

ANTSREC: A Semantic Recommender System Based on Ant Colony Meta-Heuristic in Electronic Commerce

Mojtaba Salehi¹, Abdolhossein Fathi² and Fardin Abdali-Mohammadi³

¹*Department of Computer Engineering, Islamic Azad University of Khorramabad, Khorramabad, Iran*

^{2,3}*Department of Computer Engineering, Razi University, Kermanshah, Iran*

¹*salehi_mojtaba@ymail.com, ²a.fathi@razi.ac.ir, ³fardin.abdali@razi.ac.ir,*

Abstract

A recommender system is a guide and assistance for choosing the required product or service for improving the electronic commerce systems. Most of the recommender systems use the history of customer purchase and a few are based on Semantic relatedness of purchased commodities. In this paper a semantic recommender system based on Ant Colony and Ontology dependencies is used for improvement of electronic commerce. This system comprises heuristic, stochastic, reinforcement learning in Ant Colony theory and semantic dependency in ontology characteristics. The presented system is able to recommend similar, complement and bundled products. This characteristic can overcome problems such as cold start, scalability and scarcity of information. In this paper applied tests results show the performance and efficiency of presented algorithms.

Keywords: *Ant Colony, Electronic Commerce, Ontology, Recommender System, Semantic Relatedness*

1. Introduction

The main objective of recommender systems is guidance and assistance in choosing the required product, services and information. Recommender systems improve the electronic commerce efficiency. Using these systems electronic commerce executive companies can improve customer satisfaction and purchase procedure and in the meantime stipulates the customer to purchase (some products are invisible to customer at the first glance). In this way this system increases selling of products.

Most of the recommender systems use history and customers' preferences and a few of them apply a semantic relatedness in purchased products. A suitable recommender system should recommend the complementary products as well as similar and popular ones. In case of pencil purchase for instance, this system can recommend complementary products such as an eraser and notebook.

Ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary. In defining ontology manual intervention of experts is required and therefore high accuracy in the developing of these relations is needed.

Depending on customer interests and variations of products, recommender systems should be dynamic in order to adapt with the environment. In addition to extracting knowledge from working environment these systems should be able to use heuristic approaches and test new solutions. Ant Colony algorithms is a heuristic algorithm which is based on the ants efforts seeking food in nature. These algorithms are based on stochastic procedures and reinforcement learning and are extremely satisfying in dynamic environments.

In this paper a new recommender system based on Ant Colony theory and Semantic relatedness in ontology is developed for improving the electronic commerce procedures. Considering heuristic, stochastic and potential learning of Ant Colony algorithm, this structure is used in recommending of a new system. Also since systems should recommend a valid, useful and related option, the concept of ontology is also embedded in this system. The presented system can recommend complementary products, bundled products and products which are generally categorized in one group. In this paper we use semantic relatedness of products for selection of initial recommendation. In our method, we construct implicit rating from the implicit feedback of purchasing behaviours.

The structure of this paper is presented in the following. Section 2 is the research background. Section 3 presents the concept of Semantic relatedness of two terms (products) and determination method using ontology. Section 4 represent essential concepts of Ant Colony algorithm. In Section 5 the new semantic recommender algorithm is presented. Section 6 presents test data and evaluation criteria of recommender algorithm. Conclusion and remarks is presented in Section 7.

2. Research Background

The main elements in recommender systems are customer and product. The main differences between different recommending systems stem from product and modelling customer behaviour approach and analysis methods. There are three main approaches to recommendation: Content based filtering [1, 2], Collaborative filtering [3, 4], and Economic factor-based [5]. However hybrid methods are also usual.

Content based systems use product content and semantic similarity between them. In these systems the recommendations are analogues to previous orders of a customer itself. However collaborative filtering systems are based on the analysis of customer's behaviour and the recommendations are based on customers' profile similarities and purchase background [3]. Economic factor-based systems consider product cost as the first priority. Collaborative filtering systems are divided into two "product-based" and "customer-based recommender systems".

In the recent years, application of semantic web technologies in recommender systems and mobile electronic commerce [25] has had a progressive rate. In semantic web recommender systems, ontology is a tool for overcoming the problem of non-homogenous resources, efficiency and intellectual. Ontology and application of semantic information are main tools in diagnosing recommender systems in electrical commerce.

