

## Review:

# Crowd intelligence in AI 2.0 era\*

Wei LI<sup>1</sup>, Wen-jun WU<sup>†1</sup>, Huai-min WANG<sup>2</sup>, Xue-qi CHENG<sup>3</sup>,  
Hua-jun CHEN<sup>4</sup>, Zhi-hua ZHOU<sup>5</sup>, Rong DING<sup>1</sup>

(<sup>1</sup>State Key Laboratory of Software Development, Beihang University, Beijing 100191, China)

(<sup>2</sup>National Laboratory for Parallel and Distributed Processing, College of Computer,  
National University of Defense Technology, Changsha 410073, China)

(<sup>3</sup>Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

(<sup>4</sup>College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

(<sup>5</sup>National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China)

<sup>†</sup>E-mail: wwj@nlsde.buaa.edu.cn

Received Dec. 23, 2016; Revision accepted Jan. 3, 2017; Crosschecked Jan. 5, 2017

**Abstract:** The Internet based cyber-physical world has profoundly changed the information environment for the development of artificial intelligence (AI), bringing a new wave of AI research and promoting it into the new era of AI 2.0. As one of the most prominent characteristics of research in AI 2.0 era, crowd intelligence has attracted much attention from both industry and research communities. Specifically, crowd intelligence provides a novel problem-solving paradigm through gathering the intelligence of crowds to address challenges. In particular, due to the rapid development of the sharing economy, crowd intelligence not only becomes a new approach to solving scientific challenges, but has also been integrated into all kinds of application scenarios in daily life, e.g., online-to-offline (O2O) application, real-time traffic monitoring, and logistics management. In this paper, we survey existing studies of crowd intelligence. First, we describe the concept of crowd intelligence, and explain its relationship to the existing related concepts, e.g., crowdsourcing and human computation. Then, we introduce four categories of representative crowd intelligence platforms. We summarize three core research problems and the state-of-the-art techniques of crowd intelligence. Finally, we discuss promising future research directions of crowd intelligence.

**Key words:** Crowd intelligence; Artificial intelligence 2.0; Crowdsourcing; Human computation  
<http://dx.doi.org/10.1631/FITEE.1601859>

**CLC number:** TP18

## 1 Introduction

The Internet based cyber-physical world has profoundly changed the information environment for the development of artificial intelligence (AI), thus bringing a new wave of AI research and promoting it into the new era of AI 2.0 (Pan, 2016). It empowers human intelligence to play an increasingly important role, and reshapes the landscape of AI research. As one of the major features of AI 2.0, crowd intelligence emerges from the collaborative

efforts of many autonomous individuals, showing higher intelligence than the capability of each individual. Many Internet applications, such as open source software development, Wikipedia, Web Q&A, crowdsourcing, and the sharing economy, have successfully tapped into talent pools of crowds, and demonstrate significant progress beyond the traditional paradigms. The success of these applications inspires many researchers from interdisciplinary areas such as AI, human-computer interaction, cognitive science, management science, economy, and auction theory, to investigate the research problems of crowd intelligence. Over the past decade, they have proposed a number of frameworks, models, and

<sup>‡</sup> Corresponding author

\* Project supported by the National Natural Science Foundation of China (No. 61532004)

ORCID: Wen-jun WU, <http://orcid.org/0000-0002-8659-481X>  
© Zhejiang University and Springer-Verlag Berlin Heidelberg 2017

algorithms to achieve further theoretical understanding of crowd intelligence in areas including social computing, crowdsourcing, and human computation. To summarize research efforts and practical experience in these diversified areas from a consistent perspective, we present a comprehensive survey of the theoretical research and industrial applications in this emerging field.

The rest of the paper is organized as follows. Section 2 introduces the definition of crowd intelligence, and elaborates the scope of the field in comparison with other related concepts. Section 3 proposes a taxonomy of crowd intelligence platforms. Section 4 describes the core research issues and the state-of-the-art.

## 2 Definition of crowd intelligence

### 2.1 What is crowd intelligence?

Before we can give a clear definition of crowd intelligence, it is necessary to explain these important concepts that are closely related to the topic of our paper, and elaborate the differences and similarities among them. The most important concept is collective intelligence. This has been widely adopted by researchers, in cognitive science, social psychology, and management science since the 1990s. Note that there have been many definitions of collective intelligence. Many researchers attempt to describe the meaning of collective intelligence from their own perspective. For instance, according to Wikipedia, collective intelligence is defined as “shared or group intelligence that emerges from the collaboration, collective efforts, and competition of many individuals and appears in consensus decision making” ([http://en.wikipedia.org/wiki/Collective\\_intelligence](http://en.wikipedia.org/wiki/Collective_intelligence)). This definition emphasizes the collective capability from the aspect of decision augmentation, which can elevate the decision capability of individuals (Bonabeau, 2009). Smith (1994) defined it as “a group of human beings carrying out a task as if the group, itself, were a coherent, intelligent organism working with one mind, rather than a collection of independent agents”. This definition again highlights the nature of a group intelligent mind. Pierre (1997) described collective intelligence as “a form of universally distributed intelligence, constantly enhanced, coordinated in real time,

and resulting in the effective mobilization of skills”. Every version of the above definitions intends to express its understanding about what intelligence means and how a human group works together to achieve the intelligent outcomes.

The latest definition of collective intelligence can be found in the online ‘Collective Intelligence Handbook’ (Malone *et al.*, 2009). The authors defined collective intelligence as “groups of individuals acting collectively in ways that seem intelligent”. This is a very general definition that requires collective intelligence to be manifested as intelligent group activities and behaviors. Apparently, any kind of human group activity can fall into the scope of collective intelligence, as long as the interactions among the group members demonstrate intelligent features, such as generation of new knowledge, consensus decisions, and the emergence of smart behavior patterns.

From the formal definitions of collective intelligence, one can see that it is a term coined in the context of social psychology science. How can we refine the concept of collective intelligence in the field of computer science, and more specifically, AI? In this paper, we give the following definition of crowd intelligence:

**Definition 1** (Crowd intelligence) Beyond the limitation of individual intelligence, crowd intelligence emerges from the collective intelligent efforts of massive numbers of autonomous individuals, who are motivated to carry out the challenging computational tasks under a certain Internet-based organizational structure.

The above definition shows that crowd intelligence is an essentially Internet-based collective intelligence that has the following distinguishing properties:

First, Internet-based crowd intelligence emerges from massive numbers of individuals in online organizations and communities on online platforms. None of the definitions of collective intelligence specifies the platform supporting group collaboration and coordination. Historically, any social organization from a cyber infrastructure can foster the generation of collective intelligence. For instance, conventional academic groups or lab teams are effective channels to organize scientists to work together in research projects. Pervasive Internet access further extends the forms of collective intelligence, and enables explosive growth in online communities and virtual

organizations towards a variety of collective endeavors in areas of science discovery, crowd innovation, software development, and even service markets. Our definition introduces an explicit statement on this trend, and emphasizes the importance of the Internet in the emergence of crowd intelligence.

Second, a crowd intelligence system interweaves crowd and machine capabilities seamlessly to address challenging computational problems. The origin of the term ‘collective intelligence’ does not narrow the definition within the scope of computer science and AI. Our definition attempts to give a clear description about the computational context of crowd intelligence. From the aspect of computing, the rise of crowd intelligence allows novel possibilities of seamlessly integrating machine and human intelligence at a large scale, which can be regarded as mixed-initiative intelligent systems. In such a system, AI machines and crowds can complement the capability of each other to function as enhanced AI systems.

Many computational tasks, such as image recognition and classification, are very trivial for human intelligence, but pose grand challenges to current AI algorithms. AI researchers developed a new computational paradigm called human computation (Law and Ahn, 2011), where people act as computational components, and perform the work that AI systems lack the skills to complete. The outcomes of crowd intelligent tasks, including data collection and semantic annotation, can be used to help train, supervise, and supplement their algorithms and models. In addition to the support from data annotation and enrichment, crowd intelligence enables the development of AI software systems. AI software components can be developed in an open and distributed way, in which crowd members in open source software communities can actively get engaged.

With the increase in the scale of crowd systems, it is challenging to coordinate the work process of massive crowds to handle complex tasks. AI can work as enablers to help the crowd act in a more efficient, skilled, and accurate way. Machine intelligence plays the role of ‘crowd organizer and mediator’ to stimulate the activities of individuals, assess their skill levels, assign subtasks to them, and aggregate the outcomes of their tasks. On the platforms of online communities, AI can implement knowledge sharing and management among individuals for large-scale collaborations.

## 2.2 Other related concepts

There are many concepts such as social computing, crowdsourcing, and human computation, which are similar to crowd intelligence. These notions overlap more or less with crowd intelligence in their research areas. To clarify these concepts, Fig. 1 illustrates their scopes as overlapping circles.

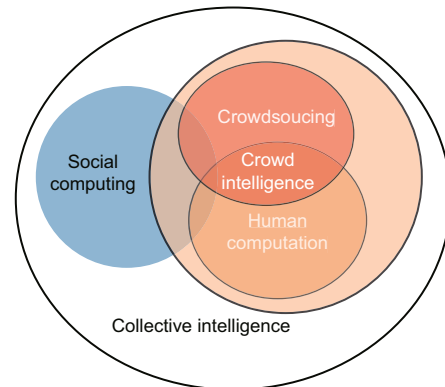


Fig. 1 Crowd intelligence and related areas

### 2.2.1 Social computing

The term ‘social computing’ is a broad concept that covers everything to do with social behavior and computing. The central idea of social computing is the use of social software to support social interaction and communication. Research topics in social computing include social network analysis, online social dynamic modeling, and design of social software.

Social computing is closely related to the notion of collective intelligence, because both terms have their roots in the area of social science. From the aspect of group collaboration, social computing is considered as a sort of collective intelligence, which provides users a way to gain knowledge through collective efforts in a social interactive environment.

### 2.2.2 Crowdsourcing

The term ‘crowdsourcing’ was defined as “outsourcing a job traditionally performed by an employee to an undefined, generally large group of people via open call” (Li et al., 2016; Tong et al., 2017). Numerous tasks or designs conventionally carried out by professionals are now being crowdsourced to the general public, who may not know each other, to perform in a collaborative manner. Specifically, crowdsourcing has been widely used for data

cleaning, identifying chemical structure, image recognition, logo design, medical drug development, taxonomy construction, topic discovery, social network analysis, and even software design and development (Cao *et al.*, 2012; 2013; Wang JN *et al.*, 2013; Tong *et al.*, 2014b; 2014a; Zhang CJ *et al.*, 2014a; Meng *et al.*, 2015; Zhang *et al.*, 2015).

Crowdsourcing activities involve accessing a large crowd of workers, who can either collaborate or compete to accomplish intelligent tasks that are typically performed by a single individual or group. These organizing crowdsourcing projects need to run their processes on Web platforms. Machine intelligence has been extensively integrated into crowdsourcing platforms to accurately rank the skills of crowdsourcing workers, and effectively ensure the quality of their work. Thus, the area of crowdsourcing should fall into the scope of crowd intelligence.

### 2.2.3 Human computation

Human computation refers to “a paradigm for using human processing power to solve problems that computers cannot yet solve” (Law and Ahn, 2011). It centers around harnessing human intelligence to solve computational problems that are beyond the scope of existing AI algorithms.

The central element in human computation is the microtask or human intelligent task. Human computation often breaks a large computational project into tiny and well-defined tasks for a crowd to work on. The majority of microtasks are designed as short-time work without time-consuming effort, such as image tagging, translating sentences or paragraphs, filling survey forms in social experiments, and adding transcripts segments.

Although human computation has different research goals and subjects from social computing, the two areas do have some intersections, because human computation systems require social behavior and interaction among a group of people. In essence, human computation is more similar to crowdsourcing because it often requires open calls for work to attract crowd workers to undertake microtasks. However, with the different focuses on crowdsourcing, human computation places more emphasis on the computational microtasks that can be assigned to a group of individuals in a crowdsourcing way. Similar to crowdsourcing, human computation needs to be orchestrated through online platforms such as online

microtask markets where a variety of human intelligent tasks can be posted and processed by crowd workers.

## 3 Typical crowd intelligence platforms

Currently, crowd intelligence has been widely applied in massive data processing, scientific research, open innovation, software development, and the sharing economy. Every application area of crowd intelligence has customized requirements for crowd tasks, organization styles, and workflows. To support such specific requirements, practitioners of crowd intelligence need to set up an online platform that connects many individuals, and coordinates their work through a specific organizational mechanism. In this section, we introduce a taxonomy of crowd intelligence platforms based on the following properties of the platform:

**Crowd task:** What kinds of tasks are available for a crowd worker? What is the nature of these tasks? Is it a tiny and light-weight microtask or a time-consuming product-design task?

**Organizational mechanism:** How can the platform motivate potential workers to actively get involved in crowd tasks? Can the platform rank workers based on their skills and perform adaptive task allocation for these workers? What kind of payment or reward can the platform give to the workers after they deliver high-quality outcomes?

**Problem solving workflow:** How can a challenging problem be divided into smaller tasks for the crowd? What’s the right pipeline for orchestrating crowd workers to collaborate on a complex task? What’s the appropriate quality assurance method to overcome the inherent variability in crowd productions?

### 3.1 Human computation and microtasking

Human computation platforms can organize humans to carry out the process of computation, whether it be performing the basic operations, orchestrating the human computation workflows (e.g., deciding what operations to execute next, or when to halt the execution of the workflow), or even synthesizing the workflows on the fly (e.g., by creating composite operations from basic operations and specifying the dependence among the operations).

### 3.1.1 Crowd tasks in human computation

We introduce crowd tasks in human computation through two well-known platforms.

Amazon Mechanical Turk (AMT) (<http://www.mturk.com>) is the most famous human computation platform that coordinates the supply and demand of micro computational tasks that require human intelligence to complete. It is an online labor market where crowd workers are recruited by requesters for the execution of human intelligent tasks (HITs) in exchange for a small-amount reward. Tasks are typically simple enough to require only a few minutes to be completed such as image tagging, audio transcriptions, and survey completion.

reCAPTCHA (von Ahn *et al.*, 2008) is a human computation system for transcribing old books and newspapers for which optical character recognition (OCR) is not very effective. It takes advantage of the preexisting need for reCAPTCHA, the distorted images of text that are used by websites to prevent access by automated programs. When a user goes to a website, instead of seeing computer generated distorted text, they see an image of a word from an old book or newspaper, whose content could not be identified by the OCR software. By typing the letters in the course of visiting the website, the user provides computational power to help with the transcription effort.

