

A crowd sensing data collection framework based on edge computing

Yanjing Liu, Xuemei Sun*, Xiaorong Yang and Yueyue Zhao

School of Computer Science and Technology Tiangong University, Tianjin, China

Abstract. With the rapid increase in the use of mobile devices equipped with built-in sensors, mobile crowd sensing (MCS) as a human driven perception mode came into being. Since a large number of users submit data to the cloud servers in parallel, it will not only increase the pressure on the cloud servers, but also lead to the problem of data redundancy. In order to solve this problem, this paper introduces edge computing into mobile crowd sensing for collecting perception data, and proposes a group intelligence perception network data collection model based on edge computing. The data is observed and sampled at the edge node through the compressed sensing algorithm, and the compressed data is transmitted to the cloud server. Using CI_BP algorithm restores the compressed data in cloud server. The results show that compared with the orthogonal matching pursuit algorithm (OMP), the data collection model based on edge cloud computing proposed in this paper can better solve the problem of data redundancy.

Keywords: Mobile crowd sensing, Edge computing, Data collection, Compressed sensing.

1 Introduction

Mobile crowd sensing is a new form of data collection, which combines the concept of crowdsourcing with the perception of intelligent devices. Through their existing intelligent mobile devices, users use the embedded sensors to collect and transmit sensing data, so as to complete large-scale sensing and information sharing [1]. Mobile crowd sensing benefits from a large user group, which makes it a source for us to obtain all kinds of rich information in the environment [2].

The implementation of traditional mobile crowd sensing service is a centralized service based on cloud [3-5]. In this way, in the case of large-scale MCS deployment, device management and real-time data processing require a lot of computing resources. However, cloud based group intelligence perception will put a great load on mobile networks and increase the traffic of running cloud servers. For real-time perception scenarios, due to the frequently changing context when a large number of devices participate in MCS tasks, it will increase the delay of data and information dissemination. In view of this phenomenon, edge computing provides a new processing method.

* Corresponding author: Sunxuemei@tiangong.edu.cn

Mobile edge computing (MEC) offloads computing near mobile devices by introducing a new middle layer [6], which is responsible for data aggregation, filtering, processing and storage. As shown in Figure 1. In the context of group intelligence perception, MEC can preprocess the original perception data collected by intelligent devices, and can also independently support real-time perception of scenes and deal with users' frequently changing context in time without contacting cloud server. The mobile crowd sensing task execution process based on edge computing [7] reduces the computational complexity of centralized cloud services, reduces the delay caused by sending notifications and alerts to mobile devices in real-time swarm intelligence aware scenarios, and reduces privacy threats.

Mobile crowd sensing system will face many problems in the process of data collection. On the one hand, it is necessary to ensure that mobile users cover all perceived areas in order to obtain reliable and complete data. However, in practice, the platform cannot afford to recruit so many users because of the large cost involved, and sometimes there are no users available in the required sensing area. At the same time, this method may lead to data redundancy and consume additional system resources. On the other hand, the platform will collect a large amount of data after publishing tasks, and processing a large amount of perceptual data will greatly increase the load of cloud server. Therefore, how to reduce the transmission of redundant information and reduce the consumption of system resources has become a major problem in the research of swarm intelligence perception system.

To sum up, this paper designs a data collection mode based on edge computing. Firstly, after processing the data collected by users by the edge server, the data is observed and sampled at the edge node through the compressed sensing algorithm [8], and the compressed data is transmitted to the cloud server. Secondly, using CI_BP algorithm restores the compressed data, so as to effectively restore the data in the target area.

2. System model and algorithm

2.1 System model

The model proposed in this paper consists of three layers, including cloud server, multiple edge servers in the region and a large number of mobile users. The task requester first issues a perception task to the cloud server. The cloud server decomposes the request according to the required location and distributes the sub request to the corresponding edge server. The edge server selects the appropriate user to perform the sensing task. Then, the user sends the collected data to the edge server through short-range communication technology. The edge server preprocesses the data, then compresses it and forwards it to the cloud server. Finally, after collecting all the sensing data, the cloud server reconstructs the data and displays the aggregated results to the requester. The flow of the whole model is shown in Figure 1.

