PERSONALIZED PRODUCT SELECTION IN INTERNET BUSINESS

Krishnamoorthy Srikumar
Infosys Technologies Ltd,
Hosur Road, Bangalore – 560100. India.
srikumar_k@infosys.com

Bharat Bhasker
Indian Institute of Management,
Off Sitapur Road, Lucknow – 226013. India.
bhasker@iiml.ac.in

ABSTRACT

Traditional product selection methods – especially for high involvement products like refrigerators, cars and diamonds – use customer specified multi-attributes of the product to select products of interest to the customer. However, such methods tend to generate lot of false positives and false negatives due to conflicting, imprecise and non-commensurable nature of product attributes. In this paper, we present a novel methodology for product selection in Internet business to effectively handle the nebulous nature of product attributes. The system enhances the customer desired product attributes by utilizing his/her past profile, which is built by using his/her past purchases in the related product category. The suggested system offers the product variants as recommendation in a ranked order with customization to individual user's needs. We experimentally evaluate the system on a real-life dataset in order to assess its potential usefulness. The methodology discussed here can be useful for consumers in making a better choice of the final product. In addition, it can also be useful for e-commerce managers in providing personalized services to their customers.

Keywords: Product Selection, E-commerce, Recommender Systems, Customer Profiling, Clustering

1. Introduction

The proliferation of Internet provides a wide variety of choices for the customer in selecting a product that meet his/her desires. However, innumerable number of choices available on the web overwhelms customers and they often find it difficult to make a final choice of the product. Recommender systems address this information overload problem by offering products of interest to the customers. The most common example of a recommender system is that of Amazon (www.amazon.com) that provide wide variety of recommendation services based on customers' past purchases and/or tastes and preferences.

We can broadly classify the recommender systems based on the type of product for which recommendations are offered as (a) Low Involvement Products (LIP) such as soaps, books, and hair care products, and (b) High Involvement Products (HIP) such as refrigerators, diamonds, and cars. Vakratsas et al, 1999 provide a definition for Low and High involvement products in the marketing literature. They provide do-feel-think model for low involvement products and think-feel-do model for high involvement products. That is, in a do-feel-think model, a customer first purchases the product on impulse, then feels about the product upon usage and finally develops an attitude about the product by thinking. Similarly, for a high involvement product, a customer first thinks and learns about the product, then develops a feeling for the product and finally makes a decision to purchase.

1.1 Review of Literature

In the case of low involvement products, the click-to-buy rates are generally higher compared to that of high involvement products. Consequently, the recommendations for low involvement products are offered with the help of customer's past purchases, demographic details, explicitly specified interests etc. Collaborative Filtering (CF) is one of the most popular techniques for recommendation generation for such products. The collaborative filtering technique essentially matches the target customer's tastes and preferences with that of all other customers to identify like-minded customers, and then offer the products purchased by them as recommendations. Many of the commercial retail websites (such as Amazon) use CF to effectively offer cross-category recommendations. That is, recommendations such as "customers who bought soaps also bought toothpastes" are generated for cross-selling. There are variety of recommender system available in the literature that use such concepts and the more recent ones are Cho et al, 2002; Lee et al, 2002; and Srikumar & Bhasker, 2004. We do not deliberate in detail on such methods, as the focus of this paper is on efficient product selection (or recommendations) in the case of high involvement products such as refrigerators, cars, and computers.

Traditionally, the systems for product selection in the case of high involvement products take as input, a set customer specified product attributes, and matches them with a set of available products in the database to generate a ranking of products most likely to be of interest to the customer. For instance, shopping assistants available on the web (such as Dealtime, www.dealtime.com; Shopping, www.shopping.com; E-pinions, www.epinions.com) provide an agent based shopping support for customers. These agents take price and a set of product features as inputs, and match them with available products on the Internet to select set of products of interest to the customer. These agents also provide services such as product ratings, customer reviews, price comparisons and details of product availability across stores. However, selecting suitable products in the vast Internet is a challenging problem [Mohanty & Bhasker, forthcoming]. Other shopping agents available on the web such as Active Buyers' Guide (www.activebuyersguide.com) takes into account the importance of product features in addition to the feature itself to select products of interest to the customers. In essence, the shopping assistants or agents available on the web use a set of customer-desired features, and match the same with the available products on the web to select a set of product for recommendations. The recommendations generated in such systems are generally product variants rather than cross-category products as in CF based methods. The basic presumption is that the chance of a customer looking for a Refrigerator to buy a Washing Machine or any other high involvement product is negligible, as customers often follow the think-feel-do model [Vakratsas et al, 1999] while choosing high involvement products.

