A Neural Network Hybrid Recommender System

Maria Nadia POSTORINO and Giuseppe M. L. SARNE' [1]
Università Mediterranea di Reggio Calabria, Italy

Abstract. Among the Web opportunities, e-Commerce processes have increased in relevance requiring the development of complex tools to support all the parties involved therein. This paper proposes a neural network hybrid recommender system able to provide customers, associated with XML-based personal agents within a multi-agent system called MARF, with suggestions about flights purchases. MARF agents continuously monitor customers' interests and preferences in their commercial Web activities, by constructing and automatically maintaining their profiles. In order to highlight the benefits provided by the proposed flight recommender, some experimental results carried out by exploiting a MARF prototype are presented.

Keywords. Multi-agent system, Neural network, Recommender system

1. Introduction

E-Commerce (EC), defined as any form of commercial activities conducted over Internet [7], plays a pivot role among the Web opportunities and more and more sellers and buyers exploit the advantages of a real open-world market by offering and purchasing products and services. When an EC transaction occurs between a merchant and a customer, it is known as a Business-to-Consumer (B2C) process. It corresponds to the traditional retail commerce, that in 2012 is expected to grow up for the US home market till 183,9 billion of dollars (www.emarketer.com).

To support B2C users' activities in a personalized way, new systems, with high levels of automation and capability to process complex tasks and large amounts of data, have been designed. Customer behavioral profiles are built so as to take into account customer's interests and preferences; a direct elicitation approach based on explicit user's ratings or implicitly the use of information that customer spreads in his/her B2C activities are used to this aim. Software agents [2] automate this task to obtain customers' profiles appearing closer to their real interests and preferences than other approaches [3]. Recommender systems [20,21,26] can exploit the information stored in such profiles to provide users with potentially useful suggestions for their purchases. The most part of

[1] Corresponding Author: DIMET, Facoltà di Ingegneria, Università Mediterranea di Reggio Calabria, Località Feo di Vito, 89122 Reggio Calabria, Italy; E-mail:sarne@unirc.it.

[2] Software agents can actively interact with the environment and carry out delegated tasks in simple, intelligent and independent manners [25,17,6,16,9,18].

[3] Note that it is difficult to represent suitably unexpected or spontaneous behaviors such as impulsive purchases and, however, an initial user's profile should be generated to solve the cold start issue.

recommender systems are *(i) Content-based*, based on the past interests shown by the user, and *(ii) Collaborative Filtering*, based on the past interests of similar users. Anyway, *Hybrids* systems, combining content-based, collaborative filtering and possibly other techniques to generate suggestions, are commonly recognized as the most performing [2].

This paper proposes a Multi-Agent Recommender Framework (MARF) to support customers, associated with personal agents, in their tourist flight choices with suggestions generated by a content-based and neural network collaborative filtering recommender system.

The paper is organized as follows. Section 2 deals with related work, while Section 3 and 4 respectively present the multi-agent framework and the recommendation algorithm exploited therein. In Section 5, an experiment able to test the proposed recommender is presented and discussed and, finally, conclusions are drawn in Section 6.

2. Related Work

Recommender systems have been designed by using software agents in a large number of models and architectures proposed in a very large variety of works. For such a reason, approaches examined in this section are those that, to the best of our knowledge, come closest to the material presented in this paper. The interested reader might refer to [20,21,25,3,26] for a more complete overview. All the systems cited in this Section *(i)* exploit the agent technology, *(ii)* store information about user's interests and preferences in an internal profile and *(iii)* implement hybrid approaches combining content based and collaborative filtering techniques to generate most effective recommendations.

Recommender systems are widely used within Web sites (e.g., Amazon.com, CDNOW.com, MovieLens, etc. [16]) to provide suggestions closed to users' interests. Different techniques are adopted, but no one is *a priori* the best for all users in all situations [26]. Recommender agent systems using users' profiles and working in a B2C scenario normally adopt hybrid approaches based on content-based and collaborative filtering techniques [1,12,22]. Within this scenario, the Handy Broker evolutionary ontology-based approach [5] takes into account the whole user's history, stored in a profile, on the basis of how many times a product has been selected; it evaluates product relevance due to tangible attributes (e.g., price). An evolutionary mechanism performs an implicit and rough collaborative filtering stage by integrating similar users’ profiles. In [13] agents assist a consumer for products rarely purchased. Domain experts are used to compute optimal products matching customers' preferences obtained by a questioner (a multi-attribute decision making method based on consumer's needs and products features is used in this case). Moreover, in order to share the experiences of other consumers social information are used to infer previous consumer suggestions.