In 2006 Khosravi and Nematbakhsh [6] developed a personalized recommender system for electronic commerce. Wang and Kong in 2007 recommended a system for personalized recommendations based on ontology and categorized knowledge of products [7]. Moosavi *et al.*, [8] presented a semantic approach for complementary products based on ontology and product catalogue. They assumed a NEEDS relation in product list which was indicative of complementary products. Defining the complementary dependency of a product to the other one was based on tracking the IS-A and NEEDS relations between products.

3. Semantic Relatedness

The first philosophical definition of ontology was: "Systematic description of an existence". During recent years several definitions for ontology are presented. On the first definitions of ontology is presented by Neches *et al.*, [9]: "A set of logical axioms designed to account for the intended meaning of a vocabulary". This definition shows the required parts of an ontology such as basic terms, relations between terms and required principles. This also

defines the terms and relations in ontology. Neches believes that ontology a set of logical axioms designed to account for the intended meaning of a vocabulary. Several years later Gruber [10] defines ontology as an explicit specification of a conceptualization. Some years later Gruber modifies this definition: "Ontology is an accurate specification of a conceptualization". In definition the term "explicit" shows that applied concepts and their constraints should be defined explicitly. Term "accurate" shows that ontology should be understandable for computer. Also the term "conceptualization" shows that ontology should be validated by a group not a specific person. Guarino [11] defines ontology as "series of logical rules for defining vocabulary or terms". As it is evident from previous definitions, ontology shows the conceptual relations and semantic dependencies between all products in range. This principle is used for semantic relatedness and using ontology for calculation of this value. The concept of semantic relatedness is very deeper than semantic similarity. It is commonly assumed that similar objects are semantically related. However it should be noted that non-similar objects can also be semantically related (*i.e.*, Contrast concepts such as cool and warm and complementary concepts such as tire and machine).

Applying ontology for defining Semantic relatedness is advantageous [10]. On the first benefits of this approach is the accuracy, since they are defined manually by experts. Also using ontology and related algorithms optimizes the process time. Several approaches are common for calculating of semantic similarities using ontology. These approaches are categorized in three: (1) Path based methods, (2) Content based methods and (3) Feature vector based method [10]. The first category is used for determination of Common generalized concept which is more compatible with the objectives of this paper. Second and third categories are used for text processing. In this paper the lower the distance between two concepts in ontology the higher is the Semantic relatedness.

The path based methods use the shortest semantic distance between two concepts or terms in ontology hierarchy. Wu and Palmer [12] have developed a criterion for measuring the semantic distance of common more explicit concept of two concepts (common ancestor in ontology hierarchy):

$$S_{W\&P}(C_1, C_2) = \frac{2H}{N_1 + N_2 + 2H} \quad (1)$$

In Equation 1, N_1 and N_2 are number of IS-A links of C_1 and C_2 concepts to more common general concept C in ontology hierarchy respectively. H is the number of IS-A links from C concept to ontology root. The determined value from this formula shows the analogy between two concepts and is a value between 0 to 1. The more this value is closer to 1 the more two concepts are analogous.

Li and others [13] use a nonlinear function for calculation of analogy between two concepts. Equation 2 shows this function.

$$S_{Li}(C_1, C_2) = e^{-\alpha L} \frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}} \quad (2)$$

In this equation L shows the shortest possible path between two concepts in ontology, α and β are control parameters of concepts relative to the depth of concepts in ontology. The calculated value from this equation ranges from 0 to 1 and indicates the analogy of two concepts. The typical value of α is about 0.2 and typical value of β is about 0.6 [13].

Leacock and Chodorow in [14] present another approach for calculation of analogy between two concepts. This approach is based on shortest possible path between two concepts (*i.e.*, $d(C1, C2)$) and normalizing them with doubling the depths of these two concepts in

ontology hierarchy (D) and finally calculation of logarithm for final result. This approach is formulized in the following equation.

$$S_{L\&C}(C_1, C_2) = -\log(d(C_1, C_2) / 2D) \quad (3)$$

For calculation purposes one unit is usually added to $2D$ and $d(C_1, C_2)$. In this way in calculation of the shortest possible way ($d(C_1, C_2)=0$), $\log(0)$ is avoided.

Mao and others in [15] have developed an approach for calculation of similarity between two concepts in which the descendants of concepts in Ontology hierarchy is considered.

