Worker Selection Towards Data Completion for Online Sparse Crowdsensing

Wenbin Liu[1,2], En Wang[1,2,*], Yongjian Yang[1,2], and Jie Wu[3]
[1]College of Computer Science and Technology, Jilin University, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China
[3]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
Email: {liuwenbin,wangen,yyj}@jlu.edu.cn, jiewu@temple.edu

Abstract*—As a cost-effective paradigm, Sparse Crowdsensing aims to recruit workers to perform a part of sensing tasks and infer the rest, which has broad applications, including environmental monitoring, urban sensing, etc. In most cases, workers will participate in real time, and thus their sensing data are coming dynamically. Taking full advantage of the online coming data to complete the full sensing map is an important problem for Sparse Crowdsensing. However, for data completion, the importance of data collected from different spatio-temporal areas is usually different and time-varying. For example, the newly obtained data in the center is often more important than the old ones from edges. Moreover, the area importance may also influence the following worker selection, i.e., selecting suitable workers to actively sense important areas (instead of passively waiting for given data) for improving completion accuracy. To this end, in this paper, we propose a framework for online Sparse Crowdsensing, called OS-MCS, which consists of three parts: matrix completion, importance estimation, and worker selection. We start from the dynamically coming data and propose an online matrix completion algorithm with spatio-temporal constraints. Based on that, we estimate the spatio-temporal area importance by conducting a reinforcement learning-based up-to-date model. Finally, we utilize the prophet secretary problem to select suitable workers to sense important areas for accurate completion in an online manner. Extensive experiments on real-world data sets show the effectiveness of our proposals.

***Index Terms*—Online Sparse Crowdsensing, Matrix Completion, Worker Selection.**

I. Introduction

With the explosive growth of portable devices, Mobile CrowdSensing (MCS) [1] becomes a promising paradigm that recruits crowd workers to perform a wide variety of sensing tasks at the target time and locations [2]. However, in some large-scale tasks [3], e.g., environmental monitoring, urban sensing, spatial crowdsourcing, the large number of sensing areas and long-term continuous sensing requirements make the traditional MCS consume a lot to cover the full map at all times. To this end, a modified cost-effective paradigm has been proposed to recruit workers to perform only a part of sensing tasks and infer the rest by exploiting the inherent data correlations, called Sparse Crowdsensing [4].

Recently, with the low cost and high accuracy, Sparse Crowdsensing has drawn increasing attention, while most of the existing works are conducted offline [5]–[12]. As shown in Fig. 1 (upper part), the offline scheme first recruits workers from a known pool to perform a few sensing tasks, and then infer the rest after receiving all the data. However, in most cases, *workers will participate in real time, and thus the sensing data provided by them are coming dynamically*. For such online scenarios, we not only have to decide whether to select one worker in an online manner, but also need to make full use of the dynamically coming data. Intuitively, we can simply and directly complete the full sensing map after receiving each new piece of data. Obviously, it costs a lot and may cause high completion latency due to heavy computation load. Another alternative is to group data into batches and infer only once after receiving the data of each batch. However, this method could not take full advantage of the online coming data because there is a lag between receiving data and exploiting data. Hence, how to effectively exploit the dynamically coming data for *online data completion* is the first challenge.

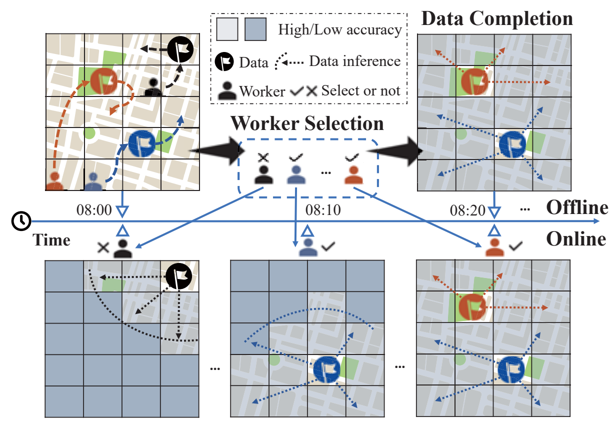


Fig. 1: Example of Offline/Online Sparse Crowdsensing.

Note that the importance of data collected from each sensing area is usually different and time-varying, which can be utilized to improve the data completion. As shown in Fig. 1 (lower part), the blue worker can collect data from a center area, which is usually more important than the corner data collected by the black worker. Similarly, the newly obtained data is obviously more useful to do data completion compared with the old one collected from the same area. Thus, to improve the data completion under ever-changing online scenarios, *area importance estimation* becomes the second challenge.

* En Wang is the corresponding author.

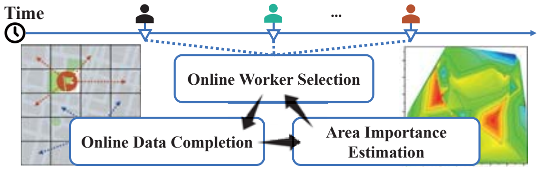


Fig. 2: Workflow of OS-MCS.

Moreover, instead of passively waiting for given data, we prefer to select suitable workers to actively sense important areas for accurate data completion by utilizing area importance estimation. However, in online scenarios, the workers and their provided data are invisible and hard to predict until they participate in Sparse Crowdsensing. Hence, how to *select the worker set in an online manner* to sense important areas for accurate data completion is the third challenge.

To tackle these challenges, in this paper, we propose an efficient framework for online Sparse Crowdsensing, called OS-MCS, which consists of three parts: data completion, importance estimation, and worker selection. As shown in Fig. 2, for each coming worker, we first decide whether to select it or not according to the model of importance estimation; if the worker is selected, we then use its collected data to complete the full sensing map; the online data and the completed full map will also be sent to keep the estimation model up-to-date. Thus, data completion, importance estimation and worker selection form a complex circular triangle relationship. For **data completion**, we conduct statistical data analysis on real-world data sets. In fact, there are many correlations in sensing data, e.g., the temporal and spatial continuity, periodicity, and similarity. Combining a low-rank representation model, we propose an online matrix completion algorithm with spatio-temporal constraints in order to complete the full map in an online manner. Based on that, for **importance estimation**, we conduct a reinforcement learning-based method to evaluate that when and where to collect data may help more on complicated data completion. Note that we not only use the model learned by reinforcement learning from the collected data, but also conduct an up-to-date training data set to deal with the ever-changing online scenarios. For **worker selection**, we utilize the data importance estimation to connect the worker selection with data completion, i.e., to select workers to actively collect important data for accurate completion. We formulate the worker selection as the classic secretary problem and introduce the prophet inequality to select the best workers according to a changing threshold from the historical records.

In summary, this paper makes the following contributions:

- *Online Sparse Crowdsensing*: We investigate the Sparse Crowdsensing under online scenarios. We select workers to perform a part of sensing tasks while inferring the rest in a more practical online manner, by dealing with the workers participating in real time and the dynamically coming data they provide.
- *Framework OS-MCS*: We propose an efficient framework for online Sparse Crowdsensing, called OS-MCS. We first propose an online matrix completion algorithm under spatio-temporal constraints. Based on that, we conduct a reinforcement learning-based importance estimation with an up-to-date model. Finally, a prophet secretary-based online worker selection strategy is presented, aiming to select workers to actively sense important areas for data completion in an online manner.
- *Extensive Evaluation*: We conduct extensive evaluations on five typical sensing tasks with real-world data sets, which verify the effectiveness of our proposed methods for online Sparse Crowdsensing.