### 3.1.2 Organizational mechanism of human computation

According to the survey of Quinn and Bederson (2011), there are four major kinds of incentive mechanisms that motivate human computation workers: payment, altruism, enjoyment, and reputation.

Financial rewards are probably the easiest way to recruit workers. However, as soon as money is involved, people have more incentive to cheat the system to increase their overall rate of pay. Also, because participants are usually anonymous, they may be more likely to do something dishonest than they would when identified. Commercial human computation platforms, such as AMT, adopt the monetary mechanism to attract an online workforce. Once a microtask worker has completed a task, the task requester needs to pay him at a very low rate, which can be as low as \$0.01, and rarely exceeds \$1.

All of the other three mechanisms are non-monetary and rely on people's inner motivations to participate as crowd workers. Note that all these incentives are mostly applicable in the context of collaboration and communities. For example, in Section 3.3, citizen science platforms often aim at promoting people's curiosity, altruism, and desire for reputation to undertake human computation tasks for scientific data collection and analysis. For altruism and enjoyment, the crowd tasks must be either interesting or important to attract Web users to participate. Reputation is often employed by online communities to motivate crowd workers to actively undertake tasks to receive public recognition for their efforts.

### 3.1.3 Problem-solving workflow in human computation

In AMT, both workers and requesters are anonymous although responses by a unique worker can be linked through an ID provided by Amazon. Requesters post HITs that are visible only to workers who meet the predefined criteria (e.g., country of residence or accuracy in previously completed tasks). When the workers access the website, they find a list of tasks sortable according to various criteria, including the size of the reward and the maximum time allotted for the completion. Workers can read brief descriptions and see previews of the tasks before accepting to work on them. Fig. 2 illustrates the human computation workflow of AMT.

One of the distinguishing properties of human computation is in the way that explicit control is exercised. Although most tasks on human

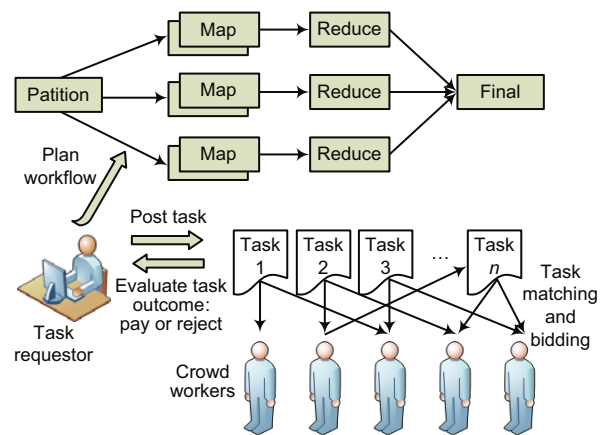


Fig. 2 Human computation and microtask platform

computation markets are simple and self-contained with no challenging coordination dependencies, they can support programmable application program interfaces (APIs) to implement explicit decomposition or assignment of tasks, and allow the developers to design quality mechanisms to ensure that the human computers tell the truth.

In a lot of human computation applications using AMT's APIs, more complicated tasks are typically decomposed into a series of smaller tasks, including the checking and validation of other workers' HITs. AMT can be viewed as large distributed computing systems in which each work is analogous to a computer processor that can solve a task requiring human intelligence. Based on this, researchers on human computation (Kittur *et al.*, 2011) introduced a map-reduce based approach to design problem solving workflows, in which a complex problem is decomposed into a sequence of simpler map and reduce subtasks. They define three types of subtask: the partition task, in which a larger task is broken down into discrete subtasks; the map task, in which a specified task is processed by one or more workers; the reduce task, in which the results of multiple workers' tasks are merged into a single output.

### 3.2 Mobile crowdsourcing platforms and sharing economy

Mobile crowdsourcing is an extension of human computation from the virtual digital world to the physical world. The 2009 Defense Advanced Research Projects Agency (DARPA) Red Balloon Challenge (Tang *et al.*, 2011) is a well-known exemplar test for the effectiveness of mobile crowdsourcing efforts. It intentionally explored how mobile crowdsourcing can be used to solve a distributed, time-critical geo-location problem. Ten red balloons were deployed at undisclosed locations across the continental USA, and a prize would be awarded to the winning team for correctly identifying the locations of all 10 balloons first. A team from the Massachusetts Institute of Technology (MIT), which used the geometric reward mechanism (recursive incentive mechanism), won in less than nine hours.

#### 3.2.1 Crowd tasks in mobile crowdsourcing

Participatory sensing proposed by Burke *et al.* (2006) intends to organize mobile devices to form

interactive and participatory sensor networks, which enable public and professional users to gather, analyze, and share local knowledge. It can be further extended into mobile crowd sensing, which leverages both sensed data from mobile devices (from the physical community) and user-contributed data from mobile social network services. Mobile crowd sensing greatly extends the capability of group perception and awareness in many areas, such as health care, urban planning and construction, as well as environmental protection. Fig. 3 illustrates the major tasks and applications in both participatory sensing and mobile crowdsourcing.

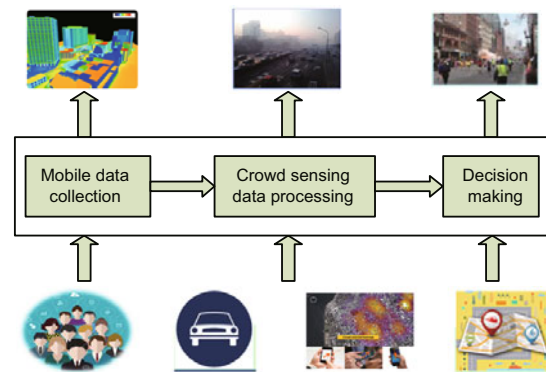


Fig. 3 Participatory and mobile crowd sensing

Transportation and traffic planning: TrafficInfo (Farkas *et al.*, 2014) implements a participatory sensing-based live public transport information service, which exploits the power of the crowd to gather the required data, share information, and send the feedback. TrafficInfo visualizes the actual position of public transport vehicles with live updates on a map and gives support to crowdsourced data collection and passenger feedback. Moreover, Zhang CJ *et al.* (2014b) adopted the mobile crowdsourcing approach to enhance the accuracy of navigation systems.

Environment monitoring: EarPhone (Rana *et al.*, 2010) presents the design, implementation, and performance evaluation of an end-to-end participatory urban noise mapping system called EarPhone. The key idea is to crowdsource the collection of environmental data in urban spaces to people who carry smartphones equipped with GPS and other sensors.

Public safety: Recently, user-contributed data has been used for crime prevention. For instance, Ballesteros *et al.* (2014) showed that the data

collected by geo-social networks bears a relationship to crimes, and proposed iSafe which is able to evaluate the safety of users based on their spatial and temporal dimensions. The spread of technology from the security cameras to the smartphones in every pocket, has proved helpful to criminal investigations. It has been reported that, during the Boston marathon explosion in Apr. 2013, photos and videos shot by onlookers after the explosion have been used as evidence in the investigation by FBI (Fowler and Schectman, 2013).

Mobile crowdsourcing marketplaces: A number of companies, for example, GigWalk (<http://www.gigwalk.com>), Field Agent (<http://www.fieldagent.net>), and TaskRabbit (<http://www.taskrabbit.com>), have sprung up to build mobile crowdsourcing marketplaces (Gao et al., 2016). These mobile crowdsourcing markets offer physical world tasks to crowd workers, such as checking street signs, verifying the existence of a pothole, and running household errands. One of the sharing economy's pioneers and largest success stories, TaskRabbit Inc., allows users to outsource small jobs and tasks to local contractors.

Mobile crowdsourcing tasks have varying complexities, which require different levels of time, expertise, and cognitive efforts to complete. For example, most tasks posted in TaskRabbit include deliveries, furniture assemblies, and house chores, which often demand high degrees of expenditure of effort. In contrast, common tasks posted on the site of GigWalk are mostly microtasks, including store audits, price checks, customer interviews, and field photography. These microtasks can be done while workers are already out and near the task locations.

### 3.2.2 Organizational mechanism of mobile crowdsourcing

Teodoro et al. (2014) conducted a qualitative study to investigate the motivations of workers in both TaskRabbit and GigWalk, shown as follows:

Motivation for joining on-demand mobile workforce: They found that monetary compensation and control of working conditions (e.g., working duration, payment rate, and the allocated tasks) are the primary factors for joining these systems. Participants seem to prefer a flexible work schedule, personal control of working tasks, and locations as

the main motivations for joining a mobile workforce service.

Motivation for selection of mobile crowdsourcing tasks: The main motivations driving task selection over other tasks involve cost-benefit analysis, physical location, geographic factors, as well as trust in the task requesters. Participants consider the costs, distance, benefits, and conditions associated with completing the physical world tasks, such as time spent traveling, gas consumption, familiar/safe locations, and potential payout. For example, how far people have to travel appears to influence their attitudes toward a task. This is a reflection of the important principle of distance decay in geography. In addition, statistical modeling shows that both the distance to a task and the socioeconomic status of a task area influence whether a worker is willing to accept it, and that distance is the dominant predictor of task price (Thebault-Spieker et al., 2015). In addition to location factors, workers consider situational factors about the task, such as their availability of time, the timing of the task, and even weather conditions. They express preferences for tasks posted by requesters who have pictures and information on their profiles verifying their identities.

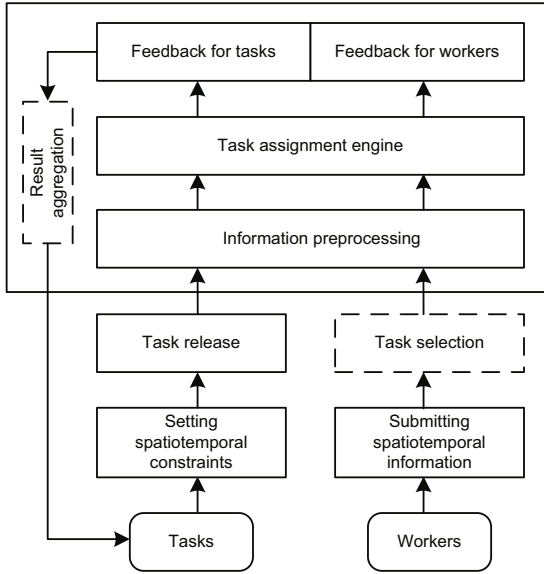
### 3.2.3 Problem-solving workflow in mobile crowdsourcing

In this subsection, we first introduce a general problem-solving workflow in mobile crowdsourcing, and then further discuss different problem-solving models in real mobile crowdsourcing applications.

Problem-solving workflow: A general problem-solving workflow is shown in Fig. 4 (Tong et al., 2017). The participants in mobile crowdsourcing include 'requesters of tasks' and 'workers', who are connected through mobile crowdsourcing 'platforms'. We next introduce the workflow from the perspectives of the requesters of tasks, workers, and platforms.

1. Requesters of tasks. The requesters set the spatiotemporal constraints of their tasks at first. Next, the requesters submit the tasks to the platforms. Then, the requesters need to wait for a feedback from the platforms.

2. Workers. To perform tasks, the workers should submit their spatiotemporal information first. Then, they can select tasks autonomously or wait to



**Fig. 4 Problem-solving workflow in mobile crowdsourcing**

be assigned some tasks by the platforms. Workers also need to wait for feedback from platforms.

3. Platforms. Platforms are central to the workflow. After receiving the information from the requesters/workers and preprocessing the information, the platforms transfer the processed information to the task assignment engine, whose duty is to assign the tasks to appropriate workers. According to the requirements of different types of tasks, the platforms can feed back the result of the tasks to the corresponding requesters with or without aggregation.

**Problem-solving models:** As shown in the aforementioned workflow, the core component in the problem-solving process is the task assignment engine, which chooses different task assignment strategies to assign tasks to suitable workers. According to different task assignment strategies, the existing problem-solving approaches in mobile crowdsourcing can be described as the following two models: matching-based model and planing-based model.

1. Matching-based model. This model uses the bipartite graph matching model as the task assignment strategy, where each task and worker corresponds to a vertex in the bipartite graph, and the relationship between the task and the worker is regarded as the edges in the bipartite graph (Tong *et al.*, 2016c; 2016b). Based on different optimization goals and constraints, there are some variants of the matching-based model. For example, She *et al.*

(2015b; 2016) proposed a conflict-based matching model to address the spatiotemporal conflict among different vertices of a bipartite graph. Tong *et al.* (2015b; 2016a) integrated the social relationships of different workers into the matching-based model. In particular, Tong *et al.* (2016c) used the online minimum matching model to explain the rationality that real-time taxi-calling services adopt the nearest neighbor approach to assign taxis to taxi-callers. Therefore, the matching-based model has a wide range of applications, such as the taxi-calling service and last-mile delivery.

2. Planning-based model. This aims to provide the optimal planning for each worker, according to the given spatiotemporal information and constraints of the tasks. She *et al.* (2015a) proposed a general planning-based model, which provides travel planning for a set of workers based on the spatiotemporal conflicts of the other set of tasks. In particular, the planning-based model can be widely used to the on-wheel meal-ordering service, real-time ride-sharing, etc.

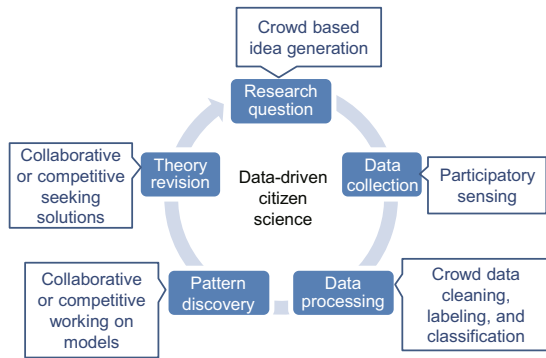
To sum up, with rapid development of mobile Internet and the sharing economy, various mobile crowdsourcing platforms are playing important roles in our daily life. Hence, research on mobile crowdsourcing will definitely have more and more significance in the future.

### 3.3 Citizen science platforms

Citizen science (also known as crowd science) is the scientific research conducted, in whole or in part, by amateur or nonprofessional scientists. It encourages the members of the public to voluntarily participate in the scientific process. Whether by asking questions, making observations, conducting experiments, collecting data, or developing low-cost technologies and open-source code, members of the public can help advance scientific knowledge and benefit the society.

