Teaching Crowdsourcing: An Experience Report

Hui Guo, Nirav Ajmeri, Munindar P. Singh

Abstract

Crowdsourcing is the process of accomplishing a task by using a typically open call to invite members of the public (the "crowd") to work on one's task. The authors describe a project assignment in which students received the opportunity of practicing crowdsourcing to accomplish a hummed song recognition task, yielding improved comprehension of the concept and high student satisfaction.

Keywords— Crowdsourcing, social computing, computer science education, human computation, song recognition

Introduction

Crowdsourcing is about marshalling human knowledge and intelligence to solve tasks that are natural for humans but cannot be effectively performed by a computer. In addition to the idea of using humans, a distinguishing feature of crowdsourcing is to distribute the work to multiple humans, potentially including those with no specific credentials other than common sense, and usually from outside of one's organization. The term "crowd" is applied to the human workers to indicate that they may be nonspecialists and are selected from the public, although in practice the workers are chosen from eligible pools and may possess special knowledge or credentials. Leading applications of crowdsourcing include data gathering and analysis, crowdfunding, and idea generation.

Although the idea of outsourcing work to the public dates back centuries, it was not until recently that distributing computations through microtasks became important in computing practice [6]. This paper is about the latter, narrower sense of crowdsourcing as a computing paradigm.

This paper describes an approach to teaching crowdsourcing with a practical orientation via an assignment we incorporated in our social computing course, which is offered to a mixture of graduate and undergraduate computer science students. Our course provides an introduction to the rich variety of social computing applications, and identifies the concepts for their modeling and realization. Crowdsourcing is a key topic in social computing. Our instructional objectives were to familiarize students with the basic principles and logistics of crowdsourcing projects and introduce important elements of human computation, e.g., ways to motivate participation, such as deploying dynamic, adaptive, and personalized rewards (including incentives in terms of money and recognition) [12]. The assignment we adopted was aimed to provide students hands-on experience on the technology, with respect to designing, deploying and analyzing their own crowdsourcing projects and responses, as well as participating as a crowd worker for other projects. Deeper aspects of crowdsourcing, such as the differences between quality evaluation strategies, incentive mechanisms, and task setup, were introduced, but not within the scope of our learning goals. Our survey of the students showed a significant increase in their understanding of the concepts and workflow of crowdsourcing projects after our lectures and assignment.

Crowdsourcing has been taught in colleges. Instructors usually introduce crowdsourcing by adopting the requester role and using the students as a crowd. Davidson [5] introduced crowdsourcing to students by outsourcing part of the grading and teaching to students and received positive results. Proper peer grading, which we incorporate in our approach, has been proven to be beneficial to instructors and students [11]. However, we think it is imperative that the students practice this technique as requesters via commercial platforms, such as Amazon Mechanical Turk, so that they are better prepared for when they need to employ crowdsourcing techniques. One of our students reported that his internship involved crowdsourcing and a proper exercise in class would be valuable for preparing for his work. Bigham et al. [4] argue for the importance of students participating in crowdsourcing projects both as requesters and crowd workers. They organize multiple projects throughout their Crowd Programming course where students experience aspects of crowdsourcing separately. Compared to their course, the time and budget dedicated to crowdsourcing in ours are much more limited.

Organizing a project assignment on crowdsourcing can be challenging. Our previous course assignments were primitively designed in that students were not inherently encouraged to take the task seriously and there was not a natural way for managers to evaluate the quality of work. Therefore, result aggregation phases did not necessarily achieve high quality and student did not report high satisfaction. In addition, we did not adopt a popular crowdsourcing platform. The challenges lie in the choice of suitable tasks and logistics during the process of the assignment. Students need to accomplish a computational task by using a crowd, which means the task should not be easily solvable by automated tools or students' own expertise. Also, crowdsourcing should be able to produce reliable and satisfying results for all or most students, which imposes requirements on both the quality of the specific task and the targeted crowd, especially with no funds being allocated. We propose that students use their classmates as the crowd, which requires the instructors to monitor the progress of the assignment closely, and resolve any logistics problems promptly. It can be nontrivial to make sure all students have a smooth and enjoyable experience when they have to act as employers, workers, and analysts.