II. Related Work

Mobile CrowdSensing is a promising paradigm that utilizes crowd of workers to perform various sensing tasks [1]. To achieve accurate and complete sensing results, traditional MCS works have to recruit a large number of workers to cover all tasks [2], which obviously consumes a lot and still cannot deal with the tasks with no available workers. To this end, a modified cost-effective paradigm, called Sparse Crowdsensing [4], is proposed that only needs to perform a part of sensing tasks while inferring the rest. However, most of the existing Sparse Crowdsensing works are conducted offline [5]–[12], which ignore the more practical online scenarios that the workers usually participate in real time, and thus the sensing data provided by them are coming dynamically.

In Sparse Crowdsensing, researchers mainly focus on data completion. Recently, with the rapid development of sparse representation and other technologies, compressive sensing [5]–[9] and matrix completion [10]–[12] have gradually become the *de facto* choices in Sparse Crowdsensing. However, these existing works have not completely explored the spatio-temporal correlations in the sensing data. There are also some works on cell selection [13]–[17], which mainly measure the uncertainty of subarea by entropy or reinforcement learning-based methods, but not when and where to collect data. For worker selection, which is actually the foundation of Sparse Crowdsensing, there only exists few works that use the cost order [6], [7], [10] or try to estimate the data inference accuracy [8] to select workers. However, all of these above works cannot be directly used in such online scenarios. In this paper, we aim to conduct an efficient framework for online Sparse Crowdsensing, which conducts online data completion, keeps importance estimation up-to-date, and selects workers in a totally online manner, aiming to recruit suitable workers to actively sense important areas for data completion.

III. Model, Problem, and Framework Overview

A. System Model

We first introduce the system model of our online Sparse Crowdsensing, whose main notations are listed in Table I.

Tasks. We consider a general MCS scenario, where m target sensing areas should be sensed with a duration T. Considering the continuity and stability in sensing data (will be illustrated in Section V), we can split the duration T into n cycles. We assume that the cycles are short enough that the sensing data in each cycle will not change much. In this way, the MCS

TABLE I: Main notations

Notation	Meaning
m, n	Number of sensing areas and cycles.
T, t	Duration and current time.
Y, Y', M	Full matrix, sensed matrix, and mask matrix.
$\hat{Y}$, ε	Completed matrix and completion error.
w, k	Number of total workers and selected workers.
W, μ	Worker set and selected worker set.
B, c	Budget and cost.
U, V, r	Latent spatio-temporal matrices and rank.
S, A, R	State, action, and reward.

campaign actually can be seen as a combination of $m \times n$ sensing tasks, each of which should be performed at a specific location and time.

Data. Let $Y_{m\times n}$ be the full sensing matrix with m sensing areas and n cycles, where each $Y[i,j] = y_{i,j}$ denotes the true value in the i-th sensing area at the j-th cycle. In Sparse Crowdsensing, we only need to perform a part of tasks while inferring the rest. We record the actually sensed matrix $Y' = Y \otimes M$, where $\otimes$ denotes an element-wise product and $M[i,j] = \{1,0\}$ records whether the task is completed or not. Under online scenarios, Y' and M will change along with time t, and thus can be denoted as Y'_t and M_t, where we consider a discrete time model, i.e., $t = \{1, 2, \ldots, T\}$, for simplicity.

Importance. We use the data completion method $f(\cdot)$ to complete a full map $\hat{Y}$ from the actually sensed matrix Y', aiming to achieve an approximation of Y, i.e., $f(Y') = \hat{Y} \approx Y$. The error of online data completion at time t is denoted as $\varepsilon(Y, \hat{Y}_t) = \sum_{i=1}^{m} |Y[i,t] - \hat{Y}[i,t]|$, which is actually the completion accuracy of the current sensing cycle and thus reflects the importance of the current sensed Y'_t.

Workers. Let $W = \{u_1, u_2, \ldots, u_w\}$ be the worker set with a total of w workers, each with a cost c_i. In the online scenarios, a worker u_i will participate in MCS campaign at time t_i with location l_i, and we should immediately decide whether to select her to perform the task with it location and time[1], according to the estimated importance and the remaining budget B. The selected worker set is denoted as μ with cardinality k, and thus we have $\sum_{i=1}^{k} c_i \le B$.

B. Problem Formulation

Problem [Worker Selection Towards Data Completion for Online Sparse Crowdsensing]: *Given a set of tasks with m sensing areas and n cycles, with a budget B and a duration T, our problem is to select a set of sequential participating workers μ, with the objective of minimizing the total completion error during the whole online processing*:

$$\text{minimize} \quad \sum\nolimits_{t=1}^{T} \varepsilon(Y, \hat{Y}_t) \tag{1}$$

$$\text{subject to} \quad \hat{Y}_t = f(Y'_t),\ \mu \subseteq W,\ \sum\nolimits_{u_i \in \mu} c_i \le B \tag{2}$$

[1]To simplify the problem, we assume that each selected worker will immediately and successfully perform the task at its time and location (matches the sensing cycle and area). The more complicated situations can be easily modified in our worker selection method.

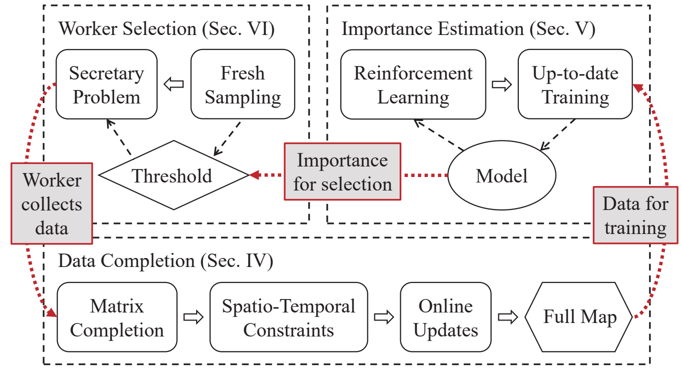


Fig. 3: Overview of OS-MCS.

Lemma 1. *The online worker selection problem is NP-hard.*

Proof: Without loss of generality, we ignore the complicated data completion but consider that adding a new worker/data will reduce the error. Then, we can change the monotonous objective function from minimizing error to maximizing negative error. Further consider that the workers have the same cost c, and thus we can select at most $k = \lfloor B/c \rfloor$ workers. Then, the online worker selection problem is reduced to the subset selection problem, i.e., selecting k-element subset to maximize a set function, which is a classic NP-hard problem [18]. Thus, the simplified case is NP-hard. Consequently, further considering the complicated data completion and ever-changing online scenario, our online worker selection problem is NP-hard. ■

C. Framework Overview

To solve this problem, in this paper, we propose a framework for online Sparse Crowdsensing, called OS-MCS, which consists of three parts: data completion, importance estimation, and worker selection, as illustrated in Fig. 3.

