adaptive voting scheme with majority voting, demonstrating evidence for a strict dominance of the former. Our model is simple yet flexible and provides the foundation for analysis of more complex settings.

1 Introduction

Binary labeling tasks are widely used in crowdsourcing. A commonly used scheme is *adaptive voting*, where workers assign labels to an item until the votes for one class outnumber the votes for the other class by a threshold δ. We call a voting scheme *adaptive* to indicate that the number of workers is not determined in advance, but depends on the distribution of votes as they arrive sequentially.[1]

We present a model that examines how workers of different accuracy levels should be paid, comparatively to each other, under such a voting scheme. There is no lack of empirical results for such questions (Chilton et al. 2013; Dai et al. 2013; Mortensen, Musen, and Noy 2013). However, the simple, stylized theoretical model proposed herein can guide design decisions. In our analysis, the voting process is modeled as a Markov random walk, and we show:

- How to calculate the expected accuracy of the final answer in a model with an adaptive voting scheme.
- How to estimate the number of votes necessary for the process to finish.
- How to derive a *fair* payment for two groups of workers with different accuracy rates in a way that guarantees that the task is associated with the same cost and generates results of the same quality when assigned to either group.

Our model is simple yet flexible and provides the foundation for analysis of more complex settings.

2 The Model

Consider the problem of assigning binary labels to items on a crowdsourcing platform. For every item, we solicit votes from workers who assign the correct label with probability p and the incorrect label with probability $1 - p$. We keep asking workers to assign labels to an item until the absolute difference between the numbers of correct and incorrect votes exceeds a pre-defined *consensus threshold* δ.

Example. We set $\delta = 2$, so the process will stop when the tuples $\langle n_1, n_0\rangle$ composed of the counts of correct and incorrect votes reach one of the following stages: $\langle 2, 0\rangle$, $\langle 0, 2\rangle$, $\langle 3, 1\rangle$, $\langle 1, 3\rangle$, $\langle 4, 2\rangle$, $\langle 2, 4\rangle$, and so on. A consensus vote obtained in one of the states $\{\langle 2, 0\rangle, \langle 3, 1\rangle, \langle 4, 2\rangle, ...\}$ is correct, while a consensus vote obtained in one of the states $\{\langle 0, 2\rangle, \langle 1, 3\rangle, \langle 2, 4\rangle, ...\}$ is incorrect.

Since the only desideratum for consensus is the difference between the counts of two types of votes, the process can be modeled as a Markov random walk. Define the current state as the difference between the numbers of correct and incorrect votes. If the difference is δ or $-\delta$, the process terminates with a correct or an incorrect consensus label, respectively. In all other states, we procure an additional vote, which will be correct with probability p, moving the process from state i to state $i + 1$. This process has a state diagram illustrated in Figure 1 and is known as *Gambler's Ruin* model[2]. Be-

[1] This is in contrast to non-adaptive schemes, e.g. "majority out of 5" where we always select five votes and the majority label ends up being the consensus. The broader family of adaptive voting schemes contains more sophisticated workflows, where at each iteration (i.e. with each added vote) some of the workflow parameters are updated (e.g., (Khetan and Oh 2016)).

[2] The *Gambler's Ruin* model is a common introductory example for random walks. The models is describing the probability of a gambler's winning a certain amount in the casino, vs. the probability of losing everything. See (Feller 1968, page 344) for details.

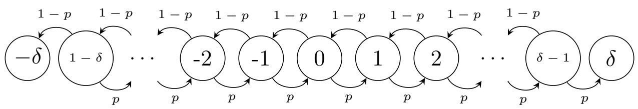

Figure 1: A graph of state transitions in the process of voting for a single item. Node labels mark the difference between the numbers of correct and incorrect votes, $n_1 - n_0$ in a corresponding state. A correct or an incorrect consensus vote is reached in absorbing states δ and $-\delta$, respectively.

low we provide several analytical results pertaining to this model. We omit proofs for space considerations.

2.1 Quality of Consensus Vote

The first quantity of interest is the quality of the final consensus votes. The process always starts at state 0 (no votes). Let us denote as C a random variable indicator of whether the consensus vote (in the event that it was reached) corresponds to the ground truth correct label of an item. We want to calculate the probability of reaching terminal state δ (voters agree, ground truth correct label is assigned, i.e. $C = 1$) vs. reaching terminal state $-\delta$ (voters agree, ground truth incorrect label is assigned, i.e. $C = 0$). In all other states, the voting continues. Independently proving the results in (Feller 1968, page 344), we have the following:

Theorem 1. The probability $Q(\phi, \delta)$ that the consensus vote c is correct, for an item classified using an adaptive voting scheme with worker accuracy p and a consensus threshold δ, is:

$$Q(\phi, \delta) = P(C = 1|\phi, \delta) = \frac{\phi^\delta}{1 + \phi^\delta}\ , \tag{1}$$

where $\phi = \frac{p}{1-p}$ are the *odds* of a single worker classifying the item correctly.

