A Hybrid Similarity-Aware Clustering Approach in Cloud Manufacturing Systems

Jian Liu(✉) and Youling Chen

College of Mechanical Engineering, Chongqing University, Chongqing, China
liujiancqu@l26.com

Abstract. With the rapid development of cloud manufacturing (CMfg), a lot of cloud services are emerging on the Internet, which leads to cloud service clustering a critical topic. However, most existing approaches suffer from the low clustering quality due to the data sparsity condition, and are thus prone to the unreal result. To handle this problem, we put out a hybrid approach called HCA for cloud service clustering. At the first, we utilize Pearson Correlation Coefficient (PCC) and Proximity-Significance-Singularity (PSS) to compute the user similarity. Then, a similar group of users can be obtained using K-medoids algorithm, in which an ensemble model is established by incorporating those two user similarities. Based on two real-world data sets, the results show that the effectiveness of HCA.

Keywords: Cloud manufacturing · Clustering · Manufacturing systems · User similarity

1 Introduction

Nowadays, cloud manufacturing (CMfg) [1, 2] is becoming an emerging platform, has been put forward utilizing the modern information technologies, involving Big Data [3], Cloud Computing [4], Internet of Things [5]. The main purpose of CMfg are to put the distributed resources and idle capabilities into cloud pool, and provide cloud service to meet user's requirements (e.g., faster execution time, higher reliability, lower cost and so on.) [6].

However, the growing number of similar cloud services available in the CMfg systems raises a new research problem: it is costly and time consuming to make a cloud service selection of the large range of options. Under this circumstance, cloud service clustering is becoming a critical topic recently.

Clustering is an unsupervised classification algorithm, which aims to reduce the search space and find similar grouping of pattern [7]. In general, there are three classic clustering algorithms in the CMfg systems: K-means [7], K-medoids [8] and Hyper-graph partition algorithm [9].

For example, Ghazanfar et al. [7] found a similar group of users by integrating the user similarity into clustering algorithm, namely K-means. However, the results of this clustering algorithm mentioned above may be unreliable by averaging all the attribute values. To cope with the problem, Guo et al. [8] utilized a novelclustering algorithm called K-medoids to develop a multi-view recommendation system.

C.-F. Chien et al. (Eds.): *IE&EM 2019*, pp. 101–108, 2020.
https://doi.org/10.1007/978-981-15-4530-6_11

But it also shows a relatively poor clustering quality. Then, Yang et al. [9] designed a vertex hypergraph partitioning algorithm, EQHyperpart, to generate better partitioning results. The hypergraph partitioning algorithm has a faster execution time, but limited to low coverage and partitioning quality.

Above all, we put out a novel hybrid clustering approach, HCA, whose main purpose is to improve the clustering quality. In the HCA, firstly, we integrate the PCC and PSS into our improved K-medoids algorithm to achieve a better result. And the experiment part demonstrates the superiority of HCA.

2 Methodology

Considering the fact that PCC is more understandable and easy to implement, PCC becomes a common method to compute the user similarity, and has been widely used in the CMfg systems recently [10, 11]. And PCC is computed as follows:

$$PCC(u, v) = \frac{\sum_{s \in S} \left(r_{u,s} - \bar{r}_u\right)\left(r_{v,s} - \bar{r}_v\right)}{\sqrt{\sum_{s \in S} \left(r_{u,s} - \bar{r}_u\right)^2} \sqrt{\sum_{s \in S} \left(r_{v,s} - \bar{r}_v\right)^2}} \tag{1}$$

where $S = S_u \cap S_v$ is the set of co-rated services by user u and v, $r_{u,s}$ represents the value of service s rated by u, and $\bar{r}_u$ denotes the average value of all service s rated by u.

Despite high accuracy of PCC, there are still some problems remained that will lead to unsatisfactory results, since the user similarity is computed only based on the co-invoked users or services.

To overcome the problem and better compute the non-linear relationship between users, we also employ the Proximity-Significance-Singularity (PSS) model [12] to compute the user similarity which not only relies on the co-invoked services.

And the PSS model in this paper is defined as below:

$$\begin{aligned} PSS(u, v) = {} & Proximity(r_{ui}, r_{vj}) \times \\ & Significance(r_{ui}, r_{vj}) \times Singularity(r_{ui}, r_{vj}) \end{aligned} \tag{2}$$

where $PSS(u, v)$ denotes the similarity between u and v, r_{ui} and r_{vj} denote the rating value of i and j, respectively.

