Goldilocks: Learning Pattern-Based Task Assignment in Mobile Crowdsensing

Jinghan Jiang[1], Yiqin Dai[2], Kui Wu[1](✉), and Rong Zheng[3]

[1] Department of Computer Science, University of Victoria, Victoria, BC, Canada
{jinghanj,wkui}@uvic.ca

[2] Department of Computer Science, National University of Defense Technology, Changsha, China
daiyq98@gmail.com

[3] Department of Computing and Software, McMaster University, Hamilton, ON, Canada
rzheng@mcmaster.ca

Abstract. Mobile crowdsensing (MCS) depends on mobile users to collect sensing data, whose quality highly depends on the expertise/experience of the users. It is critical for MCS to identify right persons for a given sensing task. A commonly-used strategy is to "teach-before-use", i.e., training users with a set of questions and selecting a subset of users who have answered the questions correctly the most of times. This method has large room for improvement if we consider users' learning curve during the training process. As such, we propose an interactive learning pattern recognition framework, Goldilocks, that can filter users based on their learning patterns. Goldilocks uses an adaptive teaching method tailored for each user to maximize her learning performance. At the same time, the teaching process is also the selecting process. A user can thus be safely excluded as early as possible from the MCS tasks later on if her performance still does not match the desired learning pattern after the training period. Experiments on real-world datasets show that compared to the baseline methods, Goldilocks can identify suitable users to obtain more accurate and more stable results for multi-categories classification problems.

Keywords: Mobile crowdsensing · Learning pattern recognition · Task assignment

1 Introduction

Mobile crowdsensing relies on the sensing capacity of mobile devices and the intelligence of mobile users to create innovative solutions to many real-world problems [1]. One example is environmental monitoring, where the participants are required to take photos of specific objects (e.g. some endangered species or poisonous mushrooms). Such MCS tasks normally involve both mobile users'

X. Chu et al. (Eds.): QShine 2019, LNICST 300, pp. 63–83, 2020.
https://doi.org/10.1007/978-3-030-38819-5_5

spatial-temporal information as well as their capability of securing correct, high-quality sensing data.

There is always a cost in the MCS applications of the above kind. This is because the participants (aka, crowd workers) may need to be rewarded, and even if they are purely volunteers, it would require substantial efforts for the decision maker to clean incorrect data if the crowd workers are not competent for the given tasks. As a result, "finding right people for right tasks" has become one of the most fundamental principles in MCS.

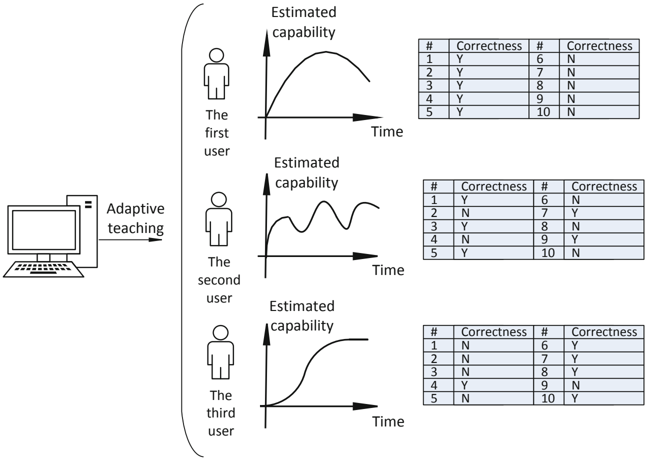
Indeed, many approaches [5,16] have been proposed to identify the qualification of crowd workers. A widely-used method is (randomly) inserting some questions with known answers during their participation. The samples with known ground truth are gold instances [2], which are used to evaluate a crowd worker's domain knowledge and accordingly take proper actions on their answers. This approach, however, has a well-known pitfall: embedding gold instances in *every* worker's annotation process may incur a high and sometimes unnecessary cost.

Another type of broadly-used method is teaching the crowd workers before assigning them a give MCS task [15,17]. This type of methods overcome the problem of embedding gold instances in the whole annotation process of crowd workers because a crowd worker without enough domain knowledge can get trained and crowd workers not performing well after the training can be excluded earlier. Nevertheless, in most existing solutions in this category, a fixed teaching set is used to train the crowd workers and the final selection of the crowd workers is mainly based on the number of questions they correctly answer in the teaching phase [15]. We, via the following illustrative example, argue that we can do much better if we consider each individual user's learning pattern.

Motivating Example. Butterflies have been widely used by ecologists as model organisms to study the impact of habitat loss and fragmentation, and climate change. To understand the ecological changes in a certain area, ecologists need to monitor different butterfly species at several specific times in a day. This is clearly too demanding if the monitored area is large. MCS fits this application perfectly but needs to solve the problem that most people do not have the expert knowledge of butterfly species. If we use the teaching-before-using method, the number of correctly answered questions during the teaching process may not be a good indicator for our final selection of crowd workers.

Consider three different learning patterns in the teaching phase, as shown in Fig. 1. The horizontal axis represents the time (or rounds of teaching), while the vertical axis denotes the capability estimation on the basis of user's correctly-answered questions over time. The three users have identical performance w.r.t. the total number of correctly-answered questions during the teaching phase (e.g. 5 in the example) and they should be treated equally in the final selection of crowd workers. 1 Nevertheless, they exhibit sharply different learning patterns: The first user correctly labels five questions in the beginning, but then makes mistakes in the follow-up questions; the performance of the second user varies all the time during the teaching process; for the third user, the number of correctlylabelled questions begins to increase after she learns a few examples and then her capability remains stable at a good level.

The reasons for different patterns could be many. For example, the first user may get confused or get tired quickly with more teaching instances; the second user may not learn effectively at all and consistently makes mistakes over time; the third user can quickly learn new knowledge and remain stable at a good capability level. Anyway, the hard-to-validate conjecture is irrelevant, since we are only interested in the observed learning patterns during the teaching phase. The point here is that the learning pattern can comprehensively reflect a user's capability of accepting, memorizing and applying new knowledge, which cannot be captured by the oversimplified metric – the total number of correctly answered questions. Clearly, users whose learning pattern in the teaching stage conforms to the third type in Fig. 1 should be a better choice than users whose learning pattern follows the other two curves, particularly when we only have limited teaching samples and cannot afford training the users for a long time.