According to simulation results presented in [10, 24] the difference between all aforementioned methods is negligible and all methods result in similar analogy between concepts. In this paper Wu and Palmer algorithm is used for determination of semantic relatedness of two concepts. For determinations of semantic relatedness all links in ontology should be taken into account. In other words since the recommender system is able to suggest complementary and similar products, IS-A, Part_of and Needs [8] relations are considered in product catalogue. In this way calculation of shortest way includes these links as well.

4. Ant Colony Optimization

Ant colony algorithm is a group of heuristic optimization algorithms which are based on ants' food seeking effort in the nature [16, 20]. In nature on the way to home, ants dispose Pheromone on their tracks. In action enables them to find the food location easier. Disposed Pheromone evaporates gradually. Ant colony problems are usually modelled using a graph. In these graph each edge represents the path and weight of each edge represents the disposed Pheromone. In each node ants choose the next node using a probability (the probability of path choose). After that they alter the amount of Pheromone in each edge using "Pheromone updating procedure". Using this mechanism each ant tries to find the shortest way in the graph.

In Ant colony algorithms first each ant is placed on one node stochastically. This ant has a memory to record partial solutions until the current time. Starting from starting node, each ant moves from one node to another. In i th node, k th ant chooses the next node (j th node) which is not travelled by this specific ant, based on a "path selection probability function". After each ant completes its turn the Pheromone amount of that turn should be updated. In Ant colony algorithm the updating procedure first comprises of reducing the Pheromone amount by a constant rate (evaporation) and then applying an extra amount by each ant in each travelled edge. The evaporation constant ranges between 0 and 1. The updating procedure avoids infinite accumulation of Pheromone and enables the algorithm to forget the previous "bad decisions". Algorithm guides more ants to shorter paths and in this regard the amount of Pheromone in these paths increases which in turn increases the probability of choosing the shorter ways in future.

The difference between different Ant colony algorithms is in path selection and updating Pheromone amount by ants. ACS (Ant Colony System) [17, 18, 19] is one the best algorithms in Ant colony. In ACS algorithm not only ants update the Pheromone amount in each edge but also after each turn the best path travelled updates the travelled path once more. The main cause behind using ACS algorithm in this paper is transparent heuristic seeking mechanism. In other words in contrast with other algorithms in Ant colony family, in ACS algorithm ants use 2 formula for choosing the next node among which one of them uses a stochastic variable. This means that the ant which is located in node " r ", chooses the next node " s " based on the following formula:

$$s = \begin{cases} \max_{j \in N_i^k} \{ \tau_{ij}(t) \eta_{ij}^\beta \} & \text{if } q < Q_0 \\ \text{according to AS equation} & \text{otherwise} \end{cases} \quad (4)$$

In which N_i^k are the unvisited neighbours of k th ant, $\tau_{ij}(t)$ is Pheromone amount in (i,j) edge during t turn and η_{ij} is the distance between node i to node j and " q " is a stochastic variable with uniform distribution. Q_0 is a threshold value in which initializing procedure begins. While Q_0 closes 1, data exploitation would be preferred to data exploration. In contrast when Q_0 closes 0 values, the principle of choosing the next node is similar to path selection principles in AS [16]. Also in ACS algorithm in each turn the ant with the best travelled path updates Pheromone amount of its path using the following equation:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \Delta \tau_{ij}^{best}(t) \quad (5)$$

In which $\Delta \tau_{ij}^{best}(t)$ is the Pheromone amount of ant with the best travelled path in turn " t " on (i,j) edge. The ant with the best travelled path is either the best ant in the current turn or the best ant from the first t current turn.

In order to use Ant colony algorithm in different problems, a graph model should be first developed.

5. ANTSREC: A Semantic Recommender System based on ACS

In this section a semantic recommender system based on Ant colony algorithm is described. This algorithm is called: "AntSRec (Ant Colony based Semantic Recommender system)". In most of recommender systems the history and preference of customers is considered. Customers usually are interested in bundled and complementary products. Using ontology principles recommending such a specification is possible. In ontology all adjectives and related concepts of products are included and recommended algorithm uses ontology structure, Ant colony mechanism and semantic distance concept. In the following general structure and algorithms are presented.

