ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1704

Adaptive Condorcet-Based Stopping Rules Can Be Efficient

Omer Reingold[1] and **Nina Narodytska**[2]

Abstract. A crowdsourcing project is usually comprised of many unit tasks known as Human Intelligence Tasks (HITs). As answers to each HIT varies between workers, each HIT is often contracted to more than one worker to obtain a reliable and consistent enough answer. When implementing a project, an important design decision is how to formulate HITs and how to aggregate workers' answers. These decisions have strong impact on the quality of results and cost of elicitation process. One way to design an efficient elicitation procedure is to use *adaptive stopping rules*, which allows terminating the elicitation process as soon as a high quality result is guaranteed.

Adaptively deciding how many times to issue a HIT is mostly well understood for the case of binary-answer HITs, thanks to the work of Abraham *et al.* [3, 2, 1]. In this line of work the authors focused on plurality-based stopping rules and provided their theoretical analysis. As a decision rule (when many alternatives are offered), it is well known that plurality may be inferior to other rules, such as the Condorcet method. We argue that for large number of possible answers, plurality-based stopping rules may also be terribly inefficient. In other words, one may need to elicit answers from many workers (at least linear in the number of answers) in order to get any reasonable approximation of the plurality answer. Somewhat surprisingly, we show that Condorset-based stopping rules may be much more efficient (with the number of workers to find the approximate Condorset winner depending only logarithmically on the number of answers). Moreover, in an important case of restricted domains, namely single-peaked domains, we show that the stopping time to find an approximate Condorcet-based winner does not depend on the number of answers at all. Overall, our results suggest that both crowdsourcing platform developers and HIT designers, should consider Condorcet-based adaptive stopping rules as a useful tool in their toolboxes.

Introduction Crowdsourcing is a simple and efficient way to gather information from a large population of people. A typical crowdsourcing platform is a marketplace for two types of users: requesters and workers. The central notion at the marketplace is a human intelligence task (HIT). HIT is a unit task that can be allocated to a worker. The requester formulates their question as a HIT and assigns a monetary reward for *answering* this question. Each time a worker answers a HIT, they receive the reward and the requester obtains their answer.

Some of the most popular crowdsourcing tasks are data classification and data ranking [9, 14], e.g., labeling an image as child-appropriate or adult-only, tagging tweets with relevant sentiments [7], ranking search results based on relevance, etc. Such tasks are used, in particular, to provide input to analytics algorithms, including machine learning, natural language processing and information retrieval algorithms. For example, image labeling is an important preprocessing step for training deep neural networks, which typically requires a training set of 1-2 million labeled images. Obtaining such a set is a labor demanding task. Thanks to crowdsourcing platforms, like Amazon Mechanical Turk or CrowdFlower, large volumes of data can be processed efficiently and with reasonable budget.

Being a gateway to a large population of workers, the crowdsourcing framework is unable to evaluate the qualification of workers or to monitor the quality of their answers. Therefore, ensuring high-quality answers is one of the first problems that any requester faces in dealing with such frameworks. One natural way to improve the quality of results is to obtain answers from many different workers. Implementing such redundancy in practice can be costly as the requester does not know in advance how many workers to assign to the same task. In this context, Abraham *et al.* [3, 2, 1] studied *adaptive* stopping rules that at each stage, given previous workers' answers, decide whether to stop or to ask for an additional answer. Stopping as early as possible saves costs and is therefore highly motivated. They focus on a single HIT stopping rule, where each HIT is a binary question and they perform a thorough theoretical investigation of the proposed stopping rule.

In this work we consider non-binary or multi-answer HITs. Non-binary tasks naturally occur in data classification and data ranking problems. For example, a tweet tagging task requires the worker to choose the most relevant emotional labels from a list of labels. Another example is to choose the most appropriate label for a picture. Next we overview our contributions.