#	Correctness	#	Correctness
1	Y	6	N
2	Y	7	N
3	Y	8	N
4	Y	9	N
5	Y	10	N

#	Correctness	#	Correctness
1	Y	6	N
2	N	7	Y
3	Y	8	N
4	N	9	Y
5	Y	10	N

#	Correctness	#	Correctness
1	N	6	Y
2	N	7	Y
3	N	8	Y
4	Y	9	N
5	N	10	Y

Fig. 1. Example learning patterns in the teaching phase

Motivated by the above observation, we propose Goldilocks[1], an interactive capability assessment framework for MCS, which adopts judicious user selection strategy to eliminates unsuitable users from the current MCS job at the early stage. Our goal is to select the participants whose performance follows the desired learning pattern to maximize the overall performance in an MCS task. Goldilocks adopts an adaptive teaching model to identify the users with stronger ability of generalizing new learned knowledge in the given task. The adaptive teaching strategy is tailored for each individual and is designed to exert the greatest

[1] The term is meant to emphasize that our ultimate goal is identifying the right people for right tasks rather than evaluating people's general learning capability.

learning potential of the user. Our philosophy is that if the “ultimate” capability of one user is still not satisfactory after the tailored teaching, we have enough evidence to believe that the user would not be a good candidate for the task and thus should be excluded from task assignment at the early stage.

This paper makes the following contributions:

1. We propose a new framework, Goldilocks, for adaptive learning pattern recognition in participatory MCS. This framework is largely different from previous work that uses either gold instance in the whole process or simplified criteria (e.g. the total number of correct answers) in the participant selection.
2. Goldilocks integrates the teaching and selection phases in a unified framework, and selects the qualified users as early as possible without assigning all the questions to each user. In this way, it significantly saves time and cost for a given MCS task.
3. Based on the work in [9], we develop a new web service over Amazon Web Services (AWS), which obtains the model parameters and automatically adjusts questions to maximize individual participants’ learning performance. The web service also collects data to profile the participants’ learning pattern, which can be used for task assignment over MCS. Experiments on real-world datasets show that Goldilocks outperforms the baseline methods in both user profiling and the final participant selection.

2 Related Work and Background

Based on the degree of human participation, MCS can be roughly classified into two categories: participatory sensing and opportunistic sensing, as shown in Fig. 2.

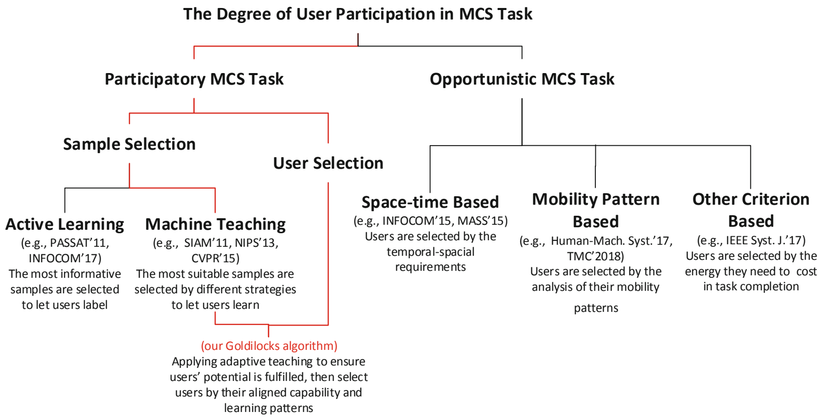

The Degree of User Participation in MCS Task
Participatory MCS Task
Opportunistic MCS Task
Sample Selection
User Selection
Active Learning
(e.g., PASSAT’11, INFOCOM’17)
The most informative samples are selected to let users label
Machine Teaching
(e.g., SIAM’11, NIPS’13, CVPR’15)
The most suitable samples are selected by different strategies to let users learn
(our Goldilocks algorithm)
Applying adaptive teaching to ensure users’ potential is fulfilled, then select users by their aligned capability and learning patterns
Space-time Based
(e.g., INFOCOM’15, MASS’15)
Users are selected by the temporal-spacial requirements
Mobility Pattern Based
(e.g., Human-Mach. Syst.’17, TMC’2018)
Users are selected by the analysis of their mobility patterns
Other Criterion Based
(e.g., IEEE Syst. J.’17)
Users are selected by the energy they need to cost in task completion

Fig. 2. Classification of MCS

2.1 Opportunistic Sensing

The user involvement is low in opportunistic sensing tasks. For example, asking for the continuous location information without the explicit action of the user is one type of opportunistic sensing [6].

Opportunistic sensing can be classified into three groups: space-time based, mobility pattern based, and others. *In the first group*, Karaliopoulos et al. [11] proposed a recruitment strategy in opportunistic networking to generate the required space-time paths across the network for collecting data from a set of fixed locations. Li et al. [12] presented a temporal-spatial model for participant recruitment in scenarios where the tasks arrive in real time and have different temporal-spatial requirements. *In the second group*, Guo et al. [8] selected crowd workers based on their movement patterns and the task's sensitivity to time. The recruitment strategy proposed in [20] is based on the user's probability of moving to a destination. *In the third group*, Liu et al. [13] considered the energy needed to complete tasks and used a participant sampling behavior model in the participant selection. Yang et al. [22] presented a personalized task recommend system which recommends tasks to users based on their preference and reliability to different tasks.

2.2 Participatory Sensing

Unlike opportunistic sensing, participatory sensing requires the active involvement of crowd workers. Active learning and machine teaching are frequently used to interact with users in their labeling process, so the performance of users can be observed in real time.

The authors in [25] introduced a method to automatically identify the most valuable unlabeled instances as well as the samples that might benefit from relabeling. Zhang et al. [24] combined both labeled and unlabeled instances in the interaction with users, and proposed a bidirectional active learning algorithm by querying users with both the informative unlabeled samples and the unreliable labeled instances simultaneously. The distributed active learning framework in [21] takes the upload and query cost into consideration when interacting with different users, with the goal of minimizing the prediction errors. In addition to the above active learning frameworks, various studies in machine teaching also provide methods to deal with the user interaction problem in the labeling process. Du et al. [3] presented a teaching strategy to achieve more effective learning, which interacts with users by using a probabilistic model based on their answers in each round. Zhu [26] employed Bayesian models to find the optimal teaching set for individuals. Johns et al. [9] developed an interactive machine teaching algorithm that enables a computer to teach users challenging visual concepts by probabilistically modeling the users' ability based on their correct and incorrect answers.

Among the machine teaching methods, adaptive teaching is directly related to our work, and hence we introduce more background in adaptive teaching in the next section.

2.3 Adaptive Teaching

Since many MCS tasks require specific domain knowledge, users usually need training before they are assigned tasks. In order to select right people for a given task, we should consider two problems: (a) how to teach people so that they can generalize the learned knowledge as soon as possible? (b) how to profile users' learning pattern so that better candidates for the MCS task can be determined?