Content-Boosted Collaborative Filtering (CBCF) [15] is both a content-based predictor to process user data rated in six classes of relevance by the same user, and a collaborative filtering approach based on a neighborhood-based algorithm to produce personalized suggestions by weighting ratings of similar users. Multi Agent System Handling Adaptivity (MASHA) [19] is a hybrid multi-agent recommender system that provides suggestions by considering the device currently exploited as well. MASHA generates personalized Web site presentations containing suggestions derived both from the profile of the current user and the profiles of the other users that, during the past site visits, exploited the same type of device. Note that MASHA can also work without utilizing its advanced features. Duine [24] implements a switching method to choose among more techniques and able to provide more accurate recommendations. Unlike other hybrid recommender systems, Duine is domain-independent.

Structural properties of the graphs (i.e., nodes and links) can model user's interests and preferences. In [10] a two-layer graph model, adaptable to different techniques, is implemented to generate content-based, collaborative filtering and hybrid recommendations. Users-products information uses nodes, intra and inter layer links that model in two layers *(i)* customers and products and *(ii)* transactions and similarities. The authors exploit this model with three different techniques (i.e., "direct retrieval", "association mining" and "high-degree association retrieval") to obtain content-based, collaborative filtering and hybrid recommendations.

3. The Agent Framework

In this section, the *Multi-Agent Recommender Framework* (MARF) is shortly presented. It is conceived to support the personalized suggestions generation for customers' tourist flight choices. In MARF each customer *C* (merchant *M*) is associated with a personal agent *c* (*m*) logged in an agency (*Ag*). All the agents share the same vocabulary (*V*), implemented with a XML-Schema [4], where each trip destination (*d*), represented by a pair *(id code, textual description)*, is defined as an "element" and each flight (*f*) going to *d* is its "element instance". Agent communications occur by means of messages exchanged when they are: *(i)* explicitly requested; *(ii)* the result of a monitored action carried out in MARF; *(iii)* the result of an automatic process.

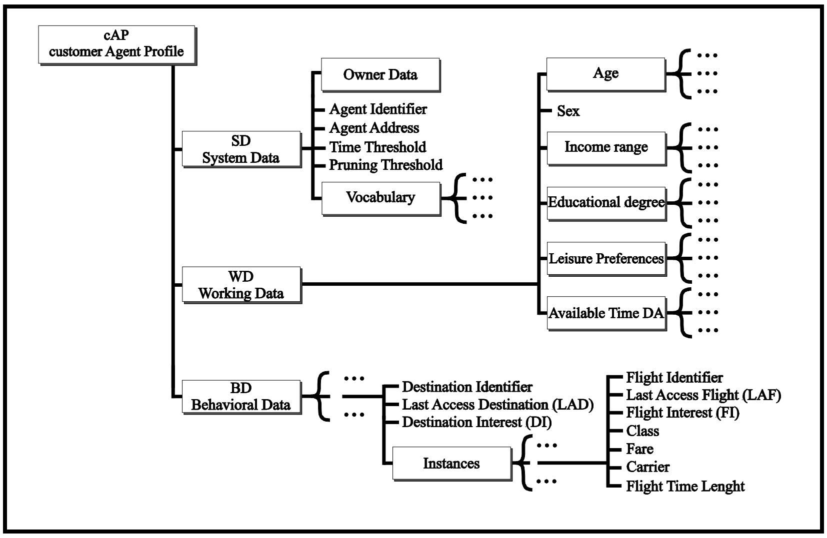

Figure 1. The customer Agent Profile (c.AP).