As part of our assignment, students exercised the workflow of a typical crowdsourcing project on a real-world platform. Doing so involved setting up a task; obtaining information from the crowd; rating and rewarding workers; analyzing responses; and computing an answer. They met the challenges arising in each of the above components. Most students reported that this assignment was interesting and educational, as well as valuable to their learning of crowdsourcing.

Task and Setting

Amazon Mechanical Turk (MTurk) is a popular crowdsourcing platform, widely used for data collection or as a subject recruitment tool for behavioral [7] and political sciences [1]. MTurk is therefore an appropriate platform for instruction. We adopted MTurk Sandbox, a closed variant of MTurk that provides the same interfaces for managing expenses, rating answers, giving rewards but without payments—and hence is easier logistically. We instructed students to serve both as workers for MTurk HITs (Human Intelligence Tasks) and as requesters using others as workers.

In order for students to experience the power of crowdsourcing, each student's assigned task should be relatively easy for a crowd of students, but hard for an individual one. Also, each microtask should be enjoyable, since each student may be tasked to finish a large number of microtasks. The task we adopted in this assignment—namely, the identification of music from snippets—is well suited to crowdsourcing [8]. More specifically, we chose hummed song recognition as the target task.

Hummed song recognition challenges current computational techniques. First, the absence of lyrics makes it difficult to search song names. Wang et al.'s [14] query by singing/humming (QBSH) system can misclassify humming clips as singing, leading to erroneous output. Second, the accuracy of song recognition depends on the comprehensiveness of the existing dataset. Music recognition algorithms, e.g., Shazam [13], identify snippets of existing recordings. Item-to-item similarity calculations that accommodate differences in timing and tempo scale poorly to datasets of millions of instances [2]. Third, since hummed recordings vary across users, extracting audio features that accurately represent music content remains challenging [10]. For example, Yang et al. [15] use absolute pitch values, which produce inaccurate results when people use dissimilar start tones. These challenges make crowdsourcing an appropriate approach for hummed song recognition. In our setting, hummed song recognition through crowdsourcing produced high accuracy for most students.

Structure of the Crowdsourcing Assignment

To lower the probability of one person being able to correctly recognize all songs, we generated a list of 1,000 songs, of which 200 are classical music pieces, such as Beethoven's *Für Elise*; 200 are old favorites (for holidays or for children, such as *Santa Claus Is Comin' To Town*); and 600 are pop music. We sampled 200 from the 1,000 songs, covering these classes with the same ratios. We recorded a humming of 10 to 20 seconds for each song, with the intention of producing recordings that would be easy to recognize. These recordings involved recognizable parts of songs such as the opening or chorus. A worker should be able to determine whether he or she recognizes the song instantly.

We published the 1,000 song names, each with a numeric ID, so that only IDs could be used subsequently, which avoids the challenges of textual input, such as variability in input. Each student was assigned links to five recordings, with the ID of one of them revealed, and was asked to identify the remaining four. Students would crowdsource tasks to their fellow classmates.

Students act as both requesters and workers. One individual student's success requires the student's devotion as well as the timely cooperation from his or her classmates. We broke the assignment into three parts with different deadlines to make sure that all students kept up with the schedule. As Figure 1 shows, the three parts followed Kamoun et al.'s [9] five-stage lifecycle for crowdsourcing, namely, initiation, preparation, engagement, evaluation, and commitment. We set three deadlines to motivate students' participation throughout the workflow and to discourage them from putting off work until the end.

As part of the initiation phase, before this assignment, students had learned in class about the merits, basic workflow, and difficulties of crowdsourcing. We had given four lectures on crowdsourcing and other topics related to human computation, in which we had offered students extensive information on the theoretical background of gathering information from and outsourcing computation to a crowd of people. For example, we discussed in length the merits of negative surveys against positive surveys, the concept and examples of vox populi, a social mobilization example, gamification of certain human computation tasks, and so on. Students were given a list of relevant papers and notes as reading materials. Additional information regarding our teaching can be found here: go.ncsu.edu/teaching-crowdsourcing. Upon the announcement of this assignment, we ex-