Fig. 5 displays the research paradigm of citizen science. A scientific research project normally consists of five major phases: hypothesis formation, data collection, data processing, pattern discovery, and model and theory revision based on new evidence. In each of the steps, crowd intelligence can greatly accelerate the scientific research processes, enrich scientific data sets, and extract valuable insights into phenomena and observations.





**Fig. 5 Citizen science research paradigm**

We next use two examples to introduce the scientific data collection and processing.

### 3.3.1 Scientific data collection

Networks of human observers play a major role in gathering scientific data. They continue to contribute significantly in astronomy, meteorology, or observations of nature. eBird (Sullivan *et al.*, 2009) is a Web-enabled community of bird watchers who collect, manage, and store their observations in a globally accessible unified database. Birders, scientists, and conservationists are using eBird data worldwide to better understand avian biological patterns and the environmental and anthropogenic factors that influence them. Developing and shaping this network over time, eBird has created a near real-time avian data resource, producing millions of observations per year.

### 3.3.2 Scientific data processing

Galaxy Zoo (Lintott *et al.*, 2008) is a project involving over 250 000 volunteers who help with the collection of astronomical data, and who have contributed to the discovery of new classes of galaxies and a deeper understanding. It was launched in the summer of 2007, to process the large amount of image data of distant galaxies that are made available by the Sloan digital sky survey (SDSS). Volunteers are asked to sign up, read an online tutorial, and then code six different properties of astronomical objects visible in SDSS images. Seven months after the project was launched, about 900 000 galaxies had been coded, and multiple classifications of a given galaxy by different volunteers were used to reduce the incidence of incorrect coding, for a total of roughly 50 million classifications. For an individual scientist,

50 million classifications would have required more than 83 years of full-time effort. The Galaxy Zoo data allowed Lintott's team to successfully complete the study they had initially planned.

The success of Galaxy Zoo sparked interest in various areas of science and the humanities. In 2009, Lintott and his team established a cooperation with other institutions in the UK and USA to run a number of projects on a common platform 'the Zooniverse' (Borne and Zooniverse Team, 2011). The Zooniverse platform currently has more than one million registered volunteers, and hosts dozens of projects in fields as diverse as astronomy, marine biology, climatology, and medicine. Recent projects have also involved participants in a broader set of tasks and in closer interaction with machines.

## 3.4 Crowd-based software development

Software development is a kind of intellectual activity, which involves both the creation and manufacturing activities of the crowds (Wang *et al.*, 2015). In the process of software development, various types of tasks, such as the requirement elicitation and bug finding, may rely on the creativity and talents of developers. Meanwhile, ensuring the efficiency and quality of the outcome of software development requires rigorous engineering principles to be applied in the entire life cycle of software development. Open source software and software crowdsourcing have significantly changed our understanding of software development, and presented a successful demonstration of crowd-based software development.

### 3.4.1 Crowd tasks in crowd-based software development

Some software development activities, which have presented the difficulties for traditional methods for several decades, are now becoming an active arena for crowd intelligence. Essentially, software development tasks demand consistent efforts from programmers with software expertise and rigorous testing, verification, and validation process. The complex nature of software engineering determines that crowd intelligence for software development has many unique features and issues different from general crowd intelligence. Specifically, it needs to support the following three points:

1. The rigorous engineering discipline of software development, such as rigid syntax and semantics of programming languages, modeling languages, and documentation or process standards.
2. The creativity aspects of software requirement analysis, design, testing, and evolution. The issue is how to stimulate and preserve creativity in these software development tasks through crowd intelligence.
3. The quality aspects including objective qualities, such as functional correctness, performance, security, reliability, maintainability, safety, and subjective qualities such as usability.

#### 3.4.2 Organizational mechanism of crowd software development

There are two major approaches to crowd software development: open source and software crowdsourcing (Wu *et al.*, 2013). In fact, the two approaches are highly similar because they emphasize the openness of crowd software development, but from different aspects.

Open source for crowd software development: Open source software (OSS) has reshaped our understanding of software development. Open source project hosting services like GitHub (<http://www.github.com>), Google Code (<http://www.code.google.com>), and SourceForge (<http://www.sourceforge.net>) make it possible for anyone to create and manage OSS projects at any time. Developers across the world can easily access the source codes, documents, and test cases of these OSS projects, and participate in the entire development process. This meritocracy drives both professionals and amateurs to actively share their ideas, experience, and source codes in OSS communities, and to create novel software products collaboratively.

Users from all over the world, no matter what professional level they are at, can become architects through design discussion, developers through code contribution, or testers through bug reporting. The Mozilla OSS project, which has produced the famous Firefox browser, has actually gathered such a large crowd that the number of officially acknowledged contributors is over 1800. The open source method values the openness of source code among software development communities, and encourages contributions from community members through intrinsic and altruism incentives, such as the

reputation, opportunity to learn programming skills, and willingness to address user needs.

Software crowdsourcing for crowd software development: Software crowdsourcing is an open call for participation in any task of software development, including documentation, design, coding, and testing. It features the openness of the software process, distributing all kinds of development tasks to communities. Software crowdsourcing platforms often adopt explicit incentives, especially monetary reward such as contest prizes, to stimulate the participation of community members. Therefore, software crowdsourcing can be regarded as an extension of open source, which generalizes the practices of peer production via bartering, collaboration, and competition. However, it does not necessarily distribute end-products and source-material to the general public without any cost. Instead, it emphasizes community-driven software development on the basis of open software platforms, the online labor market, and a financial reward mechanism.

TopCoder (<http://www.topcoder.com>) is a software crowdsourcing example, which creates a software contest model where programming tasks are posted as contests and the developer of the best solution wins the top prize. Following this model, TopCoder has established an online platform to support its ecosystem, and gathered a virtual global workforce with more than 250 000 registered members, and nearly 50 000 active participants. All these TopCoder members compete against each other in software development tasks, such as requirement analysis, algorithm design, coding, and testing.

App Store as online software market: Apple's App Store is an online IOS application market, where developers can directly deliver their creative designs and products to smartphone customers. These developers are motivated to contribute innovative designs for both reputation and payment by the micro-payment mechanism of the App Store. Around the App Store, there are many community-based, collaborative platforms for the smartphone applications incubators. For example, AppStori (<http://www.crunchbase.com/organization/appstori>) introduces a crowd funding approach to build an online community for developing promising ideas about the new iPhone applications. Despite different needs and strategies adopted by crowd

software development processes, they actually share much commonality in terms of platform support.

### 3.4.3 Problem-solving workflow in crowd software development

As an online labor market where a crowd workforce can autonomously choose a variety of software development tasks requested by project initiators, a software crowdsourcing platform needs to effectively facilitate the synergy between two clouds—the human cloud and the machine cloud. Many core services pertaining to labor management and project governance must be incorporated into the platform, including expertise ranking, team formatting, task matching, rewarding, as well as crowd funding. Moreover, each individual should be able to easily initialize a virtual workspace with design and coding software tools customized for specific tasks in a crowd software project. All these common elements are encapsulated in the reference architecture of a crowd software development platform (Fig. 6).

**Software development tools:** In any crowd software project, the crowd workforce needs modeling, simulation, and programming language tools, such as compilers and intelligent editors, design notations, and testing tools. An integrated development environment (IDE) for crowd workers can integrate these tools for requirements, design, coding, compilers, debuggers, performance analysis, testing, and maintenance.

**Knowledge sharing and collaboration tools:** Facebook, Twitter, wikis, blogs, online forums, Question-Answer sites, and other Web-based collaboration tools allow participants to communicate for knowledge sharing and collaboration. For example, Facebook profiles can be enhanced to facilitate the formation of a crowd team, even if the participants do not previously know each other. StackOverflow enables crowd developers to post questions and answers to technical problems to share development experiences and knowledge.

**Quality assurance and improvement tools:** Software artifacts contributed by crowd developers have to be fully evaluated and tested before they can be integrated into software products. Software testing, performance profilers, and program analysis tools are very important to crowd project managers for running automatic checking on crowd submissions.

**Project management tools:** Crowd software project management should support project cost estimation, development planning, decision making, bug tracking, and software repository maintenance, all specialized for the context of the distributed and dynamic developer community. In addition to these regular functions, it needs to incorporate crowdsourcing specific services such as ranking, reputation, and reward systems for both the products and participants.

### 3.5 Summary of crowd intelligence platforms

The summary of crowd intelligence platforms is shown in Table 1.

## 4 Research problems in crowd intelligence

In this section, we discuss some research problems in crowd intelligence.

**Research problem 1** (Effectiveness of crowd organization) How can individuals with different backgrounds and skills be effectively organized in a distributed way to generate measurable and persistent crowd intelligence? Crowd intelligence may involve a wide range of crowd tasks, such as collective data annotation, collaborative knowledge sharing, and crowdsourcing-based software development, which demands different levels of expertise and dedication. To achieve the goals of these crowd tasks, one can adopt interaction patterns including collaboration, coordination, and competition to connect individuals and provide mediation mechanisms for them to work in a socialized environment.

**Research problem 2** (Incentive mechanism of crowd intelligent emergence) Through the decentralized ways of collaboration, competition, and incentive, individuals are linked as a crowd. The crowd has a complicated pattern of behavior and distinctive intelligence, which is greater than the sum of the intelligence of the individuals. However, because the individuals have high autonomy and diversity, the time, strength, and cost of the crowd intelligence have high uncertainties. Therefore, a key scientific problem is how to grasp the pattern of crowd intelligence in different scenarios, reveal its internal mechanism, and guide the incentive mechanism and operation method to realize predictable crowd intelligent emergence.

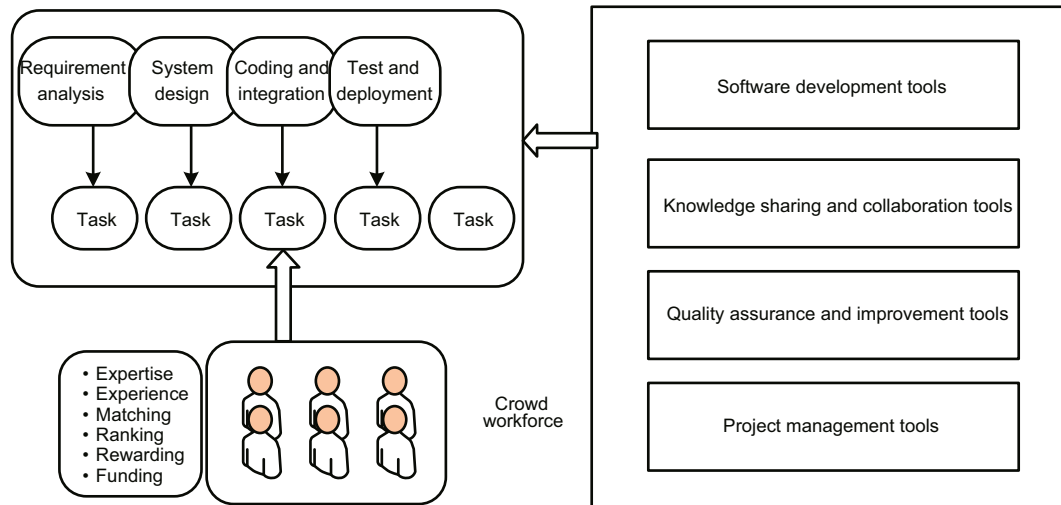


Fig. 6 Crowd-based software development platform

Table 1 Crowd intelligence platforms

Platform type	Human computation	Mobile crowd-sourcing		Citizen science	Software development
		Participatory sensing	Sharing economy		
Crowd task	Microtasks, mostly simple data processing tasks	Spatiotemporal microtasks for data gathering	Providing services or sharing goods	Simple analysis of scientific data	Software development tasks
Organizational mechanism	Monetary payment	Reputation, physical location, situation, and self-control schedule	Auction and market	Reputation curiosity learning	Open source or competition
Problem-solving workflow	Data-driven map-reduce	Data-driven map-reduce	Pricing model for optimization	Map-reduce	Crowd-based software process model

**Research problem 3** (Quality control of crowd intelligence) The submissions of crowds, such as data labels and ideas of product designs, will be further processed in the system. The quality of the submissions therefore has a great impact on the effectiveness of the system, which, however, often varies among different workers. How to assess, control, and guarantee the quality of the work, and how to use the work even of low quality are important issues. To address these problems, new mechanisms and methods are needed for quality control to assure the quality of the whole crowd intelligent system.

#### 4.1 Crowd organization

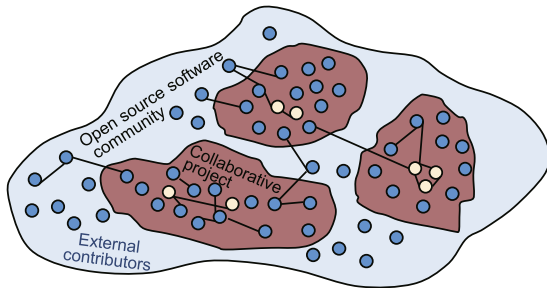
As a typical and most complex crowd intelligence process, OSS development heavily relies on a massive number of developers participating in a distributed collaboration to create software innovations.

How to organize such autonomous crowds to achieve efficient collaboration and effective intelligence emergence is quite a challenging problem. Most successful open source projects and communities suggest a similar type of organization that is not completely liberal, but with a hierarchical institution embedded. In this subsection, we introduce the crowd organization through open source practice.

##### 4.1.1 Organization structure of open source crowds

Although a strict hierarchical structure does not exist in OSS communities, developers playing different roles are collaborating with each other under the ‘onion’ structure (Ye and Kishida, 2003); i.e., a project team consists of a small but strongly organized core team, and a large-scale but unorganized periphery of contributors (i.e., the crowds), as shown in Fig. 7. Generally speaking, there are eight roles

(Nakakoji *et al.*, 2002) in the communities. In this paper, we summarize the five key roles as follows:



**Fig. 7 Collaborative structure in open source software communities**

The project leader/owner has initiated the project, and takes the responsibility for the development direction and vision of the project.

The core member is responsible for guiding, managing, and coordinating the developments of the project. A core member usually has been involved with the project for a relatively long time and made significant contributions.

The active contributor finishes most of development tasks in the project, such as contributing new features and fixing bugs.

The peripheral contributor occasionally makes contributions to the project, such as submitting a code patch or reporting a bug to the projects.

The watchers/readers/users have not directly contributed to the project, but they are the project users by reusing the code, modules, or the whole software. Also, because of the high quality of an OSS project, they try to deeply understand how the project works by analyzing its architecture and design pattern, or learn how to program by reading the source code.

