2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovations

A Synergistic Cloud Service Approach for Cold Start Problems

1st Rui-dong Qi, 2nd Jian-tao Zhou*
College of Computer Science
Inner Mongolia University
Hohhot, China
csqrdong@mail.imu.edu.cn

3rd Xiaoyu Song
Department of Electrical and Computer Engineering
Portland State University
Portland, OR, USA
songx@pdx.edu

***Abstract*—More and more new cloud users use the personalised cloud service combination strategy (CSCS). Solving the Cold Start problem of the cloud environment becomes intractable. The paper represents a novel method of choosing the most optimal combinatorial features based on the attenuation function to cluster, and integrating multi level sampling method to cope with the pure Cold Start for cloud users. By means of every clustering process with different combinatorial features, then using the relatively stable number of clusters for every clustering obtains the optimal combinatorial features, which presents the tendency of the whole society of cloud users who use the CSCS. Meanwhile, we propose the function of periodic attenuation that enhances the degree of recommendation for CSCSs which have been issued recently. We harness the vectors of preference and disfavour to calculate the similarity of cloud users. An improved cluster algorithm of CFSFDP is employed. Moreover, it is worth selecting the most representative features to cluster which demonstrates effectively. In addition, the attenuation function can increase the probability of recommendation of recent CSCS, and the multi level sampling method has been used to heighten the diversity of recommendations. The method of ours can enhance the effectiveness and intelligence of recommendation for the pure Cold Start problem.**

***Keywords*-Cold start problem; cloud service combination; clustering algorithm; multi level sampling; recommendation**

I. Introduction

With the rapid development of cloud computing, more and more people have become cloud users[1]. Different cloud users have different demands for cloud services, which may include real-time monitoring, the threshold alarm, state prediction and other service resources [2]. Cloud service providers need know what kind of cloud service combination strategies (CSCSs) is beneficial.

In general, cloud users are not able to apply for appropriate CSCSs to service providers directly while they give a simple description of their demands only. Thus recommender systems play an important role [3], in which different CSCSs given by cloud service providers are recommended to cloud users to meet their requirements.

It is well known that collaborative filtering (CF) is one of commonly used techniques to generate recommendations[4]. The CF methods utilize a miraculous amount of data collected from user behavior in the past and predict intelligently what item users would like. Nevertheless, the inherent issue endures from data sparsity and the Cold Start (CS) problem[5].

Some state-of-the-art methods, such as social graphs[6], trust network[7], demographic attributes[8], Warm Start (WS)[9]etc. have been integrated with CF to solve the problem of Cold Start and improve the quality of service (QoS), but they cannot resolve the pure Cold Start problem completely, because all these methods need the information of the cloud user. In the case of the pure Cold Start, however, the recommended system has no way to obtain any information of the cloud user. Therefore, they can do nothing. At this moment, the classification and extraction of the cloud user will play a significant role in the cloud recommended system.

In this work, we propose the function of periodic attenuation. The function can increase the probability of recommendation for the recent CSCSs. Moreover, we use the number of clusters for clustering to find the optimal combination of cloud services. In addition, we present the recommender model. It uses the improved clustering algorithm and integrates multistage stratified sampling to solve the pure Cold Start problem of cloud environment intelligently.

The rest of this paper is organized as follows. Section II gives a brief overview of related research on CS from which the research gap is identified and motivating out present work. Section III, the proposed approach is introduced elaborately, and we emphasize the advantages of our method in principle. Section IV, we take experiments on Movelens data set to verify the effectiveness of our methods in the Cloud Cold Start (CCS). Finally, in section V, we will sum up all we have done throughout this entire process and provide an outlook of our work.

II. Related Work

In order to have a clear picture of the cloud recommender system and the Cold Start (CS) problem of recommender system, the subsection (II-A) will give a review of cloud

* Corresponding author.

978-1-5386-9380-3/18/$31.00 ©2018 IEEE
DOI 10.1109/SmartWorld.2018.00107

recommender systems. Subsection (II-B) will give a systematic literature review on the Cold Start (CS) problem of recommendation.

A. Recommender system of the cloud environment

In [10], a cloud recommender system to recommend an optimal cloud configuration to users based on accurate estimates is presented. The algorithm in [11] classifies cloud services into the different numbers of groups based on selected quality attributes and ranks them accordingly, which can assist each type of cloud user for choosing a cloud service.

In [12], this thesis proposes an intelligent method to meet the system for selecting cloud-based infrastructure service. A cloud recommender system can be divided into five categories. It had given the advantage and disadvantage systematically for each categorization[13].

In[14], a two-stage cross-domain recommendation for the Cold Start problem in cyber-physical systems (CPS) is presented. Using a meta-heuristic approach for the cloud resource allocation based on the bio-inspired coral-reefs, they optimize the model cloud elasticity in a cloud-data center[15].

B. Cold Start in the average Recommendation System

Many approaches have been proposed in the present literature to resolve the Cold Start (CS) problems. We can roughly divide them into the following six categories.

1) Modeling the preferences of new users can be done most effectively by taking a short interview[16], more than that, or let users participate in choosing the items [17].

2) Using the demographic data can help to identify other users with similar behaviors[18]. They present a new method for generating user's profiles that takes advantage of a large-scale database of demographic data [19].

3) They use social networks information to find similarities among users which are called communities, and different communities have different features, therefore, network can be divided into different communities [20].

4) They tackle the Cold Start problem with a switching hybrid solution, and switching hybrid content-aware (SHCA) algorithm to specific issues [21]. They exploit the W-SHCA model to solve the mixed Cold Start problem[22].

5) They use a deep learning approach to solve the Cold Start problem and sparsity problems in CF based on recommendation systems and propose techniques can be applied on top of any existing CF based on recommendation engine without changing the CF core [23].

6) They formalize the Cold Start problem as an optimization problem, and budget constraint uses monotone-super modular object function to improve efficient optimal design [24].