The agent behavior consists of: *(i)* simple operations carried out to affiliate or delete the agent with MARF; *(ii)* all the actions related to customer's commercial activities and/or required by the platform and automatically carried out by the agent. Each agent controls three management areas in its *Agent Profile* (*AP*), in terms of insertion, deletion and updating. More in detail, the AP customer agent (Fig. 1) consists of:

- *system data* area storing: agent platform information, vocabulary V and some parameters used in the customer's interests measures (see below);
- *working data* area storing: customer's age, sex, income range, educational degree, leisure preferences, available time for discretionary activities;
- *behavioral data* area storing: identifier, last access data and interest measure for each destination and flight identifier, last access data, class, fare, carrier and flight time length (representing a proxy distance measure) for each corresponding destination instance (i.e., flight).

Differently, the AP of a merchant agent collects: *(i)* working data information of each customer agent that visited in the past the merchant's Web site, stored in the working data section and *(ii)* all the information related to visits and/or purchases performed by the customer over the site, for each destination and its instances, stored in the behavioral data area.

Particularly, the interest in a destination instance, *f*, is measured by the *Flight Interest* (*FI*), ranging in (0,1) and computed as:

$$\mathrm{FI} = \begin{cases} \mathrm{FI} + \Delta\,, \text{ where } \Delta = \begin{cases} 0.00 \text{ if none activity is performed} \\ 0.05 \text{ for page visits} \\ 1.00 \text{ for purchases} \end{cases} & \text{iff } \mathrm{FI} + \Delta \le 1 \\ 1, \text{ otherwhise} \end{cases} \qquad (1)$$

The interest in a destination *d* is measured by the Destination Interest (DI), computed as the average of all the FI values of the destination instances *f* associated with *d*. In this way, on the customer (i.e. seller) side, DI takes into account the whole customer's history (i.e. customers' histories). Automatically, each interest value DI (FI) decreases if its last access LAD (LAF) is older than the Time Threshold (TT), a system parameter expressed in days. Furthermore, DI (FI) is set to 0 when it is less than the Pruning Threshold (PT), another system parameter.

The agency manages (in terms of insertions, deletions and updating) its *Agency Profile* (*AgP*), where system data and the identifiers of all the MARF agents (Fig. 2) are stored. The agency carries out, without human intervention, all the operations required to manage the framework as affiliation, deletion, limitation of the potential inactive affiliate growth and so on. Moreover, the agency provides agents with a broadcasting message and a yellow page services.

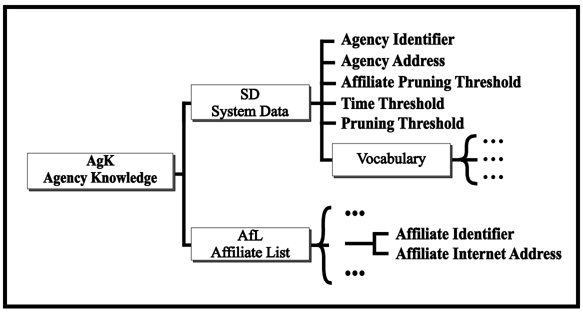

Figure 2. The Agency Profile (AgP).

4. The Recommendation Algorithm

Recommender systems are powerful tools to promote businesses by providing potential buyers with personalized suggestions for their purchases. To this end, a hybrid recommender system has been designed; it is able to act both as content-based and collaborative filtering to select those instances matching interests and preferences of the current customer to those of other similar customers. In the recommendations generation the merchant agent interacts with the customer by exploiting representations of customers' interests and preferences based on the Web actions they performed on the merchants' site.