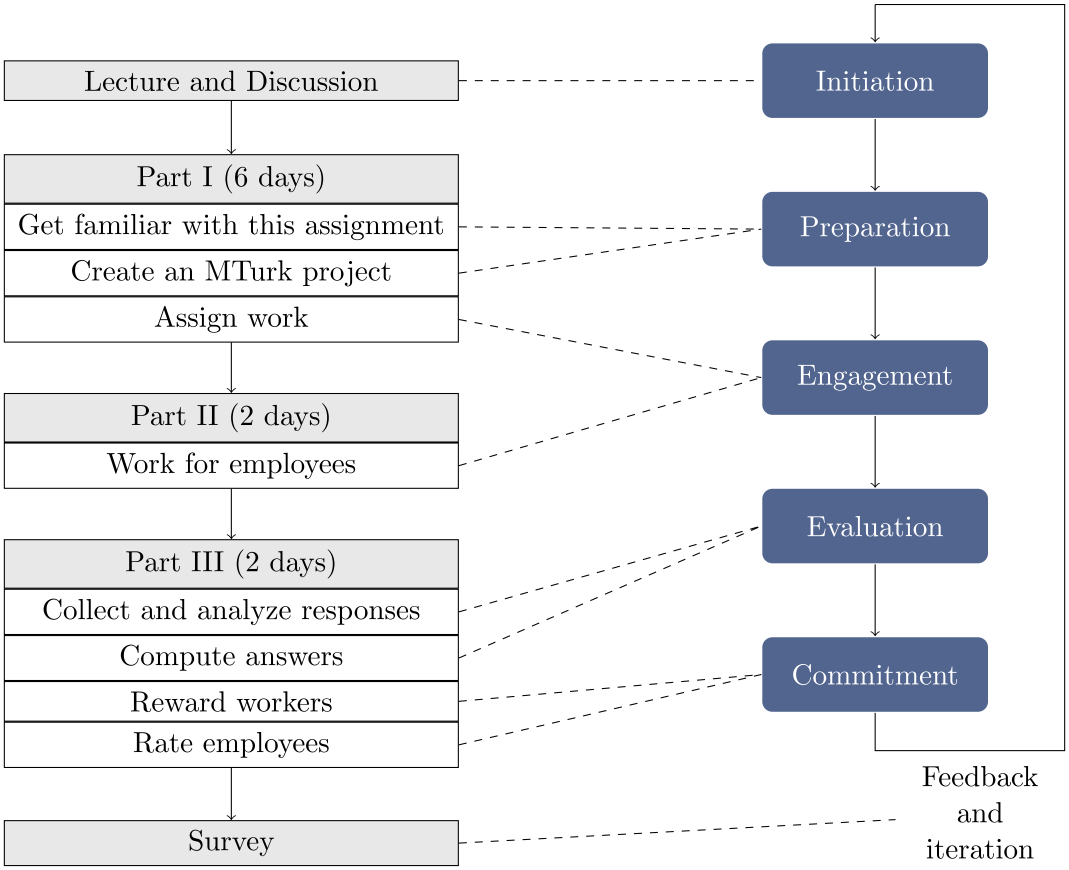

Figure 1: Crowdsourcing lifecycle [9] and assignment parts.

plained the rationale of using crowdsourcing for hummed song recognition. To enable coordination, we provided a Google Sheet available to all students.

Part I: Create a Project

Students became familiar with the assignment and understood activities and deliverables. Each student was required to complete the following in six days.

- Set up a survey using Google Form for workers, authenticated at the university, to give answers (IDs) for the hummed snippets.
- Set up a project on MTurk Sandbox, linking to the aforementioned Google Form, and advertise it on the Google Sheet with a link to his or her MTurk HIT to recruit workers. For uniformity, we limited each project to a budget of $10 in notional money.
- Sign up as a worker for other projects on the Google Sheet. Select at least 10 projects, taking projects that currently have fewer than 10 workers (as a way to ensure all projects receive some workers).

We encouraged the students to give thoughts to the qualification strategies. Students were free to design their own tasks; they could ask the questions in any way and include any additional questions. We also encouraged them to advertise their projects on the course's online message board, which was public to the whole class.