- **Worker Selection:** for each coming worker, we first use the worker selection component to decide whether to select it or not. We first conduct a fresh-looking sampling from the historical records and the workers coming ahead. Then, we calculate a threshold according to the samples' (workers') spatio-temporal areas by importance estimation. Finally, we select suitable workers by a prophet-secretary-based strategy, aiming to actively sense important areas for data completion.
- **Data Completion:** if the coming worker is selected, we then use its newly provided data, combining the previous collected ones, to complete the full map in the data completion component. To make full use of the dynamically coming data, we propose a matrix completion algorithm with spatio-temporal constraints and online updates.
- **Importance Estimation:** we present the importance estimation component to measure when and where to collect data are more helpful, in order to improve the data completion accuracy and guide the active worker selection. Specifically, based on the completion algorithm, we utilize a reinforcement learning-based method to train a model to estimate the importance of spatio-temporal

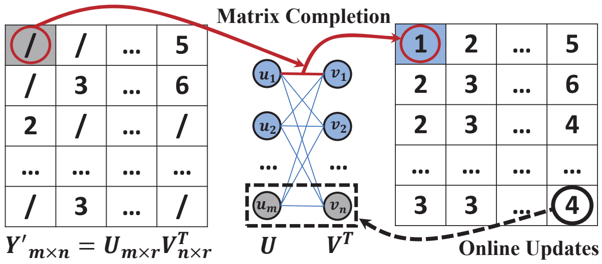


Fig. 4: Data Completion with Online Updates.

areas. Furthermore, we also use the online coming data to keep the trained model up-to-date.

In the following, we first introduce the foundation online data completion in Section IV; based on that, we present the up-to-date importance estimation in Section V; finally, we propose the more active online worker selection (instead of passively waiting for given data) in Section VI.

IV. Online Data Completion

We first study the basic data completion component, where spatio-temporal constraints and online updates are added, to complete the full matrix from a part of online coming data.

A. Matrix Completion

In the physical world, the sensing data naturally exist with some correlations, which leads that the full data matrix can be approximated to a low-rank matrix [4]. Given the incomplete sensing matrix Y', we can complete the full matrix $\hat{Y}$ by utilizing the low-rank property:

$$\min \ \mathrm{rank}(\hat{Y}), \qquad \text{s.t.}, \hat{Y} \otimes M = Y'. \tag{3}$$

Since the above optimization is nonconvex and thus hard to be solved, as shown in Fig. 4 (red lines), we then factor the low-rank matrix $\hat{Y}$ into the product of two tall matrices, i.e., $\hat{Y}_{m\times n} = U_{m\times r}V^T_{n\times r}$, where U and V can be seen as the latent spatio-temporal feature matrices and r is the latent rank that $r \le \min\{m, n\}$. Note that minimizing the rank of $\hat{Y}$ is equivalent to minimizing $\|U\|_F^2 + \|V\|_F^2$ under certain conditions [19], we then revise Eq. 3 as:

$$\min \ \|(Y' - UV^T) \otimes M\|_F^2 + \lambda(\|U\|_F^2 + \|V\|_F^2), \tag{4}$$

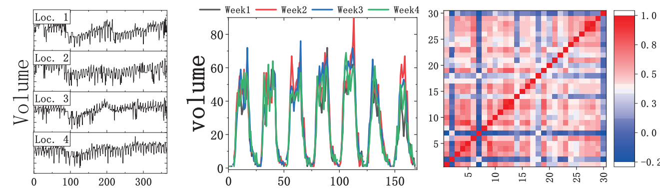
where $\|(Y' - UV^T) \otimes M\|_F^2$ represents the completion error based on the Frobenius norm $\|\cdot\|_F$ and λ makes a trade-off between rank minimization and accuracy fitness.

B. Spatio-Temporal Constraints

The optimization problem in Eq. 4 is the basic model for matrix completion, which only relies on the global low-rank property. Since the sensing data usually exhibit strong spatial and temporal correlations, we would like to exploit them to further improve the completion accuracy.

As shown in Figs. 5 (a-c), we conduct a simple statistical data analysis on *Traffic*[2], which reflects some common and effective spatio-temporal correlations, i.e., the continuity, periodicity, and similarity. Typically, such correlations can be

[2] *Traffic* is a real sensing application that records the real-time traffic volume in different subway stations, which will be described in detail in Section VII.



(a) Continuity (b) Periodicity (c) Similarity

Fig. 5: Statistical Data Analysis on *Traffic*.

formulated as the spatio-temporal constraints and added into Eq. 4, as follows:

$$\min \|(Y' - UV^T) \otimes M\|_F^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) + g(\mathbb{T}) + h(\mathbb{S}), \tag{5}$$

where $g(\mathbb{T})$ and $h(\mathbb{S})$ are spatio-temporal constraint functions.

For temporal correlations, we first consider the temporal continuity. Let $\mathbf{y}^T$ be the column vector of the matrix Y, i.e., $Y = [\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_n^T]$. As shown in Fig. 5 (a), we observe that the values of sensing data only change in a small range between adjacent times, thus we have

$$\min \sum\nolimits_{i=2}^{n} \|\mathbf{y}_i - \mathbf{y}_{i-1}\|_2. \tag{6}$$

Let $\mathbf{v}_i$ be the column vector of the tall matrix V, we have

$$\begin{aligned} \|\mathbf{y}_i - \mathbf{y}_{i-1}\|_2 &= \|(U\mathbf{v}_i^T)^T - (U\mathbf{v}_{i-1}^T)^T\|_2 \\ &\le \|U\|_2\|\mathbf{v}_i - \mathbf{v}_{i-1}\|_2 \le \|U\|_2\|\mathbf{v}_i - \mathbf{v}_{i-1}\|_1, \end{aligned} \tag{7}$$

where ℓ_1 norm is more robust on noise [20]. Then, we can use a matrix $\mathbb{T}_c =$Toeplitz$(0, 1, -1)$ to obtain the temporal constraint function, as follows:

$$g(\mathbb{T}) = \|V^T\mathbb{T}_c\|_1, \tag{8}$$

where

$$\mathbb{T}_c = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 1 & -1 & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & 1 \end{bmatrix}_{n\times(n-1)}. \tag{9}$$

This matrix intuitively constraints that values from same location at two continuous cycles are usually similar.

As shown in Fig. 5 (b), except for continuity, the sensing values also exist strong periodicity from hour to hour, day to day, etc, which can be learned by the classic time-series analysis in statistics [21]. Thus, we further utilize it to modify the temporal constraint matrix $\mathbb{T}$, by moving 1 to the interval line, i.e., $\mathbb{T}_p[i,:] = [\ldots, -1, \ldots, 1, \ldots]^T$, $\forall 1 \le i \le n$. Also, 1 and -1 can be changed to the similar ratios to reflect the trend. Then, we revise the temporal constraint matrix as $\mathbb{T} = (1 - \lambda_p)\mathbb{T}_c + \lambda_p\mathbb{T}_p$ and reformulate the Eq. 5, as follows:

$$\begin{aligned} \min \ & \|(Y' - UV^T) \otimes M\|_F^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) \\ & + \lambda_t\|V^T\mathbb{T}\|_1 + h(\mathbb{S}), \end{aligned} \tag{10}$$

where λ_p is used to balance the continuity and periodicity, and λ_t is the temporal weight. Actually, $\mathbb{T}_c$ maintains the stability and $\mathbb{T}_p$ reflects the characteristics of the sensing data.