The recommendation algorithm is encoded in the function *recommendations* (Fig.3) running on the merchant side. Its input is a customer agent *c* and its outputs are the lists *L3* and *L4* of destination instances (i.e., flights) for *c*. The function *extract_mD*, called by *recommendations*, returns the list L1 containing all the destinations on sale over the merchant's Web site (stored in the merchant AP) ordered in a decreasing manner on the basis of their interest values. In order to preserve the customer's privacy, the list *L1* is sent to *c* by calling the function *send*. As consequence, *c* provides *m* with the lists *L2* and *bL* by using the function *receive*. The list *L2* includes the *max_d* (a parameter set by *c*) destinations with the highest *c* scores, while the list *bL* (exploited in the collaborative filtering task) includes the destinations refused by the customer, usually chosen by him/her in the past. When *L2* is received, the function *contentBasedF* is called and returns the list *L3* enclosing the first *max_f* (another parameter set by *c*) destination instances compatible with *L2* and having the highest scores. Note that *L2* is a qualitative representation of the customer's interests that also considers the customer's activities carried out over other sites.

```
void recommendations(customerAgent c, ListOfInstances L3, ListOfInstances L4) {
    ListOfElements L1 = extract_mD(m.AP.BD);
    send(L1, c);
    ListOfElements L2, bL = receive( );
    ListOfInstances L3 = contentBasedF(L2, m.AP.BD, max_f);
    ListOfInstances L4 = collaborativeFilteringF(L2, bL, m.AP.WD, m.AP.BD, first_f);
    return; }

ListOfElements = elementOfInterest(ListOfElements L1) {
    ListOfElements bL = deselectD(c.AK.BD);
    ListOfElements L5 = extract_cD(c.AK.BD, bL);
    ListOfElements L6 = intersectionD(L1, L5);
    ListOfElements L2 = selectD(L6, max_d);
    return L2, bL; }
```

Figure 3. The recommendation algorithm.

The generation of collaborative filtering recommendations is often an expansive computational task requiring complex algorithms to analyze large amounts of data and complex tuning phases to obtain satisfactory results. As a consequence, the collaborative filtering stage often has to be carried out partially or fully off-line. In order to simplify such a task, a single neural network[4] (NN) has been used to learn relationships among customers' preferences and flights. To this aim, a back-

[4] A deeper analysis on NNs is out of the aim of this paper, given the considerable and well established literature on the topic, among the others [8].

propagation NN has been chosen; the data-set has been built by taking into account the customers' preferences and activities of the users that in the past have interacted with the merchant Web site, stored in the AP of its *m* agent. To simplify the learning task, only binary inputs have been adopted by structuring data in classes and using an input neuron for each class. The training set considers all the customers' Web site visits or purchases; for each of its items (i.e., a site visit or a purchase) it consists of 32 input data and 1 output data. More in detail, the input data (neurons) are the following: age (4); sex (1); income range (3); educational degree (3); leisure preferences (7); available time for discretionary activities (3); class (2); fare (3); carrier (1) and flight time length (5). The output of the training set simply is the FI score computed on the merchant side. Many NN topologies have been trained off-line with different data sets; the best topology has been a three-layer NN with 32 input neurons, 65 neurons in the hidden layer and 1 neuron as output layer. The neuron activation function is the hyperbolic tangent in the hidden layer and the sigmoid one in the output layer.

This unique back-propagation NN, trained on customers' preferences and interests, works within the function *CollaborativeFilteringF* to generate suitable suggestion for all them. This function accepts as inputs the lists *L2* and *bL* supplied by a *c*, with the WD and BD sections of the *m* profile, the integer *first_f* (a parameters arbitrarily set by *c*) and returns the list *L4*. More in detail, the characteristics of the current customer and all the product instances offered by the merchant, purified from those having a destination included in the black-list *bL* provided by *c* (i.e., the refused destinations for *c*), are the inputs of the trained NN. Then the *first_f* destination instances having the highest output value produced by the trained NN are inserted in the list *L4*.

On the customer agent side, when the list L1 coming from the merchant agent *m* is received, the function *elementOfInterest* (Fig.3) is executed. It calls the function *deselectD* to obtain the list *bL* storing the customer's refused destinations. Then the function *extract_cD* computes the list *L5* storing the destinations of interest for the customer but purified from the destinations present in *bL*. When the function *intersectionD* is called, the intersection of *L1* and *L5* is computed. The function *selectD* receives as input an integer *max_d* and the list *L6*. This latter is ordered in a decreasing manner, based on the interest values stored into the agent profile; then its first *max_d* destinations are inserted into the list *L2* that is the output of *elementOfInterest*.

5. Experiment

A preliminary experiment is presented in the current section aimed at evaluating the capability of the proposed hybrid recommender to generate content-based and collaborative filtering suggestions assisting customers in their flight choices.

