

Research on Recommended Method for Crowdsourcing Task of Mobile Application Based on Top-K Algorithm

Ying Liu^{*}, Tao Zhang, Kun Li, Hehe Gong

College of Software and Microelectronics, Northwestern Polytechnical University, xi'an, China

894749065@qq.com, 290515477@qq.com, 383037601@qq.com, 1807388494@qq.com

Keywords: Mobile application crowdsourcing testing, Top-K algorithm, task recommendation

Abstract. Crowdsourcing testers don't have to sign any contract, and the anonymous crowdsourcing testers determine the quality of tests. To match the recommended tasks with testers, a two phase task recommendation method based on Top-K algorithm was proposed. Category was introduced to reduce time complexity of Top-K algorithm. By classifying the tasks and calculating the category matching scores, the task types which were most suitable were obtained. After calculating the similarity between tester portrayal and tasks, the top K tasks were recommended from selected categories. Experiment shows that the proposed Top-K-Task algorithm can greatly improve the matching degree between testers and the recommended task.

1. Introduction

Mobile application crowdsourcing testing is a distributed solution that outsourcing mobile application testing tasks to anonymous network users in a voluntary way[1]. Compared with the traditional testing methods, crowdsourcing testing has advantages of freedom, high innovation and low cost[2]. However, because the lack of effective task recommendation mechanism, the matching degree between tasks and testers is low. Not only does the problem affects the test efficiency and quality, but also raise testers' enthusiasm.

Table 1 shows that existing crowdsourcing platforms often provide preliminary sorting and searching mechanisms for users. But traditional method of sorting and searching makes it difficult for testers quickly find out interesting testing tasks, and it is difficult to guarantee the quality of testing.

Table1. Task searching and ranking method

| Crowdsourcing platform | key word | region | category | time | price |
|------------------------|----------|--------|----------|------|-------|
| Mturk | √ | | √ | √ | √ |
| CloudCrowd | √ | | | √ | √ |
| Naoliku | √ | | | √ | √ |
| Zhubajie | √ | √ | √ | | √ |
| Sandaha | √ | | √ | √ | √ |
| testin | | | √ | √ | |

To solve these problems, the paper studies the recommendation method of crowdsourcing testing tasks. First, an improvement on Top-K algorithm was proposed by introducing category. After calculating the matching score, the most suitable task types were selected. Then the top k tasks were

recommended after the calculation of similarity between tester portrayal and tasks.

2. Task recommendation Algorithm of mobile application crowdsourcing testing

Top-K algorithm is a real-time recommendation algorithm proposed by Mejdil Safran[3]. The algorithm predicts the user's preferences by setting the weights of the user's different attributes[4], returning the previous K results based on the user's preferences[5]. Because the algorithm can extract useful information from a large amount of data, it has been widely used in many fields such as e-commerce, search engines and so on. The paper will study and improve Top-K algorithm, and propose a method for task recommendation.

2.1. Top-K algorithm based on category

In the application scenario of Top-K algorithm, there are a lot of Crowdsourcing tasks and testers data. An effective intermediate mechanism is introducing category to avoid a large-scale operation. Category acts as an intermediary mechanism between crowdsourcing and task. Depending on the skills required, tasks can be put into different categories, and testers can be associated with a variety of categories based on their own features, historical task information. The crowdsourcing task can be recommended as a limited 1-K match between testers and tasks. The data structure is shown in figure 1.

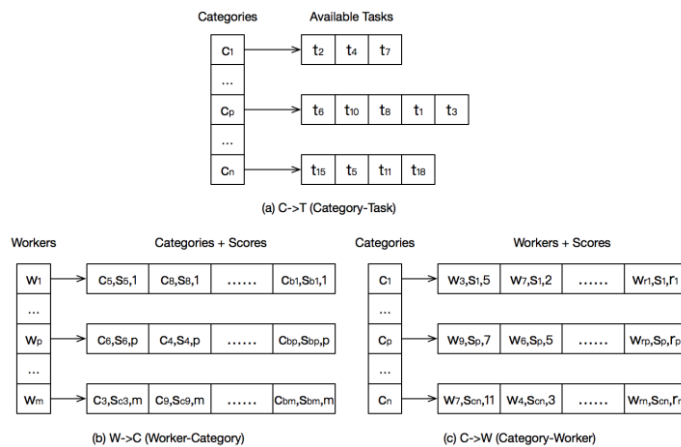


Figure1. Data structure of Top-K algorithm based on category

$C \rightarrow T$ (category-task) is introduced as an array of many categories, each category c_i contains a large number of tasks, the data structure is shown in Figure 1 (a). Figure 1 (B) shows a worker W_j in $W \rightarrow C$ (worker-category) points to many categories. However, it is not enough to get the $W \rightarrow C$ data structure because $W \rightarrow C$ only represents the relationship from the testers to categories. Additional data structures are needed to match the appropriate categories with the appointed testers, as is shown in Figure 1 (C).