The works in the literature on product selection problem include [Ryu, 1999; Lee & Widmeyer, 1986; Lee et al, 2002; Mohanty & Bhasker, forthcoming; Saward & O'Dell, 2000; Yager, 2003]. Ryu, 1999 presents a methodology for construction of dynamic taxonomy hierarchy based on customer specified attributes. The system searches for products that satisfy customer specified attributes on the Internet. If the products are available, then it is presented to the customer. Otherwise, i.e. in product shortage situations, the next best alternative product is provided to the customer. Lee & Widmeyer, 1986 suggest another methodology for efficient product selection on the Internet. However, it uses a static product hierarchy for product classification unlike Ryu's approach.

Saward & O'Dell, 2000 provides a Case-Based Reasoning (CBR) approach to efficient product selection. As per this approach, every product in the database is represented as a case consisting of set of features and attributes. The customer's requirement is also captured using the same feature vector used for case representation. The closeness of the customer requirement to the product case (in the database) is assessed using nearest neighbour similarity metric. Then the products with higher similarity scores are offered as recommendation to the customer. They discuss variety of CBR methodologies and their implications for product selection. They also suggested an iterative relaxation CBR cycle to effectively capture the customer requirements.

Lee et al, 2002 presents a set of recommender system for products that are purchased frequently as well infrequently. Their system for less frequently purchased products involves the utilization of customer's ephemeral needs for retrieving optimal products. An iterative procedure is being suggested to improve the product selection performance, in terms of offering the best available product to the customer.

Yager, 2003 provide a fuzzy logic based methodology for constructing recommender systems. This approach utilizes a single individual's preferences for generating recommendations as opposed to CF based methods that uses preferences of collaborators. The suggested method is called as reclusive methods. The author has concluded that reclusive methods can act as complementary to CF based methods. Yager, 2003 has also pointed out that optimal recommender systems should be based on a combination of collaborative and reclusive methods.

Recently, Mohanty & Bhasker, introduced a methodology for efficient product classification using the concepts of fuzzy logic [Dubois & Prade, 1981; Dubois & Prade, 1978; Mohanty, 1998]. Their system use the basic concepts of Ryu's approach and builds upon it to include the linguistic quantifiers [Kacprzyk & Yager, 1984] in order to enhance the customer satisfaction levels.

Closely related to product selection problem is the multi-criteria decision analysis problem [Lee et al, 2001]. Works in this area include SMART [Edwards, 1977], SWING [von Winterfeldt and Edwards, 1986] and AHP [Saaty, 1980]. These methods use utility theory to evaluate a set of alternatives.

1.2 Problem Statement & Contributions

We consider the major shortcomings of all of the above approaches are that they do not take into consideration the customer's profile to personalize the recommendations. By customer profile, we mean the use of customer's past purchase histories in the related product category. The related product used for building customer profile may either belong to low or high involvement products. However, it is to be noted that low involvement product purchases (typically having a higher click-to-buy rates) provide rich source of data for constructing customer profiles, even though high involvement product purchases have very low click-to-buy rates. So, we investigate the use of customer profiles, which are built using the customer's past purchases in the related product category to improve the performance of product selection for high involvement products.

The suggested system is distinct from conventional Collaborative Filtering (CF) based recommender systems in the following ways: (1) The focus of our system is on generating recommendations for high involvement products where customer's involvement is higher. (2) It uses customer specified multi-attributes of the product in addition to their related product purchases. On the other hand, CF based methods utilize customer's general tastes, preferences and/or past purchases. (3) Product variants are offered as recommendations in a ranked order rather than cross-category/other related product recommendations as in CF based methods.

The major contributions of this paper include (a) introduction of a novel methodology for product selection with the help of customer profiles, which are built using customer's past purchases in the related product category, (b) experimental evaluation of the system on a real-life data and demonstration of the potential usefulness of the suggested approach.