For spatial correlations, similar with the temporal ones, we mainly consider the spatial continuity and similarity. In most cases, the closer locations usually have the similar values of sensing data. As shown in Fig. 5 (c), we use the Pearson

Correlation Coefficient to calculate the similarity between locations. We can find that not only the nearby locations, but also some far away locations have similar surroundings, or other similar conditions have similar sensing values. Thus, inspired by the temporal constraint matrix $\mathbb{T}$, we first use the Euclidean distance to model the spatial continuity constraint matrix $\mathbb{S}_c$ for $m \times m$ pairs of locations, as follows:

$$\mathbb{S}_c[i,j] = \begin{cases} d_{i,j} / \sum_{t \neq i} d_{i,t}, & i \neq j \\ -1 & i = j \end{cases} \tag{11}$$

where we set $d_{i,j} = e^{-\mathrm{dis}(loc_i, loc_j)/\sigma_c^2}$ and $\mathrm{dis}(loc_i, loc_j) = \sqrt{(loc_i.x - loc_j.x)^2 + (loc_i.y - loc_j.y)^2}$.

Note that not only the adjacent locations, but also some relatively distant locations may have similar sensing values. As shown in Fig. 5 (c), Loc. 1 is far away from Loc. 3, but they still have similar trend and values. The reason is that both Locs. 1 and 3 are teaching buildings, i.e., they share the same physical attributes characteristics. Similar, other factors, such as the similar surroundings, PoIs, topography, etc, may also lead to similar values. In addition to prior expert knowledge, we can also learn such spatial correlations from the values. We conduct the similarity constraint matrix $\mathbb{S}_s$ as:

$$\mathbb{S}_s[i,j] = s_{i,j} / \sum\nolimits_{t \neq i} s_{i,t}, \; i \neq j, \tag{12}$$

where $s_{i,j} = e^{-|y_i - y_j|/\sigma_s^2}$ and $\mathbb{S}_s[i,i] = -1, i = 1, \ldots, m$. Then, doing the same with the temporal constraint function, we revise the spatial constraint matrix as $\mathbb{S} = (1-\lambda_q)\mathbb{S}_c + \lambda_q \mathbb{S}_s$ and reformulate the Eq. 10, as follows:

$$\begin{aligned} \min \; & \|(Y' - UV^T) \otimes M\|_F^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) \\ & + \lambda_t \|V^T \mathbb{T}\|_1 + \lambda_s \|\mathbb{S}U\|_1, \end{aligned} \tag{13}$$

where λ_q and λ_s are nonnegative weights to balance the corresponding terms. Similarly, $\mathbb{S}_c$ holds the stability and $\mathbb{S}_s$ portrays the characteristics. In this way, we directly impose the spatio-temporal constraints on latent feature matrices, which further guide the direction to help data completion.

C. Online Updates

In order to deal with the problem in Eq. 13 under online scenarios for practical applications, several heuristics have been proposed, which first approximately complete the full matrix and conduct a fast update for each new coming data. One of those is Stochastic Gradient Descent (SGD) [22], which is a well known fast algorithm that updates the spatio-temporal matrices towards the new data. However, to achieve a good performance, SGD requires a tedious adjustment and fine tuning of the stepsize [23]. Thus, we utilize a modified Alternating Least Squares (ALS), which updates the spatio-temporal matrices U and V iteratively according to Eq. 13, in order to obtain the completed full matrix by $\hat{Y} = UV^T$.

ALS is a two-step iterative method, which first fixes U to calculate V and then turns around iteratively. Suppose that the spatio-temporal matrices U' and V' have already been estimated based on the current Y' after many iterations. When a new data $y_{i,j}$ is coming, it is quite intuitive that the new estimated U and V are close to the previous U' and V'.

Algorithm 1 Matrix Completion with Online Updates

Input: collected data: M, Y'; new data: $y_{i,j}$; previous latent spatio-temporal matrices: U, V

1: fix $\mathbf{u}_i$, calculate $\mathbf{v}_j$ according to $y_{i,j} = \mathbf{u}_i \mathbf{v}_j^T$;
2: **while** not convergent **do**
3:     fix $\mathbf{v}$, calculate $\mathbf{u}_i$ according to Eq. 13;
4:     fix $\mathbf{u}$, calculate $\mathbf{v}_j$ according to Eq. 13;

Actually, as shown in Fig. 4 (black line), only the rows $\mathbf{u}_i$ and $\mathbf{v}_j$ in U' and V' will be impacted by the new data $y_{i,j}$. Thus, the online updates should be conducted on such two rows of two tall matrices. Note that such efficient online updates are very popular and well applied in the recommender systems domain. Thus, for the online Sparse Crowdsensing, the matrix completion algorithm with online updates is summarized in Algorithm 1. We first fix the relatively stable spatial vector $\mathbf{u}_i$ and use the new data $y_{i,j}$ to calculate the temporal vector $\mathbf{v}_j$ (line 1). Then, we update these two vectors iteratively until convergent (lines 3-4). In addition, since the latent rank r is unknown, we also should conduct a grid search to balance the accuracy and efficiency in practice.

V. Importance Estimation

Based on the above online data completion, we further study the importance estimation to measure when and where to collect data can help more on data completion, which will further be used to guide the worker selection in Section VI.

A. Reinforcement Learning-based Estimation

Considering the spatio-temporal correlations, the data collected from important spatio-temporal areas may play a significant role in improving the completion accuracy. For example, when we have collected data from the same area for many cycles, the data far away from this area can often provide more information for data completion. However, due to the complicated data completion method and ever-changing online scenarios, we can hardly identify when and where to collect data can help more on data completion.

To tackle that, we propose to use Reinforcement Learning (RL) to connect the spatio-temporal area with completion accuracy directly, in order to estimate its importance for completion. RL can be abstracted as 'an *agent* decides the next *action* under a certain *state* to maximize the total reward', which is correspondingly interpreted as 'the *Sparse Crowdsensing* decides the next *spatio-temporal area* under the *already collected data* to maximize the *completion accuracy*'. Note that our proposed RL-based method is not used to select the next sensing area and cycle, we mainly use the model learned by RL to estimate the importance, i.e., the expected completion accuracy, of each spatio-temporal area, which is used to guide us to select suitable workers to actively sense importance spatio-temporal areas for data completion. As shown in Fig. 6, we formally model the three key concepts.

