AN INTELLIGENT RECOMMENDATION SYSTEM BASED ON FUZZY LOGIC

Shi Xiaowei
Philips (China) Investment Co.,Ltd. Shanghai R&D Centre
38F,Tower1 Kerry Everbright City 218 Tian Mu Xi Road Shanghai, P.R.C.200070

Keywords: Recommendation, User profile, Fuzzy-logic, Multi-agent, Metadata

Abstract: An intelligent recommendation system for a plurality of users based on fuzzy logic is presented. The architecture of a multi-agent recommendation system is described. How the system simulates human intelligence to provide recommendation to users is explained. The recommendation system is based on the fuzzy user profile, fuzzy filtering and recommendation agents. The user profile is updated dynamically based on the feedback information. Fuzzy logic is used in fuzzy filtering to integrate different types of features together for a better simulation of human intelligence. Ambiguity problems can be solved successfully in this system, e.g., deducing whether a programme with both interesting features and uninteresting features is worth recommending or not. The application scenario shows that it is more convenient for users to find programmes of their interest with the proposed recommendation system.

1 INTRODUCTION

Due to the digitalisation of television, in the near future we will be able to receive hundreds of channels via satellites, terrestrial antenna, cable and even phone lines. At the same time, it is becoming increasingly challenging for television viewers to identify television programmes of their interest. Recommendation systems, such as TV-Advisor, Personal TV, etc, have been studied to help viewers to find, personalize, and organize the contents [1,2,3,4,5].

The following recommendation process is considered: matching Electronic Programme Guide (EPG) metadata to the user preference knowledge, filtering out the tedious programmes and recommending interesting programmes to users, and updating the user profiles based on feedback information.

The traditional methods to evaluate if a programme is good enough to be recommended are based on the explicit (i.e. non-fuzzy) inference. In other words, the programme evaluation result can either be “interesting” or “non-interesting”. As we know, explicit mathematics cannot intelligently simulate human’s flexible inference. Especially, it is difficult to decide whether a programme with both interesting features and uninteresting features should be recommended by the existing filtering and recommendation methods.

This paper presents an algorithm about the integration of fuzzy logic into a multi-agent recommendation system to better recommend interesting programmes to users. The architecture of the fuzzy recommendation system is described. Moreover, the reason why fuzzy logic is adopted in the recommendation system is explained. The fuzzy recommendation algorithm is also developed. Finally, the recommendation process is illustrated with an example.

2 RECOMMENDATION SYSTEM

2.1 System Architecture

The recommendation system is provided to generate programme recommendations for multiple users based on programme metadata and user profiles. The architecture of this system is illustrated in Figure 1. The system uses a central recommendation server unit that includes a Fuzzy User Profile Database, Fuzzy Filtering Agent, Fuzzy Recommendation Agent, Profiling Agent, and Interface Agent.

Xiaowei S. (2004).
AN INTELLIGENT RECOMMENDATION SYSTEM BASED ON FUZZY LOGIC.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 180-184
DOI: 10.5220/0001136301800184
Copyright © SciTePress

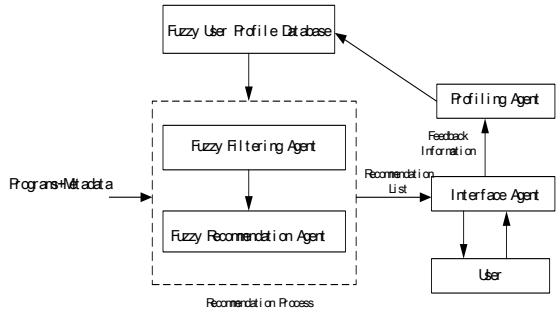

Figure 1: System architecture.

- Fuzzy Filtering means to gather and make the incoming live programme broadcasted based on programme metadata match to fuzzy user profiles, which is based on fuzzy logic, and then filter the programmes based on fuzzy threshold;
- Fuzzy Recommendation generates an optimal recommendation list to every user according to the learned user preference knowledge, and transmits it to user terminals;
- Profiling updates the user profile based on both the explicit and implicit feedback information from the Interface Agent;
- The Interface Agent handles the interactions with the user.

2.2 Information Description

In the context of TV-Anytime [7], metadata consists of two kinds of information: (a) content description metadata; (b) consumer metadata. Programme content description metadata includes attributes of a television programme, such as the title, genre, list of actors/actresses, language, etc. These data are used to make a search.