To address the first problem, adaptive teaching [10] is a proper way to help people learn more effectively by posing training questions that are adaptively chosen based on their existing performance. In [3], the next teaching image for a user is the one that the user's answer is predicted to be the farthest from the ground truth. An offline Bayesian model is applied to select adaptive teaching samples in [17]. The teaching strategy presented in [9] chooses a teaching sample that has the greatest reduction on the future error over the rest of unlabeled samples. The goal of adaptive teaching is to stimulate a user's learning potential as much as possible. We hence have a very good reason to eliminate the users who still perform poorly for a given MCS task even after adaptive teaching.

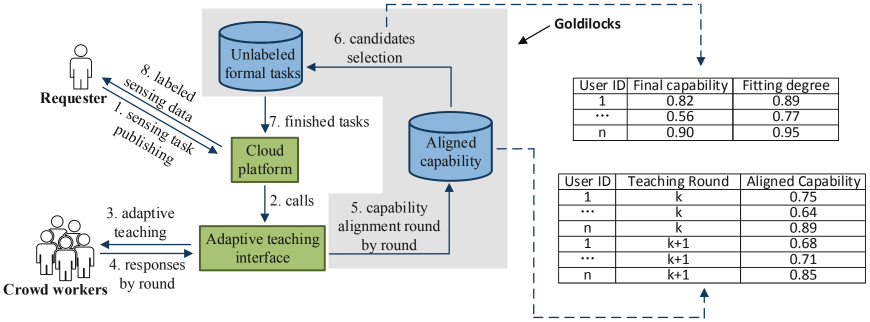
Regarding the second problem, we need to extract users' learning styles through their training processes. Although learning pattern recognition algorithms are studied before [7], they normally assume that abundant labeling data in a given domain are available and then apply different machine learning classification models on the historical learning data to extract learning patterns. This does not match our scenario where learning profile should be built on the fly as a new question is asked and answered. In addition, the learning pattern recognition methods in existing work are more focused on predicting the accuracy of new labels only based on whether or not a user's past questions are labeled correctly.

In this paper, we consider the above two problems and propose an interactive user selection framework for participatory MCS platforms. Note that *we build on the existing solution* [9] *for adaptive teaching but develop a new method for profiling users' learning patterns*. Unlike other work, our work focuses on the selection of users by doing adaptive capability acquisition round by round[2] for each worker. To match the right user to right task, we firstly select the most appropriate teaching samples for each user to ensure that they can learn as much as possible according to their own learning patterns. And then we select the users for the job by fitting their learning pattern to the non-linear (sigmoid) [23] curves.

3 Overview of Goldilocks

The role of Goldilocks in the workflow of MCS is illustrated in Fig. 3: an organization or individual requests sensing data from the cloud platform (Step 1), then an adaptive teaching interface is called to teach crowd workers according to their

[2] A round means that the user answers a question and then is told whether her answer is correct or not as well as what the ground truth is.



User ID	Final capability	Fitting degree
1	0.82	0.89
···	0.56	0.77
n	0.90	0.95

User ID	Teaching Round	Aligned Capability
1	k	0.75
···	k	0.64
n	k	0.89
1	k+1	0.68
···	k+1	0.71
n	k+1	0.85

Fig. 3. The role of Goldilocks in the MCS platform; solid lines denote the sensing data requesting procedure; dash lines denote the intermediate results obtained by Goldilocks; the grey box denotes the main functions of Goldilocks.

performances (Steps 2, 3, 4). Adaptive teaching will select the questions that can maximize the user's probability of correctly answering the questions in the future round by round. After each round, we compare the user's estimated performance based on her learning performance so far with her actual performance in labeling the new question (Step 5). Steps 3, 4, and 5 will *repeat until the specified number of teaching samples are taught* to each user. Then we adopt a two-stage candidate selection strategy based on the acquired performance indicators returned by Step 5, to select the best user subset for the formal tasks (Step 6). For the users who are not selected, we will not continue to teach or hire them. Finally, the formal questions without ground truths will be completed by *the selected users*, and the sensing data from this user subset will be returned to the cloud platform, which will then be provided to the original requester (Steps 7 and 8). The above procedure is denoted by the solid lines in Fig. 3.

The core component of Goldilocks is the design of Step 5 and Step 6, denoted in the grey box in Fig. 3. They describe the learning pattern of each user by adjusting a user's estimated capability on the basis of her true performance on the teaching sample in each new round. The dash lines refer to the data stored in the "Round-by round performance indicator" database, and the high-level indicators of user's learning pattern calculated by the "Round-by round performance indicator" database.

The two core steps are denoted by the grey box in Fig. 3: "capability adjustment" and "candidate selection", which will be introduced in Sects. 4 and 5, respectively. The notations used in this paper are listed in Table 1.

4 Capability Adjustment in Goldilocks

We adjust the estimation of user's capability round by round. If we measure the user's capability with a *performance indicator*, its value should be adjusted over time, based on the questions that the user have answered and the current question at the current round. Using M_j^k to indicate user k's capability estimated

Table 1. Summary of main notations

Notation	Description
i	The question number of labeled questions $(0 < i < j)$
j	The question number of the current question
m	The number of extracted keypoint feature vector of a question
n	The number of questions in the teaching set
$\mathbf{x_{iw}^k}$	The wth $(0 < w < m)$ keypoint feature vector of the ith question labeled by user k
$\mathbf{A_i^k}$	The vector set consists of the m keypoint feature vectors of the question i labeled by user k
d_{ij}^k	The value of $j - i$ of user k
s_{ij}^k	The similarity between question i and question j answered by user k
a_i^k	Binary value, $a_i^k = 1$ if the ith question is correctly labeled by user k, else $a_i^k = 0$
t_i^k	The time of labeling the ith question by user k
I_i^k	The ID of the ith question labeled by user k
δ_j^k	The offset on user k's capability after the jth question is labeled
α_{ij}^k	The impact of the user k's previously answered question i on her current to be answered question j
ϕ_j^k	The value of $\sum\limits_i \alpha_{ij}^k$
M_j^k	The accumulated capability of user k after the jth question is labeled

after she answers the j-th question, we need to design an algorithm that calculates $\{M_1^k, M_2^k, \cdots, M_n^k\}$, where n is the total number of training questions for the user. We use superscript k to denote user k to emphasize that the sequences of teaching questions for different users are different.