III. Models and Their integration

In this section, we propose the model of classification using the different service combination clustering. We use this model to obtain the cloud user classification. Moreover, we obtain the preference of cloud users in that times. Finally, we obtain these clusters of cloud users and integrate the modified multistage stratified samplings to recommend. Detailed descriptions and the insights of our methods will be presented in this section.

A. Modeling of classification

1) Variable declaration: For clarity, we introduce a number of notations to describe the pure cold start problem of cloud. U is the set of all cloud users. I is the set of all the CSCSs. F is the set of all the features. R is the set of all the ratings. Furthermore, we denote the symbols u, v for the cloud users and i, j for the CSCS, respectively. $t_{u,i}$ is the time of u rated i. t_n is the current time. T_i represents the issued time of a CSCS i. T_0 represents the current time. We define $|U| = n$, $|CSCS| = m$, $|R| = n \times m$, and $|F| = K$. $r^t_{u,i}$ represents a rating given by u on CSCS i at the time t. r^{avg}_u represents the mean value of cloud user u. Similarly, variables of max and min are given as r^{max}_u and r^{min}_u, respectively.

2) Parameters: We need to handle cloud users ratings when cloud users give the range of rating from one to five points. A cloud user A gives one, two or three points of ratings. Another cloud user B gives one, three or five points of ratings. We do not consider that a cloud user A with three points and B with three points are at the same level. So we should do the normalization processing for all the ratings of cloud users. We consider the normalization process as:

$$r^{t^*}_{u,i} = \frac{r^t_{u,i} - r^{arg}_u}{r^{max}_u - r^{min}_u}, \tag{1}$$

where $r^{t^*}_{u,i}$ belongs to (-1, 1). We can see that the bigger the value of $r^{t^*}_{u,i}$, the more cloud users like it. As the recent CSCS is more appealed to cloud users, we give the attenuate similar degree function of time as:

$$T^s(T_i) = \frac{1}{1 + (\alpha^{\left\lfloor \frac{T_0 - T_i}{T_*} \right\rfloor})}, \tag{2}$$

$$t^a(t_{u,i}, t_{v,i}) = (T^s)^{\left| \frac{t_n - t_{u,i}}{t_*} - \frac{t_n - t_{v,i}}{t_*} \right| + \ell}, \tag{3}$$

where T_* and t_* present the periodic attenuations, $T_* \neq t_*$, $\alpha > 1$, inspired from the nuclear decay. $\left\lfloor \frac{T_0 - T_i}{T_*} \right\rfloor$ represents the lower bound of $\frac{T_0 - T_i}{T_*}$. If $\left\lfloor \frac{T_0 - T_i}{T_*} \right\rfloor = 0$, CSCS is issued in n years ($0 < n \leq 10$), then $\alpha^{\left\lfloor \frac{T_0 - T_i}{T_*} \right\rfloor} = 1$. If

$\left\lfloor \frac{T_0-T_i}{T_*} \right\rfloor > 0$ and $\alpha^{\left\lfloor \frac{T_0-T_i}{T_*} \right\rfloor} > 1$. The larger the value of $\left\lfloor \frac{T_0-T_i}{T_*} \right\rfloor > 0$, the smaller the value of T^s. T^s ($T^s < 1$) is monotonic decreasing function. When $\left| \frac{t_n-t_{u,i}}{t_*} - \frac{t_n-t_{v,i}}{t_*} \right| > 0$, the function of t^a is monotonic decreasing, so the time of rating is longer, the degree of the attenuation is much bigger. If a CSCS is issued in recent years, and this CSCS has the smaller degree of the attenuation. In this cases, when $\left| \frac{t_n-t_{u,i}}{t_*} - \frac{t_n-t_{v,i}}{t_*} \right|$ is invariable, the larger the value of T_s, the larger the value of t^a.

We define the attenuated function of $t^a(t_{u,i}, t_{v,i})$ which characterises the similarity of different cloud users who rate the same CSCS. We define the attenuated function of the rating time of cloud users:

$$T^a_{u,i}(t_{u,i}) = (T^s)^{\left\lfloor \frac{t_0-t_{u,i}}{t_*} \right\rfloor + \ell}. \tag{4}$$

The attenuated function of rating and rating for the same CSCS will be defined as follows:

$$R^{T^a}_{u,i}(r^{t^*}_{u,i}) = \begin{cases} e^{(r^{t^*}_{u,i}+\varepsilon)} T^a_{u,i} \ ,(r^{t^*}_{u,i}) \geq 0 \\ -e^{(\left|r^{t^*}_{u,i}\right|+\varepsilon)} T^a_{u,i} \ ,(r^{t^*}_{u,i}) < 0 \end{cases}, \tag{5}$$

$$R^{t^a}_{u,i}(r^{t^*}_{u,i}) = \begin{cases} e^{(r^{t^*}_{u,i}+\varepsilon)} t^a \ ,(r^{t^*}_{u,i}) \geq 0 \\ -e^{(\left|r^{t^*}_{u,i}\right|+\varepsilon)} t^a \ ,(r^{t^*}_{u,i}) < 0 \end{cases}, \tag{6}$$

the recent rating has a higher value, which means that the recent rating has a higher weight during the process of calculation.

Definition 3.1: For a feature $f_k \in F$ $(1 \leq k \leq K)$, $Set^{f_k}_u$ is the set of CSCSs such that the user u rates CSCSs which has the feature of $f_k \in F$, and $F^{u+}_{f_k} = \frac{\sum_{i \in Set^{f_k}_u}(R^{T^a}_{u,i}+\gamma)}{\left|Set^{f_k}_u\right|}$, where $R^{T^a}_{u,i} \geq 0$ and $\gamma < 1$. In a similar way, $F^{u-}_{f_k} = \frac{\sum_{i \in Set^{f_k}_u}(R^{T^a}_{u,i})}{\left|Set^{f_k}_u\right|}$.