5.1. AntSRec Algorithm

The main components of Ant colony theory are graph, nodes, edges, distance between nodes, Pheromone and selection function (decision function for selecting next node). In AntSRec algorithm available products construct a graph. Each node in this graph indicates a product and each node has a unique identity. The weight of (i,j) edge represents the similarity or relatedness between two i and j products. This weight ranges from 0 to 1. As indicated in Section 2, NEEDS relations [8] as well as IS-A and Part_of relations are used in determination of relatedness between products. Each node in this graph comprises information related to corresponding product. Sample information is product rating and satisfaction level of products. For all edges a threshold value is considered. If the value of similarity between two products is less than this value, no edge would be considered for between them. Table 1 shows Ant colony components and corresponding elements in the AntSRec algorithm.

Table 1. Components in Ant colony Algorithm and Corresponding the AntSRec Algorithm

<i>Ant colony algorithm</i>	<i>AntSRec algorithm</i>
<i>Graph</i>	<i>Product graph</i>
<i>Node</i>	<i>Product</i>
<i>Edge</i>	<i>Minimum similarity between two products</i>
<i>Edge weight</i>	<i>Semantic distance between two products</i>
<i>Pheromone amount</i>	<i>Reinforcement value</i>
<i>Updating mechanism</i>	<i>Rating mechanism</i>

When a customer enters the system, an agent is created for it automatically. Customer purchase history is loaded into this agent. The main function of this agent is recommending preferred products to the customer. When a customer selects a specific product, its agent settles on the corresponding product node in the products graph. In this way customer should travel in the graph using available customer background and recommends the best products to him. The agent applies following formula to seek in product graphs and recommends candidate products:

$$s = \begin{cases} \max_{j \in N_i^k} \{ \tau_{ij}(t) \cdot r_j(t) \cdot \eta_{ij}^\beta \} & \text{if } q < Q_0 \\ \text{according to equation no. 7} & \text{otherwise} \end{cases} \quad (6)$$

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta [r_j(t)]^\gamma}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta [r_l(t)]^\gamma} \quad (7)$$

In fact each node can be considered as an energy point. The customer agent tends to find a way to a point with higher energy level. If the similarity between purchased product and attended product is more than a specific value (more than threshold value) then the customer agent records node (or corresponding product) in the recommendation list. Then the agent continues to search in graph using equations 6 and 7. Power β in equation 6 means that similarity between two products is more important than other information in graph.

If the relatedness value between two nodes (products) is lower than the threshold value, system exits the attended node and seeks a new node with more contrast with attended node (and thus more similar to purchased product (node)). In this case the agent uses the following formula to select the next node:

$$s = \begin{cases} \min_{j \in N_i^k} \{ \tau_{ij}(t) \cdot \eta_{ij}^\beta \} & \text{if } q < Q_0 \\ \text{according to equation no. 9} & \text{otherwise} \end{cases} \quad (8)$$

$$p_{ij}(t) = 1 - \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad (9)$$

In equations 6 to 9, $\tau_{ij}(t)$ is reinforcement value of edges, η_{ij} is the similarity between i th and j th products, N_{ik} is the series of unvisited products in node i . q and Q_0 are introduced in section 3 and r_j is the rating of j th product.

In the beginning product rating is constant and non-zero. In contrast with available recommender system, the developed recommender system has no problem with cold-start. This system uses semantic similarity in the absence of preliminary customer background and

product rating. This system has no scalability constraints and adding more nodes to this system can be easily achieved. Also this action does not result in increase in recommending time. The scarcity of information does not harm this system and semantic information on edges is used for system decisions.

In order to forget bad decisions this system applies Equation 5 and after completion of purchases balances the value on edges. In the following sections other aspects of this algorithm are investigated.

5.2. Adding and Removing Products

When a product is added to the system, one node is correspondingly added to the graph. In this case the semantic distance between this node and other nodes in the graph is calculated and required links are drawn. The stimulation value for all new edges is equally unit (one). Similarly removing a product from the list causes removal of its node and related links.

5.3. Products Rating

In many of recommender systems, product rating is obtained from customer. In this way customer may not necessarily reply honestly and may answer the questions accidentally. On some special occasions direct questioning may not be possible (e.g., cases when time is very valuable for customer). Some new implicit rating mechanisms are presented to overcome these problems [21]. In this paper a new simple implicit method is presented for product rating.