Part II: Work for Others

Since each student was required to work for at least 10 projects and each project contained five recordings, each student worked on at least 50 recordings. A student submitted responses, i.e., recognized song ID for each recording, on the Google Form, and completed the HIT on MTurk Sandbox for each project.

Part III: Solution and Closing

Students produced song IDs for four recordings based on the answers they received on their projects within two days:

- Close the MTurk project.
- Gather responses on their Google Forms.
- Evaluate the responses, optionally using the song with the known ID for qualifying answers.
- Give due rewards to workers, summing up to $10.
- Analyze the responses to determine song IDs for the four recordings.

Project reports asked for

- Statistics and quality of answers received;
- Report on the analysis, including (1) the method of computing IDs, and (2) computed IDs of four song recordings;
- Rewards given to each worker, summing up to $10;
- Ratings of their employers, summing up to 10, or 1000%; and
- Comments and suggestions.

We encouraged students to write their own software for analysis and include their source code in their submission.

We asked the students to give ratings of their employers, as a way to compute each employer's reputation, similarly to how they would give rewards. The average reward for a worker from one employer is $1, and the average rating for an employer from one worker is 1 or 100%. Students were graded based on their performance in each part, incorporating the reward they received as workers and ratings as employers. We encouraged them to complete a post-survey regarding this assignment after they had completed it.

Results and Discussion

Thirty-four students participated in this assignment, crowdsourcing the recognition of 170 recordings. Most of them were graduate students majoring in computer science, and approximately 20% were undergraduate students. On average, each student worked for 12 others, meaning that each recording received 12 answers. Thirty participants recruited 10 or more workers. Twenty-seven students completed a post-survey (not everyone answered every question) about their experience.

We now present recognition results produced by our students, their performance and experience with the assignment, and recommendations for improvement.

Hummed Song Recognition

The accuracy of song recognition resulted from a combination of the quality of our recordings and the students' ability to identify them. Of the 170 recordings, 163 (96%) received at least one correct response. If we use the mode response for a recording, 153 (90.0%) were correctly recognized, which confirms that the recordings were of high quality. The actual recognition rate was slightly higher, since some students used their own expertise or additional validation (e.g., listening to all proposed songs to break a tie, which was a violation of crowdsourcing. We discuss this aspect of the findings below).

Of all the responses received from workers, only 52% were correct. Not surprisingly, incorrect responses were generally independent; hence, the correct response was likely to be the mode, if not the majority. Sixteen recordings were correctly recognized with only two correct responses. Figure 2 shows the results for different classes. Overall, students accomplished the task with high recognition rates, which justifies the adoption of crowdsourcing for this task.

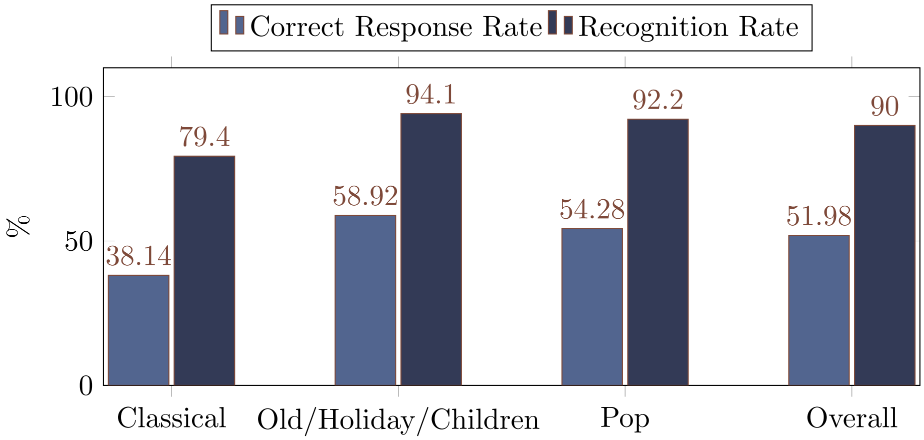

Figure 2: Accuracy of hummed song recognition.