The rest of the paper is organized as follows: In section 2, we describe the definitions, notations and assumptions used in this paper. We have deliberated on our novel methodology in section 3. A step-by-step illustration is also provided in section 3 for better understanding of the methodology. Section 4 is devoted to some of the experimentation carried out on the suggested methodology. Finally, we conclude with a summary and directions for future research work in section 5.

2. Definitions, Notations And Assumptions

In this paper, the domain of analysis is assumed to be Internet Retailing. Let us also assume that we are given customer's product purchase history of High Involvement Products (HIP) and their related products.

We represent the products (HIP) in the database on a set of features, that is, $\boldsymbol{P}=\{\boldsymbol{F_1, F_2 \ldots F_m}\}$, where $\boldsymbol{m}$ is the number of features required to represent a product. To capture the large number of feature variants of products, we map the products in feature space on to attribute space, $\boldsymbol{A = \{a_1, a_2 \ldots a_n\}}$, where $\boldsymbol{n}$ ($\geq \boldsymbol{m}$) is the total number of attributes. That is, the product $\boldsymbol{P_i}=\{\boldsymbol{a_{i1}, a_{i2} \ldots a_{in}}\}$, where $\boldsymbol{a_{ij}}$ is the value of $\boldsymbol{j}^{th}$ attribute for product $\boldsymbol{P_i}$. If the product $\boldsymbol{P_i}$ possess the attribute $\boldsymbol{a_{ij}}$, then $\boldsymbol{a_{ij}}$=1 otherwise, zero. For better clarity of product representation in terms of features and attributes, refer to Table 1. The product database in Table 1 has 5 features viz. Make ($\boldsymbol{F_1}$), Capacity ($\boldsymbol{F_2}$), Energy Consumption ($\boldsymbol{F_3}$), Type ($\boldsymbol{F_4}$) and Door opening ($\boldsymbol{F_5}$). Each of these features can take on multiple attribute values. For example, the feature $\boldsymbol{F_1}$ has three attributes Samsung ($\boldsymbol{a_1}$), LG ($\boldsymbol{a_2}$) and Videocon ($\boldsymbol{a_3}$). Similarly, feature $\boldsymbol{F_2}$ has two attributes viz. 350 ltr ($\boldsymbol{a_3}$) & 200 ltr ($\boldsymbol{a_4}$); feature $\boldsymbol{F_3}$ has three attributes viz. Low ($\boldsymbol{a_6}$), Medium ($\boldsymbol{a_7}$) & High ($\boldsymbol{a_8}$); feature $\boldsymbol{F_4}$ has two attributes viz. Top freezer ($\boldsymbol{a_9}$) & Bottom freezer ($\boldsymbol{a_{10}}$) and feature $\boldsymbol{F_5}$ has two attributes viz. Left ($\boldsymbol{a_{11}}$) and Right ($\boldsymbol{a_{12}}$). So, total numbers of product variants that are possible are 72 (3x2x2x2x2). These 72 products are represented in the product database as shown in Table 1. The products are represented with twelve attribute values. For instance, product $\boldsymbol{P_1}$ in Table 1 implies, a refrigerator with Samsung Make, having a capacity of 350ltrs, energy consumption of low range, refrigerator type of top freezer and left door opening.

Table 1: Product Database for Refrigerator

Product ID	**Make (F_1)**			**Capacity (F_2)**		**Energy Consumption (F_3)**			**Type (F_4)**		**Door opening (F_5)**	
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}
P_1	1	0	0	1	0	1	0	0	1	0	1	0
P_2	1	0	0	1	0	0	1	0	1	0	0	1
P_3	1	0	0	1	0	0	0	1	0	1	1	0
P_4	1	0	0	0	1	1	0	0	0	1	0	1
P_5	1	0	0	0	1	0	1	0	1	0	1	0
P_6	1	0	0	0	1	0	0	1	1	0	0	1
…												
…												
P_{72}	0	0	1	0	1	0	0	1	0	1	0	1

The target customer and target product are referred to as ***TgtC*** and ***TgtP*** respectively. The target product (***TgtP***) is the intentional requirement communicated by the customer during his/her visit to the site. A customer communicates his intentional requirements in the form of multiple attributes. We represent, $\boldsymbol{TgtP} = \{\boldsymbol{x_1, x_2 \ldots . x_n}\}$

where $\boldsymbol{x_i} \in \boldsymbol{A}$. That is, $\boldsymbol{x_i}$'s are attribute values of product and it can take on the value of either 0 or 1 (depending upon whether the customer prefers a particular attribute or not). The target customer (***TgtC***) is captured using the login information provided by the customer.

