REACTIVE MULTI-AGENT MODEL FOR COLLABORATIVE FILTERING - BASED RECOMMENDER SYSTEMS

Tran Thi Ngoc Trang[1], Le Viet Man[2], Nguyen Minh Duc[2]

[1]*Hue University; mantrang27@gmail.com*

[2]*College of Economics, Hue University; lvman@hce.edu.vn, ducnm@hce.edu.vn*

Abstract - In recent 20 years, using multi-agent models has been developed in many research fields, especially in social science. These multi-agent models allow simulating and studying a complex part of real world by performing insilico test, or called real simulation. Recently, some research has also proposed multi-agent model for Information Retrieval problems and has achieved some remarkable results. In this paper, we introduce a reactive multi-agent model as a new approach for recommender systems in order to overcome some common limitations of recommender systems, especially recomputation problems when new data is added to the system. Experimental results also indicate that the proposed model can be applied for recommendation problems and our model performs more stably than collaborative filtering based recommender systems.

Key words - Collaborative filtering; recommender systems; multi-agent systems; reactive multi-agent model; reactive agent; attractive force; repulsive force.

1. Introduction

In daily life, people usually rely on recommendations from other people by spoken words, reference letters, news reports from news media, general surveys, travel guides, and so forth. Recommender systems (RS) assist and augment this natural social process to help people sift through available books, articles, webpages, movies, music, restaurants, jokes, grocery products, and so forth to find the most interesting and valuable information for them. The most common technique used for recommendations is collaborative filtering (CF). CF-based RS predict user preferences for products or services by learning past user-item relationships from a group of user who share the same preferences and tastes. Although owning many advantages in comparing to other techniques, CF has been facing many problems needed to be solved, such as data sparsity, scalability, similar items, grey-sheep, black-sheep, false recommendations, privacy,….

Until now, there have been many methods proposed to tackle all the problems of CF approach, such as hybrid RS [15], graph-based RS [11], especially multi-agents based RS [2, 7]. In this research, we propose a reactive multi-agent model for RS in which user-rating list and the methods for computing similarity are used based on Item-based CF technique. This solution is an new approach for RS which offers precise recommendations based on particular preferences of users with better performance than CF- based RS.

The rest of this paper is organized as follows. In section 2, we review some existing works about CF approach and multi-agent systems. Next, in section 3, we first give an overview of proposed model, reactive agents and then the method for determining attractive and repulsive forces as well as self-organized model. The results of an experimental evaluation are presented in section 4 with the use of a movie database called MovieLens 100K. The paper ends with a discussion of the limitations of the work and an outlook on possible directions for future work.

2. Related works

2.1. Collaborative filtering-based recommender systems

Most of RS basically rely on three methods: content-based, knowledge-based and CF-based where CF is the approach which has been used most widely. CF-based RS provide personalized recommendations according to user preferences. They maintain data about active users' purchasing habits or interests and use this data to identify groups of similar users. They then recommend items liked by similar users. CF systems offer two major advantages: Firstly, they do not take into account content information, and secondly, they are simpler and easier to implement. Further, ignoring content information allows CF systems to generate recommendations based on user tastes rather than the objective properties of domain items. This means that the system can recommend items very different from those that the user had previously shown a preference.for.

Mathematically, CF algorithms represent a user as an M-dimensional vector of items, where M is the number of distinct catalog items. By computing the similarity of users, a set of "*nearest neighbours*" whose known preferences correlates significantly with a given user are found. Preferences for unseen items are predicted for an active user based on a combination of the preferences known from the nearest neighbours. Filtering these neighbours is equivalent to computing the distance among M-dimensional vectors. Accordingly, CF algorithms are categorized as *memory-based filtering* and *model-based filtering*. Memory-based filtering computes distance between vectors by using Euclide distance, Pearson correlation, … whereas model-based filtering is considered as an approach to solve some limitations of memory-based filtering, especially scalability problem. In particular, machine learning techniques (such as PCA – Principal Component Analysis [10], MDS – Multi-dimensional scaling [14] or SOM – SelfOrganizing Maps [9]) are used in model-based filtering in order to map M-dimensional vectors into 2 or 3-dimensional space in order help the process of computing distance, clustering or classification to be easier. Despite getting some effective results, these techniques still have some disadvantages, such as data sparsity, data change, computing complexity, decline of recommendation quality,…

