First International Congress on Technology, Communication and Knowledge (ICTCK 2014)
November, 26-27, 2014 - Mashhad Branch, Islamic Azad University, Mashhad, Iran

Solving cold start problem in tag-based recommender systems using discrete Imperialist Competitive Algorithm

Mohammad Hossein Jafari
Department of Software Engineering
Mashhad Branch, Islamic Azad University
Mashhad, Iran
m.h.jafari@mshdiau.ac.ir

Ghamarnaz Tadayon Tabrizi
Department of SoftwareEngineering
Mashhad Branch, Islamic Azad University
Mashhad, Iran
tadayon@mshdiau.ac.ir

Mehrdad Jalali
Department of SoftwareEngineering
Mashhad Branch, Islamic Azad University
Mashhad, Iran
jalali@mshdiau.ac.ir

***Abstract*—Recommender systems detect users' favorites based on their past behavior and provide them with proper suggestions; however, these systems would encounter problems while dealing with users with low or empty usage data. This issue leads to the most prominent challenge of such systems called cold start. In thispaper, we proposea system based on which a modified discrete imperialist competitive algorithm where tags are clustered using K-medoids algorithm. When a new user logs in and enters his/her tags then the system will suggest just a few sources with the largest weight. Experimental results demonstrate improvement of evaluation criteria for recommender system in comparison with other methods.**

I. INTRODUCTION

During last decades, internet, information technology and digital information developments have resulted in increasing growth of knowledge, which have in turn, increased information overload. It is important to provide helpful information for users form this wide range of information. Thus, several research works have focused on personalization [1]or customization[2, 3] of information according to users' favorites. In this way, users are able to find their desired information with lower cost and time. Numerous methods have been proposed to realize this criterion one of which is information retrieval[4].In fact, information retrieval process is execution of the first command of user. It includes finding information from database to match users' demand and to return results to the users[3]. In order to overcome shortcomings of information retrieval, information filtering was introduced [5]. Information filtering is one of the most effective tools for reducing extra information basedon analysis of users' behaviors to obtain their priorities and favorites in this situation where it is able to filter and display their required information.

Thedifferencebetweeninformation retrieval and information filtering is thatinformation retrieval must passively wait for query command from users before proceeding further; in contrast, information filtering can actively assist users to find the relevant information they are interested in.

Recommender systems are indeed a sub-class of information filtering systems [6].The goal of a recommender system is to generate good suggestions for a set of users about products that they may like. A good recommendation is to provide sufficient information and to understand customers' preference for finding what they want. Therefore, recommendation in essence is usually important as personalized recommendation. The mainpurposeofrecommendation aims to timely provide the suitable and valuable information according to users' demand and such information will be used as reference for supplementing decision-making[7].

Recommender systems provide the most relevant and precise recommended items to user according to users' preference.These systems are usually classified into five categories: rule-based filtering[3], demographic filtering[8], content-based[9], collaborative filtering [3], hybrid approaches[9].

Although there are some advantages in collaborative filtering, it has its particular limitations, which one of themoccurs in the face of new members.When a new user enters the system, there is not enough information to generate suggestions, which leads to cold start problem.

Common solutions are based on using hybrid recommendation technologies, which combine content data with collaborative data [10, 11] and sometimes they include asking some basic information such as age, location and favorites.Three problems regarding cold start might be distinguished[12]:

978-1-4799-8021-5/14/$31.00 ©2014 IEEE

1- Cold society: it refers to the startup problem of a recommender system associated with achieving enough amounts of data for generating proper suggestions.

2- Cold items: usually new items do not have any information such as score and tag; therefore, they are not so probable forrecommendation. Thus, they will not be entered recommendations list and hidden from users.

3- Cold user: since new users in recommender system have not provided any information, they cannot receive personal recommendations based on collaborative filtering.