The $F^{u+}_{f_k}$ is a *u* who prefers CSCSs, and these CSCSs have the features of $f_k (1 \leq k \leq K)$ which exceeds the average rating, on the contrary, $F^{u-}_{f_k}$ is below the average rating.

Definition 3.2: For $F^{u+}_{f_k}$, a cloud user who rated CSCS which has $k(1 \leq k \leq K)$ features, $U^+_u = (F^{u+}_{f_1}, F^{u+}_{f_2}, \cdots, F^{u+}_{f_K})$ defines the fact that the cloud user u prefers feature vector. Similarly, let $U^-_u = (F^{u-}_{f_1}, F^{u-}_{f_2}, \cdots, F^{u-}_{f_K})$ as disfavored feature vector.

Definition 3.3: For a cloud user who has U^+_u and U^-_u, let $U^*_u = (U^+_u, U^-_u)$. U^*_u $(u \in U)$ represents the cloud user for feature vector with preference and disfavour.

Definition 3.4: For a feature $f \in K$, let $Set^s_{Cf} \subseteq F(1 \leq s \leq K)$ be the combination of features, where F is the features set.

Definition 3.5: Let C^s_K be the combinatorial number of features which selects s mutually different elements number from K, and $\left|Set^s_{Cf}\right| = C^s_K$.

In this way, we obtain Set^s_{Cf}, and we use these features to cluster, because the number or the category of features for each $Set^s_{Cf}(1 \leq s \leq K)$ is different. We take the attitude from which we can find the optimum Set^s_{Cf}. The set of $Set^{\cap}_{u,v}$ presents cloud users u and v have features in common.

3) Modeling: We define the vectors for every cloud users which can know the cloud users preference or disfavour. We use the Lance Williams distance[25] which can gain the similarity of features, so we define the function of similarity of features:

$$sim^{f^+}_{u,v}(U^*_{u,v}) = \begin{cases} \frac{\sum_{f_k \in Set^s_{Cf}} \frac{\left|F^{u+}_{f_k} - F^{v+}_{f_k}\right|}{F^{u+}_{f_k} + F^{v+}_{f_k}}}{\left|Set^{\cap}_{u,v}\right|} \ ,F^{u,v+}_{f_k} \neq 0 \ , \\ 0 \ , \quad otherwise \end{cases} \tag{7}$$

$$sim^{f^-}_{u,v}(U^*_{u,v}) = \begin{cases} \frac{\sum_{f_k \in Set^s_{Cf}} \left|\frac{F^{u-}_{f_k} - F^{v-}_{f_k}}{F^{u-}_{f_k} + F^{v-}_{f_k}}\right|}{\left|Set^{\cap}_{u,v}\right|} \ ,F^{u,v-}_{f_k} \neq 0 \ . \\ 0 \ , \quad otherwise \end{cases} \tag{8}$$

The Eq.7 and Eq.8 represent the similarity of preference and disfavor of cloud users *u* and *v*. If they have rated the same CSCS, they should have the similarity of preference or disfavour which must have a higher priority level than any other similarity.

We use $Set^{f_k}_a$ to represent cloud users u and v who rated the same CSCSs which have feature f_k, using Definition(3.1) $F^{u+}_{f_k} = \sum_{i \in Set^{f_k}_a}(R^{t^a}_{u,i} + \gamma)$ and $F^{u-}_{f_k}$, the same with Definition (3.2) $U^{a^+}_u = (F^{u+}_{f_1}, F^{u+}_{f_2}, \cdots, F^{u+}_{f_K})$ and $U^{a^-}_u$. With Definition (3.3), $U^{*a}_u = (U^{a^+}_u, U^{a^-}_u)$. In order to obtain this level of similarity, we define the similar function of rating the same CSCS as follows:

$$sim^{f^a+}_{u,v}(U^{*a}_{u,v}) = \begin{cases} \sum_{f_k \in Set^{f_k}_a} \frac{\left|F^{u+}_{f_k} - F^{v+}_{f_k} + \varepsilon\right|}{F^{u+}_{f_k} + F^{v+}_{f_k}} \ ,F^{u,v+}_{f_k} \neq 0 \\ 0 \ , \quad otherwise \end{cases}, \tag{9}$$

$$sim^{f^a-}_{u,v}(U^{*a}_{u,v}) = \begin{cases} \sum_{f_k \in Set^{f_k}_a} \left|\frac{F^{u-}_{f_k} - F^{v-}_{f_k} + \varepsilon}{F^{u-}_{f_k} + F^{v-}_{f_k}}\right| \ ,F^{u,v-}_{f_k} \neq 0 \\ 0 \ , \quad otherwise \end{cases}, \tag{10}$$

$$sim^{f^a}_{u,v}(U^{*a}_{u,v}) = \begin{cases} \sum_{f_k \in Set^{f_k}_a} \left|\frac{F^{u}_{f_k} - F^{v}_{f_k} + \varepsilon}{F^{u}_{f_k} + F^{v}_{f_k}}\right| \ ,F^{u,v}_{f_k} \neq 0 \\ 0 \ , \quad otherwise \end{cases}. \tag{11}$$

The Eq.9 and Eq.10 are calculated the similarity of cloud users u and v with the same $Set^{f_k}_a$ ($Set^{f_k}_a \subseteq Set^s_{Cf}$) , and they are the monotone decreasing functions. The Eq.11 means that if cloud user u and user v have the different opinions about the same CSCS. The $F^{u,v+}_{f_k} \neq 0$ presents that $F^{u+}_{f_k} \neq 0$ and $F^{v+}_{f_k} \neq 0$, $F^{u,v-}_{f_k}$ is handled in a similar way.