From the social perspective, crowd contributors would spontaneously gather into different sub-communities in terms of their social activities. By constructing and mining follow-networks, the typical patterns of social behavior have been investigated (Yu *et al.*, 2014). For example, a famous developer is followed by a large number of users, but almost never follows back to others, which is called star-pattern (Yu *et al.*, 2014). Bird *et al.* (2006; 2008) confirmed the consistency between social behavior and technical collaboration behavior, by mining email social network and development activity history.

#### 4.1.2 Governance structure of open source crowds

Crowd intelligence usually emerges in certain online organizations and communities. However, the autonomy of the crowds requires effective organizational mechanisms to ensure the outcome of crowd creation.

In crowd software development, especially in successful OSS projects, their governance mechanisms play a very crucial role. A software foundation is commonly adopted as an effective governance institution in open source communities.

As one of the most famous open source organizations, the Apache communities are governed by the Apache Software Foundation (ASF), which embodies the principle of meritocracy (Castaneda, 2010), and forms community-driven governance known as the Apache way. The governance structure in the Apache community consists of two types of hierarchy: corporate governance and technical governance.

Corporate governance includes roles of ASF member, board of directors, executive and corporate officer, and project management committee (PMC), which makes ASF work as a corporation. Corporate governance is fairly simple: the ASF members elect a board of directors, the board appoints various officers and creates PMCs, the PMCs report periodically to the board, and most other officers report through the president, and then to the board.

Technical governance includes roles of commit-ter and PMC. PMC manages the Apache projects independently using the Apache Way, and determines the general direction (especially the technical direction) of the project and its releases. Committees are elected for every project and granted write access to it.

These different roles are chosen by self-selecting individuals, and they are authorized with power according to their sustained contributions over time (Erenkrantz and Taylor, 2003). Such hierarchical governance with different roles in Apache has proved to have a direct influence on code production (Mockus *et al.*, 2002).

Many other OSS like Linux, Eclipse, and Open-Stack establish their foundations to govern the corresponding communities for crowd creation. These institutions often have a hierarchical structure, and provide similar functions like determining technical directions, providing legal frame-

works, and business supports. Such governance is of great importance for inspiring and guiding the continuous emergence of crowd intelligence in a promising direction, and forms a prosperous creation ecosystem.

## 4.2 Incentive mechanism in crowd intelligence

A well-designed incentive mechanism is an effective way to degrade the uncertainty in crowd intelligent emergence. We introduce monetary and non-monetary incentive mechanisms in this subsection.

### 4.2.1 Monetary-driven incentive mechanisms in crowd intelligence

Most monetary incentive mechanisms are designed based on the auction theory. We first introduce the preliminaries of an auction-based mechanism. Then the relationship between auctions and monetary incentive mechanisms is discussed. Finally, we survey some representative studies of auction-based incentive mechanisms for crowd intelligence emergence.

#### 1. Preliminaries of auction-based mechanisms

Auction theory (Krishna, 2009) is a branch of game theory, which studies the properties of auctions and human behavior in auction markets. Briefly, an auction is the process of buying and selling goods by negotiating the monetary prices of the goods. In the process of an auction, based on the bids submitted by the users (i.e., buyers or sellers), the auction selects a subset of users as winners and determines the payment of each winning user. In particular, auctions have two properties: (1) An auction is a decentralized market mechanism for allocating resources and may be used to sell any item; i.e., auctions are universal. (2) An auction does not depend on the identities of the bidders, so auctions are anonymous.

The existing research on auctions can be classified roughly into two categories, i.e., regular auctions and reverse auctions. In regular auctions, a group of buyers bid on an item sold by a seller, and the highest bidder wins the item. Conversely, in reverse auctions, a buyer offers a reward for executing a task, and a group of sellers offer bids to complete the task. During the reverse auction progress, the price decreases due to the competition of the sellers, and thus the task is finally assigned to the lowest bidder(s). Reverse auction is more appropri-

ate for modeling the negotiation process in crowd intelligence, and many monetary incentive mechanisms based on reverse auctions have been proposed (Lee and Hoh, 2010; Jaimes *et al.*, 2012; Krontiris and Albers, 2012; Yang *et al.*, 2012; Subramanian *et al.*, 2013; Yang *et al.*, 2013; Feng *et al.*, 2014; Luo *et al.*, 2014; Zhao *et al.*, 2014; Zhu *et al.*, 2014; Gao *et al.*, 2015; Han *et al.*, 2016). Therefore, we focus only on discussing the reverse-auction based incentive mechanisms for crowd intelligence.

#### 2. Emergence of the relationship between auction and crowd intelligence

Auctions have been extensively used in crowd intelligence emergence, because they can effectively model the economic behaviors between crowd intelligent platforms and their users. Specifically, on the one hand, the participants in the crowd intelligent systems earn rewards using their intelligence to complete the tasks. In other words, these participants can be considered as selling their intelligence. As discussed above, since auctions are universal for any trade, the auction theory can be used to design a monetary incentive mechanism to provide reasonable and acceptable prices for crowd intelligence. On the other hand, auctions are anonymous, and crowd intelligent emergence platforms often recruit anonymous participants only to contribute their intelligence and finish the tasks. In addition, most existing studies usually believe that the following four characteristics should be considered when a reasonable and robust auction-based incentive mechanism is designed under crowd intelligence emergence scenarios.

**Truthfulness:** A bidder cannot improve her/his utility by submitting a bidding price different from its true valuation, no matter how others submit.

**Individual rationality:** Each bidder can expect a nonnegative profit.

**System efficiency:** An auction is system-efficient if the sum of the valuations of all the winning bids is maximized over the possible sets of valuations.

**Computation efficiency:** The outcome of the auction, i.e., the set of winners among all bidders, can be computed in polynomial time.

Specifically, truthfulness can improve the stability of the platform. Particularly, truthfulness can prevent market manipulation and guarantee auction fairness and efficiency. If auctions are untruthful,

selfish bidders may manipulate their bids to trick the system and obtain outcomes that favor themselves but hurt others. However, in truthful auctions, the dominant strategy for bidders is to bid truthfully, thereby eliminating the threat of market manipulation and the overhead of strategizing over others. With true valuations, the auctioneer can allocate tasks efficiently to the buyers who value it the most. Individual rationality guarantees that every participant can gain nonnegative profits, and thus more participants will be attracted to the platform. System efficiency is also important, because it guarantees that crowd intelligent platforms can run healthily. Last but not the least, computation efficiency requires that the computation of the platforms is efficient, which is necessary to provide a satisfactory user experience.

### 3. Representative studies of auction-based incentive mechanism

There are many auction-based incentive mechanisms that have been proposed to motivate the participants in crowd intelligent platforms. Most studies are based on reverse auctions. Based on the characteristics of different crowd intelligent platforms, different auctions, such as online auction and double auction, are applied to design incentive mechanisms.

**Basic reverse auction:** In the context of crowd intelligence, the platform which plays the role of buyer distributes tasks. The participants who play the role of sellers will offer their own bids representing their expectation for the rewards. The platform finally may choose a subset of participants with lower bids to finish the task. Lee and Hoh (2010) first applied reverse auction into incentive mechanism design. In the auction, there is an auctioneer (buyer) who wants to purchase  $m$  sensing data, and a group of crowd participants (sellers) who use their mobile devices to collect the sensing data. As the auction progresses, the price decreases as participants (sellers) compete to offer lower bids than their competitors. The mechanism is designed to minimize and stabilize the bid price while maintaining an adequate number of participants, in case some participants who were lost in previous auction rounds drop out. To attain such goals, the authors proposed a novel reverse auction based dynamic price incentive mechanism with the virtual participation credit (RADP-VPC). Specifically, the participants lost in the previous auction round will receive vir-

tual participation credit. In the current round, the bid price for a participant (used in price competition) is the participant's real bid price minus its VPC; i.e., in auction round  $r$ , the competition bid price for participant  $i$  is  $b_i^{r*} = b_i^r - v_i^r$ , where  $b_i^r$  represents the participant's real bid price, and  $v_i^r$  represents its VPC. Such credits can increase the winning probability of the participant without affecting its profit, since the platform will pay the participant its real bid price. By introducing VPC to the mechanism, many issues can be solved.

Jaimes *et al.* (2012) pointed out that an incentive mechanism should also consider the participants' location information, budget, and coverage constraints, which will make the mechanism more realistic and efficient. By analogy with the set cover problem, they formulated their problem as a budgeted maximum coverage problem, which can be regarded as an extension of the set cover problem. Then they tackled their problem by designing a new greedy incentive algorithm (GIA) that uses the existing techniques to address the classical set cover problem.

Yang *et al.* (2012) considered the fundamental property of truthfulness in incentive mechanism design, and proposed a mechanism satisfying such a property, called MSensing. MSensing consists of two phases: winner selection and payment determination. Both phases follow a greedy approach, since the objective function in their specific problem has the property of submodularity, which guarantees a good approximation when the algorithm uses a greedy strategy.

**Online auction:** As the crowd participants may be available at different times, the mechanism should handle the dynamic arrivals of participants. In the online auction, participants arrive in a sequence and the platform must decide immediately whether to choose the current participant or not. Therefore, it is desirable to assume that the mechanism which processes the current bids without knowing the future bids can still have good performance, which means that the objective function value of the online algorithms can be at most a factor of the offline optimal solution with the same input.

Han *et al.* (2016) considered for the first time the mechanism design of an online auction. Specifically, they tackled the problem called mobile crowdsensing scheduling (MCS), and then proposed algorithms

under both the offline and online settings with an optimal performance. Following their work, Subramanian *et al.* (2013), Zhao *et al.* (2014), and Zhu *et al.* (2014) explored online incentive mechanism design with different assumptions and goals.

**Multi-attribute auction:** As the sensing data submitted by the participants is not all of the same quality, the auctioneer (buyer) may have a preference. Krontiris and Albers (2012) proposed the multi-attribute auction (MAA) mechanism that not only considers the negotiated price, but also helps the auctioneer (buyer) select the sensing data of the highest quality. In fact, MAA can be regarded as an extension of the traditional reverse auction.

Specifically, the auctioneer will express its own preferences in the form of a utility function. The function takes the bid price and multiple quality attributes as the input, and translates the value of each attribute into a utility score. Suppose the  $i$ th attribute of bid  $x$  is denoted by  $x_i$ . The score is calculated using  $S(x) = \sum_{i=1}^n w(i)S(x_i)$ , where  $w(i)$  denotes the weight toward the  $i$ th attribute. Bids with higher utility scores will win the auction. The authors also give advice about other attributes: location distance, location accuracy, user credibility, etc.

**Combinatorial auction:** In real scenarios, a platform usually distributes multiple tasks which the crowd participants can finish if these tasks appear in the service coverage of the participants. Feng *et al.* (2014) proposed a combinatorial auction mechanism, which allows the participants to bid for a combination of the tasks, and assigns it to a participant. However, it is still an extension of a reverse auction.

By formulating the problem as an optimization problem, they first proved the NP-hardness for the problem and then embedded an approximation algorithm with an approximation ratio of  $1 + \ln n$ , where  $n$  denotes the number of sensing tasks, into the proposed mechanism called truthful auction (TRAC). They demonstrated many desired properties of TRAC by theoretical analysis and extensive experiments.

**All-pay auction:** In the definition of the all-pay auction, every bidder needs to pay the bid they propose, while only the bidder with the highest bid gets the prize. Luo *et al.* (2014) studied the incentive mechanism problem based on the model of an all-

pay auction, which is the first work using an all-pay auction in crowd intelligence. In particular, different from a winner-pay auction, an all-pay auction requires all the participants make their contribution to the task, while only the participant with the highest contribution gets the prize. Thus, in the setting of an all-pay auction, to maximize the contribution collected from users, the organizer needs to choose a prize as the function of the winner's contribution. In addition, Luo *et al.* (2014) considered that users may prefer a smaller but more stable prize rather than taking the risk to lose even if they have the same expected prize, which is called being risk-averse. For all-pay auctions and risk-averse, Luo *et al.* (2014) proposed a mechanism to solve the profit maximization problem. The mechanism guarantees a positive payoff for the participants, which means that it is effective.

**Double auction:** Unlike the reverse and all-pay auctions in which a single buyer exists, a double auction contains more than one buyer and one seller. Because of this property, the double auction has also been used as an incentive mechanism of crowd intelligence emergence.

Yang *et al.* (2013) studied the problem of incentive mechanism design for a  $k$ -anonymity location privacy. In this privacy,  $k - 1$  other mobile users as well as the exact user form a cloaking area to protect the location privacy. However, not all the users are willing to share their locations without a prize. Yang *et al.* (2013) designed three incentive mechanisms based on the model of double auction to motivate mobile users to join the  $k$ -anonymity process. In particular, the mechanisms were designed considering three kind of users: (1) users who have the same privacy degree, (2) users who have different privacy degrees, and (3) users who can cheat. Further study shows that the proposed mechanisms all satisfy the following properties: (1) computational efficiency; (2) individual rationality; (3) truthfulness.

**Vickrey-Clarke-Groves auction:** In addition to the above auctions, the Vickrey-Clarke-Groves (VCG) (Vickrey, 1961) auction proposed recently is also a useful incentive mechanism. In the VCG auction, the loss of a bidder results in other bidders paying this bidder. Gao *et al.* (2015) studied the sensor selection problem in time-dependent and location-aware participatory sensing systems under both future and current information scenarios. To



solve the online (current information) asymmetric sensor selection problem, they proposed an incentive mechanism based on the VCG auction. This study also showed that the mechanism is truthful and converges asymptotically to the optimal offline performance.

#### 4.2.2 Community-driven incentive mechanism in crowd intelligence

##### 1. Incentive mechanism in software development communities

OSS is usually developed by a loosely-knit community of developers distributed across the world, who contribute to a software project via the Internet without necessarily being employed or paid by an institution. The development communities of OSS have been likened to a bazaar where everyone can join and contribute (Raymond, 1999), creating an inspiring, creative, and democratic atmosphere. Following this model, hundreds of thousands of high-quality and successful pieces of software have been produced by a large number of voluntary developers collaborating with each other.