II. RELATED WORK

The closest approaches to ours for addressing cold start problem are as follows. In [13],feature taking and clustering methods were employed to justify cold start problem.This study usesthe cosine vector method for deriving similarity matrix and clustering of users. When new users log in the system the system classifies them into different groups based on registered features (such as age, gender,etc.); thus, a new user can obtain new recommendations for its class. In [14], a system is proposed where classification algorithm is utilized together with similarity methods and prediction mechanisms. Proposed system presents the results of personalized recommendations for new users utilizing their demographic features. In[15], cluster social ranking is suggested. Wecallcontent providers as leader and calls searchers as imitators. First off, by using C-means fuzzy algorithm leaders are detected and clustered into societies which they have common favorites. Cosine similarity method determines the amount of dependency on the cluster center. The system guides Users' queries to the leader society who is able to provide the best response. In [16], content based filtering and collaborative filtering are exploited as two main technologies for improvement of recommender systems. Content based filtering obtains users' favorites and generates recommendations by analyzing items or users' features.

Our paperproposes a recommender system to overcome cold start problem. In our proposed system, in offline phase, tags are clustered using modified discrete imperialist competitive algorithm with K-medoids algorithm. The K-medoids algorithm uses WordNet ontology. Furthermore, in online phase when new user's tags enter, the system extracts and weights k neighbors of new tags and elements where these neighbors have tagged these tags andelements then suggests a few sources with the largest weight.

III. PROPOSED SYSTEM

The proposed system is composed of six modules, data cleaning, imperialist competitive, empire extraction, neighbor extraction, recommendation engine and semantic similarity. Fig. 1 depicts a schematic of proposed architecture.

In offline phase,after clustering and receiving the tags in online phase, which are desired by new user this system recommends sources to new user. It is an advantage of our proposed system, as it does not need to receive extra information form user such as age and gender.

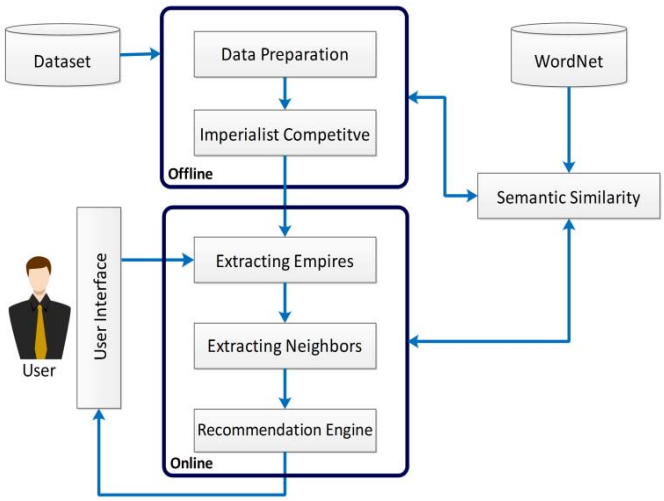

Figure 1. The architecture of the proposed system.

A. Data cleaning module

The first module, which improves the quality of output results in offline phase, is data cleaning module. First off, it eliminatesthe several types of the data items:

- Nonetagged items.
- Items consisted of tags with length less than three characters.
- Itemsconsisted of tags a combination of numbers and letters.
- Itemsconsisted of tags prepositions.

Afterwards, P-core parameterisappliedto dataset. It guarantees each user's event, source and tag in at least P annotations of database [17]. Finally, a part of data is selectedas training data and the other part is used as test data according to ratio parameter. Since in the next module the operation only involves training data set tags so extracts and useonly the tags of this dataset to the next module.

B. Imperialist competitive module

This is the last module in offline phase. This module employsthe discrete version of imperialist competitive algorithm and the K-medoids clustering for data clustering. To state mapping of imperialist competitive algorithm to clustering conceptsexplicitly, we must mention that imperialists, colonies, empires and objective function represent cluster head, cluster members, whole cluster and distance measure function, respectively.

1) Determining initial population

Firstly, initial population is determined. Since our problem is a discrete one and random generation of tags is meaningless, comparing with continuous version of imperialist competitive algorithm it is the first part, which has changed. In fact, initial population is considered as tags where are assigned to sources by users.

$$Population = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{\mathrm{M_{country}}} \end{bmatrix} \quad (1)$$

$$M_{country} = M_{imp} + M_{col} \quad (2)$$

M_{imp}, M_{col} and $M_{country}$ are the number of imperialists, colonies and initial population, respectively and x denotes country (tag). At the beginning, the algorithm randomly selectsM_{imp} countries as cluster heads.