Notice that this success may not translate to real life. A real-life recognition task may include hard-to-identify, poor quality snippets. A list of possible answers, albeit a long one, that was provided to the workers could have influenced the results.

Assignment Execution

A few students failed to meet the deadline of Part I. It is not uncommon for students not to read the instructions until a few days before the final deadline. This assignment was of unusual structure, with three separate deadlines. Some students did not realize the first deadline was actually four days before the final one. We emailed these students and they managed to keep up after that.

Most students did not make the effort of recruiting workers, although we had emphasized that recruiting was an important phase of crowdsourcing. Since we were using their classmates as the targeted crowd, students were unaware of the possibility of lacking workers. In fact, the initial assignment of workers at the beginning of Part II was highly imbalanced, which we discuss later.

The average grade of this assignment was 94.5%. Most students conducted good analyses regardless of the quality of the responses. Students' grades included the rewards and ratings they received from their employers and workers, respectively. Nearly all students reported that the rewards and ratings were fair. The average reward and reputation each student received were $10 and 10, respectively, per the instruction, but the standard deviation of the rewards was $5.5, much higher than that of the ratings, which was 2.6. Students' effort as workers was much more varied than as requesters.

Student Learning and Experience

Most students used mode as a technique to get final answers, since only a few asked additional questions to differentiate the responses. Also, the overall quality of the responses was high. This made it unnecessary for them to write their own code to obtain their final answers.

We asked the students about their understanding of crowdsourcing before the lectures (BL), after the lectures (AL), and after this assignment (AA) on a one-to-five Likert scale where one corresponds to *Know little or nothing* and five to *Expert*. Lectures and this assignment improved their understanding levels of crowdsourcing (from 2.00 to 3.77, then to 4.12), with significance of $p \leq 0.01$ and $p \leq 0.05$, respectively, in the Wilcoxon signed-rank test. Fig. 3 shows the distribution of students' Likert ratings in the three stages, as well as the box plots of their improvements by the lectures and overall. The curves show the normal distributions that match the means and standard deviations.

Even students who did not report an improvement in understanding post-assignment accepted that the assignment was valuable in learning crowdsourcing. In fact, more than 80% of the participating students agreed that this assignment was essential to the teaching of crowdsourcing. For example, one student reported: "this assignment was very fundamental to learning the basics of crowdsourcing. It taught how to prepare a task, launch a task, receive responses, and analyze the responses." We think the lectures covered the basics of crowdsourcing well, and this assignment was a great supplement to them.

Most students needed greater detail in instructions than we provided, especially about MTurk Sandbox. Most participants reflected that their experience was "fun" or "pretty good," and that the amount of work they had to do as workers was appropriate. A few students indicated that they would have preferred an additional one or two days to complete the assignment.

Opportunities for Improvement

During the lectures, we discussed the importance of participation motivation and quality-improving strategies in crowdsourcing projects. However, in this assignment, most students asked only one