To this purpose, the MARF system has been implemented in JADE [11] following its agent data structures and the recommendation algorithm descriptions. In particular, the parameters *max_d*, *max_f* and *first_f* of the recommendation algorithm have been set respectively to 3, 2 and 6 (see Section 4). The experiment has been executed by utilizing a family of 10 XML Web sites where each proposed destination instance was associated with a Web page, produced from a XML document. Some volunteers (different for age, sex, preferences and so on) accepted to test the system. On the basis of his/her characteristics and for each Web site, each user had a virtual budget to spent in purchases. In order to obtain an initial profile of the customers' interests, agents monitored the activities performed by the volunteers over the first 5 Web sites without exploiting any recommender help. Based on such profiles, some suggestions have been generated to support users for the other 5 sites.

To evaluate the relevance of the generated suggestions two lists, called *A* and *B*, have been considered for each user. In these lists the destination instances suggested by the merchant agent and the corresponding customer's choices have been collected. The associated pairs in the two lists have been compared in order to measure the effectiveness of the provided support for the customers' activities by means of the standard performance metrics *Precision*, *Recall* and *F-measure* [23]. More in detail, the Precision (2) is defined as the share of the instances actually visited by the user among those recommended by the system. The Recall (3) is the share of the instances suggested by the system among those chosen by the user. The F-Measure (4) represents the harmonic mean of Precision and Recall. Formally, these measures are defined as:

$$\mathrm{Pr}\,e(A(x)) = \frac{|A(x) \cap B(x)|}{|A(x)|} \tag{2}$$

$$\mathrm{Re}\,c(A(x)) = \frac{|A(x) \cap B(x)|}{|B(x)|} \tag{3}$$

$$F(A(x)) = \frac{2 \cdot \mathrm{Re}\,c(A(x)) \cdot \mathrm{Pr}\,e(A(x))}{\mathrm{Re}\,c(A(x)) + \mathrm{Pr}\,e(A(x))} \tag{4}$$

As shown in Table 1, the experimental results, in terms of Precision, Recall and F-measure for the global, content-based and collaborative filtering components, evidence the satisfactory quality of the generated suggestions. From their analysis, it can be argued that: *(i)* performances are probably due to the good quality of the customers' profiles; *(ii)* the adoption of a NN as collaborative filtering engine promises future interesting developments.

Table 1. Performances of the proposed hybrid recommender.

component	Precision	Recall	F-measure
global	0.574	0.541	0.557
content-based	0.495	0.445	0.469
collaborative filtering	0.403	0.374	0.388

6. Conclusion

This paper illustrates a *Multi-Agent Recommender Framework*, called MARF, fully implemented and tested. The recommender adopts a hybrid approach, both content-based and collaborative filtering (implemented by exploiting a NN), to produce suggestions about flights purchase for supporting customers and merchants in their Web activities. MARF agents build, update and exploit customers' profiles able to provide a representations of customers' orientations.

A preliminary experimental session has been carried out by using a JADE-based prototype. The performances exhibited by the recommendation algorithm confirm expectations in terms of quality of the generated suggestions. In particular, the introduction of a NN to generate collaborative filtering solution produces interesting results that should be further verified in the future with a wide experimental campaign.

References

[1] M. Balabanovic, and Y. Shoham, Content-Based, Collaborative Recommendation, *Communication of the ACM*, **40**(3), (1997), 66-72.

[2] J. Breese, D. Heckerman and C. Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proc. of the 14th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'98)*, San Fransisco, CA, USA, 43-52, Morgan Kaufmann, 1998.

[3] R.D. Burke, Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction*, **12**(4), (2002), 331-370.

[4] Extensible Markup Language (XML) Schema URL, *http://www.w3.org/XML/Schema*, 2010.

[5] S. Guan, C. S. Ngoo, and F. Zhu, Handy broker: an Intelligent Product-Brokering Agent for m-Commerce Applications with User Preference Tracking, *Electronic Commerce Research and Application*, **1**(3-4), (2002), 314-330.

[6] R. H. Guttman, A. Moukas and P. Maes, Agents as Mediators in Electronic Commerce, *Electronic Markets*, **8**(1), (1998), 22--27.

[7] R. Hake, *E-commerce, A World of Opportunity*, The British Computer Society, Wiltshire, UK, 1999.