2) Objective function evaluation

The algorithm calculates Objective function for each country. As shown in (3) the basis of this expression is the distance between tags, which is indeed the dissimilarity between tags.

$$F(x) = min_{j=1,2,\ldots,M_{imp}} \left(dis\_sim_{norm}(C_j, x) \right) \quad (3)$$

Where x denotes a country and C_j is the center of jth cluster. Moreover, $dis_sim_{norm}(C_j, x)$is the distance between two tagscanbeobtainedfrom (9).

3) Sorting initial population based on objective function

Next stage sorts countries or tags based on objective function in an incremental order. Each country with smaller objective function value is more powerful. Actually, objective function and power are inversely related.

4) Determining imperialists and colonies

As the goal is to minimize objective function, the algorithm selects a number (equal to number of initial empires) of countries with the least value of objective function as imperialists and considers other countries as colonies as well.

5) Initial assignment of colonies to imperialists

This stage assigns a number of colonies to each imperialist proportionate to its power.

6) Movement of colonies toward imperialists

Due to discrete nature of the problem, the closeness of colonies to imperialists is not meaningful and the Euclidean distance between them is not useful. Therefore, this section has also changed comparing to continuous version of imperialist competitive algorithm. For moving toward imperialists, the values of objective functions are updated.

7) Replacement of imperialist and colony

If there is a colony in the empire, which decreases empire cost in case of being imperialist, the algorithm replaces imperialist and that colony. To determine such colonies, our proposed algorithm utilizes K-medoids algorithm. Fig. 2 illustrates the pseudo code of the algorithm, which is appliedall empires.

8) Calculating the cost of each empire

The cost function of each empire is the value of objective function of imperialist added to a percent of average value of colonies objective functions.

$$TC_n = Cost(imperialist_n) + \varphi\left(mean\left(cost(coloniesofempire_n)\right)\right) \quad (4)$$

9) Competition between empires

The algorithm inversely relates the probability of assigning colonies to empires to their cost function value. The algorithm selects more probably the empire with less cost function as superior empire. When the superior empire is determined, the algorithm assigns the weakest colony from the weakest empire to it, in turn, reducespopulation and power of the weak empire.

10) Eliminating the weakest empire

During competition between empires, the weakest are eliminated. However, with a small change in the main algorithm, elimination does not mean complete elimination of the weakest imperialist; but it will become a colony of the most powerful imperialist.

11) Empires number test

The number of empires is tested and until it does not reach N, the algorithm will be repeated from step 6.

This module has some advantages as follows:

- It is not sensitive to the order of input data and presents the same clustering for all types of data.
- In K-medoids algorithm if the number of cluster members is zero, there is no way to change, modify and continue the method. This problem in this moduleis solved.
- Optimization ability of imperialist competitive algorithm is higher comparing with other optimization algorithms.
- Problem formulation and understanding the process is simple.

1- Select imperialist as initial medoid (O_{medoid}).

2- Randomly select a non-medoid from colonies (O_{random}).

3- Calculate empire cost while O_{medoid} and $O_{random}$are replaced. If it decreases replace $O_{medoid}$with $O_{random}$and update colonies objective function; otherwise go to step 2.

4- Continue steps 2 and 3 as far as all countries are selected as medoids and clusters do not change.

Figure 2. pseudo code based K-medoids.

C. Semantic similarity module

To calculate the semantic difference between two tags, firstly, WordNet ontology is checked according to existence both of two tags. If there are both of two tags in the WordNet ontology, it can useaccording to semantic similarity which based on Ontology [18] and otherwise it can use in relationof Co-Occurrence between two tags [19].