Intuitively, when a user is trained with a question i, the knowledge learned from this question may have an impact on the probability that the user correctly answers a later question. This impact comes from various reasons, e.g., (1) the questions may be similar, and (2) adaptive teaching [10] raises the next question based on the user's historical answers to previous questions. In addition, the impact may become smaller as time goes. As such, we need a method to estimate the impact of all previous questions on the user's performance of answering current question (Sect. 4.1) and consider this impact when we adjust the user's performance indicator value (Sect. 4.2).

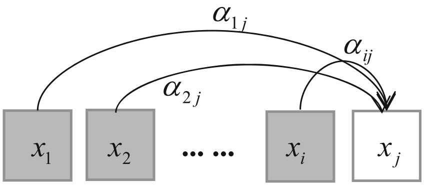


Fig. 4. The impacts of the previous questions on a user's capability of answering the current question. The shaded parts are samples that have been taught to the user

4.1 Impact of Previous Questions

We describe the teaching set for user k in the teaching phase as

$$D^k = \{(1, \mathbf{A_1^k}, t_1^k, a_1^k), \ldots, (n, \mathbf{A_n^k}, t_n^k, a_n^k)\},$$

where $\mathbf{A_i^k}$ is an extracted high-dimensional vector set to describe the features of ith question. Note that the similarity between different teaching samples and their order should be used in our model. t_i^k denotes the time that user k spends on labeling question i and a_i^k is a binary variable to record whether her answer is correct ($a_i^k = 1$ if the question is correctly labeled, otherwise $a_i^k = 0$).

As shown in Fig. 4, for each user in the teaching phase, each previously learned question has an impact on the user's ability of answering the current question j. We denote this influence value as α_{ij}^k. To estimate this influence, we need to consider parameters a_i^k and t_i^k on each previously learned question i. If question i is correctly labeled, a shorter time spent on the question implies a better skill on such type of questions, and thus user k is more likely to use the knowledge related to question i when answering the current question j. If question i is given a wrong label, the larger t_i^k implies that user k has worked harder on this question. While the reasons could vary, it is reasonable to assume that a larger t_i^k would have a more positive impact on the user's later performance compared to the situation that she spends a little time on question i with a wrong answer returned. Based on the above consideration, we suggest the following equation to calculate α_{ij}^k:

$$\alpha_{ij}^k = \frac{1}{1 + e^{-\left[\left(\frac{s_{ij}^k}{d_{ij}^k}\right)\left(1 + \frac{(-1)^{a_i^k+1}}{t_i^k}\right)\right]}}, \tag{1}$$

where $d_{ij}^k = j - i$ is the distance between questions i and j, s_{ij}^k is the similarity between question i and question j. The calculation of s_{ij}^k depends on specific applications (e.g. image, audio, and text should use different algorithms to calculate similarity). Following the human memory curve [4], a larger d_{ij} implies a smaller likelihood that the user correctly labels question j.

When we estimate the capability of a user before she answers a current question j, we should consider the impact of *all* previous questions. As such, we calculate $\sum_i \alpha_{ij}^k$, which is denoted as ϕ_j^k for simplicity:

$$\phi_j^k = \sum_{i=0}^{j} \alpha_{ij}^k. \tag{2}$$

ϕ_j^k could be considered as a quantitative measure of *the impact of previous questions on the ability that the user can correctly answer the current question*, since ϕ_j^k encodes all the impact of previous questions before j. ϕ_j^k also implicitly reflects the knowledge that the user has gained so far and thus is useful for later adjustment of her capability estimation after we see her true answer.

Equation (2) has two advantageous properties: (a) It considers the relationship of different teaching questions. And in such context, it utilizes the user's time and correctness on those questions. (b) As the number of samples learned by the user increases, the ϕ_j^k is increased by a number in $(0, 1)$ each time. Compared to other user profiling methods, this process enables us to focus on the user's capability of applying the new knowledge she just learned, rather than simply counting the number of correctly answered questions.

Remark 1. Equation (2) is just one way, among potentially many others, of estimating ϕ_j^k. While there is no theoretical guarantee on its accuracy, it can be empirically shown to be effective in our later experimental studies.

4.2 Adjusting Capability Estimation

After the user answers the current question j, we then use ϕ_j^k and her answer to adjust the estimation of her capability. As mentioned before, we use M_j^k to indicate user k's capability estimated after she answers the j-th question, and we record $M_1^k, M_2^k, \cdots$. Essentially, we need to find the adjustment value δ_j^k to calculate $M_j^k = M_{j-1}^k + \delta_j^k$.

When the User Answers the Current Question Correctly. We propose the following formula to calculate δ_j^k when a user answers the current question j correctly (i.e., $a_j^k = 1$):

$$\delta_j^k = \frac{\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\phi_j^k - \mu)^2}{2\sigma^2}}}{t_j^k} \tag{3}$$

The above formula is proposed due to the following considerations: (1) Assuming that ϕ_j^k (i.e., the knowledge gained by a user) follows normal distribution, the numerator is the normal distribution's probability density function (pdf) with mean μ and variance σ^2, where μ and σ are empirical values (from our later experimental results, $\mu = 2.69, \sigma = 1$). (2) Assuming that the longer the time that the user spends on the question, the lower the increments on her capability estimation, we divide the probability value by t_j^k. It is worth noting that in our method it is the shape (i.e., the bell shape when t_j^k is fixed) rather than the absolute values that matter, and as such we believe other possible functions of the similar shape would also work well.

When the User Answers the Current Question Incorrectly. In this case, the higher the chance that she should answer the question correctly based on her historical records, the higher the reduction that should be posed to adjust her capability estimation. Due to this reason, we use a monotone decreasing function w.r.t. ϕ_j^k and t_j^k to obtain δ_j^k:

$$\delta_j^k = \frac{-\nu\left(\phi_j^k\right)^2 + c}{t_j^k} \tag{4}$$

where ν and c are parameters set with experimental results[3]. The 3-D functional image of a_j^k, t_j^k, ϕ_j^k and δ_j^k is shown in Fig. 5. From this Figure, we can also see that when a certain threshold of labeling time (e.g., 6 s) is exceeded, the longer labeling time will not lead to a significantly smaller subtraction on the alignment offset δ_j^k. This reflects a fact that there should be no much difference in the user's capability if she takes a long time and gives a wrong answer.