The user profile metadata in this system defines the details of a user's preferences and aversion. These descriptions are closely correlated with media descriptions, and thus enable the user to search, filter, select and consume the desired content efficiently.

2.3 Application of Fuzzy Logic Control Theory in a Recommendation System

There are many factors that influence a user if he/she wants to view a programme or not. The user's attitude to a programme is the result of some complicated reaction. In other words, it is difficult for a user to describe their emotion about a programme in quantity. Fuzzy theory can simulate human intelligence. It owns the advantage of describing this kind of indefinite object, providing a possible way to solve the problem. Hence, fuzzy theory is used in programme recommendation in this paper.

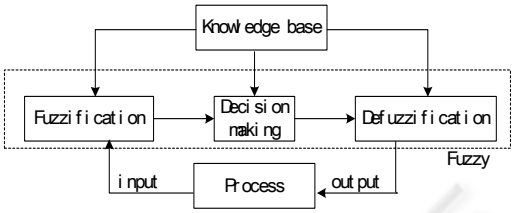

Figure 2: General structure of fuzzy inference system .

Fuzzy theory includes a series of procedures for representing set membership, attributes, and relationships that cannot be described by single point numeric estimates. The structure of a fuzzy inference system is shown in Figure 2.

Where,

Knowledge base: parameters of membership functions and definitions of rules;

Fuzzification: transformation of crisp inputs into membership values;

Decision-making: fuzzy inference operations on the rules;

Defuzzification: transformation of the fuzzy result of the inference into a crisp output

3 FUZZY RECOMMENDATION

3.1 Fuzzy Information Database

3.1.1 The Fuzzy User Profile

The user profile, UP, can be represented by a vector of these 3-tuples. If there are m distinct terms in the profile, it will be represented by:

$$UP = ((t_1, ld_1, w_1),....(t_i, ld_i, w_i).....,(t_m, ld_m, w_m)) \quad (3\text{-}1)$$

Where: t_i is a term; ld_i is the "Like_degree" of term t_i; w_i is the Weight of term t_i; i is the order of t_i in the profile.

The fuzzy user profile transforms the crisp parameters ("Like_degree", Weight) into membership values. "Like_degree" means the degree the user likes a feature. The shape and location may be different for different problems. If e_1 and e_2 represent "Like_degree" and "Weight" respectively, the fuzzy memberships can be described as Figure 3. It is known from Figure 3 that

Figure 3: The fuzzy membership function of user profile.

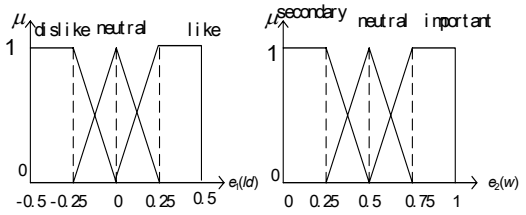

Figure 3: The fuzzy membership function of user profile

e_2 is always greater than 0. A larger e_2 indicates that this feature is more important. When $e_1 \geq 0$, it means "like" and a larger e_1 means the user likes the feature more; if $e_1 \leq 0$, it indicates that the user "dislike" this feature and the smaller e_1, the more the user dislikes it.

3.1.2 EPG Metadata

A programme can be represented by a vector of size n

$$C = (t_1, t_2, ..., t_i, ...t_n) \quad (3\text{-}2)$$

Where t_i is the i^{th} feature of the programme.

3.2 Similarity Matching

The Fuzzy Filtering Agent calculates the similarity between the programme and the user profile. If the calculated similarity is above the preset threshold, the user is considered to be interested in the programme; then the programme metadata is transferred to the Recommendation Agent.

In this system, the fuzzy similarity matching process can be divided into two steps:

- Feature matching;
- Programme matching.

3.2.1 Feature Matching

Feature matching decides how a feature of a programme is related to a user's preference. In order to get the feature "interest_degree" , the procedures is executed as shown in Figure 2.

Here, Like_degree and Weight are set as input and the feature of "interest_degree", which means how much a user likes the programme, is set as output.

The step of decision making is based on the fuzzy inference rules such as following,

I.If Like_degree is "dislike" And Weight is "secondary" Then f_i is "disgusted";

II.If Like_degree is "dislike" And Weight is "important" Then f_i is "very disgusted";

III.If Like_degree is "neutral" And Weight is "important" Then f_i is "neutral";

…

The method of "centre of gravity" takes more useful factors into consideration. It is adopted in this system to defuzzicate the feature's interest_degree.