Recently, a new approach about user preference data in RS has been proposed by representing user preference matrix in form of graph and using graph theory to solve some problems about computing the similarity between users [11]. Also, with graph-based approach, O'Donovan [13] draws a graph of user preferences in 2-dimensional (2D) space and recommendation is operated by computing the distance between user nodes in the space. In spite of reducing computation complexity, this system still uses memory-based and model-based techniques, thus it also faces common problems of CF algorithms. However, the idea about drawing a graph and computing similarity between users/items in 2D space in [8, 11] will be aslo applied for computing the similarity between items in our system.

2.2. Multi-agent systems

Muti-agent systems (MAS) refer to a computer research domain that addresses systems which are composed of micro level entities (agents), which have an autonomous and proactive behaviour and interact through an environment (either virtual environment or real environment), thus producing the overall system behaviour which is observed at the macro level [6]. Until now, MAS have been considered as an interesting and convenient way of understanding, modeling, designing and implementing different kinds of (distributed) systems [5]. Futhermore, MAS also represent a very interesting modeling alternative, compared to equation based modeling, for representing and simulating real-world or virtual systems which could be decomposed in interacting individuals [4].

There are many types of agents used in MAS, such as assistant agents, collaboration agents, mobile agents and reactive agents where reactive agents have widely used in many fields, especially information retrieval. Two typical systems which use reactive multi-agents are presented in [3, 12]. Particularly, in [12], Renault used dynamic attractive and repulsive multi-agent model which aims to organize emails in a 2D space according to similarity where each email is represented by an agent and there is no need to specify axes as well as how to organize information. The model allows agents to communicate with each other through virtual pheromones and collectively auto-organize themselves in a 2D space. Without much constraints, the system can organise (like clustering/classification) information and let the user intuitively interact with it.

Based on the idea of Renault, Cao Hong Hue et al. [3] presented a new model for image browsing and retrieval which uses a reactive multi-agent system supporting visualisation and user interaction. Each agent represents an image. These agents move freely in the space which their routes are not predefined. They just react to external stimuli sent by other agents. Each agent interact to others through forces, either attractive forces or repulsive forces. These forces are generated by the visual and textual similarities between an agent and its neighbours. Thus, the agents are attracted by similar agents and repulsed by dissimilar agents. This model is operated according to loop steps by the time. In each loop step, agents change their position in the model. Forces between agents or neighbours cause these changes. Selecting neighbours in each time step makes this model operate really slowly. That is the main limitation of this model.

The multi-agent systems proposed in [3, 12] are equivalent to the core idea of RS which use the similarity among agents to organize data. Also, RS use similarity between users or items to extract a list of recommendation items. However, in CF-based RS, selecting recommendation lists usually uses complex computing formulas whereas, using attractive and repulsive forces between agents will help computing process become easier by just finding neighbours (in 2D space) of each agent. This is the main idea used for our proposed model.

3. Reactive multi-agent model for CF method

Giải thích: Trong phần này, mô hình đa tác tử phản ứng với môi trường đã được thể hiện khá rõ qua các phần nhỏ mà chúng tôi đã nêu ở bên dưới. Việc trình bày phần toán học của mô hình chủ yếu xoay quanh việc tính độ lớn của lực và hợp lực tác động lên một agent. Theo đó, việc tính toán độ lớn của lực đã được nói rõ trong phần 3.2. Còn đối với hợp lực tác động lên một agent, để rõ hơn, chúng tôi đã có bổ sung một phần ghi chúvề việc tính tổng hợp lực tác động lên một agent dựa trên các lực tác động lên một agent và các láng giềng của nó (Figure 3).