Most research on OSS communities focuses on discovering what motivates people to join OSS even without direct compensation for the work. Starting from internal factors, Hars and Ou (2001) argued that open-source programmers are not motivated by monetary incentives, but by their personal hobbies and preferences, which confirms the need for self-actualization, i.e., the “desire for a stable, firmly based, usually high evaluation of oneself” (Maslow *et al.*, 1970). Furthermore, collaborating in OSS communities, contributors treat other participants of the community as kin, and thus are willing to do something that is beneficial for them but not for themselves. Altruistic behavior of this type is called the ‘kin-selection altruism’ by social psychological researchers (Hoffman, 1981).

For external reward, communities can provide a rapid, constructive feedback about the quality of contributors’ work. This feedback mechanism would encourage contributors to expend additional effort to perfect their code, so they can attract more favorable feedback in turn. Similarly, the survey of free/libre and open source software (FLOSS) (Ghosh, 2005) pointed out that 53.2% of the developers express social or community-related motives, such as sharing knowledge and skills. In the long run, the abilities

and skills of OSS contributors can undergo significant improvement, which leads to better job opportunities and higher salaries (Hertel *et al.*, 2003).

##### 2. Incentive mechanism in software knowledge sharing communities

Crowds of enthusiasts and experts are becoming an integral part of the support network for online communities, and knowledge sharing is a typical activity in crowd intelligence. The motivations for crowds spending time and effort to contribute knowledge can be either intrinsic or extrinsic. The intrinsic motivations include the sense of personal ownership of one’s knowledge, self-interest, enjoyment, and feelings of gratitude and respect. The extrinsic motivations are these factors that are outside the individual, such as community reputations, ratings, and monetary reward (Raban and Harper, 2008). To inspire the crowds to contribute their experience and knowledge, such motivations should be transferred into an effective incentive mechanism, which is of great importance for the emergence of crowd intelligence. In practice, various non-monetary mechanisms are widely employed on the online information sites.

The largest programming Q&A site, StackOverflow, is a typical crowd intelligence community which earns its place in software development by high-quality answers and fast response. The incentive system consists mainly of reputation and privilege hierarchy, badge hierarchy, and voting and ordering mechanism.

**Reputation:** Reputation hierarchy represents how much a user is trusted by the community. It is measured by reputation points which are earned by convincing peers through asking questions, answering, commenting, editing, etc.

**Privilege:** Privilege hierarchy controls what a user can do in the site and corresponds to the user’s reputation. For example, voting others’ posts, editing others’ questions and answers, accessing to site analytic, and other activities that require different numbers of reputation points. The more reputation points a user gains, the more privilege the user will have.

**Badges:** Badges are another honor for users who are viewed as being especially helpful in the community. Badges will appear on the profile page, flair, and posts to motivate users to post high-quality content.

Voting and ordering: The quality of posted questions and answers is measured by their readers, who can vote up or down a post if they view it as useful or not, and the authors will get or lose corresponding reputation points. Accordingly, the questions and answers are ordered by the number of votes and the activeness. The more votes a post gets, the more chances it will be read, and the more reputation points the author will get.

The incentive system in StackOverflow effectively motivates users to actively participate and provide high-quality knowledge. It attracted more than 6.4 million registered users who contributed more than 33 million technical posts, and achieved an over 92% answer rate and about 11 minutes of the first answer time, which is much better than other similar sites (Mamykina *et al.*, 2011).

In summary, the success of most knowledge sharing communities like StackOverflow relies mainly on the powerful incentive mechanisms. It combines reputations, interests, and competitions effectively, and forms a productive loop such that high-quality and quick responses attract more crowds to take part in, while this in turn provides more knowledge. However, how to motivate the crowds to continuously engage in is still a great challenge for a non-monetary incentive system. As has been experienced in StackOverflow, developers who earn enough reputation points may reach a plateau and experience a subsequent reduction in participation. Thus, such external motivations like reputation are only part of the consideration in designing the incentive system. How to discover the crowds' intrinsic needs and introduce them into the incentive mechanism reasonably should be the great research issue.

### 4.3 Quality control of crowd intelligence

Due to the diverse nature of crowd tasks, different kinds of crowd applications need to adopt different quality control methods. In this subsection, we introduce methods of quality control for data processing, decision making, and software innovation.

#### 4.3.1 Crowd intelligence for data processing

The aim of quality control for data processing is to obtain the most useful and reliable data with the minimum cost. To achieve this goal, we need methods derived from multiple points of view. First,

when only low-quality data is available, we need to be able to extract the correct information from that. Second, better crowdsourcing mechanisms can be designed. Moreover, fundamental questions on what can be done in crowd intelligence need to be addressed. These aspects are briefly summarized below:

1. Learning correct labels from crowd-sourced labels

Often, only low-quality data is collected from crowd workers. In this case, we have to extract the correct information from the crowd-sourced data. A frequently considered situation is that workers give labels for the data to extract a supervised model from the data and the labels. Low-quality data means that the labels may not be correct. To address this problem, the most popular methods are based on the Dawid-Skene model (Dawid and Skene, 1979) and inference methods such as the expectation maximization (EM) algorithm (Dempster *et al.*, 1977). The Dawid-Skene model is a probabilistic model on how a crowd of workers label the tasks. It assumes that the performance of each worker is independent of the specific task, and that the labels given by the workers are conditionally independent of each other given the ground-truth label. In this case, each worker corresponds to an unknown confusion matrix. Then, the EM algorithm is used to infer the confusion matrices and the ground-truth label of each task. Similar to Dawid and Skene (1979), Raykar *et al.* (2009; 2010) used a two-coin model to model the ability of each worker. They assumed that the ground-truth labels of the tasks were generated by an unknown linear classifier. Then, a Bayesian method based on the EM algorithm was used to infer the confusion matrices and the parameters of the linear classifier. In Whitehill *et al.* (2009), a model considering both worker quality and task difficulty with the conditional independence assumption was proposed, and the most probable label can be inferred based on the EM algorithm. Subsequently, Welinder *et al.* (2010) introduced a generalized probabilistic model with a high-dimensional formulation representing task difficulty, worker quality, and worker bias. Liu *et al.* (2012) transformed the crowdsourcing problem into a graph-based variational inference problem. The labels can be inferred with tools such as belief propagation and mean field. However, the usefulness of the method largely depends on

the prior knowledge of the worker's reliability, and the mean field (MF) form of their method is closely related to EM. A major drawback of the methods based on the EM algorithm is the lack of guaranteed convergence. In Zhang Y *et al.* (2014), an alternative algorithm combining EM and a spectral method was proposed, which uses EM for model initialization, and uses the spectral method for parameter tuning to guarantee the convergence.

The minimax entropy principle has also been introduced into crowdsourcing. Zhou *et al.* (2012) assumed that the labels were generated by a probability distribution over the workers, tasks, and labels. The task difficulty and the worker performance are inferred by maximizing the entropy of this distribution, and the ground-truth labels are inferred by minimizing the entropy of the distribution. Tian and Zhu (2015) presented a max-margin formulation of the most popular majority voting estimator to improve its discriminative ability, which directly maximizes the margin between the aggregated score of a potential true label and that of any alternative label. Yan *et al.* (2011) combined the active learning with crowdsourcing. In their work, a probabilistic model providing the criterion for selecting both the task and worker was proposed. When the crowd is dominated by spammers, an empirical Bayesian algorithm based on EM was proposed in Raykar and Yu (2012) for iteratively estimating the ground-truth label and eliminating the spammers. Traditionally, the label inference process and the machine learning process with the inferred labels are treated separately in crowdsourcing. To unify both steps, Wauthier and Jordan (2011) proposed a Bayesian framework under the name of Bayesian bias mitigation.

2. Designing smart crowdsourcing mechanisms for saving the budget

In a crowd intelligent system, each task is presented to multiple workers and each worker is also presented with multiple tasks. The system must decide to assign which tasks to which worker. In Karger *et al.* (2011), a task assignment algorithm was proposed. Karger *et al.* (2011) used the Dawid-Skene model to characterize the tasks and workers, and further formulated task assignment into edge assignment on random regular bipartite graphs. Khetan and Oh (2016) generalized the result into the generalized Dawid-Skene model which considers the heterogeneous tasks with different difficulties. In

Ho and Vaughan (2012) and Ho *et al.* (2013), the problem of assigning heterogeneous tasks was solved with an online primal-dual technique achieving near-optimal guarantees. Chen *et al.* (2013; 2015) formulated the budget allocation of crowdsourcing into a Markov decision process, and characterized the optimal policy using dynamic programming. They proposed an efficient approximation based on the optimistic knowledge gradient method.

Besides task assignment, more crowdsourcing mechanisms have been proposed to reduce the budget. In Shah and Zhou (2015), when labeling the tasks, the workers were allowed to skip the question if their confidence in their answer was low. In Wang and Zhou (2015), conditions had been derived for when allowing the skipping of questions is useful using an 'unsure' option. In Zhong *et al.* (2015), it was observed that the workers labeling multiple labels, instead of one label per object, tend to act in an effort-saving behavior; i.e., rather than carefully checking every label, they would prefer just scanning and tagging a few most relevant ones. A designated payment mechanism was proposed to incentivize the worker to select the correct answer based on his/her true confidence. They proved that the mechanism is the uniquely incentive compatible one under the 'no free lunch' assumption. Similar results were extended to allow workers to select multiple possible answers (Shah *et al.*, 2015) or to correct their answers (Shah and Zhou, 2016). Furthermore, Cao *et al.* (2012) adopted the Poisson binomial distribution to model the uncertainty that a crowd of workers complete a task correctly, and used the various uncertain mining techniques to discover the set of reliable workers (Cao *et al.*, 2012; Tong *et al.*, 2012a; 2012b; 2015a; 2016d).

3. Theoretical guarantees

More fundamental questions of crowdsourcing need to be addressed. Gao and Zhou (2013) established the minimax optimal convergence rates of the Dawid-Skene model, and proposed a nearly optimal projected EM method. Wang and Zhou (2016) emphasized the importance of model performance and bridged the gap between label cost and model performance by an upper bound of the minimal cost for crowdsourcing. They also disclosed the phenomenon that the label performance is not theoretically and empirically the higher the better. Li and Yu (2014) showed the error rate bounds from the

view of voting, and proposed an iterative weighted majority algorithm, which outperforms majority voting empirically and theoretically. In Ok *et al.* (2016), the belief propagation algorithm was proved to be optimal when each worker is assigned at most two tasks. Dekel and Shamir (2009) considered the case where each task is labeled by only one worker. They presented a two-step method to prune low-quality workers in the crowd. First, a classifier is trained using all the training data. Then, the workers who provide the label very differently from the predictions of the classifier are identified as of low quality. The labels provided by these workers are deleted before a new classifier is trained. They also presented the theoretical analysis for this two-step process. Tran-Thanh *et al.* (2012) proposed a multi-arm bandit algorithm for crowdsourcing with limited budget. They used a small portion of the budget to estimate the worker's quality, and maximized the utility of the remaining budget based on these estimations. An upper bound on the regret for the algorithm was derived. Abraham *et al.* (2013) considered a special crowdsourcing problem, called the bandit survey problem, in which each worker can be treated as a bandit arm. In this work, multi-arm and Lipschitz bandit algorithms were used to solve the bandit survey problem with theoretical guarantees. In Jain *et al.* (2014), bandit setting was employed to deal with task assignment. In Zhou *et al.* (2014), bandit arm identification was employed for worker selection.

#### 4.3.2 Crowd intelligence for decision making

Decision making is an important activity for human beings. It accompanies us from our daily life to the activity of changing nature and our society. Traditional decision-making is limited to a single individual, or is in the form of a small group of domain experts. It does not consider the challenges in open environments where the influencing factors are unclear, the information is dynamically changing, or the feedback is not on time. As a consequence, traditional decision making is unable to effectively solve challenging problems in open environments and to make correct decisions.

Crowd intelligence opens up new ways for more informed decision making. We can mine the massive pieces of the information provided by millions of crowd participants on the Internet, to analyze the sensible and computable factors that influence

the activity and credibility of individual crowd participants, to build corresponding theoretical models, and to design methods for informed decision making by aggregating diverse information from crowd participants.

For example, crowd intelligence has been used to decide when and where to dispatch resources and rescuers after earthquakes (Sakaki *et al.*, 2010), which flight to take to avoid delay (Li *et al.*, 2012), where to obtain living resources after a typhoon (Wang *et al.*, 2014), when to visit a target place to avoid crowdedness or long waiting lines (Ouyang *et al.*, 2015), and which route to take when traveling (Dantec *et al.*, 2015).

However, there are also significant challenges. First, information about the same target provided by different crowd participants may be conflicting. However, the reliability of each crowd participant is unknown a priori. Moreover, crowd participants may be affected by each other through social interaction (e.g., simply retweet others' tweets). It is difficult to decide which information to trust. Thus, decision making faces significant risks without any risk-controlling mechanism. Second, the number of crowd participants is huge. Processing all the pieces of information by these participants is unnecessary, and may reduce the decision accuracy due to the involved noises. Selecting a small set of reliable crowd participants for decision making will possibly further increase the decision accuracy and reduce the cost. Third, the information source for crowd decision making is not diversified and typically not well calibrated. Consequently, it is challenging to design an appropriate model to combat the uncertainty caused by diverse information sources, particularly for the prediction-oriented decision making tasks.

Therefore, the following research problems need to be addressed when leveraging crowd intelligence for decision making: (1) risk controlling in decision making given diverse, conflicting, and dependent information; (2) selection of a small set of reliable information sources from a tremendous number of crowd participants; (3) determining the reliability or credibility of the information source for decision making.

##### 1. Risk controlling in decision making

Risk controlling in decision making aims to make the most accurate decision whose risk (e.g., error or cost) is the lowest.

Researchers from the University of Tokyo, Japan proposed a temporal model and a spatial model, for accurately determining when and where an event (e.g., earthquake) happens for rescue deployment from diverse and conflicting crowd-provided information (e.g., tweets) (Sakaki *et al.*, 2010). They modeled the event occurrence time using an exponential distribution, and modeled that each crowd participant has a false positive rate. After parameter learning from data, they calculated the probability of an event occurrence at a given time  $t$ .

To determine the event location, they view the location over time as a state sequence  $\{x_t\}$  and  $x_t = f_t(x_{t-1}, u_t)$ , where  $u_t$  is a process noise sequence. They then explore the Kalman filter and the particle filter to determine the location of an event at time  $t$  (e.g., the location of an earthquake center or the trajectory of a typhoon). Through experiments, they showed that 96% of the earthquakes in Japan Meteorological Agency (JMA) seismic intensity scale three or more can be accurately detected merely by monitoring tweets from crowd participants. Notification and rescue decisions are made much faster than those broadcasted by JMA. The average error in event location estimation is around 3 km.