State (denoted as S) represents the data collection. We consider the mask matrix M as the state that records when

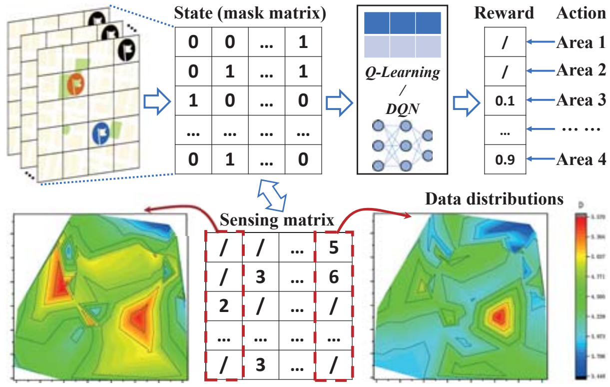


Fig. 6: Reinforcement learning-based importance estimation.

and where we have collected data, i.e., $M[i,j] = \{1, 0\}$ means that $y_{i,j}$ has been collected or not. Note that we don't add the collected data $y_{i,j}$ and other possible factors in state. On the one hand, it avoids dimension explosion; on the other, we focus on the inherent spatio-temporal correlations in the sensing data, and other factors can be easily modified.

Action (denoted as A) represents the possible spatio-temporal sensing area. Considering the online scenarios, we cannot collect data from the previous cycles and thus we use the current sensing areas as the actions, denoted as $A = \{a_1, a_2, \dots, a_m\}$. Note that if we further consider that workers can collect data at future several cycles, the actions can be set as the area-cycle pairs.

Reward (denoted as R) represents the completion accuracy of each action under a certain state. We directly use the completion error to formulate the reward, as follows:

$$R = e^{-\varepsilon(Y,\hat{Y}_t)/\sigma_e^2} = e^{-\sum_{i=1}^{m}|Y[i,t]-\hat{Y}[i,t]|/\sigma_e^2}. \quad (14)$$

Note that we don't add some penalty terms since the reward is not used to guide selection but to estimate completion accuracy, and such estimations are not the accurate values of errors but only used to compare them. We also use the total error but not the average one to increase the range.

With the above state, action, and reward, we then propose our RL-base importance estimation method. Specifically, under a certain state, we need to learn a Q-function that can output a reward for each possible action. In general, Q-function can be formulated as a Q-table or neural networks, known as the famous Q-Learning (QL) [24] and Deep Q-Network (DQN) [25], [26], which records the mappings between the spatio-temporal sensing areas to the completion accuracy in our problem. Next we formally introduce them respectively:

Q-Learning is a traditional RL method that records the rewards in Q-table $Q[s,a]$ for each action $a \in A$ under a state $s \in S$. Further consider that the spatio-temporal areas not only determine the current data completion accuracy, but also impact the future ones. Thus, we should iteratively add the future reward to update the Q-table $Q[s,a]$, as follows:

$$Q[s,a] = (1-\alpha)Q[s,a] + \alpha\big(R + \gamma \max_{a'} Q[s',a']\big), \quad (15)$$

where s' and a' are the next state and action, α is the learning rate, and γ indicates the discount factor of future rewards. With the trained Q-table, we can search it to obtain the importance of actions under a certain state, i.e., when and where to collect data may help more on completion. However, in practice, the spaces of states and actions are usually very large that the storage and search costs a lot.

Algorithm 2 Up-to-date Training

Input: model: Q_θ; training data: $\tilde{Y}_{m\times k}$; new data: y

Update $\tilde{Y}_{m\times k}$ according to the new data y;
while not convergent **do**
  Randomly select $s \in S$ and $a \in A$, obtain s';
  Get $Q_\theta(s,a)$, a', and $Q_\theta(s',a')$ according to Q_θ;
  Obtain R according to Eq. 14 and Alg. 1;
  Update Q_θ according to Eqs. 16 and 17.

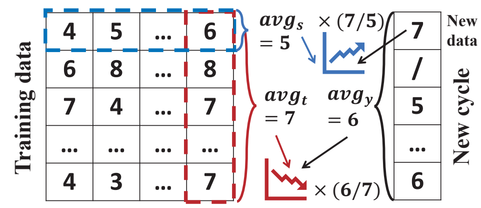


Fig. 7: Example of updating the training data.

Deep Q-Network is an emerging RL method in recent years, which utilizes the neural networks to replace the Q-table by learning a model to estimate the rewards for actions under a state, as follows:

$$Q(s,a) = \mathbb{E}\big[R + \gamma \max_{a'} Q(s',a')\big]. \quad (16)$$

We use the stochastic gradient algorithm to learn the model θ and thus obtain the loss function:

$$L(\theta_t) = \mathbb{E}[\big(R + \gamma \max_{a'} Q_\theta(s',a') - Q_\theta(s,a)\big)^2]. \quad (17)$$

In this way, DQN actually uses the model θ to approximately achieve $Q_\theta(s,a) \approx Q[s,a]$ in QL, which is exactly the estimated importance.

B. Up-to-date Training

By utilizing the above QL and DQN, we can use the training data and our proposed data matrix completion method to calculate the reward and then learn the importance estimation model. However, the sensing data may change a lot over time, which are impacted by many factors, e.g., weather, seasons, holidays, and even some emergencies. Obviously, as shown in the lower part of Fig. 6, such changing data distributions make the importance estimation model not always work well. Therefore, we should keep the model up-to-date through updating the training data, as summarized in Alg. 2.

Specifically, for each coming data, we update our training data from both spatial and temporal aspects. The basic idea is to utilize the continuity of spatio-temporal data shown in Figs. 5 of Section IV.B, i.e., the gradual changes in adjacent cycles and areas. We provide a toy example in Fig. 7. For the temporal aspect, we calculate the average values of the training data in the last cycle $avg_t = avg(\mathbf{y}_{-2}) = 7$ and the collected data in the current cycle $avg_y = avg(\hat{\mathbf{y}}_{-1} \otimes M[:,-1]) = 6$. Then, we use a classic Zoom method used in process images and information spaces [27], [28], which can hold the data

Algorithm 3 Fresh-looking Sampling

Input: selected workers: μ; new workers: u_i, c_i; number of workers: j; time: T; budget: B; historical workers: F

```
Conduct X = {f(μ, x_1), f(μ, x_2), ..., f(μ, x_{j−1})} from F according to t and T;
while i ≤ j and c_i ≤ B do
    if f(μ, u_i) ≥ max{X} then Return μ ∪ {u_i}
    else
        X ∪ {f(μ, u_i)}
        Randomly select z from X, and X = X \ {z}
        Wait for the next worker u_{i++}
```

distribution but not the values for updating the training data:

$$\mathbf{y}_t = \mathbf{y}_{-2} \otimes \big(\frac{avg_y}{avg_t} \times (1 - M[:, -1]^T)\big) + \hat{\mathbf{y}}_{-1}. \tag{18}$$

Similarly, for the spatial aspect, we mainly consider the ratios of the current data and previous data from the same sensing area to update the training data by Zoom, as follows:

$$\mathbf{y}_s = \overrightarrow{avg_s} \otimes \big(\mathbf{y}_{-1}./\overrightarrow{avg_s} \times (1 - M[:, -1]^T)\big) + \hat{\mathbf{y}}_{-1}, \tag{19}$$

where $\overrightarrow{avg_s} = \{avg_1, avg_2, \ldots, avg_m\}$ records the average values of the training data from each sensing area. Then, by combining the Eqs. 18 and 19, we obtain the update values as $\mathbf{y}_{-1} = \lambda_t \mathbf{y}_t + \lambda_s \mathbf{y}_s$, where $\lambda_t + \lambda_s = 1$ and they represent the spatio-temporal weights respectively. In addition, we only keep k cycles as training data, i.e., $\tilde{Y}_{m \times k}$, in order to utilize the gradual changes. Also, for a new cycle, we drop the oldest and add the new one to $\tilde{Y}$.