In order to calculate semantic similarity based on ontology, let C be the set of concepts of an ontology,the set of taxonomical features that are describing the concept a is defined in terms of the relation ≤ as:

$$\omega(a) = \{c \in C | a \leq c\} \quad (5)$$

In order to calculate thenormalizeddissimilarity (6) is proposed.

$$dis_{norm}(a,b) = log_2\left(1 + \frac{X+Y}{X+Y+Z}\right) \quad (6)$$

Values ofX, Y andZ in(6) are calculatedbythe followingequations.

$$X = |\omega(a) - \omega(b)| \quad (7)$$

$$Y = |\omega(b) - \omega(a)| \quad (8)$$

$$Z = |\omega(a) \cap \omega(b)| \quad (9)$$

Whereas the words may have a plurality of meanings and appear in different Synsets as well, thus, (10) was proposed to solve the problem.

$$dis_{generalized}(a,b) = \min_{\substack{\forall \acute{a} \in A \\ \forall \acute{b} \in B}} dis_{norm}(\acute{a},\acute{b}) \quad (10)$$

Where theAisseries of concepts(such as synonymousterms) for thewordsa, Bis set ofconceptsforthe wordb.

This semantic distance measurement is only applicable to words and it is not suitable for sentences. To calculate semantic similarity between two sentences X and Y, pseudo code of fig. 3 is proposed.

1- Extract all words from X and Y.
2- Calculate semantic distance between all words using (10).
3- For each word in X find the most similar word in Y and add their dissimilarity with sim_x.
4- For each word in Y find the most similar word in X and add their dissimilarity to sim_y.
5- Calculate the dissimilarity of two sentences according to (11).

$$dis\_sim_{Overall}(X,Y) = \frac{(sim_x + sim_y)}{(|x|+|y|)} \quad (11)$$

Figure 3. Pseudo code for calculating dissimilarity of two sentences.

Although WordNet ontology provides a good coverage, less than 1 percent of words in test database do not exist in WordNet ontology. To solve this problem Co-Occurrence relation between tags is used [19]. Equation (12) calculates this relation between tags of all sources in social tagging system. More details about (12) could be found in[19].

$$P_{t_y}(t_z) = \sum_{d \in C} q(t_z \mid d) Q(d \mid t_y) \quad (12)$$

Since Co-Occurrence equation returns non-normalized similarity value between two tags, suppose(13) achieves normalized dissimilarity between two tags, t_z and t_y.

$$P_{t_y}(t_z)_{norm} = \left(1 - \frac{P_{t_y}(t_z)}{|C|}\right) \quad (13)$$

In this equation, C is the number of sources annotated by both t_z and t_y. Finally, to calculate the distance between two tags $t_z$and t_y, we suggest(14)

$$dis\_sim_{norm}(t_z,t_y) = \quad (14)$$

$$\begin{cases} dis_{generalized}(t_z,t_y) & \exists\, t_z, t_y \in Wordnet \\ P_{t_y}(t_z)_{norm} & otherwise \end{cases}$$

D. Empire extraction module

The set of new user's tag consist of two types of tags.

First type includes tags existing inside the clusters.T_1denotes this set of tags and marks them as new country.

The second type consists of tags not existing in clusters. We show this set by T_2. Thus, the members of this set are considered as new colony. At the end, we calculate dissimilarity between T_2members with cluster centers so that it may achieve the lowest semantic similarity.

E. Neighbor extraction module

In this moduleencounterswithmembersof T_1 and T_2as follows:

1- T_1 tags: first of, k neighbors of this set with the highest similarity are extracted. N shows this set of tags and members of T_1; because, definitely there are some sources in data set, annotated by T_1 tags. Besides,dissimilarity of Tags N extracts and shows by S.

2- T_2 tags: firstly, the algorithm extracts k neighbors inside this set with the least dissimilarity and calls them as N.Besides,dissimilarity of Tags N extracts and shows by S.

Now extracts all non-repeated sources annotated with N members and showsthem as R. Since it is possible that there are some sources annotated with several members of N, the weights of all sources are calculated and shown by W. For this purpose, we exploit the average of dissimilarity tags of each source.

F. Recommender engine module

To present recommendations, the algorithm monitors W set and suggestsλ sources with the minimum weights from R.