For the given tester W_j and categories c_i , the matching score S_{ij} between them is defined as the product of three factors: the acceptance rate, the proficiency level, and the user satisfaction:

$$S_{ij} = AR_{ij} \times SP_{ij} \times CS_{ij} \tag{1}$$

Acceptance rate AR_{ij} is defined as the ratio of tasks number accepted by tester W_j in the category c_i to the total number of accepted tasks of W_j :

$$AR_{ij} = \frac{AT_{ij}}{ACCT_{ij}} \tag{2}$$

Proficiency level SP_{ij} is defined as the ratio of successful completion task in the category c_i to all of the task completed successfully by W_j :

$$SP_{ij} = \frac{CT_{ij}}{FT_{ij}} \quad (3)$$

User satisfaction is the ratio of average customer satisfaction in c_i to the average customer satisfaction of all tasks:

$$CS_{ij} = \frac{ST_{ij}}{CST_{ij}} \quad (4)$$

2.2. Similarity between tester portrayal and tasks

The above can only screen out the types of tasks which is most suitable for the testers. So it is important to consider how to recommend the most suitable tasks, and the introduction of Cosine similarity can solve the problem.

Cosine similarity is a commonly used similarity measurement, especially for high-dimensional spaces[6]. The Cosine similarity is more applicable than Euclidean distance when the dimensions of different attribute values are inconsistent[7]. The range of Cosine similarity is $[-1,1]$, and the bigger the result is, the higher the similarity is[8]. The Cosine similarity between $a = [a_1 \dots a_n]^T$ and $b = [b_1 \dots b_n]^T$ are:

$$similarity(a, b) = \frac{\sum_{k=1}^h a \times b}{\sqrt{\sum_{k=1}^h a^2} \times \sqrt{\sum_{k=1}^h b^2}} \quad (5)$$

Similarity between tester portrayal and tasks can be defined as:

$$similarity(p_j, r_i) = \frac{\sum_{k=1}^h p_{jk} \times r_{ik}}{\sqrt{\sum_{k=1}^h p_{jk}^2} \times \sqrt{\sum_{k=1}^h r_{ik}^2}} \quad (6)$$

Where tester portrayal p_j is the collection of test experience p_{j1} , number of certificates p_{j2} , response speed p_{j3} ; r_i is the collection of requirements of tasks.

2.3. Top-K-Task recommendation algorithm

With the increasing scale of crowdsourcing testers and tasks, testers will spend more time to find tasks which are suitable for their skills and interest, and it is difficult to ensure that if the recommended tasks is suitable. The purpose is to help testers quickly find the best matching tasks. Therefore, the proposed Top-K-Task algorithm aims at recommending the most suitable task for crowdsourcing testers.

Supported by the data structure of the Top-K algorithm based on the task category, the recommendation for the specific testers by the Top-K-Task algorithm is shown in figure 2.

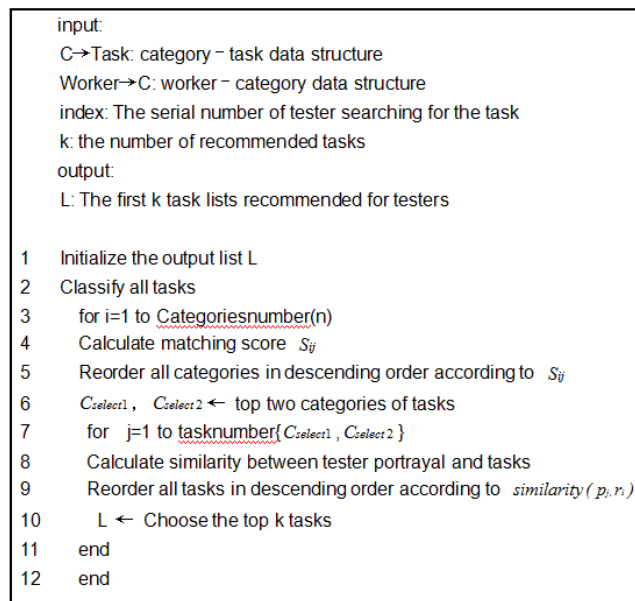


Figure 2. the Top-K-Task algorithm

The proposed algorithm firstly classifies the test tasks, and calculates the matching scores, then the categories are sorted in descending order to screen out the top two suitable categories. By computing similarity between tester portrayal and tasks in these two categories, we can get the top k tasks that is most recommendable.