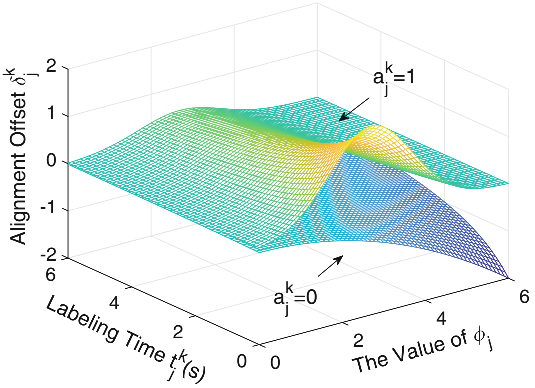


Fig. 5. The 3-D functional image of a_j^k, t_j^k, ϕ_j^k and δ_j^k, where the upper half represents the relationship of t_j^k, ϕ_j^k and δ_j^k when $a_j^k = 1$, and the other part denotes the relationship of them when $a_j^k = 0$.

4.3 Pseudocode

To summarize, the main idea of user's capability estimation includes: (a) estimating the impact of previous learned questions and (b) based on the estimation

[3] In our later experiments, to make Eqs. (4) and (3) have the same range values, we $\nu = 0.011$ and $c = -0.018$.

Algorithm 1. Capability Adjustment Workflow

```
Input: The number of question $n$, and the first teaching sample labeled by the $k$th
    user $T^k = \{(1, I_1^k, t_1^k, a_1^k)\}$, where $I_1^k$ is the ID of the first question labeled by the
    $k$th user. $t_1^k$ is the time the $k$th user spends on the first question. $a_1^k = 1$ if the first
    question is correctly labeled by user $k$, else $a_1^k = 0$
Output: The user's performance indicator after each round and the final accumulated
    capability indicator of the user $M^k = \{M_1^k, M_2^k, \cdots, M_n^k\}$
$M_1^k \leftarrow 0$
for $j = 2 \rightarrow n$ do
    $(j, I_j^k, t_j^k, a_j^k) \leftarrow \mathrm{EER}(j-1, I_{j-1}^k, a_{j-1}^k)$
    $T^k \leftarrow T^k \cup (j, I_j^k, t_j^k, a_j^k)$
    $\alpha_{ij}^k \leftarrow 0$
    for $i = 1 \rightarrow j-1$ do
        $s_{ij}^k \leftarrow \Psi(I_i^k, I_j^k)$
        $d_{ij}^k \leftarrow j - i$
        $\alpha_{ij}^k \leftarrow \alpha_{ij}^k + \mathrm{U}(s_{ij}^k, d_{ij}^k, a_i^k, t_i^k)$
    end for
    $\delta_j^k \leftarrow 0$
    if $a_j^k = 0$ then
        $\delta_j^k \leftarrow \prod_0(\alpha_{ij}^k, t_j^k)$
    else
        $\delta_j^k \leftarrow \prod_1(\alpha_{ij}^k, t_j^k)$
    end if
    $M_j^k \leftarrow M_{j-1}^k + \delta_j^k$
end for
return $M^k = \{M_1^k, M_2^k, \cdots, M_n^k\}$
```

and the latest user performance to adjust the current capability estimation. The pseudocode is outlined in Algorithm 1.

In this workflow, EER$(\cdot)$ (i.e., Expected Error Reduction) denotes the interactive teaching algorithm [9] that we adopt to determine the next sample which can stimulate user's learning potential in the greatest extend. To be more specific, the next teaching sample is the one that, if labeled correctly, can reduce the probability of total error in the greatest level over the samples that are still not selected into the teaching set. This selection problem is formulated based on the conditional probability function and solved by a graph-based semi-supervised method named Gaussian Random Field (GRF) [27]. $\Psi(\cdot)$ represents the feature detection algorithm to obtain the similarity between any two questions. $\mathrm{U}(\cdot)$ is the impact measure function of how the previous learning questions act on user k, which has been introduced in Sect. 4.1. $\prod_0(\cdot)$ and $\prod_1(\cdot)$ stand for the different capability adjustments when the user gives a correct or wrong answer to a new question j, respectively, as discussed in Sect. 4.2.

5 Candidate Selection in Goldilocks

To accurately select the high-quality users, we divide the selection process into two stages based on the learning curve (drawn by M^k returned by Algorithm 1) of each candidate.

- Firstly, we keep the a percentage ($p\%$) of candidates according to their final capability (i.e., M_n^k) and exclude the others, where p is a tunable parameter. We adopt this step because the final capability is a reflection of a user's final learning result, and thus we need filter out the ones whose final capability is below a certain threshold.
- Next, we fit the learning curve of each remaining candidate with our desired curve. We order these candidates according to the *degree of match* and eliminate the bottom candidates until the desired number of candidates are left.

Regarding the calculation of *degree of match*, after the first step, we only care about the shape of each remaining candidate's learning curve. For this reason, we avoid using a static, fixed curve as the benchmark. Instead, we use the shape of logistic regression (sigmoid) curve[4] $y = \frac{b}{1+ae^{-kx}}$ (where a, b, k are positive parameters) as the desired shape, and evaluate different measures on the degree of match in Sect. 6.

6 Experimental Evaluation

6.1 Implementation

To evaluate the performance of Goldilocks, we need real-world data that reflect people's learning patterns. For this, we deploy a Python-based Django website over Amazon Elastic Compute Cloud (Amazon EC2), which interacts with various users and collects the corresponding real-time data. Based on the work [9], we modify its database structure to collect more information such as a user's time spent on each question and the user's exact annotation for each question in teaching and testing phases, respectively.

Table 2. Summary of the experiment datasets

Information	Dataset	
	Butterfly	Oriole
#Teaching samples	20	16
#Testing samples	10	8
#Classes	5	4
#Samples per class	300	60

In our experiment, the user will firstly be presented a fixed number of teaching samples one by one. In the teaching phase, the user will be provided the ground truth after she submits the answer of each question, and then our website chooses the next teaching sample to her using the adaptive learning algorithm introduced

[4] This function is just one possible candidate, and people can adopt other desired shape here.

in [9]. In the testing phase (i.e., formal task), no ground truth will be provided for each question and the questions are selected randomly to each user. The setups of our two experiment datasets are shown in Table 2.

It is worth mentioning that in our experiments, we found that users often answered questions quickly, basically around 2 s. Small time values can lead to a steep increase in the function defined in (1) and quick fluctuations on user's capability estimated with Eq. (2). This problem can be easily avoided by using a "virtual" time system, e.g., using 2000 ms instead of 2 s or increasing the real response time by a constant. The purpose of using virtual time is to smooth the fluctuation in a way that we can easily identify a user's learning pattern. Since too-large virtual times (e.g. thousands) will take a heavy toll on using Eq. (2) to distinguish different learning patterns, in our experiment the virtual time is determined by adding a small constant (2 s) to real response time.