2.2 Data compression based on compressed sensing algorithm

This section will compress the data in the region at the edge layer through the idea of compressed sensing. Then, the filled complete data of all users from time 1 to time t can be compressed. For a single user P , the perception data obtained after t times is $X_p = [x_1^p, x_2^p, \dots, x_t^p]^T$. The collected perception data can be expressed as $X_m = [x_1^m, x_2^m, \dots, x_t^m]^T$. The union of areas covered by different users at different perception times as $X = X_t^1 \cup X_t^2 \cup \dots \cup X_t^m$.

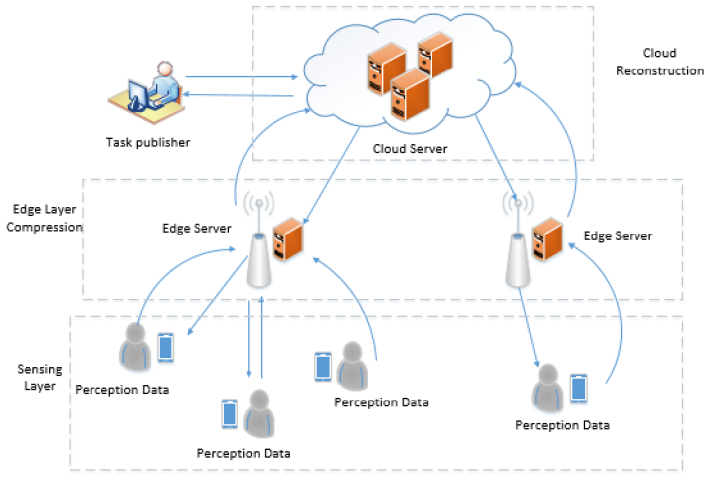


Fig. 1. Data collection model based on edge computing.

When using the compressed sensing principle for data collection, according to the requirements of CS, if you want to compress the collected data, first analyze the sparsity of the collected sensing data to determine whether the collected data is compressible, and then use discrete cosine transform (DCT) to analyze the sparsity of the collected data. If the data can be compressed, some time points should be selected through the row vector in the observation matrix. At the selected time point, the weighted measurement value multiplied by the collected perceptual data and the corresponding elements in the observation matrix is generated in the edge server. Therefore, the measurement value submitted by a single user can be expressed as (1):

$$Y_i = \sum_{j=1}^t \phi_{ij} x_j \quad (1)$$

Where ϕ_{ij} is the collection coefficient.

The perceptual data x_1 collected by the user at time t_1 is multiplied by the collection coefficient, and then the data is sent to time t_2 ; At time t_2 , the sum of the observed value at time t_1 and the perceived data x_2 collected by the user at time t_2 multiplied by the collection coefficient and the added weight is sent to the next time until the user collects the data within his perceived time. After t^* times of observation, the observation value formed by user P at the time of t^* is shown in (2), where $t^* \ll t$

$$Y = \Phi \times X \begin{pmatrix} \phi_{11} & \dots & \phi_{1t} \\ \vdots & \ddots & \vdots \\ \phi_{t1} & \dots & \phi_{tt} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_t \end{pmatrix} = [y_1, y_2, \dots, y_t]^T \quad (2)$$

2.3 Data reconstruction algorithm based on CL_BP

In the previous section, after the edge node has compressed the data, this section uses the reconstruction algorithm of base pursuit cloud (cl_bp) of cloud server to restore the original data. In the discrete cosine transform domain, the data X collected by the user meets the compressibility, and then the data construction method of the original data \hat{X} can be described by using the compressed sampling observation value Y . then the estimated value \hat{X} of the original data X can be reconstructed to solve the optimization problem in (3):

$$\hat{X} = \operatorname{argmin} \|X\|_0 \quad \text{s.t. } Y = \Phi X \quad (3)$$

Where \hat{X} is the estimated value of the original data X , and X is sparse in the transform domain, then the sparse vector s can be determined. According to (4), it can be equivalent to solving (3):

$$\hat{s} = \operatorname{argmin} \|s\|_0 \quad \text{s. t. } Y = \Phi\Psi s \quad (4)$$

Among them \hat{s} is sparse data s estimates, s is sparse coefficient of original data X in Ψ , Ψ is a sparse transformation base in this study.