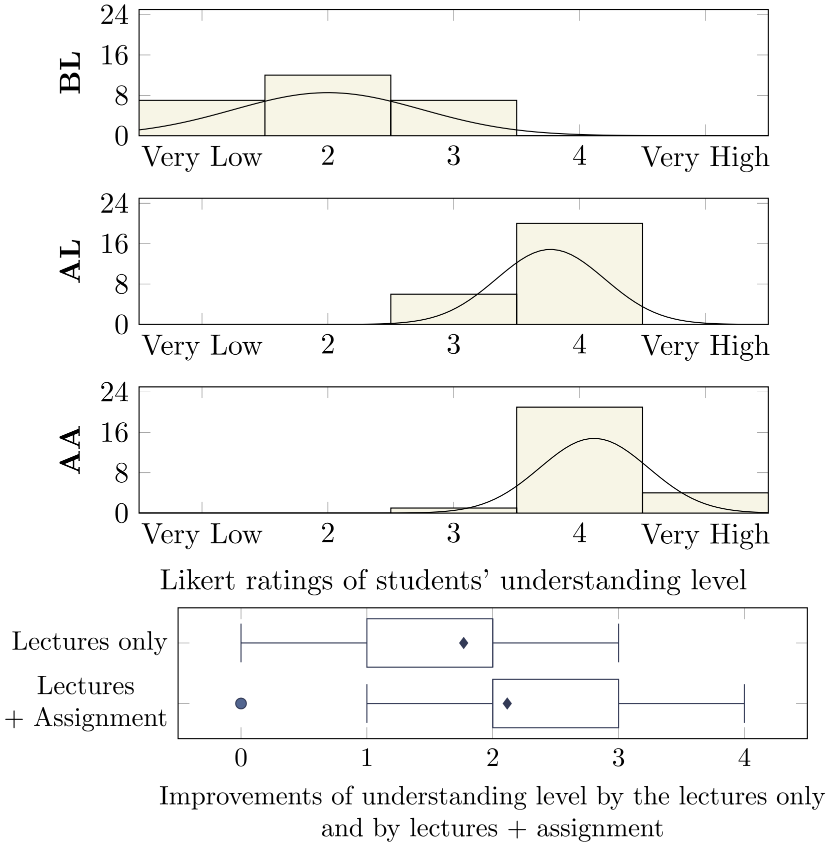

Figure 3: Students' understanding levels of crowdsourcing before the lectures (BL), after the lectures (AL), and after assignment (AA), and box plots of the improvements

straightforward question for each song in their projects, i.e., "what is the ID of this song?" Only a few students implemented additional questions to improve the quality of the responses, such as "how confident are you about your answer?" In the post-survey, almost all students realized that it was important to employ strategies to improve response quality, and more than two-thirds of them would ask different or additional questions were they to do this assignment again. We should emphasize quality-improving measures more intensively in the instructions, possibly by making them part of the grade.

The initial assignment of workers was highly imbalanced with some requesters' projects obtaining more than twenty workers and some fewer than five. We traced this problem to our placing creating projects and signing up for work in the same work segment (same deadline). Those students who wanted to complete the segment early found only a few tasks available when they signed up, and decided to ignore our guideline about balancing the work assignments. Therefore, the early tasks ended up with more than twice the target number of workers. Simply breaking the project creation (as requesters) and signing up (as workers) steps could have averted this problem. We recovered by offering extra credit for additional work so that each task would receive at least 10 workers. Consequently, some projects had more than 10 workers, on average, those who worked on exactly 10 projects received less than $10 overall.

Since we gave points for accuracy, some students applied their own expertise to identify a song. This is against the spirit of crowdsourcing. In future iterations, we will explicitly limit computed results to be based on workers' answers.

Conclusions and Recommendations

This paper illustrates an approach for teaching crowdsourcing in a college course. The assignment led to good results in terms of demonstrating the value of crowdsourcing to students, improving their knowledge, and keeping them interested.

Although more than half of the participants thought that MTurk Sandbox was necessary, some thought its complexity was unnecessary because this assignment didn't involve a public crowd. Workers drawn from a public crowd may be less reliable than those working on a shared task [3]. Some suggested using a public crowd to enhance realism. It would be straightforward to modify the instructions. However, a public crowd would introduce challenges such as stricter qualification tests and arranging for payments—which is administratively nontrivial in a university. We cannot impose an expense upon the students nor can we provide the funds.

Our assignment was simplified by providing high-quality snippets along with a list of possible answers. Future assignments on this theme can easily introduce interesting tasks that require human insight and adjust the level of difficulty. Some students suggested vision tasks, such as celebrity recognition from images. In addition, future assignments can emphasize parts of crowdsourcing that this assignment simplified, such as (1) coming up with a strategy for selecting workers, (2) dynamically determining the number of workers needed for a certain task, and (3) dealing with greater numbers of wrong responses.

The post-survey was filled by the students after all aspects of this assignment, including the grading, had concluded, and only self-assessments were conducted, which might have biases on students' actual understanding of the topic. However, the reported relative increases in students' understanding levels can in fact reflect their acknowledgement and validation of the effect of our teaching. In future offerings, rigorous tests in different stages may reflect students' improvement with greater validity.

Practice on crowdsourcing is necessary to learning it. Based on our exploratory attempt of a crowdsource-practicing assignment, we came to find some pointers that can help instructors give students better experience while achieving their course learning objectives.