When a customer logs into the site, the system profiles him and generates a set of nearest neighbors, referred as ***TgtC_NN***. We represent, ***TgtC_NN*** = {$\boldsymbol{SU}_\mathrm{i}$, ∀i = 1 to ***SimU***}, where $\boldsymbol{SU}_\mathrm{i}$'s are the similar users and ***SimU*** is the number of similar users. The system generates nearest neighbors based on the customer's related product purchases.

Using the nearest neighbors computed (i.e. ***TgtC_NN***) and their purchases in the high involvement product category, the system generates a profile, referred as ***TgtProfile***, for the target customer. We represent ***TgtProfile*** = {$\boldsymbol{y_1, y_2 \ldots y_n}$}, where $\boldsymbol{y_i} \in \boldsymbol{A}$. $\boldsymbol{y_i}$'s take on the value of either 0 or 1 (based on whether the attribute is of interest to the customer or not). For each of the attributes of interest identified, a weight is also derived by the system. The computed weights are denoted as, $\boldsymbol{w = \{w_1, w_2 \ldots w_n\}}$, where $0 \le w_i \le 1$. The details of computation of profile and its weights are deferred till section 3. Subsequently, the customer's intentional requirements (i.e. ***TgtP***) are matched with that of the profile generated by the system (i.e. ***TgtProfile***). If both of them are not closer, then the system iteratively gets user inputs to capture the customer's real requirements. The closeness of ***TgtP*** and ***TgtProfile*** is measured in terms of the number of common requested attributes. If the number of common attributes is more than 60% (say), then ***TgtP*** is considered as closer to ***TgtProfile***, otherwise not. The presumption is that, if the generated profile is not closer to ***TgtP***, then the system is unable to capture the target customer's profile based on his/her related product purchases. In such cases, the customer is explicitly asked to evaluate the profile generated and verify his requirements in order to generate better recommendations.

The target profile built by the system is matched with all other products in the database to identify a set of similar products. For identifying similar products, we use similarity metrics (denoted as ***SimMetric***) such as tanimoto (also referred as jaccard) coefficient. Although there are varieties of similarity metrics available in the literature, we use tanimoto coefficient since there is no work – to the best of our knowledge – that suggest a single similarity metric as superior. However, Strehl et al, 2000 have observed that tanimoto coefficient and cosine metric performs better compared to other similarity metrics studied. Furthermore, Haveliwala et al, 2002 claims that tanimoto coefficient outperforms all other measures in identifying similar words in documents.

In this paper, we incorporate weights for each of the attributes in the tanimoto coefficient and define modified tanimoto coefficient as,

$$Sim(TgtProfile, P_i) = \frac{\sum_{j=1}^{n} w_j . y_j . a_{ij}}{\sum_{j=1}^{n} w_j a_{ij} + \sum_{j=1}^{n} w_j y_j + \sum_{j=1}^{n} w_j . y_j . a_{ij}}$$

The similarity threshold value used for identification of similar products is denoted as ***Sim_threshold***. We conduct experiments with different ***Sim_threshold*** values in section 4 to assess the performance of the system.

The number of products to be selected for recommendation is denoted as ***N***. Also, note that the products offered as recommendation are product variants within a product category rather than cross-category products.

3. Our Methodology

First, we provide an agent based architecture for product recommendation and describe its various elements in detail. Second, a recommendation methodology is suggested and a step-by-step illustration is also provided to enlighten the procedure.

3.1 Architectural Elements

Agent based architecture for the suggested system is provided in Figure 1. We describe various architectural elements of the system in this section.