Approximate-Condorcet winner. The plurality-based stopping rule proposed by Abraham *et al.* [3] has several disadvantages that can be avoided when additional information about the structure of the preferences is available. For example, if the probability of the plurality answer is small then the rule is inefficient, even if the plurality answer is twice as likely as any other answer. To overcome these issues, we propose to natural extension of the plurality-based stopping rule to a Condorcet-based stopping rule. We use the notion of approximate-Condorcet winner which is defined as an alternative that wins or effectively ties with any other alternative. We say that an alternative x effectively ties with another alternative x' if the probability that a random worker prefers x to x' is at least $\frac{1}{2} - \mu$, where μ is a parameter. Intuitively, if the probability that one alternative is preferred to another is within a μ circle of $\frac{1}{2}$, we decide that these alternatives are equally preferred by the workers. We introduce this notion with dual motivation. First, it allows us to determine a socially acceptable answer even in cases where an exact Condorcet winner does not exist. Second, it decreases the expected stopping time significantly. To be able to find an approximate Condorcet winner and perform

[1] Samsung Research America, email: o.reingold@samsung.com
[2] Samsung Research America, email: nina.n@samsung.com

the search efficiently, we make two technical extensions of existing stopping rules.

Extensions of stopping rules. Given a k-answer HIT, the Condorcet winner determination procedure finds $O(k^2)$ outcomes of pairwise comparisons between alternatives. Each of these comparisons can be seen as an independent competition between two corresponding alternatives. Hence, we propose an extension of plurality-based stopping rule [3] that can deal with parallel processes. We show that the expected stopping time depends only logarithmically on the number of parallel processes. The second extension helps us to deal with cases when several alternatives are very close to be Condorcet winners in workers' preferences. These alternatives might effectively tie in several pairwise competitions. Applying the existing stopping rule [3] to establish the winner in these competitions is costly. In this case, we augment our stopping rule with the ability to output the 'too close to call' answer. Such enchantment achieves a significant improvement in the expected stopping time and allows us to find an approximate Condorcet winner efficiently.

Single-peaked preferences. In the general case, when HIT's answers are unrelated, we cannot reduce the elicitation cost per worker. However, for multi-answer HITs where possible answers are related we can reduce the elicitation cost by requiring workers to provide only their top choice, as in the case of plurality-based stopping rules. In particular, we show how to reduce the stopping time needed to identify approximate Condorcet winners in single-peaked domains to be *completely independent* of the number of answers and only depend on the approximation parameter.

Designing individual HITs within a larger project involves many considerations, e.g. the attention span of a typical worker, budget constraints, workers' qualification. We suggest that the cost of reaching a high-quality answer is another important consideration, and we offer our quantitative analysis as a design tool for task requesters to take into account. In particular, our results reveal that expected stopping time to find (an approximate) Condorcet winner is not very expensive and, in case of single-peaked domains, no more expensive than finding the plurality winner (for binary HITs). In other words, our results show that, especially in case of single-peaked domains, the requester can find a Condorcet winner without the significant budget overhead.

Related work The related work can be divided into two main categories. The first category is on adaptive stopping rules by Abraham *et al.* [3, 2, 1]. The main goal of this line of work is to reduce the total cost of crowdsourcing procedure by using adaptive stopping rules. In particular, the proposed algorithm decides whether to stop or to ask one more worker at each round. The crowdsourcing platform stops if it can provide a high-quality answer with high probability. The main contribution of our work in to extend these plurality-based rules to Condorcet-based rules efficiently.