3.1. Model overview

The proposed model uses reactive agents in which each agent represents an item[1] and actions of each agent depend on list of users' ratings for that item. The agents move freely in a 2D environment which has no pre-defined axes or meaning (*Figure 1*). They are reactive and only react to outside stimuli sent by other agents. Each agent interacts with others through forces (either attractive forces or repulsive forces). Forces originating between agents are computed based on the similarity. Two agents attract each other when their similarity is high and repulse each other when their similarity is low. According to the sum of attractive and repulsive forces acting upon an agent, these agents will move to the new position in the space. Ihere, agents interact to new agents and then continue moving. The movement of agents will be ended when they reach to stable status. This helps to create a self-organized model in 2D space. At steady status, two closed-agents are similar to each other and they can be used for recommendation process.

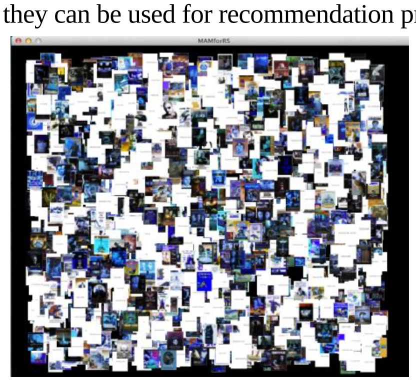

Figure 1. *The environment of agents. Each agent is represented by an image which corresponds to a poster of a movie*

[1]Item mentioned here is an item in RS

As presented above, at each time step, an agent interacts with its neighbours, gets forces from them and moving reactively. Hence, computing forces only can be done when we get list of neighbours for an agent. In our model, neighbours can be chosen according to four methods including *proximity*, *sampling*, *random* and *defined area* (*Figure 2*).

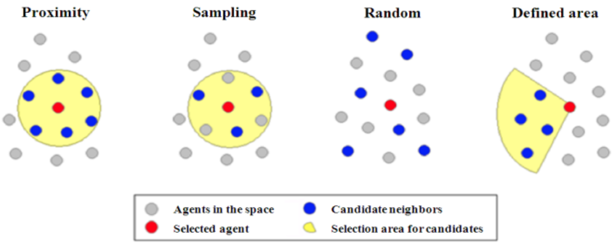

Figure 2. *Methods for choosing neighbors. Proximity: Choose neighbors in fixed-radius; Sampling: Choose randomly some neighbors in the list of closed-area; Random: Choose randomly all agents in the model; Defined area: Choose agents from a specific area*

From experimental process, in local level, we choose neighbours by using proximity approach which allows selecting agents in fixed radius. And in global level, we select agents according to random approach which randomly pick agents from all agents in the model.

Once the neighbour list for an agent is known, then this agent can simply compute the forces received from all these neighbours and react according to them (*Figure 3*).

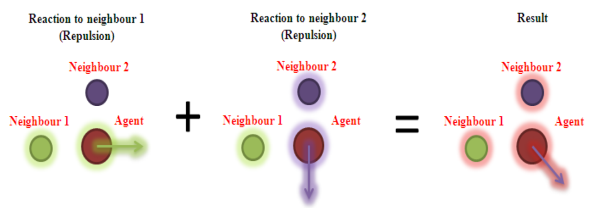

Figure 3. *An example of reaction of an agent toward two neighbours. This image shows the rule for summing forces, each agent interacts toits neighbours. The force generated from these interactions will be combined for making the final global force. This final global force for an agent is simply the vectorial summation of all forces between that agent and its neighbous*

3.2. Attractive and repulsive forces (item-based forces)

A force applied between two agents can be attractive or repulsive and is characterised by a vector with *direction* and *magnitude*. However, firstly, we need to determine the similarity between agents. In item-based CF method, the similarity between agents is usually computed by using Pearson correlation. Implementation results obviously show that this method is widely used in the CF research community and gives better results than other methods [13]. The similarity between items is computed according to the following formula:

$$w_{i,j} = \frac{\sum_{a\in U}(r_{a,i} - \overline{r_i})(r_{a,j} - \overline{r_j})}{\sqrt{\sum_{a\in U}(r_{a,i} - \overline{r_i})^2 \sum_{a\in U}(r_{a,j} - \overline{r_j})^2}}$$

where $w_{i,j}$ is the similarity between item *i* and item *j*, *U* is the set of users rating for both item *i* and item *j*, $r_{a,i}$ is rating value of user *a* for item *i* and $\overline{r_i}$ is the average rating value of all users for item *i*, $r_{a,j}$ is rating value of user *a* for item *j* and $\overline{r_j}$ is the average rating value of all users for item *j*.