[8] S. Haykin, A Comprensive Foundation, *Macmillan College Publishing Company NY*, 2000.

[9] M. He, N.R. Jennings and H. Leung, On Agent-Mediated Electronic Commerce, *IEEE Transaction Knowledge Data Engineering.*, **15**(4), (2003), 985-1003.

[10] Z. Huang, W. Chung and H. Chen, A Graph Model for e-Commerce Recommender Systems, *Journal American Society Information Science Technology*, **55**(3), (2004), 259-274.

[11] Java Agent DEvelopment framework (JADE) URL, http://jade.tilab.com/, 2010

[12] R.D. Lawrence, G.S. Almasi, V. Kotlyar, M.S. Viveros and S. Duri, Personalization of Supermarket Product Recommendations, *Data Mining Knowledge Discovory*, **5**(1/2), (2001), 11-32.

[13] W.P. Lee, Towards Agent-based Decision Making in the Electronic Marketplace: Interactive Recommendation and Automated Negotiation, *Expert Systems with Applications*, **27**(4), (2004), 665-679.

[14] P. Maes, Agents that Reduce Work and Information Overload, *Communication of the ACM*, **37**(7), (1994), 30-40.

[15] P. Melville, R. Mooney and R. Nagarajan, Content-boosted Collaborative Filtering for Improved Recommendations, In: *Proc. of the 18th National Conf. on Artificial Intelligence, Edmonton, Canada*, 187-192, AAAI/IAAI, 2002.

[16] M. Montaner, B. Lopez and J. de la Rosa, A Taxonomy of Recommender Agents on the Internet, *Journal on Web Semantics (JWS)*, **19**(4), (2004), 285-330.

[17] H.S. Nwana, Software Agents: An Overview, *Knowldege Engineering Review*, **11**(3), (1996). 11-40.

[18] L. Palopoli, D. Rosaci and D. Ursino, Agents' Roles in B2C e-Commerce, *AI Communication*, **19**(2), (2006), 95-126.

[19] D. Rosaci, and G.M.L. Sarnè, MASHA: A Multi-Agent System Handling User and Device Adaptivity of Web Sites, *User Modelling User-Adaptive Interaction*, **16**(5), (2006), 435-462.

[20] B. Sarwar, G. Karypis J. Konstan and J. Riedl, Analysis of Recommendation Algorithms for E-Commerce, In: *Proc. of the 2nd ACM Conf. on Electronic Commerce (EC '00), New York, NY, USA*}, 158-167, ACM, 2000.

[21] J. Schafer, J. Konstan and J. Riedl, E-Commerce Recommendation Applications, *Data Mining Knowldege Discovery*, **5**(1-2), (2001), 115--153.

[22] K. H. L. Tso and L. Schmidt-Thieme, Evaluation of Attribute-Aware Recommender System Algorithms on Data with Varying Characteristics, In: *Proc. of the 10th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining (PAKDD 2006), Singapore, Vol. 3918 LNCS*, 831-840, Springer, 2006.

[23] C.J. van Rijsbergen, Information Retrieval, *Butterworth*, 1979.

[24] M. van Setten M., Supporting People in Finding Information. Hybrid Recommender System and Goal-based Structuring, *Technical report*, Telematica Instuut, 2005.

[25] C. Wei, M. Shaw and R. Easley, E-Service: New directions in Theory and Practice, Chapt. *A Survey of Recommendation Systems in Electronic Commerce*, Armonk, NY, USA, ME Sharpe, 2002.

[26] K. Wei, J. Huang and S. Fu, A Survey of E-Commerce Recommender Systems, In: *Proc. of the 13th Int. Conf. on Service Systems and Service Management*, Washington, DC, USA, 1--5, IEEE, 2007.

[27] Y. Wei, L. Moreau and R. Jennings, A Market-based Approach to Recommender Systems, *ACM Transaction Information System*, **23**(3), (2005), 227--266.

[28] M. Wooldridge and N.R. Jennings, Agent Theories, Architectures, and Languages: A Survey, In: *Proc. of the Workshop on Agent Theories, Architectures, and Languages {(ECAI-94), Amsterdam, Vol. 890 LNCS*, 1-39, Springer, 1995.