The second related work category consists of finding sampling bounds and aggregation procedures in computational social choice [6, 10, 4, 12, 11]. The main difference from our work from the work is that we focus on adaptive stopping rules for the elicitation procedure. Hence, the main advantage of our work that we can stop dynamically, often before reaching the requesters static bound on the number of HIT assignments. For example, adaptive stopping rules are particularly useful if the crowd significantly prefers x over other alternatives and can efficiently produce the correct answer with a high probability. Moreover, in most of the work, a different access model to users' preferences is considered, based on comparison queries (the elicitor can ask an agent which of two alternatives they prefer), while we assume that voters can provide their top choices. Hence, we will briefly overview related work in this line of research. Kruger *et al.* [10] considered several aggregation procedures, including the plurality rule and the bias-correcting method. The authors provided axiomatic analysis of these aggregation methods and conducted an experimental study on Amazon Mechanical Turk. Procaccia *et al.* [13] present a new aggregation method that assumes that workers' answers are noisy. Lee *et al.* [12], introduced the notion of ϵ-approximate winners which is closely related to our approximation notion. They show how to elicit these approximations in time $O(klog(k))$, where k is the number of alternatives. These methods can return an approximate winner for ϵ-Borda and ϵ-Condorcet with high probability. Our notion of approximate Condorcet winner is different from ϵ-Condorcet [12] as we require that an alternative to win or tie in all pairwise competitions rather than a fraction of them. Later these results were extended to a larger class of voting rules [11]. Conitzer [5] focuses on elicitation of voters preferences in single-peaked domains. He showed preferences can be elicited using only a linear number of comparison queries if the underlying order with respect to which preferences are single-peaked is known. Goel and Lee [8] extended results of Conitzer [5] to the case when the single-peaked axis is unknown assuming that candidates and participants coincide.

REFERENCES

[1] Ittai Abraham, Omar Alonso, Vasilis Kandylas, Rajesh Patel, Steven Shelford, and Aleksandrs Slivkins, 'How Many Workers to Ask? Adaptive Exploration for Collecting High Quality Labels', Technical report, http://arxiv.org/abs/1411.0149, (2014).

[2] Ittai Abraham, Omar Alonso, Vasilis Kandylas, Rajesh Patel, Steven Shelford, and Alex Slivkins, 'Using Worker Quality Scores to Improve Stopping Rules', in *Second AAAI Conference on Human Computation and Crowdsourcing*, (2014).

[3] Ittai Abraham, Omar Alonso, Vasilis Kandylas, and Aleksandrs Slivkins, 'Adaptive Crowdsourcing Algorithms for the Bandit Survey Problem', in *Conference on Learning Theory*, pp. 882–910, (2013).

[4] Ioannis Caragiannis, Ariel D Procaccia, and Nisarg Shah, 'When do noisy votes reveal the truth?', in *Proceedings of the fourteenth ACM conference on Electronic commerce*, pp. 143–160. ACM, (2013).

[5] Vincent Conitzer, 'Eliciting single-peaked preferences using comparison queries', *JAIR*, **35**(1), 161–191, (2009).

[6] Ulle Endriss and Raquel Fernández, 'Collective annotation of linguistic resources: Basic principles and a formal model', in *BNAIC 2013: Proceedings of the 25th Benelux Conference on Artificial Intelligence, Delft, The Netherlands, November 7-8, 2013*, (2013).

[7] Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze, 'Annotating named entities in Twitter data with crowdsourcing', 80–88, (June 2010).

[8] Ashish Goel and David Lee, 'Triadic Consensus', in *Internet and Network Economics*, 434–447, Springer, (2012).

[9] Jonathan Kraus, 'Lessons Learned from Large Scale Crowdsourced Data Collection for ILSVRC', Technical report, (2015).

[10] Justin Kruger, Ulle Endriss, Raquel Fernández, and Ciyang Qing, 'Axiomatic analysis of aggregation methods for collective annotation', in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1185–1192, (2014).

[11] David Timothy Lee, 'Efficient, Private, and eps-Strategyproof Elicitation of Tournament Voting Rules', in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, (2015).

[12] David Timothy Lee, Ashish Goel, Tanja Aitamurto, and Helene Landemore, 'Crowdsourcing for participatory democracies: Efficient elicitation of social choice functions', in *Second AAAI Conference on Human Computation and Crowdsourcing*, (2014).

[13] Ariel D Procaccia, Nisarg Shah, and Yair Zick, 'Voting rules as error-correcting codes', *Artificial Intelligence*, **231**, 1–16, (2016).

[14] Luis von Ahn and Laura Dabbish, 'Labeling images with a computer game', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems(CHI'04)*, pp. 319–326. ACM Press, (2004).