Here, *Proximity* is the absolute difference between r_{ui} and r_{vj}. *Significance* aims to measure the impact of the rating pair to the similarity values. *Singularity* denotes the difference between one rating pair to others. And the formulas of three functions are presented as below:

$$Proximity(r_{ui}, r_{vj}) = 1 - \frac{1}{1 + \exp\left(-\left|r_{ui} - r_{vj}\right|\right)} \tag{3}$$

$$Significance(r_{ui}, r_{vj}) = \frac{1}{1 + \exp\left(-\left|r_{ui} - \bar{r}_u\right| \cdot \left|r_{vj} - \bar{r}_v\right|\right)} \tag{4}$$

$$Singularity\,(r_{ui},\, r_{vj}) = 1 - \frac{1}{1 + \exp\left(-\left|\frac{r_{ui} + r_{vj}}{2} - \frac{\bar{r}_i + \bar{r}_j}{2}\right|\right)} \tag{5}$$

Based on the PCC and PSS, we can establish a novel ensemble model to compute user similarity, and the formula is presented as below:

$$Sim\,(u, v) = \lambda \times PCC(u, v) + (1 - \lambda) \times PSS(u, v) \tag{6}$$

where $Sim\,(u, v)$ is the hybrid user similarity between u and v. $\lambda\,(0 < \lambda < 1)$ denotes a specific parameter to affect how much the hybrid user similarity focus on the $PCC\,(u,\, v)$ and $PSS\,(u,\, v)$, and the impact of λ will be further studied in the following section.

Then, we use the improved clustering algorithm to find the similar subgroups of users by incorporating hybrid user similarity into K-medoids (see in Fig. 1). Different from the existing approaches, this improved clustering algorithm considers a true user as the center and then keep up the group characteristics [8].

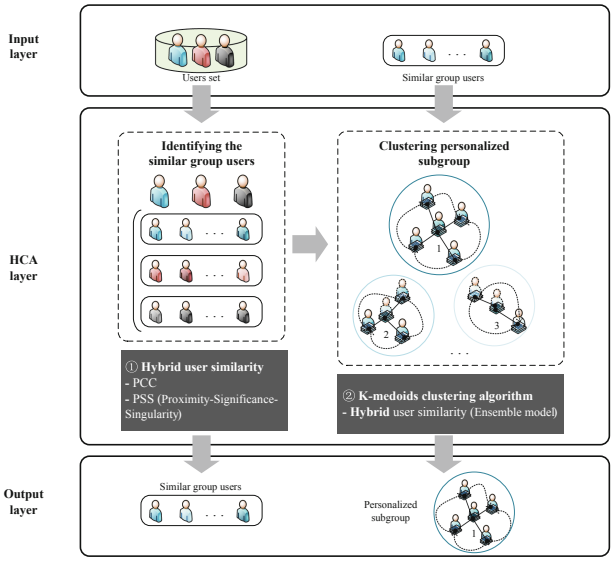

Fig. 1. Overview of the main steps of HCA.

The objective function is defined as below:

$$J = \min \sum_{c \in C} \sum_{u,v \in c} d(u, v) \tag{7}$$

Algorithm 1. K-medoids clustering algorithm

Input: distance matrix D_t ; number of clusters K

Output: user subgroups C

$p \leftarrow 0$;

randomly select K medoids m_t from users, $\theta_t^0 \leftarrow m_t$;

$C_t^0 \leftarrow u$, given $\min\left(d(u,m_t)\right)$; /* Calculate the distance using Eq. (8) */

while medoids changed **and** $< maxiterations$ **do**

  $p \leftarrow p+1$;

  $\theta_t^p \leftarrow \theta_t^{p-1}$;

  $swap(m_t, v)$, $v \in C_t^{p-1}$;

  calculate $sum_t(v) = \sum_u d(u,v)$, $u \in C_t^{p-1}$;

  if $sum_t(v) < sum_t(m_t)$ **then**

    $m_t \leftarrow v$;

    $\theta_t^p \leftarrow m_t$;

  $C_t^p \leftarrow u$, for $\forall u$, find m_t $s.t.$ $min\left(d(u,m_t)\right)$;

return $C \leftarrow C_t^p$;

where C denotes the user subgroups in which u and v belong to the subgroup $c \in C$, and $d(u, v)$ denotes the key distance between u and v.

In order to find the user subgroups, hybrid user similarity is replaced as the distance metric to cluster the similar users. More specifically, the higher user similarity means the users are closer than others [6].

Therefore, the distance can be calculated as follows:

$$d(u, v) = 1 - Sim(u, v) \tag{8}$$

Based on the above analysis, the pseudocode of our clustering algorithm is clearly presented in Algorithm 1.

3 Results

In this part, like our previous paper [6], we carry out a series of experiments to validate the effectiveness of our novel approach HCA.