IV. EXPERIMENTAL RESULTS

The test database belongs to CiteUlike website consisting information from November 2004 to November 2009. Table I presentsthe characteristics of this database.

TABLE I. THE SPECIFICATIONS OF CITEULIKE DATA SET

#User	#Article	#Tag	#Annotation
41,246	1,254,406	210,385	4,802,916

To state precise dimensions of database, we investigateit from three perspectives; distribution of user on the number of separate tags, distribution of users on the number of separate articles and distribution of articles on the number of separate tags.

The first dimension is the distribution of user on the number of distinct tagsdepicted in fig. 4. Generally, 66.7% of users have used 10 or less distinct tags. Additionally, 28.1% of users have utilized 11-100 tags and only 5.2% of users have exploited more than 100 distinct tags.

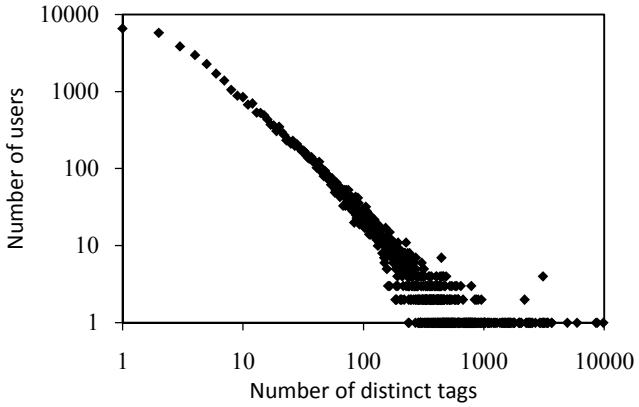

Figure 4. Distribution of user on the number of separate tags.

Fig. 5 shows the second dimension,which is the distribution of users on the number of distinct articles. Overall, 69.9% of users have tagged 10 or less number of articles. Moreover, 22.5% of users have tagged between 11 and 100 articles and merely 7.6% of users have tagged more than 100 articles.

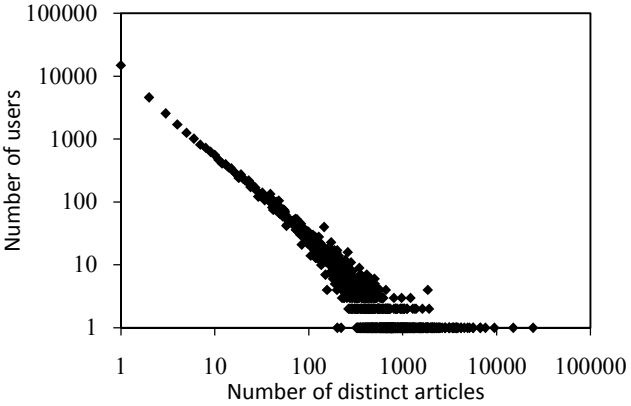

Figure 5. Distribution of user on the number of Distinct Articles.

Fig. 6 illustrates the last dimension, which is the distribution of articles on the number of distinct tags. Generally, there are tagged 96.4% of articles with 10 or less distinct tags. 3.5% of articles have been assigned to 11 -100 distinct tags and only 0.1% of articles are tagged with more than 100 distinct tags.

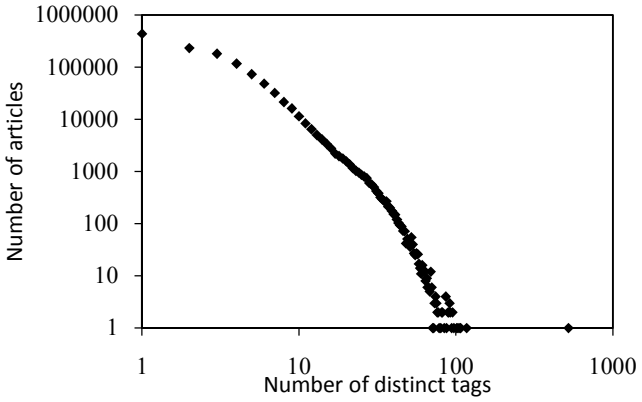

Figure 6. Distribution of Articles on the number of Distinct tags.