VI. Online Worker Selection

With the importance estimation model, we finally study the worker selection to select suitable workers to actively sense important spatio-temporal areas for accurate data completion.

A. Secretary Problem-based Worker Selection

In the online scenarios, the workers participate in real time and we should immediately decide whether to select it or not according to the importance of its covered sensing areas, without knowing the future information. If we can only select one worker, the worker selection problem is actually the classic secretary problem, i.e., select the best one out of w secretaries that come in sequence. By further considering the cardinality constraints, i.e., we can select at most k workers, and the submodular utility function, e.g., considering their overlapping skills, the classic secretary problem can be extended to the submodular k-secretaries problem [29], interpreted as 'select k out of w workers to maximize their group utility online'. The competitive ratio for this problem is proved as $(1 - 1/e)/7$.

In this paper, further considering the budget constraint and the worker's utility on completion accuracy, we formulate our worker selection problem as a budgeted multi-secretaries problem towards data completion, i.e., select a worker set μ under a budget $\sum_{u \in \mu} c_u \leq B$ to collect data from important spatio-temporal areas, in order to maximize the online data completion accuracy. Based on the importance estimation

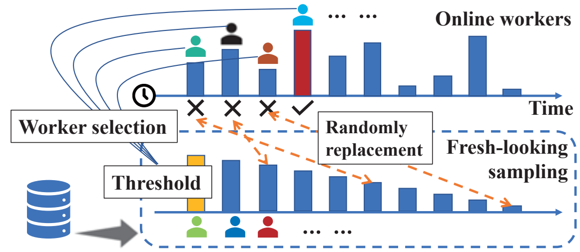


Fig. 8: Example of fresh-looking sampling.

model in Section V, we can estimate a worker u's utility according to its sensing area a_u under the already covered areas s_μ by the selected set μ, as follows:

$$f(\mu, u) = (T - t_u) \cdot Q(s_\mu, a_u), \tag{20}$$

where the worker u will collect data at time t_u and its active time for online data completion is denoted as $T - t_u$. Since there exists some spatio-temporal areas that are more important for data completion, the utility function in Eq. 20 is time-varying and generally non-submodular.

To tackle that, we propose a segmented worker selection strategy. Specifically, we first use the historical records to roughly estimate the number of selected workers k under the budget B. In general, the costs are mainly caused by the sensing devices, which have relatively accurate distributions for estimating k. Then, we divide w workers into k segments and try to select the best one from each segment, i.e., observe the first $\frac{1}{e} \cdot \frac{w}{k}$ workers and select the first one who has a larger utility than the observed workers. Finally, after selecting one worker, we re-adjust the segmented strategy, i.e., re-estimate the number k and then divide and select workers as mentioned above. Such adjustment is used to correct the inaccurate k. Note that we won't select a worker who may reduce the overall accuracy, the utility function $f(\cdot)$ thus can be seen as a monotone non-submodular function. By introducing the submodularity ratio γ of $f(\cdot)$, our proposed segmented worker selection strategy can be proved to achieve an expected competitive ratio of $\gamma^2(1-e^{-1})(1-e^{-\gamma/2})/7$ [30].

B. Fresh-looking Sampling

In the traditional secretary problem, workers are coming in random order. Thus, we should take an observation, i.e., observe and discard the first $\frac{1}{e} \cdot 100\%$ workers, to learn their utility distribution. This observation is necessary but still seems a little wasteful. However, in practice, we usually have some historical records, which can be used to learn such distributions instead of discarding workers. Thus, in this paper, we further introduce the prophet problem, i.e., select the best worker from known distributions, into our budgeted multi-secretaries problem, which no longer discards workers.

The basic idea of the prophet secretary problem is to construct a sample worker set to approximate the discarded workers. Based on the observation that $O_\varepsilon(j)$ samples can provide a sufficiently good approximation to the $1/e$-quantile of the distribution of $\max\{f(u_1), f(u_2), \ldots, f(u_j)\}$ [31], we can conduct a $j-1$ samples for each coming worker to provide the approximation on the discarded workers [32].

For simplifying and utilizing the online coming workers, we then introduce the fresh-looking sampling method, as shown in Alg. 3 and Fig. 8. Specifically, we first conduct a sample set of size $j-1$ from historical records with the same period (line 1). Then, for each coming worker, if we can afford (line2), we compare its utility with the max one from the sample set, i.e., the threshold, to decide whether to select it or not (line 3). Note that if we don't select the current worker, we add it to the sample set and randomly delete one for fresh-looking sampling (lines 5-6). Actually, since we delete the one with largest utility, the threshold decreases in expectation over time. In this way, we actually conduct a sample set of size $j-1$ approximated to the discarded workers to estimate the threshold for each coming worker.

VII. Performance Evaluation

A. Data set

We conduct extensive evaluations on five typical sensing tasks with real-world data sets, including environmental monitoring (**PM2.5** [33], **Temperature**, **Humidity** [34]) and urban sensing (**Traffic** [35], **Parking** [36]). Specifically, PM2.5 is collected by 36 air quality monitoring stations from Beijing. Temperature and Humidity are sensed by 57 static sensors deployed in the EPFL campus. Traffic collects the traffic volumes of 30 subway stations in New South Wales. Parking contains the occupancy rates of 73 car parks from Birmingham. The detailed statistics are shown in Table II.

Note that the above data sets are collected by static sensors or stations, which can also be obtained by mobile devices. To evaluate our framework, especially on worker selection, we also conduct the corresponding worker set based on some real-world traces collected from city (for PM2.5, Traffic, and Parking) and campus (for Temperature and Humidity).

B. Comparison Algorithms

Our proposed framework consists of three parts, we conduct evaluations on them respectively. For **data completion**, we propose an online algorithm with spatio-temporal constraints (ON). We compare it with its offline version (STMC), the classic matrix completion (MC) [10], and the deep matrix factorization (DMF) [12]. Similarly, for **importance estimation**, we compare our proposed up-to-date model (UTD) with the classic Deep-Q Network (DQN) [13] and the query-by-committee (QBC) [5]. For **worker selection**, we compare the prophet-secretary-based strategy (PRO) with the secretary-based strategy (DYN) [30], the near-optimal offline strategy (OFF), the online incentive mechanism (OMZ) [37], and the basic random method (RAN).