Interface Management Agent

A customer communicates his intentional requirements with the help of user interface provided by this agent. This agent primarily identifies the target customer (based on login information), i.e. ***TgtC***, and the target product, i.e. ***TgtP***. The additional role of this agent is to update the product feedbacks provided by the user. The product feedback helps in avoiding the less useful product being recommended during the customer's subsequent visit to the site.

Web Log Analysis Agent

This agent periodically monitors the web log and extracts the customer purchase information for both HIP and its related products. The extracted data are preprocessed and stored in the customer database. However, in this paper, we presume that the customer database is available.

Data Transformation Agent

This agent extracts data from web log analysis agent and transforms the data suitable for Data Mining Agent. In this case, related purchase data are organized on customer identifiers and passed to the Data Mining Agent when requested.

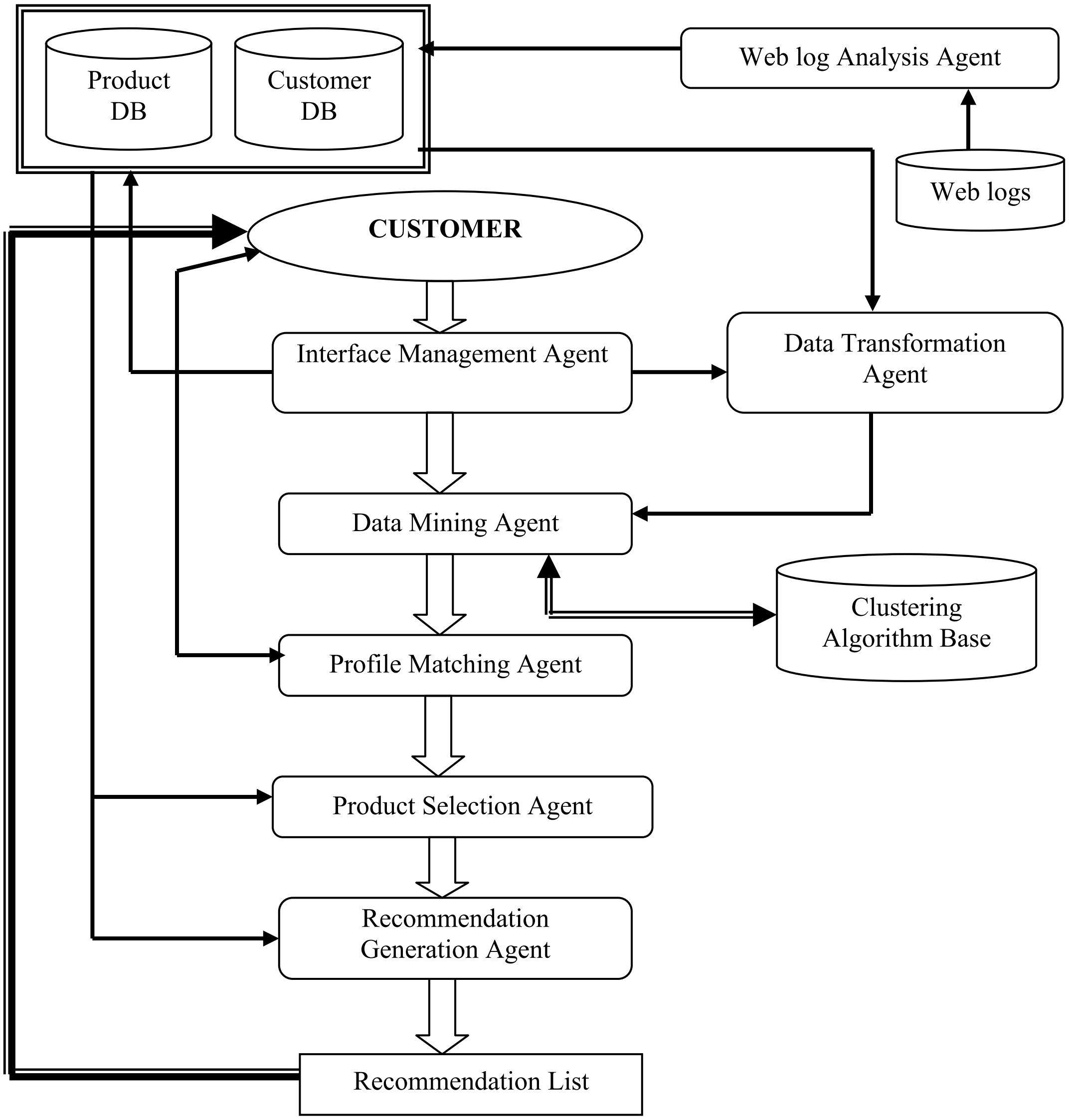


Figure 1: Agent Based Architecture

Data Mining Agent

Data Mining Agent gets its inputs from Interface Management Agent and Data Transformation Agent. This agent clusters customers in the database using past purchase histories of related products. Subsequently, the set of customers who belong to the target customer's cluster and the product purchased by them in HIP are selected. The products are assigned scores based on the similarity values derived in the clustering step. Finally, this agent extracts a target profile, ***TgtProfile***, from the list of products selected above. The weights, ***w,*** are assigned to the attributes based on the product scores. The actual computational details of weights are explained in section 3.2.1 on illustration.

Profile Matching Agent

This agent compares the ***TgtProfile*** (generated by the system) and ***TgtP*** (communicated by the customer) and assesses the number of common requested attributes. If the number of common attributes is more than 60% (this percentage can be chosen flexibly), then product selection agent is called and recommendations are generated.

Otherwise, the customer is alerted to verify the profile generated by the system and requested to modify the profile or its weights to generate better recommendations. The resulting profile can be stored in the system for future use (i.e. for customer's subsequent visits).

```
PS_Agent( )

        Inputs: TgtProfile, SimMetric, Sim_threshold, Product database, w

1.  For each $P_i$ in the database
2.      Identify the similarity between $P_i$ and TgtProfile using the similarity