Force direction is characterized by the type of forces (either attractive forces or repulsive forces). These forces show that the behavior of an agent is toward or away from other agents. In local level, agents' behavior is determined by the similarity or dissimilarity among agents.

Accordingly, force direction is determined as follows:

- If two agents are similar then they will attract each other. It means that they tend to be closer.
- If two agents are not similar then they will repulse each other. It means that they tend to be separated.

Force magnitude belongs to the similarity and the distance among them is combined to form the force characteristic. However, in practice, it is difficult to define exact value of the similarity and distance between forces. Thus, we determine force magnitude according to continuous approach as showed below (*Figure 4*).

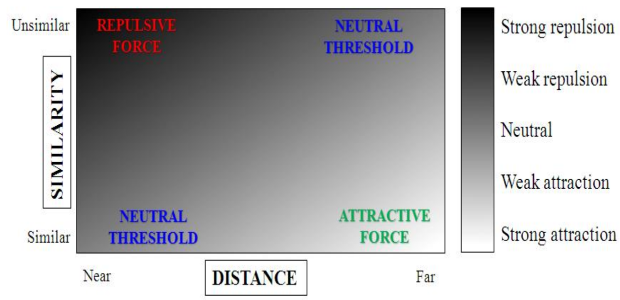

Figure 4. *Force characteristic and magnitude basing on similarity and distance (continuous approach)*

As clearly seen from the Figure 4, there is always a neutral threshold of forces. This threshold is the basic to determine force types:

If results are higher than neutral threshold, we have repulsive forces which are computed by:

$$f = \frac{w - w_{min}}{(w_{max} - \overline{w})} \times d$$

If results are lower than neutral threshold, we have attractive forces which are computed by:

$$f = \frac{w - \overline{w}}{(w_{max} - \overline{w})} \times d$$

where w is the similarity between two agents; $\overline{w}$, w_{max}, w_{min} are respectively mean value, maximum value and minimum value for active agent's neighbors; *d* is the distance between two agents computed by Manhattan [22].

3.3. Self-organized model

During the evolution of the model, agents gradually move to a status position with indefinite route. Thus, our model is similar to self-organized model in machine learning. However, also unlike the model proposed by Cao Hong Hue et al. [5], our model uses two levels: local level and global level.

Local level: Agents choose their neighbors according to proximity approach which divides the space into separate areas for operating independently. Local force generated from agents helps to create clusters which are disposed sparsely in the space. However, local level does not offer the high accuracy for the model. So, a global level is needed to break down the local connections and collect small groups together in order to enhance the accuracy of the model.

Global level: Agents choose their neighbors with random positions in a large area. Force originated in this level are called *global force* which is combined to a local force to form an associated force (according to force association show in *Figure 3.* above).

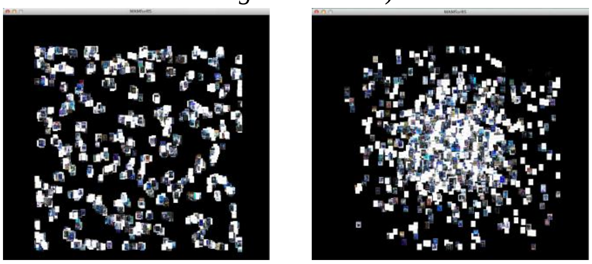

Figure 5. *Simulation on local level (a) and global level (b)*

4. Experimental results

4.1. System implementation

Giải thích: Theo yêu cầu của phản biện, ở phần này chúng tôi bổ sung thêm một kết quả của quá trình cài đặt thực nghiệm trên hệ thống tư vấn film nhằm mô tả trực quan kết quả của quá trình tư vấn (Figure 7). Hình này mô tả danh sách các bộ phim mà một người dùng nào đó có thể thích, kèm theo đó là giá trị dự đoán cho từng bộ phim đó.