For an example of the theorem's application, consider a pool of workers with accuracy $p = 0.75$ (i.e., $\phi = 3$). If we set $\delta = 2$, the expected quality of the overall classification will be $Q(3, 2) = 0.9$.

2.2 Quality Equivalence by Varying Threshold Parameter δ

The result from Section 2.1 can be used to answer the following question. Suppose that we have two sets of workers: one with accuracy p_1 and another with accuracy p_2. The first set of workers operate an adaptive voting scheme with threshold parameter δ_1. How must one choose the value of δ_2, so that the set of workers with accuracy p_2 would generate the same quality of the results as the workers in the first scheme? By setting $Q(\phi_1, \delta_1) = Q(\phi_2, \delta_2)$ and solving for δ_2, we get the following result:

Corollary 1. If an item is classified by workers with accuracy p_1 and threshold δ_1, we can achieve the same quality of results by a set of workers with accuracy p_2 if we set the threshold δ_2 to be:

$$\delta_2 = \delta_1 \cdot \frac{\ln \phi_1}{\ln \phi_2}, \tag{2}$$

where $\phi_1 = \frac{p_1}{1-p_1}$ and $\phi_2 = \frac{p_2}{1-p_2}$ are the *odds* of a single worker classifying the item correctly.

2.3 Expected Number of Votes

The next question is whether the process is guaranteed to terminate, and how many votes we expect to collect until reaching a consensus. We can estimate the number of votes it takes to reach state δ or state $-\delta$ in terms of transitions in the Markov chain as follows. The number of remaining steps E_0 from state 0 is one step it takes to reach the next state (i.e., state 1 with probability p or state -1 with probability $1 - p$) plus the number of remaining steps from this resulting node. By solving, we get:

Theorem 2. The expected number of votes N_{votes} it takes to reach a (correct or incorrect) consensus when classifying an item using an adaptive voting scheme with worker accuracy p and consensus threshold δ is:

$$E(N_{votes}|\phi, \delta) = \delta \cdot \frac{\phi + 1}{\phi - 1} \cdot \frac{\phi^\delta - 1}{\phi^\delta + 1}, \tag{3}$$

where $\phi = \frac{p}{1-p}$ are the *odds* of a single worker classifying the item correctly.

2.4 Worker Pay Equivalence

Assume that workers with accuracy ϕ are paid *pay*(ϕ) per vote. In this case, the expected cost of classifying an item is:

$$\mathit{Cost}(\phi, \delta) = \mathit{pay}(\phi) \cdot E(\mathit{votes}|\phi, \delta) \tag{4}$$

We would like to pay teams of workers with different accuracies in a way that is fair: as long as the teams can generate results of equal quality, they should be paid the same total amount. From Section 2.2, we can increase δ to compensate for a lower worker accuracy. However, as shown in Section 2.3, higher consensus thresholds also increase the expected number of votes required to reach consensus. We set $\mathit{Cost}(\phi_1, \delta_1) = \mathit{Cost}(\phi_2, \delta_2)$ and $\delta_2 = \delta_1 \cdot \frac{\ln \phi_1}{\ln \phi_2}$ to ensure equal quality. We thus get:

Theorem If workers with accuracy ϕ_1 are paid *pay*(ϕ_1) per vote, then workers with accuracy ϕ_2 will generate results of the same quality and cost if the ratio of the payments is:

$$\frac{\mathit{pay}(\phi_1)}{\mathit{pay}(\phi_2)} = \frac{\ln \phi_1}{\ln \phi_2} \cdot \frac{\phi_2 + 1}{\phi_1 + 1} \cdot \frac{\phi_1 - 1}{\phi_2 - 1}. \tag{5}$$

3 Comparison with Majority Voting

In order to gain insight into the performance of the adaptive voting scheme, we compare it with majority voting. We ran simulation studies, examining the expected quality of majority vote, as well as the estimated number of required workers for a majority voting scheme supplemented by an *early stopping* mechanism that is used when vote dominance becomes 'irreversible'. The examination of the ratio of expected cost (numbers of workers required for a consensus) between an adaptive voting scheme and a simple majority voting scheme suggest strict dominance of the adaptive voting scheme compared with majority voting.

References

Chilton, L. B.; Little, G.; Edge, D.; Weld, D. S.; and Landay, J. A. 2013. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1999–2008. ACM.

Dai, P.; Lin, C. H.; Weld, D. S.; et al. 2013. Pomdp-based control of workflows for crowdsourcing. *Artificial Intelligence* 202:52–85.

Feller, W. 1968. *An Introduction to Probability Theory and Its Applications, Vol. 1*. 3rd edition edition.

Khetan, A., and Oh, S. 2016. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *Advances in Neural Information Processing Systems*, 4844–4852.

Mortensen, J. M.; Musen, M. A.; and Noy, N. F. 2013. Crowdsourcing the verification of relationships in biomedical ontologies. In *AMIA Annual symposium proceedings*, volume 2013, 1020. American Medical Informatics Association.