6.2 Data

- **Question sets:** We select two scientific image sets to evaluate the performance of Goldilocks. The first dataset, called *Butterfly* dataset, consists of a total number of 1500 butterfly images in five categories from a museum collection [9]. The second image set, called *Oriole* dataset, includes 240 oriole images in four different species from a public dataset [19]. Correctly labeling these images requires domain knowledge, which normal users might not have in advance.
- **Data processing:** The scale-invariant feature transform (SIFT) algorithm [14] is an effective algorithm for image feature detection. We apply SIFT to calculate the keypoint feature vectors of each teaching sample, and then acquire the similarity between any two teaching samples with the following function:

  $$f(\rho) = \sum_{i=1}^{m} \mathbb{1}_{\{\rho_i > \theta\}} \tag{5}$$

  where $\mathbb{1}_{\{\rho_i > \theta\}}$ is an indicator function:

  $$\mathbb{1}_{\{\rho_i > \theta\}} = \begin{cases} 0 & \text{if } \rho_i > \theta \\ 1 & \text{otherwise} \end{cases} . \tag{6}$$

  Here function $\rho_i = \left\| \mathbf{x^k_{jw}} - \mathbf{x^k_{j'w}} \right\|_2^2$ is the Euclidean distance between the w-th $(0 < w < m)$ keypoint feature vectors of teaching images j and j', and θ is a self-defined similarity threshold between two different keypoint feature vectors.

  We implement the SIFT algorithm in C++. To make the labeling task nontrivial, we set the image blur-adjustment parameter λ of the Gaussian pyramid to 1.1, which introduces some difference to intra-category images and some similarity to inter-category images [14].

- **Data collection:** We share the Python-based Django website with 100 participants, we collect their performance data in the teaching phase and the testing phase, respectively, for both the *Butterfly* and *Oriole* image sets. While we have collected the data for the participants in both the teaching and testing phases, to compare Goldilocks and other baseline methods in terms of the quality of final returned sensing data, we can *pretend* that after the teaching phase, only X number of qualified users out of the 100 participants are selected in the testing phase, where X is a variable determined by the participant selection criteria. In this way, we have the ground truth regarding the data quality with and without those filtered users in the testing phase.

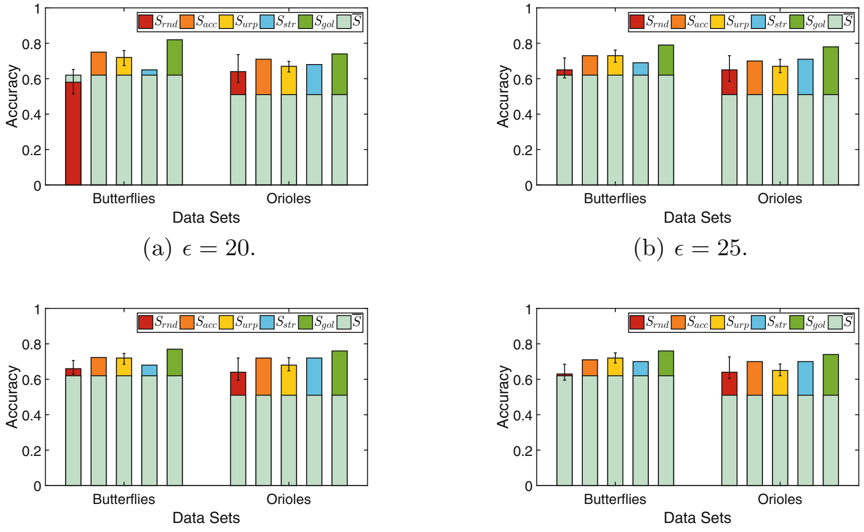
6.3 Baseline Algorithms

We compare our method with the following baseline methods:

- $\mathcal{S}_{acc}$: This method filters the candidates based on their accuracy in the teaching phase. The candidates with more correctly-labeled questions are chosen to participate in the formal tasks.
- $\mathcal{S}_{urp}$: This method [22] does not have an explicit teaching phase but it estimates users' reliability using the following *equivalent* method: After a user labels all the questions, we randomly select a small number of questions, and estimate the user's reliability based on whether or not she has correctly answered the chosen questions. In our experiment, we ask the users to label the 30 images from *Butterfly* dataset and randomly select 5 images to estimate the users' reliability. For the *Oriole* dataset, we ask the users to label the 20 images and randomly select 4 images to estimate the users' reliability. Users are then ranked based on their reliability from high to low, and we select candidates based on their reliability. We repeat this experiments 10 times and take the average over the 10 runs.
- $\mathcal{S}_{str}$: STRICT [18] is an optimized algorithm that selects all the teaching samples for a user before the training, based on the use's prior. In other words, the order of teaching samples is computed offline and *fixed* in the user's training process. It finally selects the users who have the best performances on the fixed teaching set.
- $\mathcal{S}_{rnd}$: It randomly selects the required number of candidates without any teaching. We take the average result over 50 random selections in all of our following experiments.

6.4 Evaluation Results and Analysis

Accuracy and F-Measure. Figure 6 shows the average labeling accuracy by users selected with different participant selection methods in the testing phase. For comparison, we set the number of selected users (i.e., ϵ), after the teaching phase, from 20 to 35 with the increment of 5. From the figure, we can see that on both datasets, all the methods perform better than the random selection. It is notable that when $\epsilon = 20$, the performance of random selected users is even worse than that of all the users. This is because the random selection might choose more unqualified users than qualified ones. We also observe that there is no fixed "second-best" across different methods and among different datasets. However, Goldilocks achieves the highest accuracy among all the five methods in all the ϵ settings on both datasets.



(a) $\epsilon = 20$.

(b) $\epsilon = 25$.

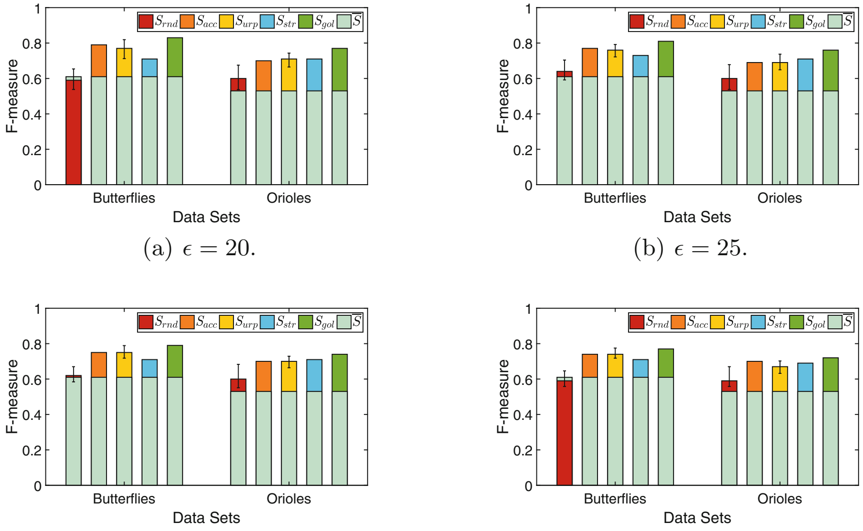
(c) $\epsilon = 30$.