In general, $sim^+_{u,v}$ is the similarity of preference of u and v. We claim that the similarity of $sim^+_{u,v}$ is composed of four parts: preference of $sim_{u,v}^{f^+}(U_{u,v}^*)$, rating similarity $sim_{u,v}^{f^a+}(U_{u,v}^{*a})$, different preference $sim_{u,v}^{f^a}(U_{u,v}^{*a})$ and ω. Hence, $sim_{u,v}^{+}$ was computed as a linear combination of the three parts:

$$sim_{u,v}^{+} = \phi^{\frac{1}{1+sim_{u,v}^{f^+}}} + \varphi^{\frac{1}{1+sim_{u,v}^{f^a+}}} - \psi^{\frac{1}{1+sim_{u,v}^{f^a}(U_{u,v}^{*a})}} + \omega, \quad (12)$$

$$sim_{u,v}^{-} = \phi^{\frac{1}{1+sim_{u,v}^{f^-}}} + \varphi^{\frac{1}{1+sim_{u,v}^{f^a-}}} - \psi^{\frac{1}{1+sim_{u,v}^{f^a}(U_{u,v}^{*a})}} + \omega. \quad (13)$$

In the Eq.12 and Eq.13, $\phi > 1, \varphi > 1, \psi > 1$ and ω are regulation parameters. The values of $sim_{u,v}^{f^+}$ and $sim_{u,v}^{f^a+}$ are smaller, the similarity is greater. For a different Set_{Cf}^{s} , we obtain the different $sim_{u,v}^{+}$ and $sim_{u,v}^{-}$, and obtain matrixes of M_{sim}^{+} and M_{sim}^{-}.

We can classify all the cloud uses by similarity, so we should choose clustering methods for our classify. Clustering plays an important role in the fields of knowledge discovery and data mining[26]. Clustering algorithms attempt to organize data into different disjoint categories.

Alex Rodriguez and Alessandro Laio was proposed a new heuristic clustering algorithm CFSFDP (Clustering by fast search and find of density peaks) in 2014[27]. Peng Wang proposed the improvements on the density-based clustering algorithm that can discover the inflection point of the decision graph to automatically identify the cluster center[28].

Each object has vectors of positive and negative of correlation. On this basis, we improve the algorithm of Peng Wang and CFSFDP, and use vectors of preference and disfavour to cluster.

Definition 3.6: d_c^+ is a cutoff distance which is local density of data points for positive correlation of data sets, d_c^- is a cutoff distance for negative correlation of the data sets.

In the algorithm of CFSFDP, the number of d_c^+ is approximately equal to $(1-2)$ percent of the total number of data points and the number of d_c^- is approximately equal to $(2-4)$.

Definition 3.7: D is a data set and $i, j \in D$, ρ is the local density, and $\rho_i = \sum_j \chi(d_{i,j}^+ - dc^+, d_{i,j}^- - dc^-)$ and the function of $\chi(x, y)$ is $\chi(x,y) = \begin{cases} 1\ , & x \le 0, y \le 0 \\ 0\ , & otherwise \end{cases}$.

We redefine the d_c^+ and d_c^-. Besides, redefine ρ_i and the function of $\chi(x, y)$.

The improved algorithm clusters with different combinations of features. We have obtained the optimal combination of cloud services set of Set_{Cf}^{M} which can express the tendency of mainstream society in the era of cloud computing, and use Set_{Cf}^{M} to cluster. When obtain all clusters, $Set_{c_m^f}^{p} \in U$ $(1 \le p \le n)$ is a set for each cluster, and $p = \left|Set_{c_m^f}^{p}\right|$ is the cluster size, furthermore, m $(1 \le m \le M)$ is the serial number of the cluster.

B. Multistage stratified sampling

The previous subsection can obtain all $Set_{c_m^f}^{p} \in U$ $(1 \le p \le n)$. On the basis of clusters, we can find the tendency of preference or disfavor of cloud users. In this section, we use the multistage stratified sampling [29] to solve the Cold Start problem, and there define sampled weight of $Set_{c_m^f}^{p}$ as follows:

$$w_{Set_{c_m^f}^{p}}^{f} = \frac{\left|Set_{c_m^f}^{p}\right|}{\sum_{m=1}^{M}\left|Set_{c_m^f}^{p}\right|}, \quad (14)$$

$$w_{Set_{c_1^f}^{p}}^{f} + w_{Set_{c_2^f}^{p}}^{f} + \cdots + w_{Set_{c_M^f}^{p}}^{f} = 1. \quad (15)$$

When we obtain $Set_{c_m^f}^{p}$ and the weight of $w_{Set_{c_m^f}^{p}}^{f}$. We use stratified sampling to select the number of cloud users from every cluster randomly, giving the number of every cluster equation for first the stratified sampling:

$$Num_{Set_{c_m^f}^{p}}^{f} = \left\lfloor \left|Set_{c_m^f}^{p}\right| \times w_{Set_{c_m^f}^{p}}^{f} \right\rfloor. \quad (16)$$

Here Eq.16 can reduce the number of each cluster for sampling and it can enhance the weight of the cluster which represents the tendency of the mainstream society in that era. The first sampling to use for the second stratified sampling. In a similar way, we can compute each clusters weight of the secondary stratified sampling:

$$w_{Set_{c_m^f}^{p}}^{s} = \frac{Num_{Set_{c_m^f}^{p}}^{f}}{\sum_{m=1}^{M} Num_{Set_{c_m^f}^{p}}^{f}}, \quad (17)$$

$$Num_{Set_{c_m^s}^{p}}^{s} = \left\lfloor \left|Num_{Set_{c_m^f}^{p}}^{f}\right| \times w_{Set_{c_m^f}^{p}}^{s} \times \iota \right\rfloor. \quad (18)$$

$Set_{c_m^s}^{p}$ represents the secondary stratified sampling set of cloud users and $Num_{Set_{c_m^s}^{p}}^{s}$ is the size of $Set_{c_m^s}^{p}$. Similarly, using the stratified sampling can guarantee diversity of recommendation, and ι is a regulation parameter.