Three examined distributions demonstrate that the most of users are cold users and the most items are cold items. Furthermore distribution of ratings [20] is derived according to (15).

$$Sparsity = \left(1 - \frac{|Ratings|}{|Users| \times |Items|}\right) \times 100 \qquad (15)$$

The distribution of rating for CiteUlike database equals to 99.99%. This number shows that how much disperse is selected database and users have just rated a small part of items.

A. Settings

Considering performed experiments the best values for parameters of this problems are N=12, M_{imp}=14 andφ=0.05.

To evaluate proposed system we apply the P-Core parameter with value 5 to database. Then, we consider the first 90% of database as training data and the rest of it as test data. As we have focused on solving the problem of cold user, we select the users who have annotated less than 10 articles. We callthese users as cold users. During experiments the length of recommendations lists wereλ={10,20,50,100,500,1000} and then evaluation measures are calculated.

B. Evaluation measures

To evaluate the quality of recommendations two major measures are utilized; recall and precision [21].Usually other related papers use these two measures as evaluation measures to assess effectiveness of information retrieval. If H includes sources annotated by the user and Sincludes sources recommended by the system, we can calculateRecall, Precision and F1 as follows.

1- Precision: the ratioof output related concepts.

$$Precision = \frac{|H \cap S|}{|S|} \qquad (16)$$

2- Recall: the ratio of output related conceptsexisting in the system.

$$Recall = \frac{|H \cap S|}{|H|} \quad (17)$$

3- F1: to have a reasonable combination of two above mentioned measures another measure is generated known as F1 which is also widely used [22]. It is a hybrid measure assigning similar weight to recall and precision.

$$F1 = \frac{2*Recall*Precision}{Recall+Precision} \quad (18)$$

C. *Compared algorithms*

The proposed system is compared to three algorithms including Social Ranking (SR), Cluster Social Ranking (CSR) and Folks Ranking (Folksonomy) (FR) which are investigated in [15]. In Social Ranking algorithm (SR), tag extension is performed by purifying information of whole society and according to potential suggestions of each user. In a large data set, it aims to find the content related to the user's query. We assume that arbitrary number of tags and users explicates such content. SR answers users' queries using traditional recommender system techniques. Social Ranking has shown the highest precision in sparse data sets. However, in large Folksonomies, its computational overload makes some difficulties. Folks Ranking algorithm, uses data as a triangular weighted graph where nodes represent users, tags and sources. FR uses a random curling strategy to recommend sources. FR can prove high precision in case of dense web data.

V. ANALYSIS AND EVALUATION

Fig. 7shows the results of comparison performed between proposed system and a mentioned algorithm for precision measure. In this figure, horizontal axis shows the number of presented recommendations and vertical axis is the average of achieved precision.

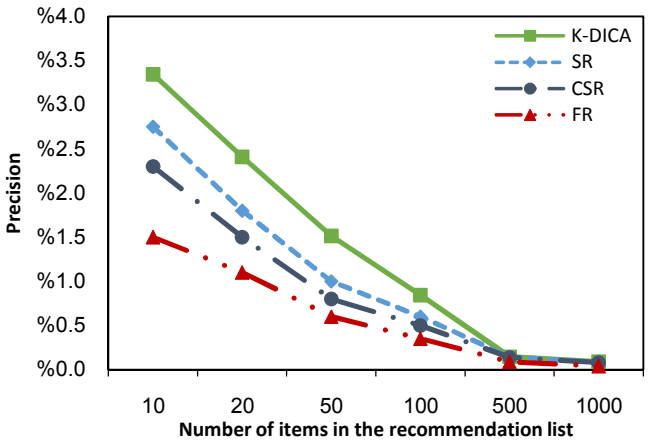

Figure 7. Precision for Cold Users.