C. Evaluation Results

1) Data completion: We first evaluate the data completion in terms of the main metric, i.e., the completion error ε, as shown in Fig. 9 (a)-(e) for five typical sensing tasks respectively. We randomly select the spatio-temporal areas and change the sparsity ratio from 0.1 to 0.5. We can see that in most cases our proposed ON is close to its offline version

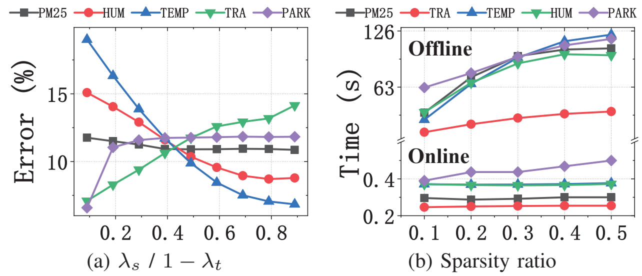


Fig. 12: Spatio-temporal weights and running time.

STMC and better than others, which proves the effectiveness of our method. However, when the sparsity ratio is low, ON usually has a poor performance, especially on Temperature and Humidity tasks. The reason is that ON relies heavily on the previous matrix completion results, i.e., the last latent spatio-temporal feature matrices U and V, which are not very accurate and change greatly when we have only a few sensing data for completion.

2) Importance estimation: For the importance estimation, since we can hardly evaluate the estimation model directly, we use the model to select and sense some spatio-temporal areas and evaluate their completion (by ON) errors. As shown in Fig. 10 (a)-(e), our UTD is always better than other comparison methods, which shows that the up-to-date training is effective and necessary. Since we use ON as the data completion method, the methods also achieve poor performances on Temperature and Humidity tasks. In addition, when the sparsity ratio is high, all of the estimation methods can't help much.

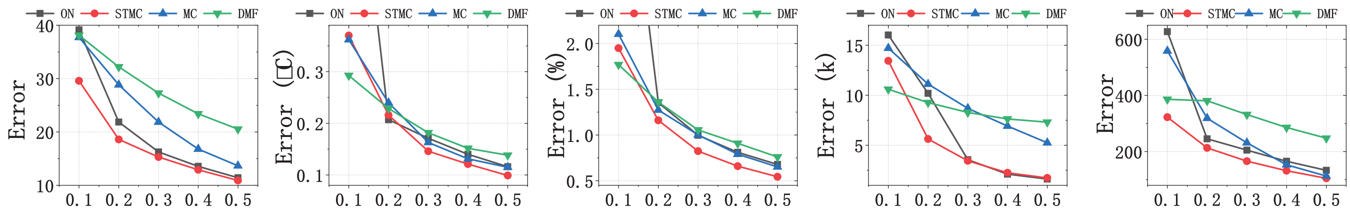
3) Worker selection: We then evaluate the worker selection towards data completion, as shown in Fig. 11 (a)-(e). We set the workers' average cost as 20 with a budget constraint $B = 300$. To effectively evaluate our PRO towards data completion, we change the total number of workers from 100 to 300. We can see that our PRO outperforms the other methods except the near-optimal OFF. Meanwhile, PRO always selects more workers and achieves fewer error than the secretary-based DYN, which shows that our prophet secretary problem really improves the performance. Similarly, in most cases, DYN and OMZ select fewer workers than RAN but have better completion accuracy, which shows the necessity of worker selection.

4) Spatio-temporal weights: In data completion and importance estimation parts, we exploit the spatio-temporal correlations in sensing data to improve the performance of our proposed methods. Thus, we conduct some evaluations on spatio-temporal weights λ_s and λ_t. Note that the above evaluations have already shown the effectiveness of these weights, we are mainly concerned with the proportion between them. As shown in Fig. 12 (a), we set $\lambda_s = 1 - \lambda_t$ and obtain the standardized error over five sensing tasks. Interestingly, we find that the temporal weights play the more important roles in urban sensing (Traffic/Park) while the spatial ones help more with environmental monitoring (PM2.5/Temperature/Humidity).

5) Running time: Finally, we compare the running time of our online framework with the offline ones, as shown in Fig. 12 (b). Actually, the offline methods cost $\sim 1-2$ minutes for

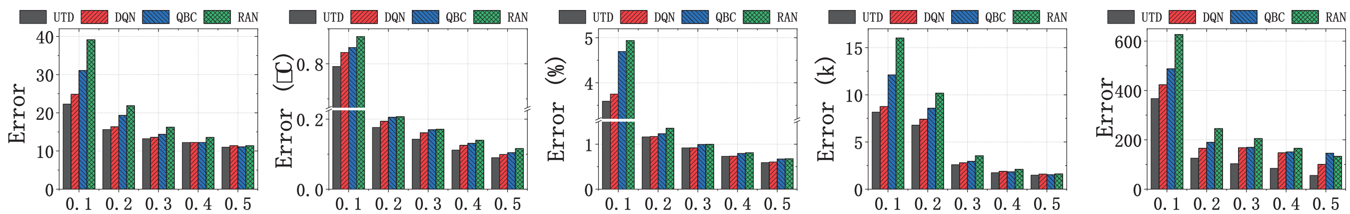
TABLE II: Statistics of five evaluation sensing tasks

	Environmental Monitoring			Urban Sensing	
	PM2.5	**Temperature**	**Humidity**	**Traffic**	**Parking**
City	Beijing (China)	Lausanne (Switzerland)		New South Wales (Australia)	Birmingham (UK)
Sensing areas	36 areas each with 1k*1km^2	57 areas each with 50*30m^2		30 subway stations	73 car parks
Cycle & Duration	1 hour & 11 days	0.5 hour & 7 days		1 day & 1 year	0.5 hour & 77 days
Mean Std.	79.11 ± 81.21	$6.04 \pm 1.87°C$	$84.52 \pm 6.32\%$	19095.73 ± 26750.79	647.97 ± 657.23



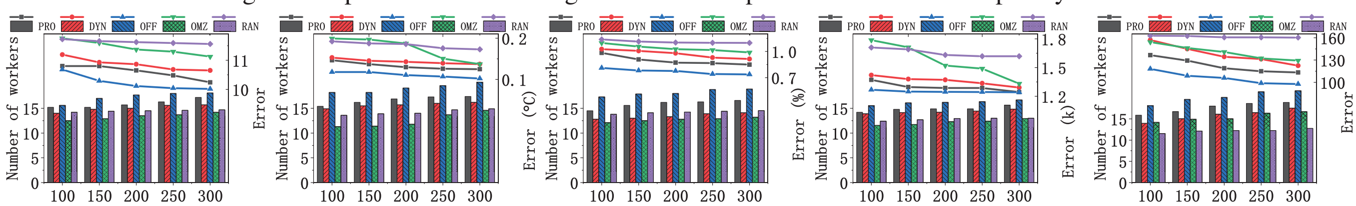
(a) PM2.5 (b) Temperature (c) Humidity (d) Traffic (e) Parking

Fig. 9: Data completion under randomly selection with different sparsity ratios.



(a) PM2.5 (b) Temperature (c) Humidity (d) Traffic (e) Parking

Fig. 10: Importance estimation-guided data completion with different sparsity ratios.



(a) PM2.5 (b) Temperature (c) Humidity (d) Traffic (e) Parking

Fig. 11: Worker selection towards data completion with different numbers of workers.