            metric (such as modified tanimoto coefficient)
3.  Sort the products by their similarity values in descending order
4.  Extract set of products whose similarity values are above Sim_threshold
5.  Call RG_Agent( )
```

Figure 2: Pseudo-code for Product Selection Agent

Product Selection Agent

This agent takes as input the ***TgtProfile*** and attribute weights generated by the data-mining agent. The ***TgtProfile*** is matched with set of other products in the database to determine a set of similar products. The match is performed using similarity metric such as modified tanimoto coefficient (refer to Figure 2 for pseudo-codes).

Recommendation Generation Agent

The set of products generated by PS_Agent (refer to Figure 2) are sorted in descending order based on their similarity to the ***TgtProfile***. The system then offers Top-***N*** products as recommendation (refer to Figure 3 for pseudo-codes).

So far, a complete description of each of the architectural elements of our methodology is outlined. Now, we present the overall recommendation strategy of our system.

```
RG_Agent( )

        Inputs: Set of products generated by PS_Agent( ), N

1.  Rank the products based on the similarity values derived by PS_Agent while
    identifying similar products
2.  Return <Top-N Products as Recommendations>
```

Figure 3: Pseudo-code for Recommendation Generation Agent

3.2 Recommendation Strategy

The proposed system selects products for recommendation as follows:

1. First, the user login information is captured and the target customer, ***TgtC***, is identified. The user then communicates his intentional requirements (in the form of set of attributes of interest). The user's requirements are captured as the target product, ***TgtP***.
2. Cluster customers based on past history (i.e. related product purchase history) to identify the target customer's nearest neighbors, ***TgtC_NN***.

3. Identify HIP products purchased by the customers identified in step 2 above and score them based on similarity values computed in the clustering step.
4. From the products identified, build the customer profile, ***TgtProfile***. Assign weights to attributes based on the product scores generated in step 3 above.
5. Match ***TgtProfile*** with ***TgtP*** and assess its closeness. If ***TgtProfile*** is not closer to ***TgtP***, get more information from customer to generate better recommendations. The customer profile is also updated in the database for future use.
6. For the identified target profile, ***TgtProfile***, set of similar products is extracted with the help of product selection agent (refer to Figure 2).
7. The resulting products are ranked and Top-***N*** products are offered as recommendation (refer to Figure 3).