Researchers from the University of Illinois at Urbana Champaign (UIUC) proposed a method for accurate decision making that takes the social relationships of crowd participants into consideration (Wang *et al.*, 2014). They modeled that each crowd participant  $i$  has two reliability parameters,  $a_i$  and  $b_i$ . The former is the probability that an event is true and the crowd participant also claims that the event is true, while the latter is the probability that an event is false but the crowd participant claims that the event is true. They also modeled that each crowd participant  $i$  has a probability  $p_{ig}$  to follow another participant  $g$ .

Based on the follower–followee relationships in social networks, they partitioned the crowd participants involved in each event into several groups, where group members retweet the group leader’s information. Each group leader is considered to be independently providing information (i.e., using parameters  $a_i$  and  $b_i$ ), while group members are considered to copy the group leader’s information (i.e., using parameter  $p_{ig}$ ). The event label (i.e., true or false) was modeled as a latent variable  $z_j$ .

They then jointly learned these parameters and inferred the event label from the observed data by maximizing the data log likelihood. In particular, they solved the maximization problem via the EM algorithm. They conducted experiments to determine the trace of hurricane Sandy by using crowd intelligence. Experimental results showed that their method can reach up to 90% accuracy in decision-making, while methods that do not consider social relationships can lead to only 60% accuracy.

Researchers from the University of California, Los Angeles (UCLA) (Ouyang *et al.*, 2015) proposed a method for accurate decision making from quantitative information, different from Wang *et al.* (2014) who worked on binary information. They modeled that each crowd participant has a bias parameter and a confidence parameter when providing a piece of quantitative information (e.g., occupancy rate in a gym or a classroom), and such information is noise-centered on the true but latent quantity value. Moreover, each latent true quantity value is associated with a difficulty level. They then built a unified graphical model to account for all these variables. They also used the EM algorithm to jointly learn all the parameters and latent variables based on the observed data. The errors of inferred quantity values were shown to be within 10%.

Researchers proposed a method to make accurate decisions with streaming data in Wang D *et al.* (2013) and to make accurate decisions for planning tasks in Dantec *et al.* (2015).

2. Selection of a small set of reliable information sources

As the number of crowd participants is huge and the information provided by them is noisy, selecting a small set of reliable crowd participants for decision making will further increase the accuracy.

Researchers from the AT&T Research Lab (Dong *et al.*, 2012) proposed a method for wisely selecting information sources. They formulated the problem as selecting the set of information sources that maximize the decision accuracy. They modeled that each information source has an accuracy parameter, and proposed a method that can estimate the accuracy of the fused decision, based on the accuracy of individual information sources. They then presented an algorithm that applies the greedy randomized adaptive search procedure (GRASP) meta-heuristic to solve the source selection problem.

Experimental results showed that selecting a subset of reliable information sources leads to even higher decision accuracy.

Researchers from the University of Fribourg (Difallah *et al.*, 2013) built profiles for each individual crowd participant, and then selected the most appropriate participants for each decision task at hand. They built profiles in three different ways, namely, category-based, expertise-based, and semantic-based. Through experiments, they observed that selecting a subset of participants can indeed increase the task accuracy. They then proposed a scheduling workflow in a subsequent study (Difallah *et al.*, 2016).

Researchers from the Université de Rennes 1, IRISA (Mavridis *et al.*, 2016) used hierarchical skills for selecting a small set of participants for each task. They modeled the skills of each crowd participant using a skill taxonomy, and proposed a method to compute the skill distance between the required skill and the skills of participants. Then, given a task, only the first few participants whose skill distances to the required one are the smallest are selected. They tested several algorithms for participant selection, including Random, ExactThenRandom, MatchParticipantFirst, ProfileHash, and HungarianMatch. They showed that the latter four algorithms result in higher decision accuracy than random ones.

### 3. Credibility of information source for decision making

For crowd decision making, the first task is to evaluate the credibility of the information source for crowd decision making, particularly for prediction-oriented decision making. For example, the failure of predicting the presidential election of the USA demonstrates that the source of information could significantly bias the prediction results.

Google Flu Trends (GFT) is a flu tracking system, developed based on the query log of their search engine. GFT achieves a much earlier prediction about the scope of flu in the USA. However, as reported, GFT generally overestimates the proportion of doctor visits for influenza-like illness than Centers for Disease Control and Prevention (Lazer *et al.*, 2014). Scientists attribute the failure of GFT to the low reliability of the information source. Shen and Barabási (2014) designed a collective credit allocation method to identify the intellec-

tual leaders of each multi-author publication. Different from existing methods that use the author list as the information source, the proposed method uses co-citation as a reliable information source, achieving a much higher accuracy at identifying the Nobel laureate from his prize-winning papers. Bollen *et al.* (2011) investigated collective mood states derived from large-scale Twitter feeds to predict the Dow Jones industrial average (DJIA). The proposed method achieves an accuracy of 86.7% in predicting the daily up and down changes in the closing values of DJIA. The success of this method also indicates the importance of a credible information source.

### 4.3.3 Crowd intelligence for innovation

Innovation was defined by Rogers (2010) as “an idea, practice, or object that is perceived as new by an individual or other unit of adoption”. OSS development is actually a continuous process for crowd innovation aggregation. The gathering and evaluation of contributions from the crowds is a key process in the emergence of crowd intelligence.

#### 1. Contribution gathering in software communities

OSS projects are primarily driven by community contribution, and contribution collection in software communities has become an important factor that affects the success of the software (Wang *et al.*, 2015).

GitHub designed a novel contribution model called a pull-based development model, which is widely used in distributed software development (Gousios *et al.*, 2015; Vasilescu *et al.*, 2015). Using this model, external contributors can easily contribute their changes to software without accessing the central repository. The core team members can merge these external contributions by processing corresponding pull-requests. Such a paradigm offers much convenience for collaborative development and greatly promotes the efficiency of contribution collection (Gousios *et al.*, 2015). Many researchers have studied the contribution gathering mechanisms. Gousios *et al.* (2015) studied the practice of pull-based software development and its challenges from a contributor’s perspective. Yu *et al.* (2016a) leveraged a regression model to quantitatively analyze what factors are determinants of pull-based development. They found that continuous integration is a dominant factor in the pull-request process.

Some researchers analyzed the contribution collection from the perspective of openness. Openness reflects how easy it is for a new developer to actively contribute to a software community. They have proposed three metrics to evaluate the openness: the distribution of the project community, the rate of acceptance of external contributions, and the time it takes to become an official collaborator on the project. These metrics have been applied on a dataset of GitHub projects. Project members and owners can use these metrics to improve the attractiveness of their projects, while external contributors can rely on these metrics to identify those projects that better fit their development needs (de Alwis and Sillito, 2009).

Some others focus on studying mechanisms for accelerating the contribution gathering process. Yu *et al.* (2016b) combined the social connections and technical factors, and designed a reviewer recommendation approach, which helps project managers find proper contribution reviewers. Many other similar works have been conducted in recent years (Rigby *et al.*, 2014; Thongtanunam *et al.*, 2015). They can be very valuable for promoting the efficiency and quality of contribution gathering. Tamrawi *et al.* (2011) paid attention to the automatic task assignment like bug triage, which aims to automatically find the right person to accomplish the right task (Jeong *et al.*, 2009; Bhattacharya and Neamtiu, 2010).

## 2. Innovation evaluation in software communities

Innovation evaluation is an indispensable step in ensuring the quality of crowd intelligence. As a typical innovation process, OSS development in communities is studied extensively for how to evaluate such innovations.

A lot of work has been done on measuring and predicting software quality by analyzing the structure of source code and its development process. Basili *et al.* (1996) and Subramanyam and Krishnan (2003) studied the relations between the complexity of source code and its defect density. Their empirical work found that, in object-oriented software, the CK metrics, like design complexity, are significantly associated with the software quality. Nagappan and Ball (2005) and Moser *et al.* (2008) analyzed the code differences between multiple versions, and found that

the change metrics can be good indicators for defect density prediction.

Some researchers studied software quality from the team behaviors' perspective. Rahman and Devanbu (2013) found that development behaviors can have great influence on software quality. Hassan (2009) used the entropy to quantify the complexity of code changes, and found it to be an important factor for introducing defects. Bird *et al.* (2011) and Rahman and Devanbu (2011) focused on studying the impact of the developer number in the team, and their experience and contribution rate on innovation quality.

In addition, some approaches have been proposed to evaluate the contributions from the community's perspective. In GitHub and many other software communities, project members often engage in extended discussions to evaluate whether an external contribution should be integrated (Tsay *et al.*, 2014b). In addition, Tsay *et al.* (2014a) studied the social and technical factors' influence on evaluating contributions in the GitHub community. They found that the strength of the social connection between contributors and the project managers, as well as the discussion density, are important factors for evaluating external contributions.

## 4.4 Future challenges

As an emerging area, there are some research challenges in crowd intelligence.

**Dynamic crowd organization:** Although the existing crowd organization techniques have already achieved good efficiency, there is little research on how to adjust the organization structure of crowd intelligence to cope with the variable external environment.

**Dynamic pricing:** The onetary incentive mechanism is important in crowd intelligence emergence. However, the supply and demand between the task requesters and workers often change. How to design an effective monetary incentive mechanism that can dynamically set a proper price for tasks, will be one of the research emphases in the future.

**Quality control on latency:** Current research of quality control mostly focuses on the assurance of the outcome of crowd intelligence. However, in some application scenarios such as mobile crowdsourcing, it is an important issue to control the latency of task

completion. This is also the research emphasis of quality control in crowd intelligence.

## 5 Conclusions

In this paper, we review extensive theoretical studies and industrial applications on crowd intelligence in AI 2.0 era. We first introduce some fundamental concepts of crowd intelligence, then illustrate various platforms of crowd intelligence, and finally, discuss the existing hot topics and the state-of-the-art techniques. In addition, we provide the potential challenges for future research. Specifically, the techniques of crowd organization and allocation, incentive mechanisms of crowd intelligence and quality control of crowd intelligence can be further improved. Since crowd intelligence is one of the core components of studies in AI 2.0 era, we hope that this paper can enlighten and help scholars working on the emerging topics of crowd intelligence.

## References

- Abraham, I., Alonso, O., Kandylas, V., et al., 2013. Adaptive crowdsourcing algorithms for the bandit survey problem. Proc. 26th Conf. on Computational Learning Theory, p.882-910.
- Ballesteros, J., Carbanar, B., Rahman, M., et al., 2014. Towards safe cities: a mobile and social networking approach. *IEEE Trans. Parall. Distr. Syst.*, **25**(9):2451-2462. <http://dx.doi.org/10.1109/TPDS.2013.190>
- Basili, V.R., Briand, L.C., Melo, W.L., 1996. A validation of object-oriented design metrics as quality indicators. *IEEE Trans. Softw. Eng.*, **22**(10):751-761. <http://dx.doi.org/10.1109/32.544352>
- Bhattacharya, P., Neamtiu, I., 2010. Fine-grained incremental learning and multi-feature tossing graphs to improve bug triaging. IEEE Int. Conf. on Software Maintenance, p.1-10. <http://dx.doi.org/10.1109/ICSM.2010.5609736>
- Bird, C., Gourley, A., Devanbu, P., et al., 2006. Mining email social networks. Proc. Int. Workshop on Mining Software Repositories, p.137-143. <http://dx.doi.org/10.1145/1137983.1138016>
- Bird, C., Pattison, D., de Souza, R., et al., 2008. Latent social structure in open source projects. Proc. 16th ACM SIGSOFT Int. Symp. on Foundations of Software Engineering, p.24-35. <http://dx.doi.org/10.1145/1453101.1453107>
- Bird, C., Nagappan, N., Murphy, B., et al., 2011. Don't touch my code!: examining the effects of ownership on software quality. Proc. 19th ACM SIGSOFT Symp. and 13th European Conf. on Foundations of Software Engineering, p.4-14. <http://dx.doi.org/10.1145/2025113.2025119>
- Bollen, J., Mao, H.N., Zeng, X.J., 2011. Twitter mood predicts the stock market. *J. Comput. Sci.*, **2**(1):1-8. <http://dx.doi.org/10.1016/j.jocs.2010.12.007>
- Bonabeau, E., 2009. Decisions 2.0: the power of collective intelligence. *MIT Sloan Manag. Rev.*, **50**(2):45-52.
- Borne, K.D., Zooniverse Team, 2011. The Zooniverse: a framework for knowledge discovery from citizen science data. American Geophysical Union Fall Meeting.
- Burke, J.A., Estrin, D., Hansen, M., et al., 2006. Participatory sensing. Workshop on World-Sensor-Web: Mobile Device Centric Sensor Networks and Applications, p.117-134.
- Cao, C.C., She, J.Y., Tong, Y.X., et al., 2012. Whom to ask? Jury selection for decision making tasks on micro-blog services. *Proc. VLDB Endow.*, **5**(11):1495-1506. <http://dx.doi.org/10.14778/2350229.2350264>
- Cao, C.C., Tong, Y.X., Chen, L., et al., 2013. Wisemarket: a new paradigm for managing wisdom of online social users. Proc. 19th ACM Int. Conf. on Knowledge Discovery and Data Mining, p.455-463. <http://dx.doi.org/10.1145/2487575.2487642>
- Castaneda, O.F., 2010. Hierarchy in Meritocracy: Community Building and Code Production in the Apache Software Foundation. MS Thesis, Delft University of Technology, Delft, Netherlands.
- Chen, X., Lin, Q.H., Zhou, D.Y., 2013. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. Proc. 30th Int. Conf. on Machine Learning, p.64-72.
- Chen, X., Lin, Q.H., Zhou, D.Y., 2015. Statistical decision making for optimal budget allocation in crowd labeling. *J. Mach. Learn. Res.*, **16**:1-46.
- Dantec, C.A.L., Asad, M., Misra, A., et al., 2015. Planning with crowdsourced data: rhetoric and representation in transportation planning. Proc. 18th ACM Conf. on Computer Supported Cooperative Work, p.1717-1727. <http://dx.doi.org/10.1145/2675133.2675212>
- Dawid, A.P., Skene, A.M., 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Appl. Statist.*, **28**(1):20-28. <http://dx.doi.org/10.2307/2346806>
- de Alwis, B., Sillito, J., 2009. Why are software projects moving from centralized to decentralized version control systems? Proc. ICSE Workshop on Cooperative and Human Aspects on Software Engineering, p.36-39. <http://dx.doi.org/10.1109/CHASE.2009.5071408>
- Dekel, O., Shamir, O., 2009. Vox Populi: collecting high-quality labels from a crowd. Proc. 22nd Conf. on Learning Theory.
- Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B*, **39**(1):1-38.
- Difallah, D.E., Demartini, G., Cudré-Mauroux, G.P., 2013. Pick-a-crowd: tell me what you like, and I'll tell you what to do. Proc. 22nd Int. Conf. on World Wide Web, p.367-374. <http://dx.doi.org/10.1145/2488388.2488421>