- Amplify the weight of the real-life platform, and give more instructions on its usage.
- Explore different tasks to solve that can emphasize the power of crowdsourcing.
- Select the targeted crowd, public or the class, based on your learning goals, task design, and allotted time and funds.
- Take into account the fact that students progress at different paces in a project that needs their collaborative work.

Author Bios

Hui Guo is a PhD student in Computer Science at NC State University. His research interests include multiagent systems, NLP, text mining and crowdsourcing. Guo has an MS in Com-

puter Science from East Carolina University, and a BS from Tsinghua University. Contact him at hguo5@ncsu.edu.

Nirav Ajmeri is a PhD student in Computer Science at NC State University. His research interests include software engineering and multiagent systems with a focus on security and privacy. Ajmeri has a BE in Computer Engineering from Sardar Vallabhbhai Patel Institute of Technology, Gujarat University. Contact him at najmeri@ncsu.edu.

Munindar P. Singh is an Alumni Distinguished Graduate Professor in Computer Science and a co-director of the Science of Security Lablet at NC State University. His research interests include the engineering and governance of sociotechnical systems. Singh is an IEEE Fellow, a AAAI fellow, a former Editor-in-Chief of *IEEE Internet Computing*, and the current Editor-in-Chief of *ACM Transactions on Internet Technology*. Contact him at singh@ncsu.edu.

References

[1] A. J. Berinsky, G. A. Huber, and G. S. Lenz. Using mechanical turk as a subject recruitment tool for experimental research. *Political Analysis*, 20:351–68, 2011.

[2] T. Bertin-Mahieux and D. P. Ellis. Large-scale cover song recognition using the 2d fourier transform magnitude. In *13th International Society for Music Information Retrieval Conference*, pages 241–246, 2012.

[3] J. P. Bigham, M. S. Bernstein, and E. Adar. Human-computer interaction and collective intelligence. In *Collective Intelligence Handbook*. MIT Press, 2014.

[4] J. P. Bigham, C. Kulkarni, and W. S. Lasecki. Crowdsourcing and crowd work. In *2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1186–1189, 2017.

[5] C. Davidson. How to crowdsource grading. https://www.hastac.org/blogs/cathy-davidson/2009/07/26/how-crowdsource-grading, July 2009. [Online; accessed September-2017].

[6] R. Gershman. Crowdsourcing: An old idea amplified by modern technology. http://www.onespace.com/blog/2016/03/crowdsourcing-old-ideaamplified-by-technology/, Mar. 2016. [Online; accessed September-2017].

[7] J. K. Goodman, C. E. Cryder, and A. Cheema. Data collection in a flat world: The strengths and weaknesses of mechanical turk samples. *Journal of Behavioral Decision Making*, 26(3):213—-224, 2013.

[8] J. S. Julià. Identification of versions of the same musical composition by processing audio descriptions. *PhD thesis, Universitat Pompeu Fabra, Barcelona*, 2011.

[9] F. Kamoun, D. Alhadidi, and Z. Maamar. Weaving risk identification into crowdsourcing lifecycle. *Procedia Computer Science*, 56:41–48, 2015.

[10] V. Kharat, K. Thakare, and K. Sadafale. A survey on query by singing/humming. *International Journal of Computer Applications*, 111(14):39–42, Feb. 2015.

[11] P. M. Sadler and E. Good. The impact of self- and peer-grading on student learning. *Educational Assessment*, 11(1):1–31, 2006.

[12] J. Vassileva. Motivating participation in social computing applications: A user modeling perspective. *User Modeling and User-Adapted Interaction*, 22(1):177–201, Apr. 2012.

[13] A. L.-C. Wang. An industrial strength audio search algorithm. In *International Society of Music Information Retrieval*, pages 7–13, 2003.

[14] C.-C. Wang, J.-S. Roger, and J. W. Wang. An improved query by singing/humming system using melody and lyrics information. In *The 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, pages 45–50, Utrecht, Netherlands, 2010.

[15] J. Yang, J. Liu, and W.-Q. Zhang. A fast query by humming system based on notes. In *INTERSPEECH*, pages 2898–2901, 2010.