The customer enters the system and directly purchases a product or uses system recommendations. The more customers seek or buy a product, the product rating value increases. Also the purchase numbers can be an indication of its favorability. In this regard product rating is defined using following equation:

$$r_i(t) = \left(\frac{Selected_i(t)}{Recommended_i(t)} + \frac{TotalBuy_i(t)}{TotalBuy_{g(i)}(t)} \right) / 2 \quad (10)$$

In which $r_i(t)$: is the rating of i th product in time t , $Selected_i(t)$: number of selecting of a recommended product and its selection up to time t , $Recommended_i(t)$: number of recommending times of product i to customers up to time t , $TotalBuy_i(t)$: number of purchases of product i up to time t and $TotalBuy_{g(i)}(t)$ is the number of purchases of bundled products by all customers up to time t . In this recommender system products are divided into 18 categories and ontology is defined.

5.4. Initial Recommendation

When a customer enters the system, an agent is created and customer information is loaded into it. Customer is located into one of the following states at the time of entering the system:

1. System is in cold start situation and no one has purchased anything. In this case the recommended products are selected considering semantic distance in customer profile.
2. For the first time customer is logged into the system and no selection is made. In this case the products are sorted according to their rating and a product with the highest rank with the highest semantic similarity with customer profile is presented to customer. This product is recommended as TOP-N to the client.

3. Customer has not purchased anything from the system but has a history of purchase. In this case system recommendations are nodes in the system with minimum semantic distance with previously purchased products. In this regard based on the mutual semantic distance of purchased products, product clustering is used and in each cluster one product is selected with minimum distance relative to other bundled products. This product should not preferably previously be selected by the customer. Then selected products from each bundle are sorted and TOP-N product is recommended to the customer.

5.5. Recommendation Process

After logging into system, system recommends initial recommendations using principles stated in section 4-4 and forms TOP-N option. After selection of a product, the customer agent is located in corresponding node and travels across the graph using equations 6 to 9. This system prepares a list of recommended products. System repeats this cycle until it can select "m" products with rank and high semantic similarity with respect to purchased product of customer. "m" is one of the parameters of this system. "m" products are replaced with "m" TOP-N products with the least semantic relations by products.

According to these mechanisms, a pseudo-code is presented in the following:

- 1- *After logging of customer to the system, a customer agent is established and customer background is loaded into it,*
- 2- *According in section 5.4, present initial recommendations to the customer,*
- 3- *Based on selected products do the following things:*
 - 3-1- *Using equations 7 to 10 equations, produce a list of products and recommend it according to section 5.1;*
 - 3-2- *After selection of products follow rating mechanism in section 5.3 and go to 3;*
- 4- *Completion of procedure and documentation of customer background;*

6. Evaluation and Model Validation

6.1. Test Data

In order to evaluate the recommended algorithm sample data from a Building Equipment Company is utilized [8]. This transaction data are recorded from 22-04-2005 to 01-08-2006. These data includes 2266 customers, 2581 products, 21662 transactions. Cement, stones and other construction materials are sample of products. In this ontology all products are categorized in 18 categories and their related ontology is defined.

6.2. Evaluation Criteria

Many of recommender systems use two well-known recall and precision evaluation criteria [6, 8]. These two criteria are usually evaluation criteria in Information retrieval. In this paper two F-measure or F-score criteria are used for evaluation of presented algorithm [22]. This criterion considers both recall and precision evaluation criteria. Also this criterion is usable in evaluation of classification algorithm performance based on false-negative and false-positive

criteria. This criterion has direct relation with true-positive and reverse relation with false-negative and false-positive. False-positive criteria are indicative of un-recommended products while they are favourable to costumers. False-Negative criteria are representative of recommended products which are not favourable to costumers. False-positive criteria are very important in electronic commerce since they may reduce the satisfaction of costumers [23].

Considering the relative importance of precision and recall in F-measure criterion, this is also called *F1* criterion. In fact *F1* criterion is harmonic average of precision and recall and can be calculated using following formula:

$$F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (11)$$

6.3. Experimental Results

In order to evaluate the system, we have divided problem data into two training and test series. Training and test data comprise 80% and 20% of whole sample respectively. In order to construct this problem graph first an ontology is formed and them semantic relatedness between each pairs of product is calculated and stored in a special table. Then using this table we can construct problem graph.