- Difallah, D.E., Demartini, G., Cudré-Mauroux, G.P., 2016. Scheduling human intelligence tasks in multi-tenant crowd-powered systems. Proc. 25th Int. Conf. on World Wide Web, p.855-865.  
<http://dx.doi.org/10.1145/2872427.2883030>
- Dong, X.L., Saha, B., Srivastava, D., 2012. Less is more: selecting sources wisely for integration. Proc. VLDB Endow., 6(2):37-48.  
<http://dx.doi.org/10.14778/2535568.2448938>
- Erenkrantz, J.R., Taylor, R.N., 2003. Supporting Distributed and Decentralized Projects: Drawing Lessons From the Open Source Community. ISR Technical Report No. UCI-ISR-03-4, Institute for Software Research, University of California, Irvine, USA.
- Farkas, K., Nagy, A.Z., Tomás, T., et al., 2014. Participatory sensing based real-time public transport information service. IEEE Int. Conf. on Pervasive Computing and Communications Workshops, p.141-144.  
<http://dx.doi.org/10.1109/PerComW.2014.6815181>
- Feng, Z.N., Zhu, Y.M., Zhang, Q., et al., 2014. Trac: truthful auction for location-aware collaborative sensing in mobile crowdsourcing. Proc. IEEE Conf. on Computer Communications, p.1231-1239.  
<http://dx.doi.org/10.1109/INFOCOM.2014.6848055>
- Fowler, G., Schectman, J., 2013. Citizen surveillance helps officials put pieces together. *The Wall Street Journal*, April 17. <http://www.wsj.com/articles/SB10001424127887324763404578429220091342796>
- Gao, C., Zhou, D.Y., 2013. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. ePrint Archive, arXiv:1310.5764.
- Gao, D.W., Tong, Y.X., She, J.Y., et al., 2016. Top-k team recommendation in spatial crowdsourcing. *LNCS*, 9658:191-204.  
[http://dx.doi.org/10.1007/978-3-319-39937-9\\_15](http://dx.doi.org/10.1007/978-3-319-39937-9_15)
- Gao, L., Hou, F., Huang, J.W., 2015. Providing long-term participation incentive in participatory sensing. IEEE Conf. on Computer Communications, p.2803-2811.  
<http://dx.doi.org/10.1109/INFOCOM.2015.7218673>
- Ghosh, R.A., 2005. Understanding Free Software Developers: Findings from the Floss Study. MIT Press, Cambridge, USA.
- Gousios, G., Pinzger, M., Deursen, A., 2014. An exploratory study of the pull-based software development model. Proc. 36th Int. Conf. on Software Engineering, p.345-355. <http://dx.doi.org/10.1145/2568225.2568260>
- Gousios, G., Zaidman, A., Storey, M.A., et al., 2015. Work practices and challenges in pull-based development: the integrator's perspective. Proc. 37th Int. Conf. on Software Engineering, p.358-368.  
<http://dx.doi.org/10.1109/ICSE.2015.55>
- Han, K., Zhang, C., Luo, J., et al., 2016. Truthful scheduling mechanisms for powering mobile crowdsensing. *IEEE Trans. Comput.*, 65(1):294-307.  
<http://dx.doi.org/10.1109/TC.2015.2419658>
- Hars, A., Ou, S., 2001. Working for free? Motivations of participating in open source projects. Proc. 34th Annual Hawaii Int. Conf. on System Sciences, p.1-9.  
<http://dx.doi.org/10.1109/HICSS.2001.927045>
- Hassan, A.E., 2009. Predicting faults using the complexity of code changes. Proc. 31st Int. Conf. on Software Engineering, p.78-88.  
<http://dx.doi.org/10.1109/ICSE.2009.5070510>
- Hertel, G., Niedner, S., Herrmann, S., 2003. Motivation of software developers in open source projects: an Internet-based survey of contributors to the Linux kernel. *Res. Polic.*, 32(7):1159-1177.  
[http://dx.doi.org/10.1016/S0048-7333\(03\)00047-7](http://dx.doi.org/10.1016/S0048-7333(03)00047-7)
- Ho, C.J., Vaughan, J.W., 2012. Online task assignment in crowdsourcing markets. Proc. 26th AAAI Conf. on Artificial Intelligence, p.45-51.
- Ho, C.J., Jabbari, S., Vaughan, J.W., 2013. Adaptive task assignment for crowdsourced classification. Proc. 30th Int. Conf. on Machine Learning, p.534-542.
- Hoffman, M.L., 1981. Is altruism part of human nature? *J. Personal. Soc. Psychol.*, 40(1):121-137.  
<http://dx.doi.org/10.1037/0022-3514.40.1.121>
- Jaimes, L.G., Vergara-Laurens, I., Labrador, M.A., 2012. A location-based incentive mechanism for participatory sensing systems with budget constraints. Proc. 10th Annual IEEE Int. Conf. on Pervasive Computing and Communications, p.103-108.  
<http://dx.doi.org/10.1109/PerCom.2012.6199855>
- Jain, S., Gujar, S., Bhat, S., et al., 2014. An incentive compatible multi-armed-bandit crowdsourcing mechanism with quality assurance. ePrint Archive, arXiv:1406.7157.
- Jeong, G., Kim, S., Zimmermann, T., 2009. Improving bug triage with bug tossing graphs. Proc. 7th Joint Meeting of the European Software Engineering Conf. and the ACM SIGSOFT Symp. on the Foundations of Software Engineering, p.111-120.  
<http://dx.doi.org/10.1145/1595696.1595715>
- Karger, D.R., Oh, S., Shah, D., 2011. Iterative learning for reliable crowdsourcing systems. Advances in Neural Information Processing Systems, p.1953-1961.
- Khetan, A., Oh, S., 2016. Achieving budget-optimality with adaptive schemes in crowdsourcing. Advances in Neural Information Processing Systems, p.4844-4852.
- Kittur, A., Smus, B., Khamkar, S., et al., 2011. Crowdforge: crowdsourcing complex work. Proc. 24th Annual ACM Symp. on User Interface Software and Technology, p.43-52.  
<http://dx.doi.org/10.1145/2047196.2047202>
- Krishna, V., 2009. Auction Theory. Academic Press, New York, USA.
- Krontiris, I., Albers, A., 2012. Monetary incentives in participatory sensing using multi-attributive auctions. *Int. J. Parall. Emerg. Distr. Syst.*, 27(4):317-336.  
<http://dx.doi.org/10.1080/17445760.2012.686170>
- Law, E., Ahn, L., 2011. Human computation. *Synth. Lect. Artif. Intell. Mach. Learn.*, 5(3):1-121.
- Lazer, D., Kennedy, R., King, G., et al., 2014. The parable of Google flu: traps in big data analysis. *Science*, 343(6167):1203-1205.  
<http://dx.doi.org/10.1126/science.1248506>

- Lee, J.S., Hoh, B., 2010. Sell your experiences: a market mechanism based incentive for participatory sensing. *IEEE Int. Conf. on Pervasive Computing and Communications*, p.60-68.  
<http://dx.doi.org/10.1109/PERCOM.2010.5466993>
- Li, G.L., Wang, J.N., Zheng, Y.D., et al., 2016. Crowdsourced data management: a survey. *IEEE Trans. Knowl. Data Eng.*, **28**(9):2296-2319.  
<http://dx.doi.org/10.1109/TKDE.2016.2535242>
- Li, H.W., Yu, B., 2014. Error rate bounds and iterative weighted majority voting for crowdsourcing. ePrint Archive, arXiv:1411.4086.
- Li, X., Dong, X.L., Lyons, K., et al., 2012. Truth finding on the deep Web: is the problem solved? *Proc. VLDB Endow.*, **6**(2):97-108.  
<http://dx.doi.org/10.14778/2535568.2448943>
- Lintott, C.J., Schawinski, K., Slosar, A., et al., 2008. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. *Month. Not. R. Astronom. Soc.*, **389**(3):1179-1189.  
<http://dx.doi.org/10.1111/j.1365-2966.2008.13689.x>
- Liu, Q., Peng, J., Ihler, A.T., 2012. Variational inference for crowdsourcing. *Advances in Neural Information Processing Systems*, p.692-700.
- Luo, T., Tan, H.P., Xia, L.R., 2014. Profit-maximizing incentive for participatory sensing. *Proc. IEEE Conf. on Computer Communications*, p.127-135.  
<http://dx.doi.org/10.1109/INFOCOM.2014.6847932>
- Malone, T.W., Laubacher, R., Dellarocas, C., 2009. Harnessing Crowds: Mapping the Genome of Collective Intelligence. MIT Sloan Research Paper No. 4732-09, Sloan School of Management, Massachusetts Institute of Technology, MA, USA.  
<http://dx.doi.org/10.2139/ssrn.1381502>
- Mamykina, L., Manoim, B., Mittal, M., et al., 2011. Design lessons from the fastest Q&A site in the west. *Proc. SIGCHI Conf. on Human Factors in Computing Systems*, p.2857-2866.  
<http://dx.doi.org/10.1145/1978942.1979366>
- Maslow, A.H., Frager, R., Fadiman, J., et al., 1970. *Motivation and Personality*. Harper & Row, New York, USA.
- Mavridis, P., Gross-Amblard, D., Miklós, Z., 2016. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. *Proc. 25th Int. Conf. on World Wide Web*, p.843-853.
- Meng, R., Tong, Y.X., Chen, L., et al., 2015. CrowdTC: crowdsourced taxonomy construction. *Proc. IEEE Int. Conf. on Data Mining*, p.913-918.  
<http://dx.doi.org/10.1109/ICDM.2015.77>
- Mockus, A., Fielding, R.T., Herbsleb, J.D., 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Trans. Softw. Eng. Meth.*, **11**(3):309-346. <http://dx.doi.org/10.1145/567793.567795>
- Moser, R., Pedrycz, W., Succi, G., 2008. A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction. *Proc. 30th Int. Conf. on Software Engineering*, p.181-190.  
<http://dx.doi.org/10.1145/1368088.1368114>
- Nagappan, N., Ball, T., 2005. Use of relative code churn measures to predict system defect density. *Proc. 27th Int. Conf. on Software Engineering*, p.284-292.  
<http://dx.doi.org/10.1109/ICSE.2005.1553571>
- Nakakoji, K., Yamamoto, Y., Nishinaka, Y., et al., 2002. Evolution patterns of open-source software systems and communities. *Proc. Int. Workshop on Principles of Software Evolution*, p.76-85.  
<http://dx.doi.org/10.1145/512035.512055>
- Ok, J., Oh, S., Shin, J., et al., 2016. Optimality of belief propagation for crowdsourced classification. ePrint Archive, arXiv:1602.03619.
- Ouyang, W.R., Kaplan, L.M., Martin, P., et al., 2015. Debiassing crowdsourced quantitative characteristics in local businesses and services. *Proc. 14th Int. Conf. on Information Processing in Sensor Networks*, p.190-201.  
<http://dx.doi.org/10.1145/2737095.2737116>
- Pan, Y.H., 2016. Heading toward artificial intelligence 2.0. *Engineering*, **2**(4):409-413.  
<http://dx.doi.org/10.1016/J.ENG.2016.04.018>
- Pierre, L., 1997. *Collective intelligence: mankind's emerging world in cyberspace*. Bononno, R., translator. Perseus Books, Cambridge, USA.
- Quinn, A.J., Bederson, B.B., 2011. Human computation: a survey and taxonomy of a growing field. *Proc. SIGCHI Conf. on Human Factors in Computing Systems*, p.1403-1412.  
<http://dx.doi.org/10.1145/1978942.1979148>
- Raban, D.R., Harper, F.M., 2008. Motivations for Answering Questions Online.  
<http://www.researchgate.net/publication/241053908>
- Rahman, F., Devanbu, P.T., 2011. Ownership, experience and defects: a fine-grained study of authorship. *Proc. 33rd Int. Conf. on Software Engineering*, p.491-500.  
<http://dx.doi.org/10.1145/1985793.1985860>
- Rahman, F., Devanbu, P.T., 2013. How, and why, process metrics are better. *Proc. 35th Int. Conf. on Software Engineering*, p.432-441.
- Rana, R.K., Chou, C.T., Kanhere, S.S., et al., 2010. Earphone: an end-to-end participatory urban noise mapping system. *Proc. 9th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks*, p.105-116.  
<http://dx.doi.org/10.1145/1791212.1791226>
- Raykar, V.C., Yu, S., 2012. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.*, **13**(Feb):491-518.
- Raykar, V.C., Yu, S., Zhao, L.H., et al., 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. *Proc. 26th Int. Conf. on Machine Learning*, p.889-896.  
<http://dx.doi.org/10.1145/1553374.1553488>
- Raykar, V.C., Yu, S., Zhao, L.H., et al., 2010. Learning from crowds. *J. Mach. Learn. Res.*, **11**(Apr):1297-1322.
- Raymond, E., 1999. The cathedral and the bazaar. *Knowl. Technol. Polic.*, **12**(3):23-49.  
<http://dx.doi.org/10.1007/s12130-999-1026-0>