(d) $\epsilon = 35$.

Fig. 6. Accuracy of different methods on different datasets when the number of selected users for the testing phase is different. $\overline{\mathcal{S}}$ denotes the performance that all users are selected for the testing phase.

F-measure is a harmonic mean of precise and recall that ranges from 0 to 1. In addition to accuracy, this metric also reflects the stability of a model. Figure 7 shows the f-measure value of different methods under different number of the selected users on the two experiment datasets. On both image sets, $\mathcal{S}_{rnd}$ has the worst f-measure performance among all the methods. Meanwhile, $\mathcal{S}_{gol}$ outperforms all the baselines across different numbers of selected users. The maximum f-measure value of Goldilocks can reach 0.83 when $\epsilon = 20$ on the *Butterfly* dataset. Also, no baseline works consistently the second best w.r.t. f-measure. From Fig. 7, we can also observe that the f-measure performance is generally poorer on the *Oriole* dataset than on the *Butterfly* dataset. This may be because image samples of orioles are more difficult to classify due to distracting environmental background.

Response Time. If a user can answer questions quickly with a high accuracy, the user must have gained good knowledge for the task. We therefore analyze



(a) $\epsilon = 20$.

(b) $\epsilon = 25$.

(c) $\epsilon = 30$.

(d) $\epsilon = 35$.

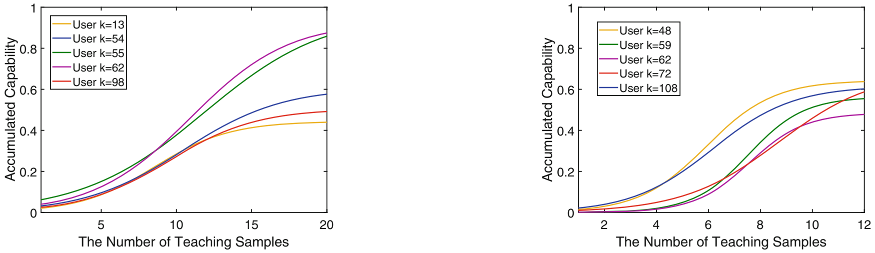
Fig. 7. F-measure of different methods on different datasets when the number of selected users for the testing phase is different. $\mathcal{S}$ denotes the performance that all users are selected for the testing phase.

the response time of those users who are chosen for the testing phase. Here the response time is defined as from the time when a selected user is shown the first image to the time when she submits the answer of the last image during the testing phase. Table 3 summarizes the average response time of users selected by different methods on two different datasets. We can observe that the users selected by Goldilocks tend to respond more quickly compared to other baselines. Considering Goldilocks' good performance in accuracy and F-measure, we have enough confidence to conclude that the users selected by Goldilocks have a better grasp of the knowledge. Although it can be seen that $\mathcal{S}_{rnd}$ and $\mathcal{S}_{urp}$ also perform well on the average response time in some ϵ settings, users' poorer performance on the accuracy and f-measure implies that the selected users may not be reliable.

Table 3. Average response time in the testing phase

		$\epsilon = 20$		$\epsilon = 25$		$\epsilon = 30$		$\epsilon = 35$	
		Butterfly	Oriole	Butterfly	Oriole	Butterfly	Oriole	Butterfly	Oriole
Method	$\mathcal{S}_{rnd}$	**41.35**	28.05	**44.32**	28.18	47.50	27.96	47.00	28.41
	$\mathcal{S}_{acc}$	47.60	34.90	46.60	35.44	45.00	34.13	47.26	33.60
	$\mathcal{S}_{urp}$	46.80	28.60	46.36	27.48	43.50	**26.80**	45.26	**26.66**
	$\mathcal{S}_{str}$	49.45	25.69	54.35	27.88	52.30	30.65	48.37	31.89
	$\mathcal{S}_{gol}$	42.96	**21.25**	45.30	**23.12**	**42.53**	27.40	**45.08**	28.03

Note: ϵ denotes the number of selected users for test



(a) *Butterfly* dataset.

(b) *Oriole* dataset.

Fig. 8. Fitted logistic regression curves of the top 5 user.

Fitting Degree of Learning Curve. Goldilocks uses the logistic function introduced in Sect. 5 to benchmark a user's learning curve. Note that a user's learning curve is obtained from the round-by-round capability adjustment (Algorithm 1). To ease illustration, we normalize the capability values with the maximum possible value. The learning curves of the selected top five users on the two datasets are shown in Fig. 8(a) and (b), respectively. We can see that all the learning curves have a similar shape of the logistic regression function, indicating the suitability of logistic regression.

Table 4. Fitting performance of Top 5 learning curves and the corresponding accuracy in the testing phase on the Butterfly dataset

User	Metrics			Accuracy
	SSE	RMSE	Adj-rsquare	
13	0.0244	0.0379	0.9638	90%
54	0.0621	0.0605	0.9702	90%
55	0.0399	0.0485	**0.9837**	**100%**
62	0.0226	0.0364	0.9721	90%
98	0.0304	0.0423	0.9715	90%

Table 5. Fitting performance of Top 5 learning curves and the corresponding accuracy in the testing phase on the Oriole dataset

User	Metrics			Accuracy
	SSE	RMSE	Adj-rsquare	
48	0.0113	0.0354	0.9869	87.5%
59	0.0146	0.0403	0.9805	87.5%
62	0.0052	0.0240	**0.9904**	**100%**
72	0.0143	0.0399	0.9776	87.5%
108	0.0080	0.0299	0.9887	87.5%

We, however, need a quantitative evaluation on the fitting degree of a learning curve towards the logistic function. Note that we only care about the shape (i.e., the trend) and thus we do not suggest a fixed logistic function. Instead, with logistic regression, different learning curves may fit to different logistic functions (i.e., the parameters in the fitted logistic functions may be different). To evaluate the fitting degree, we test the following metrics:

- *SSE*: The sum of squared errors (SSE) between the learning curve and the fitted logistic function (at discrete range values). The smaller the SSE, the better the fitting.
- *RMSE*: Root Mean Square Error. The smaller the RMSE, the better the fitting.
- *Adj-rsquare*: The adjusted R-square value according to the freedom degree of errors, where R-square is the square of the correlation coefficient between the measured data and the data obtained by the fitting curve. The higher the Adj-rsquare, the better the fitting.