TABLE III: Running time

ON	STMC	MC	DMF	KNN-S	KNN-T	GP
0.32	73.87	55.19	24.81	0.05	0.05	0.03

data completion, importance estimation, and worker selection, while our online ones only need $\sim 0.2 - 0.4$ second. We also illustrate the running time of the most time consuming part, i.e., the data completion methods, in Table III. Expect the above methods, we also add some fast ones as comparisons, i.e., the classic K-Nearest Neighbors and Gaussian Process. However, these fast methods cannot deal with the sparse data.

VIII. Conclusion

In this paper, we investigate the online Sparse Crowdsensing, where we can recruit a few workers to perform a part of sensing tasks and infer the rest. Since the workers will participate in real time, their sensing data are coming dynamically. To deal with such online workers with dynamically coming data, we propose the OS-MCS framework which consists of three parts: matrix completion, importance estimation, and worker selection. To make full use of the online data, we first propose an online matrix completion algorithm with spatio-temporal constraints. Based on that, we estimate the spatio-temporal area importance by conducting a reinforcement learning-based up-to-date model. Finally, we investigate the prophet secretary problem to select suitable workers to actively sense important areas for data completion in an online manner. Extensive experiments on five typical sensing tasks with real-world data sets have shown the effectiveness of our proposed methods and framework for online Sparse Crowdsensing.

Acknowledgment

This work is supported in part by National Key R&D Program of China under Grant Nos. 2021ZD0112501 and 2021ZD0112502, and National Natural Science Foundation of China under Grant Nos. 62102161, 61772230 and 61972450, and China Postdoctoral Science Foundation under Grant Nos. 2021T140261 and 2021M701389, and CCF-Baidu Open Fund (No.2021PP15002000), and NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[2] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE communications surveys & tutorials*, vol. 21, no. 3, pp. 2419–2465, 2019.

[3] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4w1h in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 42–48, 2014.

[4] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: challenges and opportunities," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 161–167, 2016.

[5] L. Wang, D. Zhang, D. Yang, A. Pathak, C. Chen, X. Han, H. Xiong, and Y. Wang, "Space-ta: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing," *Acm Transactions on Intelligent Systems & Technology*, vol. 9, no. 2, pp. 1–28, 2017.

[6] S. He and K. G. Shin, "Steering crowdsourced signal map construction via bayesian compressive sensing," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, April 2018, pp. 1016–1024.

[7] T. Liu, Y. Zhu, Y. Yang, and F. Ye, "alc^2 : When active learning meets compressive crowdsensing for urban air pollution monitoring," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9427–9438, 2019.

[8] W. Liu, Y. Yang, E. Wang, and J. Wu, "User recruitment for enhancing data inference accuracy in sparse mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1802–1814, 2019.

[9] D. Wu, T. Xiao, X. Liao, J. Luo, C. Wu, S. Zhang, Y. Li, and Y. Guo, "When sharing economy meets iot: Towards fine-grained urban air quality monitoring through mobile crowdsensing on bike-share system," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–26, 2020.

[10] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Active sparse mobile crowd sensing based on matrix completion," in *Proceedings of the 2019 International Conference on Management of Data*, ser. SIGMOD '19. New York, NY, USA: ACM, 2019, pp. 195–210.

[11] K. Xie, J. Tian, G. Xie, G. Zhang, and D. Zhang, "Low cost sparse network monitoring based onblock matrix completion," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[12] E. Wang, M. Zhang, X. Cheng, Y. Yang, W. Liu, H. Yu, L. Wang, and J. Zhang, "Deep learning-enabled sparse industrial crowdsensing and prediction," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6170–6181, 2021.

[13] W. Liu, L. Wang, E. Wang, Y. Yang, D. Zeghlache, and D. Zhang, "Reinforcement learning-based cell selection in sparse mobile crowdsensing," *Computer Networks*, vol. 161, pp. 102–114, 2019.

[14] Z. Zhu, B. Chen, W. Liu, Y. Zhao, Z. Liu, and Z. Zhao, "A cost-quality beneficial cell selection approach for sparse mobile crowdsensing with diverse sensing costs," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3831–3850, 2020.

[15] L. Han, Z. Yu, L. Wang, Z. Yu, and B. Guo, "Keeping cell selection model up-to-date to adapt to time-dependent environment in sparse mobile crowdsensing," *IEEE Internet of Things Journal*, 2021.

[16] P. Sun, Z. Wang, L. Wu, H. Shao, H. Qi, and Z. Wang, "Trustworthy and cost-effective cell selection for sparse mobile crowdsensing systems," *IEEE Transactions on Vehicular Technology*, 2021.

[17] T. Wang, X. Xie, X. Cao, T. B. Pedersen, Y. Wang, and M. Xiao, "On efficient and scalable time-continuous spatial crowdsourcing," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 1212–1223.

[18] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functionsi," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[19] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," *IEEE/ACM Transactions on Networking (ToN)*, vol. 20, no. 3, pp. 662–676, 2012.

[20] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[21] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[22] C. Jin, S. M. Kakade, and P. Netrapalli, "Provable efficient online matrix completion via non-convex stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 29, pp. 4520–4528, 2016.

[23] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 635–644.

[24] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[26] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, and M. Lanctot, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[27] B. B. Bederson, J. D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas, "Pad++: A zoomable graphical sketchpad for exploring alternate interface physics," *Journal of Visual Languages & Computing*, vol. 7, no. 1, pp. 3–32, 1996.

[28] D. J. Allard, F. J. Canova Jr, D. A. Johnson, C. S. Lanier, J. R. Lewis, B. K. Tiller, W. Villafana, and R. L. Yee, "Personal communicator having improved zoom and pan functions for editing information on touch sensitive display," Mar. 25 1997, uS Patent 5,615,384.

[29] M. Bateni, M. Hajiaghayi, and M. Zadimoghaddam, "Submodular secretary problem and extensions," *ACM Transactions on Algorithms (TALG)*, vol. 9, no. 4, p. 32, 2013.

[30] W. Liu, Y. Yang, E. Wang, H. Wang, Z. Wang, and J. Wu, "Dynamic online user recruitment with (non-) submodular utility in mobile crowdsensing," *IEEE/ACM Transactions on Networking*, 2021.

[31] S. Ehsani, M. Hajiaghayi, T. Kesselheim, and S. Singla, "Prophet secretary for combinatorial auctions and matroids," in *Proceedings of the twenty-ninth annual acm-siam symposium on discrete algorithms*. SIAM, 2018, pp. 700–714.

[32] J. Correa, P. Dütting, F. Fischer, and K. Schewior, "Prophet inequalities for iid random variables from an unknown distribution," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019, pp. 3–17.

[33] Y. Zheng, F. Liu, and H. P. Hsieh, "U-air:when urban air quality inference meets big data," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 1436–1444.

[34] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 1–32, 2010.

[35] T. for NSW. (2019) System of traffic volume viewer. @ONLINE. [Online]. Available: https://www.rms.nsw.gov.au/about/corporate-publications/statistics/traffic-volumes/aadt-map/index.html

[36] D. H. Stolfi, E. Alba, and X. Yao, "Predicting car park occupancy rates in smart cities," in *International Conference on Smart Cities*. Springer, 2017, pp. 107–117.

[37] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 647–661, 2014.