C. Recommendation

Using the stratified sampling to select cloud users for two times. The set of $Setc_{all}^{s}$ for cloud users is:

$$Setc_{all}^{s} = Set_{c_1^s}^{p} \cup Set_{c_2^s}^{p} \cup Set_{c_3^s}^{p} \cup \cdots \cup Set_{c_m^s}^{p}. \quad (19)$$

When the set of $Setc_{all}^{s}$ is obtained, so it selects CSCSs from I and these CSCSs are rated by cloud users who come from $Setc_{all}^{s}$, furthermore, for each cloud user $u \in Setc_{all}^{s}$, the $r_{u,i}^{t^*}$ of CSCS should be satisfied $(r_{u,i}^{t^*} \ge 0)$, so these CSCSs can be selected to recommend. When it obtains the set of CSCSs, named Set_c^i, it states for the following equation:

$$r_f = \frac{f_i^f \bigcap Set_{Cf}^{s}}{Set_{Cf}^{s}}, \quad (20)$$

$$Set_c^i = \{i \mid i \in I,\ u \in Setc_{all}^{s},\ r_{u,i}^{t^*} \ge 0 \\ ,\ r_f \ge \rho\}. \quad (21)$$

f_i^f is a set of features for i, and i is a CSCS. The value of ρ is $\frac{1}{\left|Set_{Cf}^s\right|}$ which means that the intersection of f_i^f and Set_{Cf}^s is 1. Eq. 21 obtains the set of Set_c^i, which has an army of CSCSs, so selecting CSCSs to recommend is an intractable issue, we define the vector of CSCS and find the popularity of CSCSs, and state for following equation:

$$r_{low}^{all} = \frac{num_{u,i}^r}{num_{u,i}^{r \leq \bar{r}} + 1}. \tag{22}$$

Definition 3.8: For a CSCS with $CSCS_i^r = (r_i^{arg}, r_{low}^{all})$ which rates by the cloud users of $Setc_{all}^s$, r_i^{arg} is the average rate of i and r_{low}^{all} is the degree of preference for CSCS.

$num_{u,i}^r$ ($u \in Setc_{all}^s$) is the number of cloud users who rated CSCS i. The number of cloud user who rated CSCS i and r_u^i was higher than the average rating is $num_{u,i}^{r \leq \bar{r}}$. Consequently, using $CSCS_i^r = (r_i^{arg}, r_{low}^{all})$ to select CSCS i must satisfy $r_i^{arg} \geq r_{all}^{arg}$ and $r_{low}^{all} \geq \tau$. Finally, we obtain Set_{end}^g, and choose n CSCSs from it for recommendation.

D. Main idea of model design

The algorithm 1 presents the main frame work of our approach. It takes as input the cloud user set of U, the CSCS set of I, the feature set of F and the rating set of R, and returns a set of Set_{end}^g is constituted by CSCSs.

There are mainly five phases of the procedure. When big data of history service of cloud computing obtained. The first phase calculates the vector U_u^+ of preference and the vector U_u^- of disfavor for every cloud user, then returns the $U_u^* = (U_u^+, U_u^-)$. We have obtained the feature vector of cloud users, and every feature represents a sort of cloud services. In the second phase, we calculate the similarity of each two cloud users by the each Set_{Cf}^s ($1 \leq s \leq 20, \left|Set_{Cf}^s\right| = C_K^s$) , and return the M_{sim}^+, and M_{sim}^-, the similarity of cloud users with three parts, the more details are described in the (Eq.7 - Eq.13). In this process, adding the Eq.11 which represents two cloud users has an opposite characteristic, the Eq.11 is especially emphasized. Also, we have structured the matrix of similarity of preference and disfavor by different combinational strategies of services. In the third phase, when we obtain the similarity matrices M_{sim}^+, M_{sim}^- and Set_{Cf}^s, we use the improved algorithm of CFSFDP to cluster, and obtain the number of these clusters for this time. For different cloud service features combination, we obtain different clusters. Moreover, the number of clusters is used to find the optimal cloud service features combination for recommendations. We use $(2^k - 1)$ times clusters to find the most represented features, and k is the number of cloud services. In the fourth phase, when we obtain the number of every clustering of Set_C^{Num}, we choose the steady number of clusters to calculate out the most frequent features for every Set_s, and return Set_{Cf}^M. The fifth phase, we use the features of Set_{Cf}^M and the improved algorithm of CFSFDP to cluster, and use the two times multistage stratified sampling to obtain $Setc_{all}^s$, then obtain Set_c^i. Furthermore, we use $CSCS_i^r = (r_i^{arg}, r_{low}^{all})$ to remove CSCS until the condition is met, and generate the Set_{end}^g to recommend.

Algorithm 1 OFCS-R-CS

Input: U, I, F, R.
Output: Set_{end}^g.