As can be seen in this figure the precision of proposed system is considerably better than other algorithms. It shouldbe noticed that according to investigations, each cold user averagely tags 4.83 articles meaning that the best system by providing 10 recommendations can only achieve a precision of 48.3%. On the other hand, merely 32.17% of test data articles existed in training data set. In this condition, the best system, which recommends 10 sources to the user, may achieve 15.54% precision. These are due to 99.99% sparseness of the database.

Fig.8 depicts the results of comparison between proposed system and mentioned algorithms for recall measure. In this figure, horizontal axis is the number of recommendations and the vertical one is the recall. As can be seen, the recall of proposed system is superior to other algorithms when the number of recommendations is between 10 and 50; however, when the number of recommendations is between 100-1000, no improvement is obtained comparing with SR and CSR algorithms. SR and CSR algorithms use similarity between users to modify recall, which may justify lack of improvement in large number of recommendations.

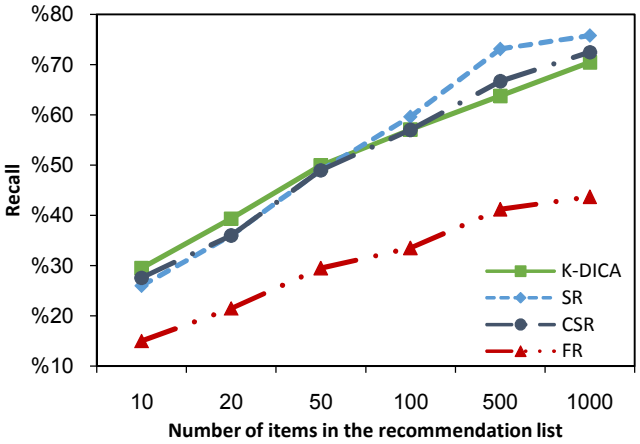

Figure 8. Recall for Cold Users.

The results of comparison between proposed system and mentioned algorithms for F1 measure are depicted in fig. 9. In this figure, horizontal axis is the number of recommendations and the vertical one is achieved F1.

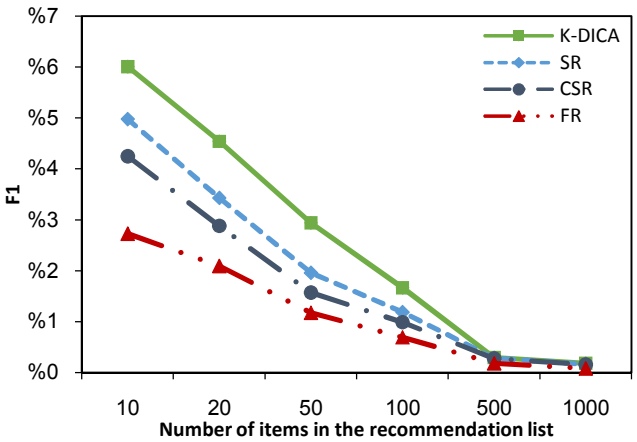

Figure 9. F1 for Cold Users

The reasons led to improvement of evaluation criteria may be summarized as follows:

- Using discrete imperialist competitive algorithm and applyingK-medoids for more precise clustering of tags.
- Using WordNet ontology andapplying Co-Occurrence relation between tags to determine dissimilarity between tags.
- Using whole tags assigned by users to sources for clustering.
- Classifying of neighbors based on their status in the model built at the end of offline phase.
- Weighting neighbors based on semantic dissimilarity.
- Data cleaning and preparation and eliminating prepositions

VI. CONCLUSION

In this paper, we proposea modified recommender system so that it could solve cold user problem. According to this, we propose a method based on imperialist competitive algorithm where the system operates in two phases: online and offline. First, the system extracts tags from data set and clusters them by using the modified imperialist competitive algorithm with K-medoids where WordNet ontology and Co-Occurrence relations are used. When a new user logs in and enters desired tags, a cluster absorbs these tags. Then, k neighbors of each new tag and elements tagged by these neighbors are extracted and weighted. Finally, recommender systems suggest a few sources with the least weights.

REFERENCES

[1] A. Merve Acilar and A. Arslan, "A collaborative filtering method based on artificial immune network," *Expert Systems with Applications,* vol. 36, pp. 8324-8332, 2009.