System is built by using Objective C and Open Graphics Library (OpenGL). To evaluate the performance of the system, we use dataset MovieLens 100K including 100000 ratings (with scale from 1 to 5) from 943 users for 1682 movies. Each user rated at least 20 movies and supplied demographics information (age, gender, occupation,…).

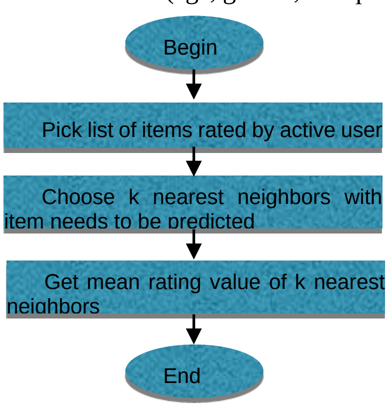

Figure 6. *Prediction algorithms with input is the item which needs to be predicted*

After the operation in 300 time steps, we recognize that forces acting upon agents are gradually decreasing to 0, agents do not move any more, the distances among agents do not change as well. At that time, the model reaches stable status. Because the proposed model is a self-organized model of agents in the space, the similarity of agents is shown exactly in this model. Hence, the result of prediction will be the rating values for nearest movies to the one needed to be recommended in the space. The prediction algorithms is illustrated in Figure 6.

After having the prediction for items which maybe liked by active user, the system collects all the films which are unseen by active userwith highest predicted ratings. The list of recommended films is described in *Figure 7*.

Figure 7. *List of recommended films with predicted ratings (according to measurement scale from 1 to 5)*

4.2. System evaluation

After offering prediction value for active user, we compute prediction accuracy (MAE) for five testing data sets. This result is illustrated in Figure 8. below:

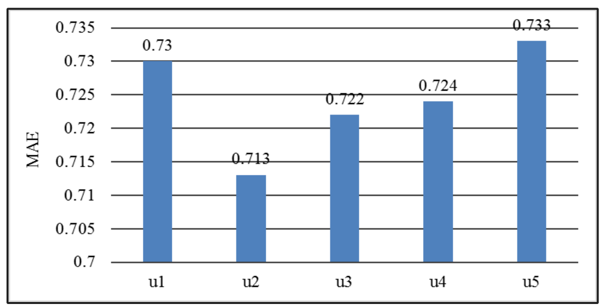

Figure 8. *MAE for five testing data sets*

The Figure 8. shows that attractive and repulsive multi-agent model give accurate prediction with the average of MAE of 0.724. Meanwhile, this value for item-based CF proposed by Badrul Sarwar et al. [2] is 0.723. It can be seen obviously that recommendation results offered by our proposed model and previous CF methods are equivalent.

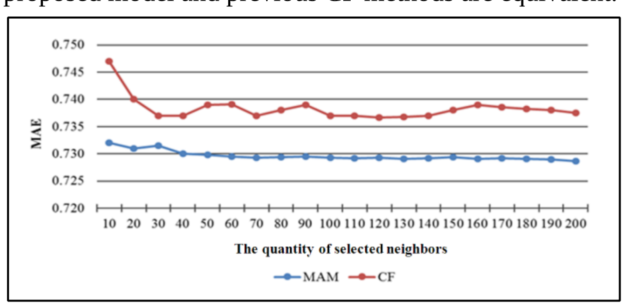

Figure 9. *MAE values for the proposed model and traditional CF technique*

Otherwise, the quantity of selected neighbors significantly influences on the MAE value. Experimental result (Figure 2) denotes that if the number of selected neighbors is under 50 then MAE value is quite high, if the quantity of selected neighbors is over 50 then MAE value is quite stable and decreases regularly. This proves that our model works more stably than traditional CF techniques.

What is more, our proposed model also overcomes the common limitations of traditional CF methods related to computation time and scalability when new item or new user is added to the system. Indeed, adding new objects to our system means that adding agents to the model, then computation is processed and agents will move in the space until they find the exact position. That is the main advantage of our model.