function USERCHARACTERISTICS(U, R)
  for each $u \in U$ **do**
    CalculateCharactPositive(U, R, U_u^+)
  end for
  for each $u \in U$ **do**
    CalculateCharactNegative(U, R, U_u^-)
  end for
  return $U_u^* = (U_u^+, U_u^-)$
end function
Calculating the Positive and Negative characteristics of cloud users.
function CALCULATESIM($U_u^*, U_u^{*a}, Set_{Cf}^s$)
  $1 \leq s \leq K$
  $Set_i \subseteq F$
  while $s < K$ **do**
    for each $s = 1, Set_i = C_K^i$ **do**
      $i = s$
      $CalculateSim(U_u^*, U_u^{*a}, Set_{Cf}^s)$
      $s++$
    end for
  end while
  return $(M_{sim}^+, M_{sim}^-, Set_{Cf}^s)$
end function
Constructing similar matrices for cloud users.
function CLUSTERNUM($(M_{sim}^{s+}, M_{sim}^{s-}, Set_{Cf}^s)$)
  for each $s = i, Set_i = C_K^i$ **do**
    $C_{Num} \leftarrow 0$
    $IMCFSFDP(M_{sim}^{s+}, M_{sim}^{s-}, C_{Num})$
  end for
  return (Set_C^{Num}, Set_s)
end function
Calculating the cluster number of each cluster for different combinatorial features.
function MOSTFREFEA((Set_C^{Num}, Set_s))
  $Set_{Cf}^s = null$
  while $\alpha \leq Num \leq \beta$ **do**
    CalcMostFreFea (Set_C^{Num}, Set_s)
  end while
  return Set_{Cf}^M
end function
Calculating the most frequently features.
function RECOMMENDATION($M_{sim}^{s+}, M_{sim}^{s-}, Set_{Cf}^M$)
  $Setc_{all}^p = null$
  $Set_{c_m^f}^p = IMCFSFDP(M_{sim}^{s+}, M_{sim}^{s-}, Set_{Cf}^M)$
  $FirstMultStratSamp(Set_{c_m^f}^p)$
  $SecondMultStratSamp(Set_{c_m^s}^p)$
  $Setc_{all}^s = Recommendation()$
  return Set_{end}^g
end function
Recommendation.

IV. Performance evaluation

In this section, we compare and evaluate the performance of our proposed model with the random selection CSCS model and the most popular CSCS model. The preparation and the results data of experiment are presented and discussed.

A. Experiment dataset and settings

A large real-world dataset provided by the MovieLens[30], we used the dataset of ml-latest-small which was generated on October 17, 2016.

In our experiment, $\alpha = 2$. $T_* = 5$ years represents the periodic attenuation for CSCS is five years, and $t_* = 3$ years represents the periodic attenuation of rating for users is three years. We set ϕ, φ and ψ is 1 and $\omega = 0$. $\ell = 1$ and $\gamma = 0.2$ are harmonic parameters, then we set $K = 20$ which means we have 20 categories of cloud services.

If the CSCS is produced recently, the probability of recommendation is higher. So we design the function of periodic attenuation (Eq.2) and we define 5 years as a periodic attenuation cycle in our experiment.

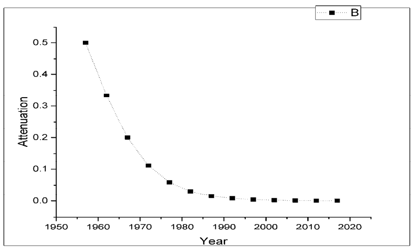


Figure 1. The periodic attenuation.

The periodic attenuation showed Fig.1, the slope of the adjacent two points become smaller with increasing of year. The recent CSCS has the higher probability of recommendation, and the attenuation is smaller. The function of periodic attenuation is innovative for any other functions of attenuation.

B. Find the optimal features (cloud resources) combination for recommendation

1) The experiment for all combinatorial features clustered: In our experiment, the category of cloud services is 20. Moreover, we cluster by different combinations of features, so we take $(2^{20} - 1)$ times clusters. Furthermore, the number of the combinations of features from 1 to 20, accordingly, we have obtained 19 figures of combination of features, which presents the number of clusters for every clustering. We show the four figures of most representative, in the following Fig.2. In subgraph of (Fig.2(a) - Fig.2(d)), the increment of the abscissa is 0.001.

Additionally, the headline of the Fig.2(b) is ten features, which means we have clustered by ten feature from twenty features, and the number of the combination of ten features is C_{20}^{10} ($C_{20}^{10} = 184756$).

In Fig.2, with the number of features increasing, the curve becomes smooth and steady gradually. Furthermore, the number of data points between the second and the third "turning point" increases gradually. And the two "turning points" become more and more obvious, furthermore, the fitting degree of all curves become similar.

We use the statistical method to discover the most eleven frequent features between the second and third "turning points" of all curve. The statistical result has been given in the Table I, which shows the frequency of every feature, and we obtain the most using features.

2) The selecting of combination of features: Above all, we do the experiment of all combination of features, and have found out the number of points of every curve between the second and the third "turning points". These points present the number of clusters have become steady. Every steady state of every curve is longer, so we choose this to calculate.

We calculate the frequency of features, then using statistical method obtains the most frequent features. In Table I, the first row represents the number of every combinatorial feature, and $(10 - F)$ represents choosing ten features from twenty features, and others are similar. The first column represents features from the first to the twentieth, and (1) represents the first feature, and so on. The number on the intersection of the *i-th* row and *j-th* column is the frequency of *i-th* feature in the *j-th* combinatorial features. We choose the frequent group of features is $(8th, 5th, 6th, 17th, 16th, 2nd, 4th, 11th, 13th, 12th, 20th)$, and use this group of features to cluster.

Table I
THE FREQUENCY OF FEATURES FOR DIFFERENT COMBINATION

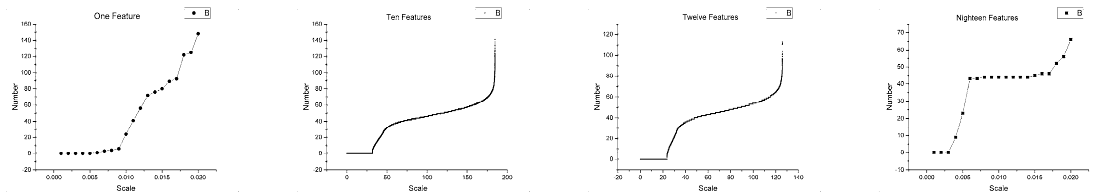
Feature	***10-F***	***11-F***	***12-F***	***13-F***	***14-F***	***15-F***	***16-F***	***17-F***	***18-F***	***19-F***
1	13613	14805	12748	8688	4699	2023	705	207	50	9
2	14986	16042	13722	9363	5133	2293	837	248	56	10
3	13420	15000	13350	9357	5181	2282	809	237	56	10
4	14227	15676	13764	9661	5398	2393	847	243	56	10
5	16497	17858	15420	10621	5836	2561	907	261	59	10
6	16187	16951	14296	9699	5360	2430	879	252	58	10
7	13665	15181	13460	9435	5212	2291	803	225	51	9
8	21323	21540	17418	11322	5957	2537	873	248	57	10
9	12646	13924	12374	8749	4920	2234	827	249	58	10
10	13862	15367	13536	9428	5190	2286	795	223	50	9
11	14016	15719	13870	9557	5192	2254	779	214	48	9
12	14230	15659	13589	9377	5214	2307	797	222	50	9
13	14092	15690	13820	9600	5281	2293	782	217	49	9
14	12997	14134	12248	8551	4893	2287	853	250	57	10
15	13946	15261	13273	9239	5169	2307	806	222	49	9
16	15813	16563	13911	9362	5051	2192	770	226	55	10
17	15620	17189	15014	10384	5709	2494	851	235	55	10
18	14129	15326	13160	9040	5023	2257	794	216	48	9
19	13159	14680	13059	9173	5056	2207	779	223	50	9
20	13862	15368	13536	9428	5190	2287	795	223	50	9