Proposed algorithm is compared with associated rule mining using response time and *F1* criterion. This latter method is common in Electronic Commerce recommender systems. The presented results are average of ten executions of presented algorithm and average value is presented to avoid probable errors.

6.3.1. Response Time Comparison: One of the main factors in recommender systems is response time. From response time point of view the proposed algorithm perform better than the associated data mining system. The data mining algorithm scans whole database for each frequent item set. While the developed algorithm required less time for developing a recommendation.

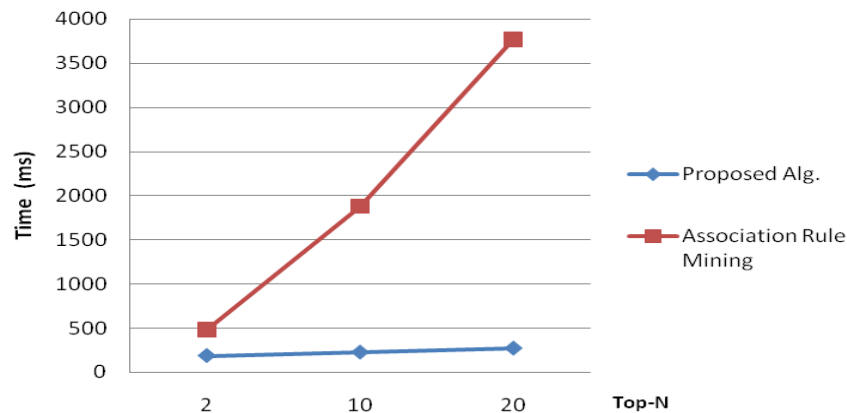


Figure 1. Comparison between Response Time of Proposed Approach and Associated Data Mining

6.3.2. Accuracy Comparison: In other Experimental the developed algorithm is compared to associated data mining approach based on F-measure criterion. Test results are presented in Figure 1. The better performance of developed algorithm comparing with associated data mining approach is evident. The main reason behind this is the nature of calculation of frequent item set in associated data mining approach. For instance ($A \rightarrow B$) is only considered

as an associated dependency in data mining approach when substantial number of customers have purchased is simultaneously. If (A,B) is lower than support value of simultaneous purchase, this would not be visible in data mining approach as well. This is also true even in case of semantic relation between them. Therefore the output of data mining approach is influenced by costumers' behavior not the nature of complementary products. However in the presented algorithm both costumer behavior and semantic relations are considered and result in more efficient recommendations for complementary products' purchase.

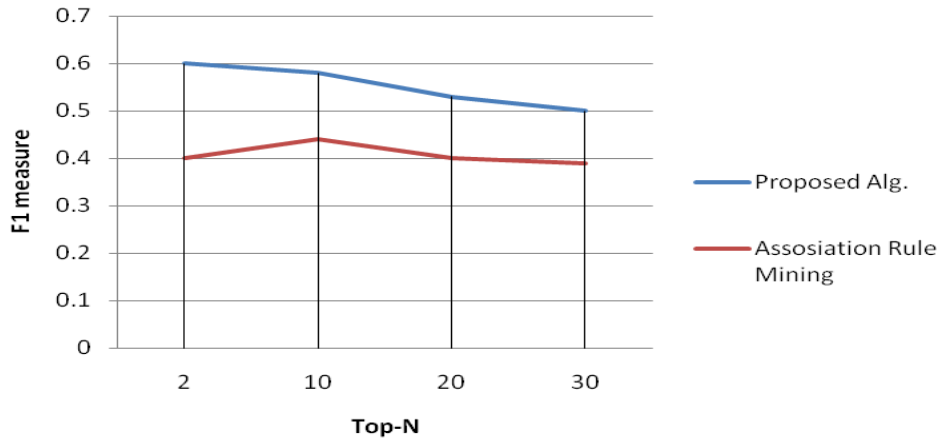


Figure 2. Comparison of F1 Criterion between Proposed Approach and Associated Data Mining

7. Conclusions

In this paper a new recommender system based on Ant colony algorithm and semantic distance in ontology is developed. This system uses semantic relations in ontology and structure of Ant colony theory. The presented algorithm is able to recommend the complementary, similar and bundled products. Typical problems such as cold-start, scalability and scarcity of information can be omitted in this system. According to the validation experimental the satisfying performance of this system is verified.