3.2.2 Programme Matching

Programme matching is to evaluate the programme interest_degree. It can be calculated by the average interest_degree of the features related with the programme.

3.3 Filtering & Ranking

Next step is to set a threshold to filter the coming programme metadata, select the interesting programme, and then rank and recommend them based on the programme interest_degree. In this system, ranking and recommendation processes are performed by the Fuzzy Recommendation Agent.

A threshold is set in the Fuzzy Filtering Agent. The threshold can be a crisp value, or a fuzzy value such as "how much does the user like". If the programme interest_degree is greater than the threshold, which means the programme is what the user wants to watch, then the Filtering Agent will transfer the programme metadata to the Fuzzy Recommendation Agent.

Based on the learned user preference knowledge, the Fuzzy Recommendation Agent generates an optimal recommendation list and sends it to the user according to interest_degree.

3.4 Profiling Agent

In the system, the feedback can be explicitly given by the user or implicitly derived from observations of the users' reaction to a recommended programme. So, the Profiling Agent revises the user profile based on both the explicit and implicit feedback information. In this section, how to update the user profile by the implicit feedback information is mainly explained.

For a recommended programme, the user always has two attitudes to it: skipping over, or watching. In other words, the user will skip (delete) the programme he/she dislikes, watch the programme he/she likes or he/she is not sure. If the user has watched the programme for a period of time, the user's profile will be refined and revised based on the viewing behaviour.

In this system, for programme i, the algorithm for updating user profile is depicted as follows:

$$Weight_i' = Weight_i + \alpha \bullet \frac{(WD_i - \theta)}{RD_i} \quad (3\text{-}3)$$

$$Like\_degree'_i = Like\_degree_i + \beta \bullet \frac{(WD_i - \theta)}{RD_i} \quad (3\text{-}4)$$

Where, WD_i : The time duration watched;
RD_i : The real time duration of the programme;
θ : The threshold of the time duration. If WD_i is less than θ, that means the user is not interested in that programme;
α and β : are less than 1. They are used to slow down the change of Weight and Like_degree. Because Weight is more stable than Like_degree, $\alpha \leq \beta$.

If $Weight'_i$ is larger than its higher-boundary, let $Weight'_i = higher\_boundary$;
If $Weight'_i$ is less than its lower-boundary, let $Weight'_i = lower\_boundary$;
If $Like\_degree'_i$ is larger than its higher-boundary, let $Like\_degree'_i = higher\_boundary$;
If $Like\_degree'_i$ is less than its lower-boundary, let $Like\_degree'_i = lower\_boundary$.

4 EXAMPLE

4.1 Similarity Matching and Preference Learning Example

Table 1 shows an assumed user profile A and upcoming programme. The problem to be solved is to determine whether the user likes the programme and how much he/she does.

Based on the above described procedures, the programme fuzzy filtering inference can be illustrated as Figure 4.

For the user profile, the actor LiQinqin's Like_degree is -0.125, which indicates the user's emotion about him is between "dislike" and "neutral". In this case, both values of $\mu_{ld=neutral}$ and $\mu_{ld=dislike}$ are 0.5. In addition, the feature of "Actor" 's Weight is 0.8, so this feature is "important" and $\mu_{important} = 1$.

Matching by fuzzy logic rules, the actor LiQinqin meets both rule II and III. For rule II , μ_{fi} is 0.5, which means the user is "very disgust" at this feature; for rule III, μ_{fi} is 0.5 and the user feels "neutral" about this feature.

Through defuzzification, interest_degree f_2 for the actor LiQinqin is about -0.4. It shows that the user's emotion about this feature is mainly "much disgusted";

Considering other features, the calculated value of the programme interest_degree P is 0.45. From Figure 4, when P (0.45) is mapped into its fuzzy membership , the emotion of the user can be obtained. In the case discussed, it is between "much interested" and "interested" ($\mu_{interested} \approx 0.2$, $\mu_{much\ interested} \approx 0.8$).

Table 1: Initial conditions.