To verify the performance of HCA, we collect two data sets in real-world. One is about web service called WSDream, which is gathered by Zheng et al. [10]. The WSDream-dataset includes 339 × 5825 rating records of two attributes (response time and throughput). Another is about CMfg service collected by Xiang et al. [13], including 463 users and 7548 services.

All the experiments were processed with Intel Core i5-4210 M 2.60 GHz processors and 4 GB RAM on a Lenovo E540 computer. The experimental parameters are presented in Table 1.

Table 1. Experimental parameters

Parameters	Web	CMfg
m: the number of users	100	100
n: the number of services	500	500
λ: how much HCA relies on PCC and PSS	0.45	0.45
K: maximum cluster number	6	8

In this section, like our previous paper [6] and [11], we employ the Normalized Mean Absolute Error, namely NMAE, as a metric to evaluate the clustering quality. Considering the fact that the huge ranges of response time in two different data sets, NMAE is computed as below:

$$\begin{cases} MAE = \dfrac{\sum_{u,i}(r_{ui} - \hat{r}_{ui})}{W} \\ NMAE = \dfrac{MAE}{\sum_{u,i} r_{ui}/W} \end{cases} \tag{9}$$

where MAE represents the average absolute deviation between the predicted value and real value according to the clustering users, r_{ui} denotes the real value of service i rated by user u, $\hat{r}_{ui}$ is the predicted rating value. W denotes the number of whole predicted values.

In order to achieve a performance comparison, we compare our hybrid clustering approach (HCA) with other popular approaches:

K-Mean: K-Mean clustering algorithm [7] is designed to search the similar group to improve the recommendation quality.

K-medoids: K-medoids [8] is another clustering algorithm to find the similar group using the PCC similarity model.

EQHyperpart: EQHyperpart [9] is a hypergraph partitioning method to obtain high quality clustering results based on information entropy modularity.

We also explore the impact of different parameters on the performance of HCA. Generally, we change a specific parameter under different condition (i.e., λ and K) while holding the others consistent. And the whole analysis process is concluded as below.

4 Discussion

The experimental results in Fig. 2 give that the value of NMAE in above approaches are decreasing as the matrix density increase, which means that the clustering quality can be significantly improved by employing more data in the rating matrix.

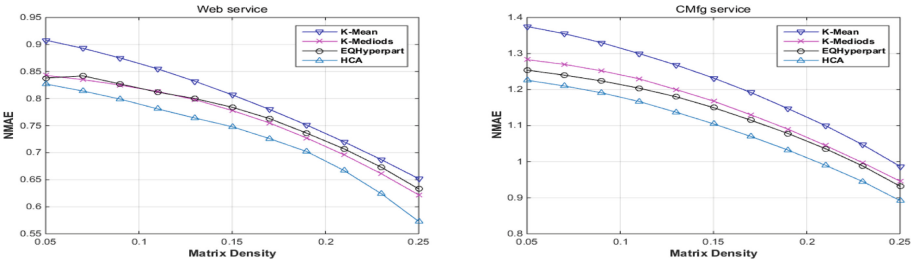

Fig. 2. Performance comparison of HCA.

Obviously, the NMAE value of HCA is relatively lower compared with other algorithms K-Mean, K-medoids and EQHyperpart. This is because HCA not only focus on the co-invoked rating records, but also the other rating information. By combining the both PCC and PSS to compute a hybrid user similarity seems to be helpful when making cloud service clustering. In addition, K-medoids show a better performance than other approaches when integrating a hybrid user similarity into it.

In Fig. 3, we can find that HCA obtains the best clustering quality in two data sets, namely web service and CMfg service, indicating that the optimal value of λ are affected by the matrix density to some extent. Since when $\lambda < 0.5$, the impact of PCC are less to be considered, while when $\lambda > 0.5$, the impact of PSS is also decreased. Thus, we set $\lambda = 0.45$ as the unchanged value in this part.

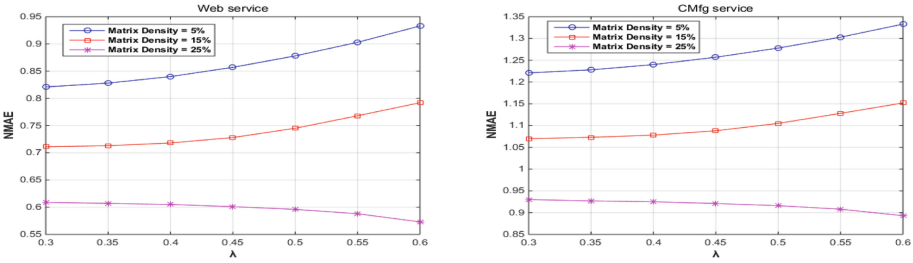

Fig. 3. Impact of λ.