Due to the (4) is a NP hard problem, Donoho put forward when the signal is sparse can use L_1 norm instead of L_0 norm, it is

$$\hat{s} = \operatorname{argmin} \|s\|_1 \quad \text{s. t. } Y = \Phi\Psi s \quad (5)$$

Therefore, in order to solve the above problems, the data reconstruction adopts CL_BP algorithm to solve, although the amount of calculation is large, the reconstruction effect of the original data is better. For (5), the L_1 norm regularization can be re expressed as a linear programming problem and can be solved effectively. CL_BP algorithm generally adopts interior point method or gradient projection method to solve this kind of problem. Gradient projection method is faster, but the result often has deviation. Compared with interior point method, the solution speed is slower, but the result is relatively accurate.

Use the interior point method to convert the above problem into the following (6):

$$\min c^T x \quad \text{s. t. } b = Ax; x \geq 0 \quad (6)$$

In the above formula, $c = [1, 1, \dots, 1]_{1 \times 2N}$, $x \in \mathbb{R}^{2N}$, $b = Y$, $A = [\Theta, -\Theta]$, solveing x , let $x = \begin{bmatrix} P_{N \times 1} \\ Q_{N \times 1} \end{bmatrix}$, and then find $s_0 = P_{N \times 1} - Q_{N \times 1}$. Finally, by solving (4) with the constraints of (6), \hat{s} can be solved.

3 Experiment and analysis

3.1 Dataset

The data used in this experiment is the real-life air quality data set [9], which collects the air quality information of Beijing, mainly PM2.5, which records the air quality collected by 36 air quality monitoring stations at the same sensing time. The monitoring station records once every hour, with a total of 9504 data points. The comparison algorithm we use is the orthogonal matching pursuit algorithm (OMP), which is one of the greedy algorithms.

3.2 Results and analysis

This paper is verified by compression ratio (CR) and normalized mean square error (NMAE), where CR and NMAE are as shown in (7) and (8):

$$CR = \frac{M}{N} \quad (7)$$

$$NMAE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2} \quad (8)$$

Where N represents the length of the original data and M represents the length of the compressed data. x_i represents real data and \hat{x}_i represents recovered data.

Figure 2 shows the reconstruction effect of cl-bp algorithm on the air quality data of one of the 36 stations at different times. It can be seen from the figure that the original data and reconstructed data are basically the same, indicating that cl-bp algorithm has good reconstruction effect and high reconstruction accuracy.

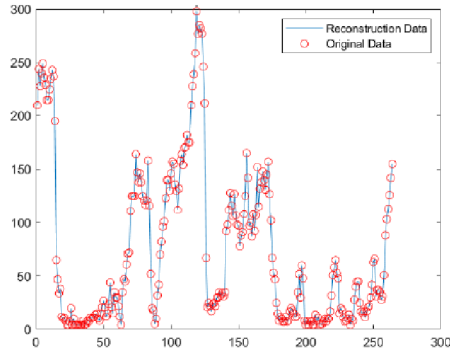


Fig. 2. Comparison between reconstructed data and original data of algorithm CL_BP.

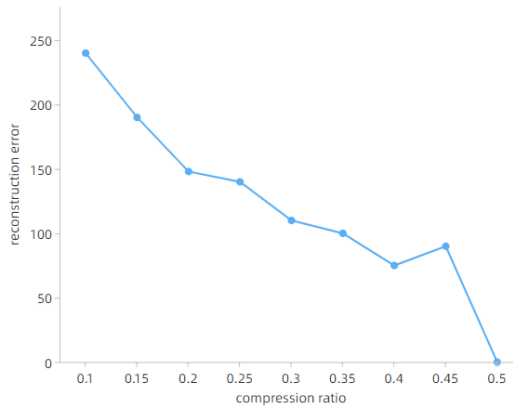


Fig. 3. Reconstruction error of OMP.