C. Recommendation to solve the Cold Start

In our experiment, we obtain the most optimal combinatorial features. These features are used to cluster for settling the Cold Start problem.

Firstly, we describe several baseline algorithms for finding a set of representative users or items. The Random Strategy is simply to randomly select a subset user or item. The Most Ratings Strategy is another simple method to select the representative the k users or items which had most observed ratings [31].

Secondly, to simulate the pure Cold Start problems, we divide the users into two disjoined subsets of training and



(a) One feature (b) Ten features (c) Twelve features (d) Nineteen features

Figure 2. The number of clusters for different combinational features.

test. We use eighty percent of the users for training to obtain the most optimal combinatorial features, and the remained twenty percent for testing which is for the Cold Start scenario. Furthermore, we adopt the recommender methods of Top-N[32] for the Cold Start problem. Yet, we compare our algorithms with other competitive algorithms: the Random Strategy and the Most Ratings Strategy.

Thirdly, to evaluate the quality of our model base on the following quantitative measures [33]. $R(u)$ is a list of the recommender items and $T(u)$ is a list of test data for cloud user:

- **Precision**: we use the $R(u)$ and $T(u)$. So the equation is showed:

$$Precision = \frac{\sum_{u \in U} |R(u) \bigcap T(u)|}{\sum_{u \in U} |R(u)|}. \tag{23}$$

- **Real-time**: $R(u)^p$ is a list of the recommending items which are issued in the ten years. The equation of the Real-time is:

$$Realtime = \frac{\sum_{u \in U} |R(u)^p \bigcap T(u)|}{\sum_{u \in U} |T(u)|}. \tag{24}$$

- **Diversity**: d_i is a set of features for recommendation and has n features ($0 \leq n \leq 20$). The $Set_D = d_1 \cup d_2 \cup ... \cup d_i$ is the total number of features for all recommendations, and the num_d ($num_d = 20$) is the number of all the features, so the equation of diversity is:

$$Diversity = \frac{|Set_D|}{num_d}. \tag{25}$$

Fig.3(a) is the precision of three algorithms. The algorithm of OFCS-R-CS is higher one thousandth than other two algorithms. Fig.3(b) is the real-time of three algorithms, we can observe our algorithm is more fifty percent than other two algorithms. Fig.3(c) is the diversity of three algorithms, furthermore, the three algorithms approach the same level of diversity.

Above all, the algorithm increases the real-time greatly, and the precision is higher than other two algorithms, and is not reduce the percent of diversity, so our models is effective.

The horizontal axis of Fig.3(a), Fig.3(b) and Fig.3(c) is from ($1T - 13T$) which represents 1, 10, 50, 100, 200, 500, 1000, 2000, 3000, 5000, 6000, 8000 and 10000 times recommendation. Furthermore, we calculate the average of the precision, the real-time and the diversity.

V. Conclusion and the Future Work

In this work, we firstly propose that we use the number of clusters for every clustering to find the optimal combination of cloud services. At the same time, we propose the function of periodic attenuation in order to increase the probability of the recommendation for recent CSCSs, and we propose the intelligent recommender model which improves the algorithm of CFSFDP and integrates the multistage stratified sampling to cope with pure Cold Start problem of cloud users. Our method can increase the diversity and precision, meanwhile, increase the real-time obviously than two algorithms of recommendation for the pure Cold Start problems, and the result of experiment proves effectively. Our experiment costs many times clustering to find the optimal combination of cloud services, so we should reduce the times of clustering in the future work.

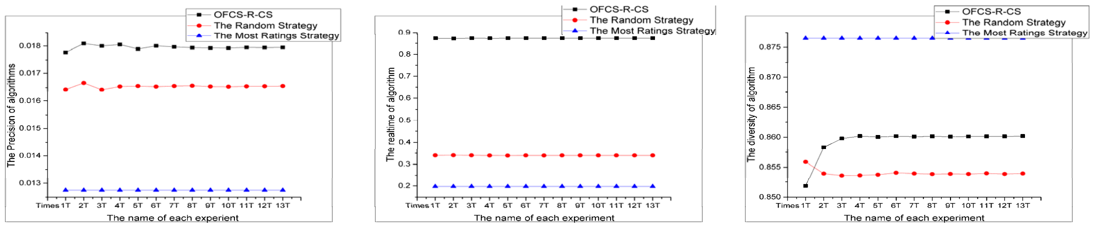
Acknowledgment

The authors wish to thank Natural Science Foundation of China under Grant No. 61662054, 61262082, Natural Science Foundation of Inner Mongolia under Grand No.2015MS0608, Inner Mongolia Science and Technology Innovation Team of Cloud Computing and Software Engineering and Inner Mongolia Application Technology Research and Development Funding Project "Mutual Creation Service Platform Research and Development Based on Service Optimizing and Operation Integrating", Inner Mongolia Engineering Lab of Cloud Computing and Service Software and Inner Mongolia Engineering Lab of Big Data Analysis Technology, and the Enhancing Comprehensive Strength Foundation of Inner Mongolia University (No.10000-16010109-25).