As shown in Fig. 4, the experimental results of K does influence the clustering quality of our approach, an appropriate K value will give out better solution in two data sets. Besides, we set $K = 6$ in the web service data set, and $K = 8$ in the CMfg service data set.

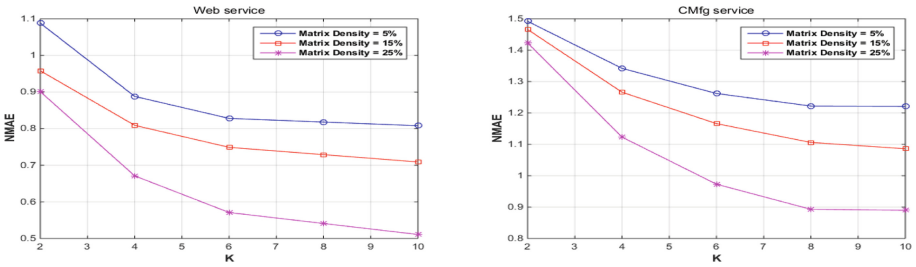

Fig. 4. Impact of K.

5 Conclusion

In cloud manufacturing systems, cloud service clustering is important for successful implementation of CMfg application. For a better clustering quality, we put out a hybrid clustering approach, called HCA. The main contribution of this manuscript lie in three-folds: (1) we establish an ensemble model to take advantage of both PCC and PSS. (2) we insert the hybrid user similarity into K-medoids clustering algorithm to acquire a high clustering quality. (3) we carry out a series of experiments on two data sets to validate the performance of HCA.

In the future, we plan to establish a comprehensive model to compute the user similarity by considering more factors in the CMfg systems.

Acknowledgment. The author would like to appreciate the editors and experts for their greatful and helpful comments which encouraged to improve the quality of the paper. And, this paper was supported in part by National Key Research and Development Program of China under grant No. 2018YFB1703002, and in part by the Fundamental Research Funds for the Central Universities under grant No. 2019CDCGJX222.

References

1. Li, B.H., Zhang, L., Wang, S.L., Tao, F., Cao, J.W., Jiang, X.D., Song, X., Chai, X.D.: Cloud manufacturing: a new service-oriented networked manufacturing model. Comput. Integr. Manuf. Syst. **16**(1), 1–7 + 16 (2010)
2. Zhang, L., Luo, Y.L., Tao, F., Li, B.H., Ren, L., Zhang, X.S., Guo, H., Cheng, Y., Hu, A.R., Liu, Y.K.: Cloud manufacturing: a new manufacturing paradigm. Enterp. Inf. Syst. **8**(2), 167–187 (2014)
3. Li, J.R., Tao, F., Cheng, Y., Zhao, L.J.: Big Data in product lifecycle management. Int. J. Adv. Manuf. Technol. **81**(1–4), 667–684 (2015)
4. Xu, X.: From cloud computing to cloud manufacturing. Robot. Comput. Integr. Manuf. **28** (1), 75–86 (2012)
5. Lu, Y.J., Cecil, J.: An Internet of Things (IoT)-based collaborative framework for advanced manufacturing. Int. J. Adv. Manuf. Technol. **84**(5–8), 1141–1152 (2016)
6. Liu, J., Chen, Y.: A personalized clustering-based and reliable trust-aware QoS prediction approach for cloud service recommendation in cloud manufacturing. Knowl.-Based Syst. **174**, 43–56 (2019)

7. Ghazanfar, M.A., Prügel-Bennett, A.: Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. Exp. Syst. Appl. **41**(7), 3261–3275 (2014)
8. Guo, G., Zhang, J., Yorke-Smith, N.: Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. Knowl.-Based Syst. **74**, 14–27 (2015)
9. Yang, W., Wang, G., Bhuiyan, M.Z.A., Choo, K.K.R.: Hypergraph partitioning for social networks based on information entropy modularity. J. Netw. Comput. Appl. **86**, 59–71 (2017)
10. Zheng, Z., Ma, H., Lyu, M.R., King, I.: QoS-aware web service recommendation by collaborative filtering. IEEE Trans. Serv. Comput. **4**(2), 140–152 (2011)
11. Liu, J., Chen, Y.: HAP: a hybrid QoS prediction approach in cloud manufacturing combining local collaborative filtering and global case-based reasoning. IEEE Trans. Serv. Comput. (2019)
12. Wang, Y., Deng, J., Gao, J., Zhang, P.: A hybrid user similarity model for collaborative filtering. Inf. Sci. **418–419**, 102–118 (2017)
13. Xiang, F., Jiang, G.Z., Xu, L.L., Wang, N.X.: The case-library method for service composition and optimal selection of big manufacturing data in cloud manufacturing system. Int. J. Adv. Manuf. Technol. **84**, 59–70 (2016)