The Top-K-Task algorithm requires $O(\alpha+k)$ to generate recommendations for appointed tester, where α is the number of task categories, and K is the number of tasks to be recommend. Obviously, the time complexity of the Top-K-Task algorithm is very low. No matter how big the data size is, the algorithm still works.

3. Experimental verification

3.1. Experimental data

In this paper, we access experimental data from the mobile application crowdsourcing test website Testin by crawling method. The paper selects data from March to April, the final experimental data is shown in table 2.

Table 2. Experimental data

| Data set | Testers number | Category | Tasks available | Tasks finished |
|----------|----------------|----------|-----------------|----------------|
| Testin | 1798 | 20 | 3612 | 9865 |

3.2. Evaluation index

The recommendation system commonly used evaluation criteria MAP (Mean Average Precision) to evaluate the experimental results, which is defined as "relevant" and "irrelevant" to assess the average correlation between recommended tasks and testers.

$$AveP(t) = \frac{\sum_{k=1}^n P(k) \times R(\text{rel}(k) > 0)}{\sum_{k=1}^n R(\text{rel}(k) > 0)} \quad (7)$$

$$R(x) = \begin{cases} 1, x = True \\ 0, x = False \end{cases} \quad (8)$$

$$MAP = \frac{\sum_{q=1}^Q AveP(t)}{Q} \quad (9)$$

AP(Average Precision) represents the average accuracy rate. P(k) represents the accuracy of k results, rel(k) indicates whether the position of k is relative, Q is the number of recommended tasks.

3.3. Experimental results and analysis

Table 3 compares the original data and the result from proposed Top-K-Task algorithm. In order to evaluate the proposed algorithm accurately and objectively, all of data is in the same parameter settings.

Table 3. Experiment results

| Data Set | MAP |
|----------------------|--------|
| Original data | 0.1123 |
| Top-K-Task algorithm | 0.1304 |
| Growth proportion | 16.12% |

From the above experiment results, the matching degree between tasks and testers in original data is low. The matching degree of Top-K-Task algorithm is higher than the original data, which proves the accuracy of the Top-K-Task algorithm.

4. Conclusion

The paper explores and improves the Top-K algorithm. The proposed Top-K-Task algorithm has the following characteristics:(1)Introducing task categories as intermediate mechanism, which can effectively avoid matching sparse matrix and large scale calculating. (2) Cosine similarity is used to calculate task similarity, which is more applicable than Euclidean distance when the dimensions of different attribute values are inconsistent.

We will further improve the Top-K-Task algorithm by considering the cold boot problem. So that the match between the testers and recommended tasks is closer, and the algorithm is more practical.

References

- [1] HOWE J. Crowdsourcing: why the power of the crowd is driving the future of busines [J]. *Journal of Consumer Marketing*, 2009, 26(4): 305 – 306

- [2] Zhiqiang Zhang, Jusheng Pang, Xiaoqin Xie, Kang Zhou. Research on Crowdsourcing Control Strategies and evaluation Algorithm[J]. Chinese Journal of Computers, 2013, 36(08): 1636-1649.
- [3] Safran M, Che D. Real-time recommendation algorithms for crowdsourcing systems[J]. Applied Computing & Informatics, 2016.
- [4] Xiang Ci, Youzhong Ma, Xiaofeng Meng. Method for Top-K Query on Big Data in Cloud[J]. Journal of Software, 2014, 25(04): 813-825.
- [5] Li-Yong Zhou. Algorithm on Top-k Keyword Search of Uncertain XML[A]. International Informatization and Engineering Associations, Atlantis Press. Proceedings of 2015 4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2015)[C]. International Informatization and Engineering Associations, Atlantis Press., 2015: 6.
- [6] Qiu Wei. Cosine similarity measures for dual hesitant fuzzy sets [A]. Proceedings of 2016 4th International Conference on Machinery, Materials and Computing Technology (ICMMCT 2016) [C], 2016: 4.
- [7] Li ZHANG, Peipei XIA, Fanzhang LI. A Method Based on Cosine Similarity in Supplier Selection.
- [8] Tingting LIANG, Chunqing LI, Haisheng LI. Top-K Learning Resource Matching Recommendation Based on Content Filtering Pagerank [J]. Computer Engineering, 2017, 43 (02): 220-226.