- Rigby, P.C., German, D.M., Cowen, L., et al., 2014. Peer review on open-source software projects: parameters, statistical models, and theory. *ACM Trans. Softw. Eng. Meth.*, **23**(4):No.35. <http://dx.doi.org/10.1145/2594458>
- Rogers, E.M., 2010. Diffusion of Innovations. Simon and Schuster, New York, USA.
- Sakaki, T., Okazaki, M., Matsuo, Y., 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. Proc. 19th Int. Conf. on World Wide Web, p.851-860. <http://dx.doi.org/10.1145/1772690.1772777>
- Shah, N.B., Zhou, D., 2015. Double or nothing: multiplicative incentive mechanisms for crowdsourcing. Advances in Neural Information Processing Systems, p.1-9.
- Shah, N.B., Zhou, D., 2016. No oops, you won't do it again: mechanisms for self-correction in crowdsourcing. Proc. 33rd Int. Conf. on Machine Learning, p.1-10.
- Shah, N.B., Zhou, D., Peres, Y., 2015. Approval voting and incentives in crowdsourcing. ePrint Archive, arXiv:1502.05696.
- She, J.Y., Tong, Y.X., Chen, L., 2015a. Utility-aware social event-participant planning. Proc. ACM Int. Conf. on Management of Data, p.1629-1643. <http://dx.doi.org/10.1145/2723372.2749446>
- She, J.Y., Tong, Y.X., Chen, L., et al., 2015b. Conflict-aware event-participant arrangement. Proc. 31st IEEE Int. Conf. on Data Engineering, p.735-746. <http://dx.doi.org/10.1109/ICDE.2015.7113329>
- She, J.Y., Tong, Y.X., Chen, L., et al., 2016. Conflict-aware event-participant arrangement and its variant for online setting. *IEEE Trans. Knowl. Data Eng.*, **28**(9):2281-2295. <http://dx.doi.org/10.1109/TKDE.2016.2565468>
- Shen, H.W., Barabási, A.L., 2014. Collective credit allocation in science. *PNAS*, **111**(34):12325-12330. <http://dx.doi.org/10.1073/pnas.1401992111>
- Smith, J.B., 1994. Collective Intelligence in Computer-Based Collaboration. CRC Press, Boca Raton, USA.
- Subramanian, A., Kanth, G.S., Vaze, R., 2013. Offline and online incentive mechanism design for smart-phone crowd-sourcing. ePrint Archive, arXiv:1310.1746.
- Subramanyam, R., Krishnan, M.S., 2003. Empirical analysis of CK metrics for object-oriented design complexity: implications for software defects. *ACM Trans. Softw. Eng. Meth.*, **29**(4):297-310. <http://dx.doi.org/10.1109/TSE.2003.1191795>
- Sullivan, B.L., Wood, C.L., Iliff, M.J., et al., 2009. eBird: a citizen-based bird observation network in the biological sciences. *Biol. Conserv.*, **142**(10):2282-2292. <http://dx.doi.org/10.1016/j.biocon.2009.05.006>
- Tamrawi, A., Nguyen, T.T., Al-Kofahi, J.M., et al., 2011. Fuzzy set and cache-based approach for bug triaging. Proc. 19th ACM SIGSOFT Symp. and 13th European Conf. on Foundations of Software Engineering, p.365-375. <http://dx.doi.org/10.1145/2025113.2025163>
- Tang, J.C., Cebrian, M., Giacobe, N.A., et al., 2011. Reflecting on the DARPA red balloon challenge. *Commun. ACM*, **54**(4):78-85. <http://dx.doi.org/10.1145/1924421.1924441>
- Teodoro, R., Ozturk, P., Naaman, M., et al., 2014. The motivations and experiences of the on-demand mobile workforce. Proc. 17th ACM Conf. on Computer Supported Cooperative Work & Social Computing, p.236-247. <http://dx.doi.org/10.1145/2531602.2531680>
- Thebault-Spieker, J., Terveen, L.G., Hecht, B., 2015. Avoiding the south side and the suburbs: the geography of mobile crowdsourcing markets. Proc. 18th ACM Conf. on Computer Supported Cooperative Work & Social Computing, p.265-275. <http://dx.doi.org/10.1145/2675133.2675278>
- Thongtanunam, P., Tantithamthavorn, C., Kula, R.G., et al., 2015. Who should review my code? A file location-based code-reviewer recommendation approach for modern code review. IEEE 22nd Int. Conf. on Software Analysis, Evolution, and Reengineering, p.141-150. <http://dx.doi.org/10.1109/SANER.2015.7081824>
- Tian, T., Zhu, J., 2015. Max-margin majority voting for learning from crowds. Advances in Neural Information Processing Systems, p.1621-1629.
- Tong, Y.X., Chen, L., Ding, B.L., 2012a. Discovering threshold-based frequent closed itemsets over probabilistic data. Proc. IEEE 28th Int. Conf. on Data Engineering, p.270-281. <http://dx.doi.org/10.1109/ICDE.2012.51>
- Tong, Y.X., Chen, L., Cheng, Y.R., et al., 2012b. Mining frequent itemsets over uncertain databases. *Proc. VLDB Endow.*, **5**(11):1650-1661.
- Tong, Y.X., Cao, C.C., Zhang, C.J., et al., 2014a. Crowd-Cleaner: data cleaning for multi-version data on the web via crowdsourcing. Proc. IEEE 28th Int. Conf. on Data Engineering, p.1182-1185. <http://dx.doi.org/10.1109/ICDE.2014.6816736>
- Tong, Y.X., Cao, C.C., Chen, L., 2014b. TCS: efficient topic discovery over crowd-oriented service data. Proc. 20th ACM Int. Conf. on Knowledge Discovery and Data Mining, p.861-870. <http://dx.doi.org/10.1145/2623330.2623647>
- Tong, Y.X., Chen, L., She, J.Y., 2015a. Mining frequent itemsets in correlated uncertain databases. *J. Comput. Sci. Technol.*, **30**(4):696-712. <http://dx.doi.org/10.1007/s11390-015-1555-9>
- Tong, Y.X., Meng, R., She, J.Y., 2015b. On bottleneck-aware arrangement for event-based social networks. Proc. 31st IEEE Int. Conf. on Data Engineering Workshops, p.216-223. <http://dx.doi.org/10.1109/ICDEW.2015.7129579>
- Tong, Y.X., She, J.Y., Meng, R., 2016a. Bottleneck-aware arrangement over event-based social networks: the max-min approach. *World Wide Web*, **19**(6):1151-1177. <http://dx.doi.org/10.1007/s11280-015-0377-6>
- Tong, Y.X., She, J.Y., Ding, B.L., et al., 2016b. Online mobile micro-task allocation in spatial crowdsourcing. Proc. 32nd IEEE Int. Conf. on Data Engineering, p.49-60. <http://dx.doi.org/10.1109/ICDE.2016.7498228>
- Tong, Y.X., She, J.Y., Ding, B.L., et al., 2016c. Online minimum matching in real-time spatial data: experiments

- and analysis. *Proc. VLDB Endow.*, **9**(12):1053-1064.  
<http://dx.doi.org/10.14778/2994509.2994523>
- Tong, Y.X., Zhang, X.F., Chen, L., 2016d. Tracking frequent items over distributed probabilistic data. *World Wide Web*, **19**(4):579-604.  
<http://dx.doi.org/10.1007/s11280-015-0341-5>
- Tong, Y.X., Yuan, Y., Cheng, Y.R., et al., 2017. A survey of spatiotemporal crowdsourced data management techniques. *J. Softw.*, **28**(1):35-58 (in Chinese).
- Tran-Thanh, L., Stein, S., Rogers, A., et al., 2012. Efficient crowdsourcing of unknown experts using multi-armed bandits. *Proc. European Conf. on Artificial Intelligence*, p.768-773.  
<http://dx.doi.org/10.3233/978-1-61499-098-7-768>
- Tsay, J., Dabbish, L., Herbsleb, J.D., 2014a. Influence of social and technical factors for evaluating contribution in GitHub. 36th Int. Conf. on Software Engineering, p.356-366. <http://dx.doi.org/10.1145/2568225.2568315>
- Tsay, J., Dabbish, L., Herbsleb, J.D., 2014b. Let's talk about it: evaluating contributions through discussion in GitHub. *Proc. 22nd ACM Int. Symp. on Foundations of Software Engineering*, p.144-154.  
<http://dx.doi.org/10.1145/2635868.2635882>
- Vasilescu, B., Yu, Y., Wang, H., et al., 2015. Quality and productivity outcomes relating to continuous integration in GitHub. *Proc. 10th Joint Meeting on Foundations of Software Engineering*, p.805-816.  
<http://dx.doi.org/10.1145/2786805.2786850>
- Vickrey, W., 1961. Counterspeculation, auctions, and competitive sealed tenders. *J. Financ.*, **16**(1):8-37.  
<http://dx.doi.org/10.1111/j.1540-6261.1961.tb02789.x>
- von Ahn, L., Maurer, B., McMillen, C., et al., 2008. reCAPTCHA: human-based character recognition via web security measures. *Science*, **321**(5895):1465-1468.  
<http://dx.doi.org/10.1126/science.1160379>
- Wang, D., Abdelzaher, T.F., Kaplan, L.M., et al., 2013. Recursive fact-finding: a streaming approach to truth estimation in crowdsourcing applications. *IEEE 33rd Int. Conf. on Distributed Computing Systems*, p.530-539. <http://dx.doi.org/10.1109/ICDCS.2013.54>
- Wang, D., Amin, M.T.A., Li, S., et al., 2014. Using humans as sensors: an estimation-theoretic perspective. *Proc. 13th Int. Conf. on Information Processing in Sensor Networks*, p.35-46.
- Wang, H.M., Yin, G., Li, X., et al., 2015. TRUSTIE: a software development platform for crowdsourcing. *In: Li, W., Huhns, M.N., Tsai, W.T., et al. (Eds.), Crowdsourcing*. Springer Berlin Heidelberg, Berlin, Germany, p.165-190.
- Wang, J.N., Li, G.L., Kraska, T., et al., 2013. Leveraging transitive relations for crowdsourced joins. *Proc. ACM Int. Conf. on Management of Data*, p.229-240.  
<http://dx.doi.org/10.1145/2463676.2465280>
- Wang, L., Zhou, Z.H., 2016. Cost-saving effect of crowdsourcing learning. *Proc. 25th Int. Joint Conf. on Artificial Intelligence*, p.2111-2117.
- Wang, W., Zhou, Z.H., 2015. Crowdsourcing label quality: a theoretical study. *Sci. China Inform. Sci.*, **58**(11):1-12.  
<http://dx.doi.org/10.1007/s11432-015-5391-x>
- Wauthier, F.L., Jordan, M.I., 2011. Bayesian bias mitigation for crowdsourcing. *Advances in Neural Information Processing Systems*, p.1800-1808.
- Welinder, P., Branson, S., Perona, P., et al., 2010. The multidimensional wisdom of crowds. *Advances in Neural Information Processing Systems*, p.2424-2432.
- Whitehill, J., Wu, T., Bergsma, J., et al., 2009. Whose vote should count more: optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, p.2035-2043.
- Wu, W.J., Tsai, W.T., Li, W., 2013. An evaluation framework for software crowdsourcing. *Front. Comput. Sci.*, **7**(5):694-709.  
<http://dx.doi.org/10.1007/s11704-013-2320-2>
- Yan, Y., Fung, G.M., Rosales, R., et al., 2011. Active learning from crowds. *Proc. 28th Int. Conf. on Machine Learning*, p.1161-1168.
- Yang, D.J., Fang, X., Xue, G.L., 2013. Truthful incentive mechanisms for  $k$ -anonymity location privacy. *Proc. IEEE Conf. on Computer Communications*, p.2994-3002.  
<http://dx.doi.org/10.1109/INFCOM.2013.6567111>
- Yang, D.J., Xue, G.L., Fang, X., et al., 2012. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. *Proc. 18th Annual Int. Conf. on Mobile Computing and Networking*, p.173-184. <http://dx.doi.org/10.1145/2348543.2348567>
- Ye, Y.W., Kishida, K., 2003. Toward an understanding of the motivation of open source software developers. *Proc. 25th Int. Conf. on Software Engineering*, p.419-429.  
<http://dx.doi.org/10.1109/ICSE.2003.1201220>
- Yu, Y., Yin, G., Wang, H., et al., 2014. Exploring the patterns of social behavior in GitHub. *Proc. 1st Int. Workshop on Crowd-Based Software Development Methods and Technologies*, p.31-36.  
<http://dx.doi.org/10.1145/2666539.2666571>
- Yu, Y., Yin, G., Wang, T., et al., 2016a. Determinants of pull-based development in the context of continuous integration. *Sci. China Inform. Sci.*, **59**(8):080104.  
<http://dx.doi.org/10.1007/s11432-016-5595-8>
- Yu, Y., Wang, H.M., Yin, G., et al., 2016b. Reviewer recommendation for pull-requests in GitHub: what can we learn from code review and bug assignment? *Inform. Softw. Technol.*, **74**:204-218.  
<http://dx.doi.org/10.1016/j.infsof.2016.01.004>
- Zhang, C.J., Chen, L., Tong, Y.X., 2014a. MaC: a probabilistic framework for query answering with machine-crowd collaboration. *Proc. 23rd ACM Int. Conf. on Information and Knowledge Management*, p.11-20.  
<http://dx.doi.org/10.1145/2661829.2661880>
- Zhang, C.J., Tong, Y.X., Chen, L., 2014b. Where to: crowd-aided path selection. *Proc. VLDB Endow.*, **7**(11):2005-2016. <http://dx.doi.org/10.14778/2733085.2733105>
- Zhang, C.J., Chen, L., Tong, Y., et al., 2015. Cleaning uncertain data with a noisy crowd. *Proc. 31st IEEE Int. Conf. on Data Engineering*, p.6-17.  
<http://dx.doi.org/10.1109/ICDE.2015.7113268>

- Zhang, Y., Chen, X., Zhou, D., *et al.*, 2014. Spectral methods meet EM: a provably optimal algorithm for crowdsourcing. *Advances in Neural Information Processing Systems*, p.1260-1268.
- Zhao, D., Li, X.Y., Ma, H.D., 2014. How to crowdsource tasks truthfully without sacrificing utility: online incentive mechanisms with budget constraint. *Proc. IEEE Conf. on Computer Communications*, p.1213-1221. <http://dx.doi.org/10.1109/INFOCOM.2014.6848053>
- Zhong, J.H., Tang, K., Zhou, Z.H., 2015. Active learning from crowds with unsure option. *Proc. 24th Int. Joint Conf. on Artificial Intelligence*, p.1061-1067.
- Zhou, D., Basu, S., Mao, Y., *et al.*, 2012. Learning from the wisdom of crowds by minimax entropy. *Advances in Neural Information Processing Systems*, p.2195-2203.
- Zhou, Y., Chen, X., Li, J., 2014. Optimal PAC multiple arm identification with applications to crowdsourcing. *Proc. 31st Int. Conf. on Machine Learning*, p.217-225.
- Zhu, Y., Zhang, Q., Zhu, H., *et al.*, 2014. Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones. *Proc. 34th Int. Conf. on Distributed Computing Systems*, p.11-20. <http://dx.doi.org/10.1109/ICDCS.2014.10>