5. Conclusion and future works

The paper proposed a reactive multi-agent model for item-based RS. With MovieLens 100K dataset, recommendation movies are the acquired result based on the analysis rating values of hundreds of former users. Experimental results also indicate that attractive and repulsive multi-agent model can be used as an alternative approach for CF techniques with more stable performance. Moreover, the model solves problem of recomputing when a new item is added to the system. This research is the basis for future works of reactive multi-agent model for RS with many improvements in the performance, the ability of visualization and interaction so as to enhance persuasiveness, transparency and satisfaction for explanations in RS. Furthermore, by combining item agents and user agents in the environment, supplementing knowledge/content into agents will help to give more intelligent and exact recommendation results.

REFERENCES

[1] Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl, *Item-Based Collaborative Filtering Recommendation Algorithms*, Proceedings of the 10th international conference on World Wide Web, (2001), pp. 285-295.

[2] Bsarry Smyth, Brynjar Gretarsson, Svetlin Bost, Tobias Höllerer, *Peerchooser: visual interactive recommendation*, Proceedings of the 2008 Conference on Human Factors in Computing Systems, Florence, Italy, (2008), pp. 1085-1088.

[3] Cao Hong Hue, G., A., *A multi-agent model for Image browsing and retrieval*, Studies in Computational Intelligence Volume 457, (2013), pp. 117-126.

[4] F. Klugl, C.Oechslein, F.Puppe, andA.Dornhaus, *Multi-agent modelling in comparison to standard modelling*. In F.J. Barros and N. Giambasi, editors, Proceedings of AIS 2002: Artificial Intelligence, Simulation and Planning in High Autonomous Systems, San Diego, CA, (2002), pp. 105–110.

[5] F. Zambonelli and H.V.D. Parunak, *From design to intention: signs of a revolution,* inProceedings of the first international joint conference on Autonomous agents and multiagent systems, ACM Press, (2002), pp. 455–456.

[6] Fabien Michel, Jacques Ferber, and Alexis Drogoul, Multi-Agent Systems and Simulation: a Survey From the Agents Community's Perspective, CRC Press, LLC, (2001).

[7] Guerra-Hernández, Alejandro, Amal El Fallah-Seghrouchni and Henry Soldano, *Learning in BDI Multi-agent Systems*, Computational Logic in Multi-Agent Systems, (2004).

[8] J. O'Donovan, B. Smyth, B. Gretarsson, S. Bostandjiev, and T. Höllerer., *PeerChooser: visual interactive recommendation*, presented at the CHI '08: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, New York, USA, (2008), pp. 1085–1088.

[9] Laaksonen, J., Koskela, M., Oja, E., *PicSOM – Self-organizing image retrieval with MPEG-7 content descriptors*, IEEE Transactions on Neural Networks 13(4), (2002), pp. 841-853.

[10] Moghaddam, B., Tian, Q., Lesh, N., Shen C., Huang, T.S., *Visualization & User-Modeling for Browsing Personal Photo Libraries*, International Journal of Computer Vision 56(1/2), (2004), pp. 109–130.

[11] Nguyen Duy Phuong, Le Quang Thang and Tu Minh Phuong, *A Graph-Based Method for Combining Collaborative and Content-Based Filtering*, Trends in Artificial Intelligence, Springer, (2008), pp. 859-869.

[12] Renault, V., Organisation de Sociétés d'Agents pour la Visualisation d'Informations Dynamiques, PhD thesis, University Paris 6 (France), (2001).

[13] Ricci F., Rokach L., Shapira B., Kantor P.B, *Recommender Systems Handbook,* 1st Edition, Springer, 845 p. 20 illus., Hardcover, ISBN: 978-0-387-85819-7, (2011).

[14] Rubner, Y., Guibas, L.J., Tomasi, C., *The earth movers distance, multi-dimensional scaling, and color-based image retrieval*, In APRA Image Understanding Workshop, (1997), pp. 661-668.

[15] Xiaoyuan Su and Taghi M. Khoshgoftaar, *A survey of collaborative filtering techniques,* Advances in Artificial Intelligence, (2009), pp. 1–20.

(The Board of Editors received the paper on 04/17/2015, its review was completed on 06/22/2015)