Table 4 shows that in the *Butterfly* dataset, both SSE and RMSE are less than 6.5% for all the 5 users' learning curves. And the values of their Adj-rsquare are all beyond 96%. Considering the high accuracy of these 5 users, we can conclude that the users whose learning curve well follow our proposed model have high-quality answers in the testing phase. This is further validated by the fact that the user who has the highest Adj-rsquare (i.e., the 55^{th} user) answers all the questions correctly. Table 5 shows the similar phenomena as in Table 4. For the *Oriole* dataset, the values of SSE are less than 1.5% and the values of RMSE are less than 4.1% for all the top five users' learning curves. The 62^{th} user has the best Adj-rsquare (99.04%) and the highest accuracy (100%).

7 Conclusions

Participant selection has always been a critical problem in MCS. To "select right people for right job", existing methods usually use the number of questions that a user answered correctly during the training phase to judge the user's qualification. We proposed an enhancement method, called Goldilocks, which (1) estimates users' learning patterns using adaptive teaching and (2) selects users based on the desired learning patterns. Experiments with real-world data disclose that Goldilocks outperforms existing baselines in terms of efficiency, accuracy, and stability.

Acknowledgement. This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant No. SPG 494083-16.

References

1. Ahlgren, B., Hidell, M., Ngai, E.C.H.: Internet of Things for smart cities: interoperability and open data. IEEE Internet Comput. **20**(6), 52–56 (2016)
2. Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B., Allahbakhsh, M.: Quality control in crowdsourcing: a survey of quality attributes, assessment techniques, and assurance actions. ACM Comput. Surv. (CSUR) **51**(1), 7 (2018)
3. Du, J., Ling, C.X.: Active teaching for inductive learners. In: Proceedings of the 2011 SIAM International Conference on Data Mining, pp. 851–861. SIAM (2011)
4. Ebbinghaus, H.: Memory: a contribution to experimental psychology. Ann. Neurosci. **20**(4), 155 (2013)
5. Fiandrino, C., Kantarci, B., Anjomshoa, F., Kliazovich, D., Bouvry, P., Matthews, J.: Sociability-driven user recruitment in mobile crowdsensing Internet of Things platforms. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2016)
6. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. IEEE Commun. Mag. **49**(11), 32–39 (2011)
7. Grantcharov, T.P., Bardram, L., Funch-Jensen, P., Rosenberg, J.: Learning curves and impact of previous operative experience on performance on a virtual reality simulator to test laparoscopic surgical skills. Am. J. Surg. **185**(2), 146–149 (2003)
8. Guo, B., Liu, Y., Wu, W., Yu, Z., Han, Q.: Activecrowd: a framework for optimized multitask allocation in mobile crowdsensing systems. IEEE Trans. Hum.-Mach. Syst. **47**(3), 392–403 (2017)
9. Johns, E., Mac Aodha, O., Brostow, G.J.: Becoming the expert-interactive multiclass machine teaching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2616–2624 (2015)
10. Jones, V., Jo, J.H.: Ubiquitous learning environment: an adaptive teaching system using ubiquitous technology. In: Beyond the Comfort Zone: Proceedings of the 21st ASCILITE Conference, vol. 468, p. 474. Perth, Western Australia (2004)
11. Karaliopoulos, M., Telelis, O., Koutsopoulos, I.: User recruitment for mobile crowdsensing over opportunistic networks. In: 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 2254–2262. IEEE (2015)
12. Li, H., Li, T., Wang, Y.: Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks. In: 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems, pp. 136–144. IEEE (2015)
13. Liu, C.H., Zhang, B., Su, X., Ma, J., Wang, W., Leung, K.K.: Energy-aware participant selection for smartphone-enabled mobile crowd sensing. IEEE Syst. J. **11**(3), 1435–1446 (2017)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
15. Patil, K.R., Zhu, J., Kopeć, Ł., Love, B.C.: Optimal teaching for limited-capacity human learners. In: Advances in Neural Information Processing Systems, pp. 2465–2473 (2014)
16. Shah, N., Zhou, D., Peres, Y.: Approval voting and incentives in crowdsourcing. In: International Conference on Machine Learning, pp. 10–19 (2015)
17. Singla, A., Bogunovic, I., Bartók, G., Karbasi, A., Krause, A.: On actively teaching the crowd to classify. In: NIPS Workshop on Data Driven Education (2013)
18. Singla, A., Bogunovic, I., Bartók, G., Karbasi, A., Krause, A.: Near-optimally teaching the crowd to classify. In: ICML, no. 2 in 1, p. 3 (2014)

19. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset. California Institute of Technology (2011)
20. Wang, E., Yang, Y., Wu, J., Liu, W., Wang, X.: An efficient prediction-based user recruitment for mobile crowdsensing. IEEE Trans. Mob. Comput. **17**(1), 16–28 (2018)
21. Xu, Q., Zheng, R.: When data acquisition meets data analytics: a distributed active learning framework for optimal budgeted mobile crowdsensing. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pp. 1–9. IEEE (2017)
22. Yang, S., Han, K., Zheng, Z., Tang, S., Wu, F.: Towards personalized task matching in mobile crowdsensing via fine-grained user profiling. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 2411–2419. IEEE (2018)
23. Yin, X., Goudriaan, J., Lantinga, E.A., Vos, J., Spiertz, H.J.: A flexible sigmoid function of determinate growth. Ann. Bot. **91**(3), 361–371 (2003)
24. Zhang, X.Y., Wang, S., Yun, X.: Bidirectional active learning: a two-way exploration into unlabeled and labeled data set. IEEE Trans. Neural Netw. Learn. Syst. **26**(12), 3034–3044 (2015)
25. Zhao, L., Sukthankar, G., Sukthankar, R.: Incremental relabeling for active learning with noisy crowdsourced annotations. In: 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, pp. 728–733. IEEE (2011)
26. Zhu, J.: Machine teaching for Bayesian learners in the exponential family. In: Advances in Neural Information Processing Systems, pp. 1905–1913 (2013)
27. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International conference on Machine learning (ICML 2003), pp. 912–919 (2003)