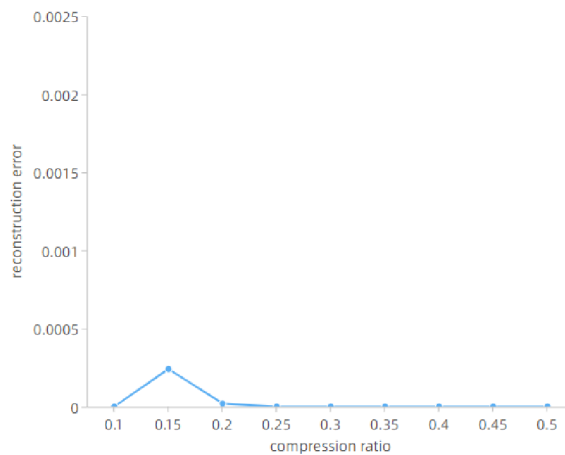


Fig. 4. Reconstruction error of CL_BP.

Figure 3 and 4 are comparative experiments of reconstruction errors of cl-bp algorithm and OMP algorithm when the compression ratio of air quality data is 0.1 to 0.5. The abscissa of the two figures is the compression ratio and the ordinate is the reconstruction error. It can be seen from the figure that with the increase of compression ratio, the reconstruction error of the two algorithms is generally decreasing. This is because with the increase of compression ratio, the amount of compressed data is also increasing, that is, the observed value is increasing, so the reconstruction accuracy is high. However, the reconstruction error of OMP algorithm is relatively large before the compression ratio is 0.45, while the reconstruction error of the proposed cl-bp algorithm tends to be stable after the compression ratio is 0.25, which is basically 0. Moreover, the reconstruction error is very small, which is better than OMP algorithm.

4 Conclusion

This paper introduces edge computing into the data collection process of mobile crowd sensing. In the edge server, the compressed sensing algorithm compresses the sensing data, which reduces the communication pressure of the cloud server and solves the network delay caused by the transmission of user data to the cloud server. Through Cl_BP algorithm recovers data in the cloud, which can reconstruct data with less data and ensure less error. The layered model based on edge computing proposed in this paper can better solve the problem of data redundancy and reduce the load of cloud servers.

References

1. Ganti R, YE F, LEI H. Mobile crowdsensing: current state and future challenges [J]. IEEE Communications Magazine, 2011, 49(11): 32-39.
2. Guo B, CHEN C, ZHANG D, et al. Mobile crowd sensing and computing: when participatory sensing meets participatory social media [J]. IEEE Communications Magazine, 2016, 54(2): 131-137.
3. Antonić A, MARJANOVIĆ M, PRIPUŽIĆ K, et al. A mobile crowd sensing ecosystem enabled by CUPUS: Cloud-based publish/subscribe middleware for the Internet of Things [J]. Future Generation Computer Systems, 2016, 56: 607-622.
4. Crowdsensing platform: Sensarena[C]// Proceedings of the 2016 13th IEEE Annual Consumer Communications & Networking Conference. 2016: 1-2.
5. Ben MESSAOUD R, REJIBA Z, GHAMRI-DOUDANE Y. An energy-aware end-to-end SHERCHAN W, JAYARAMAN P P, KRISHNASWAMY S, et al. Using On-the-Move Mining for Mobile Crowdsensing[C]// Proceedings of the 2012 IEEE 13th International Conference on Mobile Data Management. 2012:115-124.
6. Mach P, BECVAR Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading [J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.
7. Marjanovic M, ANTONIC A, ZARKO I P. Edge Computing Architecture for Mobile Crowdsensing [J]. IEEE Access, 2018, 6: 10662-10674.
8. Donoho D L. Compressed sensing [J]. IEEE Transactions on Information Theory, 2006, 52(4): 1289-1306.
9. Zheng Y, LIU F, HSIEH H. U-Air: when urban air quality inference meets big data[C]// Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013: 1436-1444.