User profile A	Programme"Cala is a dog"
Genre Weight=0.9 (Movie Like_degree=0.5 …); Actor Weight=0.8 (Ge you Like_degree=0.5 ….. LiQinqin Like_degree=-0.125);	Genre is movie Actors are GeYou, LiQinqin Duration=2hour

According to the user's behaviour, the user profile is updated. For the discussed programme, duration RD_i is 2 hours. Assumed, the threshold of the watching time duration $\theta = 20$ minutes, the time duration watched WD_i =2 hours, α =0.01 , β =0.1, then:

$$\frac{(WD_i - \theta)}{RD_i} = 0.83$$

For the fuzzy model of user profile A assumed, the user profile is updated as following:

Genre Weight=0.9083
(Movie Like_degree=0.5083
Comedy Like_degree=0.3
News Like_degree=-0.2);
Actor Weight=0.8083
(XuJing Like_degree=0.1
GeYou Like_degree=0.583=0.5
LiQinqin Like_degree=-0.125+0.083=-0.042);

4.2 Application Scenario

An application scenario is provided. Figure 5 shows the main user interface of the fuzzy recommendation system. Functions are listed in the left column. A programme recommendation list is on the topside of middle column. The description of a selected programme is presented in the bottom of middle column. For a selected programme, three choices are provided to the user, which are "Display", "Delete", and "Skip".

5 CONCLUSION

An intelligent Multi-agent recommendation system is developed to provide programme

recommendations for multiple users based on programme metadata and fuzzy user profiles.

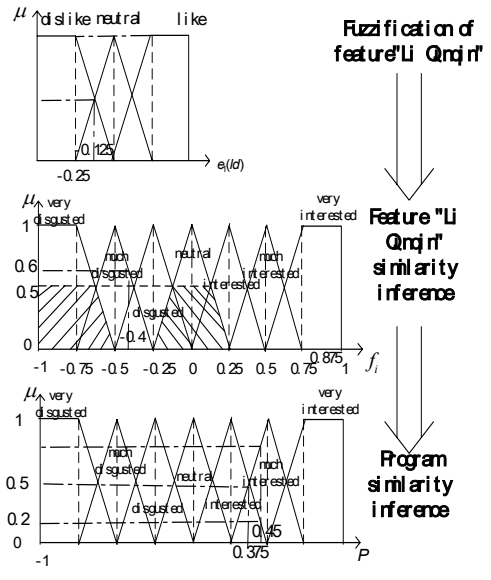

Figure 4: The programme filtering inference.

Different from traditional methods, the user profile is not based on the viewing history but on a compact and low-cost structure, including all terms that user "likes" and "dislikes". The user profile is updated dynamically, e.g. "increase" or "decrease" the corresponding preference parameters according to the feedback information. The filtering agent uses fuzzy logic to integrate different types of features together for a better simulation of human intelligence. This system shows a better capability in solving problems of ambiguities in programme recommendation, e.g., deducing whether a programme with both the interesting features and uninteresting features is worth recommending or not. The application scenario shows that this recommendation system can help users enjoy life more freely.

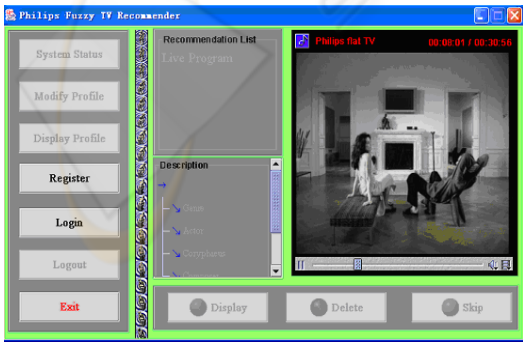

Figure 5: Main interface.

REFERENCES

Kaushal Kurapati, Srinivas Gutta, 2001. A Multi-Agent TV Recommender. *http://www.di.unito.it/~liliana/UM01/kurapati.pdf*

Belkin, N.J. & Croft, W.B., 1994. Information filtering and information retrieval: two sides of the same coin. *Communications of the ACM,1994*

Foltz, P. W. and Dumais, S. T., 1992. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM, 1992*

Ardissono, L. and Buczak, 2002. A. Personalisation in future TV". *Proceedings of TV'02, the 2nd workshop on personalization in future TV.*

Masthoff, J., 2002. Modeling a group of television viewers, *Proceedings of the workshop Future TV, 2002*

Babuska, R., 1998. Fuzzy Modeling for Control". *International Series in Intelligent Technologies,* Kluwer Academic Publishers

TV-Anytime Forum, 2003. Metadata Specification S-3 Part A: Metadata Schemas, Version 1.3, *http://www,tv-anytime.org*