References

- [1] M. Balabanovic and Y. Sholam, "Combining content-based and collaborative recommendation", Communications of the ACM, vol. 40, no. 3, (1997).
- [2] D. Kales, A. Papagelis and C. Zaroliagis, "Algorithmic aspects of web intelligent systems", N. Zhong, J. Liu, & Y. Yao (Eds.), Web intelligence. Berlin: Springer, (2003).
- [3] J. Herlocker, J. A. Konstan, L. Terveen and J. Riedl, "Evaluating collaborative filtering recommender systems", ACM Transactions on Information Systems (TOIS), vol. 1, no. 22, (2004).
- [4] T. Hofmann, "Collaborative filtering via Gaussian probabilistic latent semantic analysis", Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval, Toronto, Canada, (2003).
- [5] P. Resnick, "GroupLens: An open architecture for collaborative filtering of netnews." Proc. of the ACM Conference on Computer Supported Cooperative Work, Chapel Hill, North Carolina, United States, (1994), pp. 175-186.
- [6] H. Khosravi Farsani and M. Nematbakhsh, "A semantic recommendation procedure for electronic catalog", Transaction on Engineering, Computing and Technology, vol. 16, (2006).
- [7] R. Q. Wang, F. S. y Kong, "Semantic-Enhanced Personalized Recommender System", Machine Learning and Cybernetics: Proc. of the Int. Conference on Machine Learning and Cybernetics, vol. 7, (2007) August 19-22, pp. 4069-4074.

- [8] S. Moosavi, H. Khosravi Farsani and M. Nematbakhsh, "A semantic complement to enhance electronic market", *Expert Systems with Applications*, (2008).
- [9] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator and W. Swartout, "Enabling technology for knowledge sharing". *AI Magazine*, vol. 12, no. 3, (1991).
- [10] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing", *International Journal of Human-Computer Studies*, vol. 43, no. 5, (1995).
- [11] N. Guarino and P. Giaretta, "Ontologies and knowledge bases towards a terminological clarification", Proc. of In: 2th int. conf. of Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, University of Twente, The Netherlands, (1995) April 10-13, pp. 25-32.
- [12] Z. Wu, and M. Palmer, "Verb Semantics and Lexical Selection", *Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics*, Las Cruces, New Mexico, (1994) June 27-30, pp. 133-138.
- [13] Y. Li, Zuhair and D. McLean, "An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources", *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, (2003).
- [14] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification", Fellbaum, C., ed.: *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA. (1998).
- [15] W. Mao and W. W. Chu, "Free text medical document retrieval via phrased-based vector space model", Proc. of the American Medical Informatics Association, San Antonio, TX, (2002) November 9-13.
- [16] A. Coloni, M. Dorigo and V. Maniezzo, "Distributed Optimization by Ant Colonies", F. J. Varela and P. Bourguine, editors, *Proceedings of the First European Conference on Artificial Life*, MIT Press, Cambridge, MA. (1992).
- [17] M. Dorigo and L. M. Gambardella, "Ant colonies for the traveling salesman problem", *BioSystems*, vol. 43, no. 2, (1997).
- [18] L. M. Gambardella and M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies", *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'96)*, IEEE Press, Nagoya, Japan, (1996), pp. 622-627.
- [19] M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", *IEEE Trans. E. Comp.* vol. 1, (1997).
- [20] E. Bonabeau, M. Dorigo and G. Theraulaz, "Inspiration for Optimization from Social Insect Behavior", *Nature*, vol. 406, (2000).
- [21] L. Tong Queue, P. Young and P. Yong-Tae, "A time-based approach to effective recommender systems using implicit feedback", *Expert Systems with Applications*, (2007).
- [22] Y. H. Cho, J. K. Kim and S. H. Kim, "A personalized recommendation system based on web usage mining and decision tree induction", *Expert Systems with Applications*, vol. 23, (2002).
- [23] X. Zhang, "A Comparative Study of Ontology Based Term Similarity Measures on PubMed Document Clustering", *Lecture Notes in Computer Science*, Springer, vol. 4443, (2008).
- [24] K. Hameed, H. Shah, K. Ahsan and W. Yang, "An Enterprise Architecture Framework for Mobile Commerce", *Int. J. of Computer Science Issues*, vol. 7, no. 3, (2010).