References

[1] P. K. Senyo, E. Addae, and et al, "Cloud computing research: A review of research themes, frameworks, methods and future research directions," *International Journal of Information Management*, vol. 38, no. 1, pp. 128–139, 2018.

[2] Y. M. Afify, I. F. Moawad, and et al, "A personalized recommender system for saas services," *Concurrency Computation Practice Experience*, vol. 29, 2017.



(a) The precision of algorithms (b) The realtime of algorithms (c) The diversity of algorithms

Figure 3. The experiments of precision, realtime and diversity.

[3] F. Aznoli and N. J. Navimipour, "Cloud services recommendation: Reviewing the recent advances and suggesting the future research directions," *Journal of Network Computer Applications*, vol. 77, pp. 73–86, 2016.

[4] J. Bobadilla, F. Ortega, and et al, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, no. 1, pp. 109–132, 2013.

[5] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.

[6] A. Hannech, M. Adda, and et al, "Cold-start recommendation strategy based on social graphs," in *Information Technology, Electronics and Mobile Communication Conference*, 2016, pp. 1–7.

[7] C. Haydar, A. Boyer, and A. Roussanaly, "Hybridising collaborative filtering and trust-aware recommender systems," in *WEBIST*, 2012, pp. 695–700.

[8] L. Safoury and A. Salah, "Exploiting user demographic attributes for solving cold-start problem in recommender system," vol. 1, no. 3, pp. 303–307, 2013.

[9] G. Contardo, L. Denoyer, and et al, "Representation learning for cold-start recommendation," *Eprint Arxiv*, 2014.

[10] G. Jung, N. Sharma, and et al, "Cloud capability estimation and recommendation in black-box environments using benchmark-based approximation," in *IEEE Sixth International Conference on Cloud Computing*, 2013, pp. 293–300.

[11] T. Zain, M. Aslam, M. R. Imran, and et al, "Cloud service recommender system using clustering," in *International Conference on Electrical Engineering, Computing Science and Automatic Control*, 2014, pp. 1–6.

[12] M. Zhang, R. Ranjan, and et al, "Investigating decision support techniques for automating cloud service selection," in *IEEE International Conference on Cloud Computing Technology and Science*, 2012, pp. 759–764.

[13] F. Aznoli and N. J. Navimipour, "Cloud services recommendation: Reviewing the recent advances and suggesting the future research directions," *Journal of Network Computer Applications*, vol. 77, pp. 73–86, 2016.

[14] P. Liu, J. Cao, and et al, "A two-stage cross-domain recommendation for cold start problem in cyber-physical systems," in *International Conference on Machine Learning and Cybernetics*, 2015, pp. 876–882.

[15] M. Ficco, C. Esposito, and et al, "A coral-reefs and game theory-based approach for optimizing elastic cloud resource allocation," *Future Generation Computer Systems*, 2016.

[16] A. Kohrs and B. Merialdo, "Improving collaborative filtering for new users by smart object selection," 2016.

[17] A. M. Rashid, G. Karypis, and J. Riedl, "Learning preferences of new users in recommender systems: an information theoretic approach," *Acm Sigkdd Explorations Newsletter*, vol. 10, no. 2, pp. 90–100, 2008.

[18] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

[19] B. Krulwich, "Lifestyle finder: Intelligent user profiling using large-scale demographic data," *Ai Magazine*, vol. 18, no. 2, pp. 37–45, 1997.

[20] S. Sahebi and W. W. Cohen, "Community-based recommendations: a solution to the cold start problem," 2011.

[21] M. Braunhofer, V. Codina, and F. Ricci, "Switching hybrid for cold-starting context-aware recommender systems," pp. 349–352, 2014.

[22] V. N. Zhao, M. Moh, and et al, "Contextual-aware hybrid recommender system for mixed cold-start problems in privacy protection," in *IEEE International Conference on Big Data Security on Cloud*, 2016, pp. 400–405.

[23] J. Yuan, W. Shalaby, and et al, "Solving cold-start problem in large-scale recommendation engines: A deep learning approach," in *IEEE International Conference on Big Data*, 2017, pp. 1901–1910.

[24] O. Anava, S. Golan, and et al, "Budget-constrained item cold-start handling in collaborative filtering recommenders via optimal design," *Computer Science*, pp. 45–54, 2014.

[25] G. Jurman, S. Riccadonna, and et al, "Canberra distance on ranked lists," in *Advances in Ranking NIPS 09 Workshop*, 2009.

[26] L. Arockiam, "Clustering techniques in data mining," 2012.

[27] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, p. 1492, 2014.

[28] P. Wang and J. Wang, "A clustering algorithm based on find density peaks," in *2017 7th International Workshop on Computer Science and Engineering*, 2017.

[29] H. Xie, X. Tong, and et al, "A multilevel stratified spatial sampling approach for the quality assessment of remote-sensing-derived products," *IEEE Journal of Selected Topics in Applied Earth Observations Remote Sensing*, vol. 8, no. 10, pp. 4699–4713, 2016.

[30] MovieLens, "Ml-latest-small," http://grouplens.org/datasets/, October 2016.

[31] N. N. Liu, X. Meng, and et al, "Wisdom of the better few: cold start recommendation via representative based rating elicitation," in *ACM Conference on Recommender Systems*, 2011, pp. 37–44.

[32] M. Deshpande and G. Karypis, "Item-based top- n recommendation algorithms," *Acm Trans.inf.syst*, vol. 22, no. 1, pp. 143–177, 2004.

[33] F. Ricci and L. e. a. Rokach, *Recommender Systems Handbook*. Springer, 2011.