[2] J. W. Kim, K. M. Lee, M. J. Shaw, H.-L. Chang, M. Nelson, and R. F. Easley, "A preference scoring technique for personalized advertisements on Internet storefronts," *Mathematical and Computer Modelling,* vol. 44, pp. 3-15, 2006.

[3] T.-P. Liang, Y.-F. Yang, D.-N. Chen, and Y.-C. Ku, "A semantic-expansion approach to personalized knowledge recommendation," *Decision Support Systems,* vol. 45, pp. 401-412, 2008.

[4] P.-M. Chen and F.-C. Kuo, "An information retrieval system based on a user profile," *Journal of Systems and Software,* vol. 54, pp. 3-8, 2000.

[5] J. B. Schafer, J. Konstan, andJ. Riedl, "E-Commerce Recommendation Applications," in *Applications of Data Mining to Electronic Commerce*, R. Kohavi and F. Provost, Eds., ed: Springer US, 2001, pp. 115-153.

[6] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining,* vol. 3, pp. 1113-1133, 2013.

[7] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, "Getting to know you: learning new user preferences in recommender systems," presented at the Proceedings ofthe 7th international conference on Intelligent user interfaces, San Francisco, California, USA, 2002.

[8] M. Montaner, B. López, and J. de la Rosa, "A Taxonomy of Recommender Agents on the Internet," *Artificial Intelligence Review,* vol. 19, pp. 285-330, 2003.

[9] B. Jeong, J. Lee, and H. Cho, "An iterative semi-explicit rating method for building collaborative recommender systems," *Expert Systems with Applications,* vol. 36, pp. 6181-6186, 2009.

[10] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," presented at the Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland, 2002.

[11] X. N. Lam, T. Vu, T. D. Le, andA. D. Duong, "Addressing cold-start problem in recommendation systems," presented at the Proceedings of the 2nd international conference on Ubiquitous information management and communication, Suwon, Korea, 2008.

[12] J. Bobadilla, F. Ortega, A. Hernando ,and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems,* vol. 46, pp. 109-132, 2013.

[13] Y. Ling, D. Guo, F. Cai, and H. Chen, "User-based Clustering with Top-N Recommendation on Cold-Start Problem," in *Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference on*, 2013, pp. 1585-1589.

[14] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications,* vol. 41, pp. 2065-2073, 2014.

[15] V. Zanardi and L. Capra, "A Scalable Tag-Based Recommender System for New Users of the Social Web," in *Database and Expert Systems Applications*. vol. 6860, A. Hameurlain, S. Liddle, K.-D. Schewe, and X. Zhou, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 542-557.

[16] F. Gao, C. Xing, X. Du, and S. Wang, "Personalized service system based on hybrid filtering for digital library," *Tsinghua Science and Technology,* vol. 12, pp. 1-8, 2007.

[17] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme, "Tag Recommendations in Folksonomies," in *Knowledge Discovery in Databases: PKDD 2007*. vol. 4702, J. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenič, and A. Skowron, Eds., ed: Springer Berlin Heidelberg, 2007, pp. 506-514.

[18] D. Sánchez, M. Batet, and D. Isern, "Ontology-based information content computation," *Knowledge-Based Systems,* vol. 24, pp. 297-303, 2011.

[19] C. Wartena, R. Brussee, and M. Wibbels, "Using Tag Co-occurrence for Recommendation," in *IntelligentSystems Design and Applications, 2009. ISDA '09. Ninth International Conference on*, 2009, pp. 273-278.

[20] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowledge-Based Systems* ,vol. 57, pp. 57-68, 2014.

[21] Y.-Y. Shih and D.-R. Liu, "Product recommendation approaches: Collaborative filtering via customer lifetime value and customer demands," *Expert Systems with Applications,* vol. 35, pp. 350-360, 2008.

[22] S. K. Lee, Y. H. Cho ,and S. H. Kim, "Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations," *Information Sciences,* vol. 180, pp. 2142-2155, 2010.