3.2.1 An Illustration

Let us consider the case of refrigerators as high involvement product as they are purchased less frequently. Table 1 gives the product feature/attribute space for refrigerators. As per Table 1, refrigerator has five features viz. Make ($\boldsymbol{F_1}$), Capacity ($\boldsymbol{F_2}$) , Energy Consumption ($\boldsymbol{F_3}$), Refrigerator Type ($\boldsymbol{F_4}$) and Door opening ($\boldsymbol{F_5}$). Each of these features has set of attributes $\boldsymbol{a_1}$ to $\boldsymbol{a_{12}}$ as described in section 2.

Now let us say a customer logs in to the site and gives his intentional requirement as follows:

Make: Not specified
Capacity: 350ltrs
Energy Consumption: High
Refrigerator Type: Not specified
Door opening: Not specified

So, using the product database representation, ***TgtP*** = {0,0,0,1,0,0,0,1,0,0,0,0}. Now, we cluster the set of customers (in the database) based on their purchase history for related products. The target customer's cluster and their set of nearest neighbors are then identified. Figure 4 gives a set of sample cluster segments. The segment to which the target customer belongs is also marked as a star symbol in Figure 4. The high involvement products purchased by customers in target customers' cluster are then selected. Subsequently, a target profile is built based on the products selected and their attributes. The details of computation of profile and its weights can be referred in

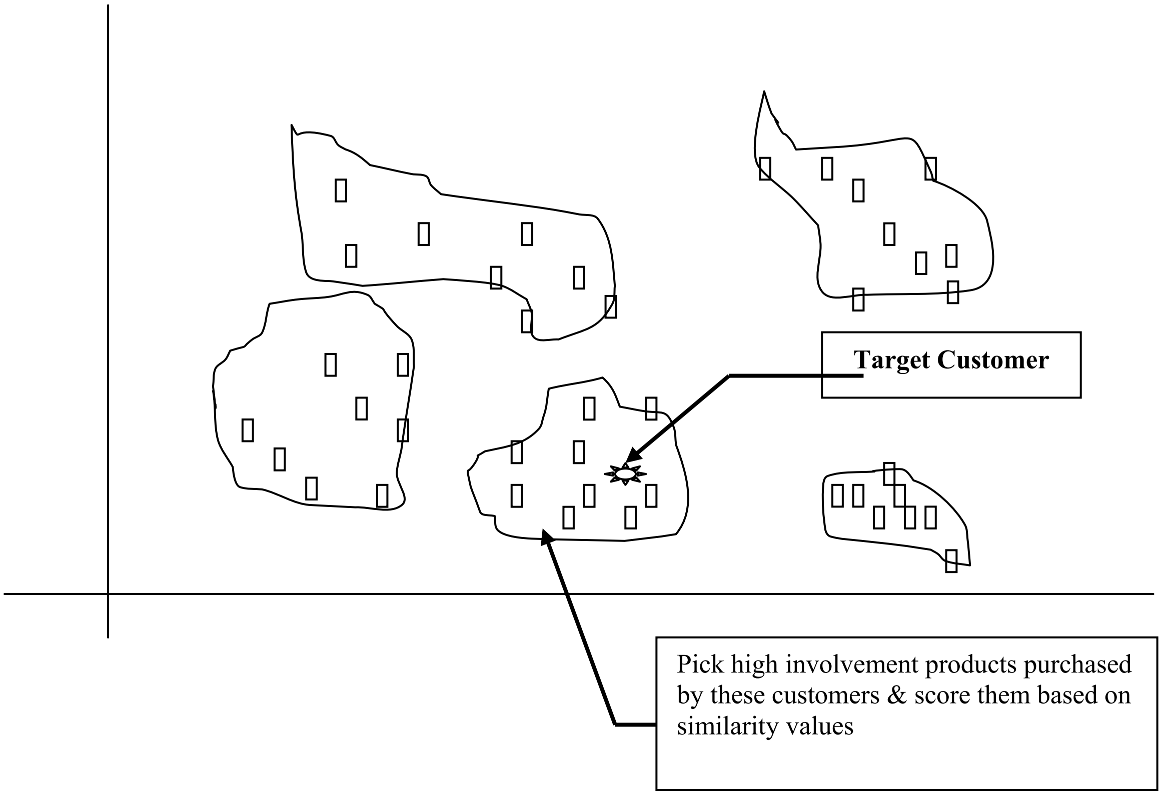
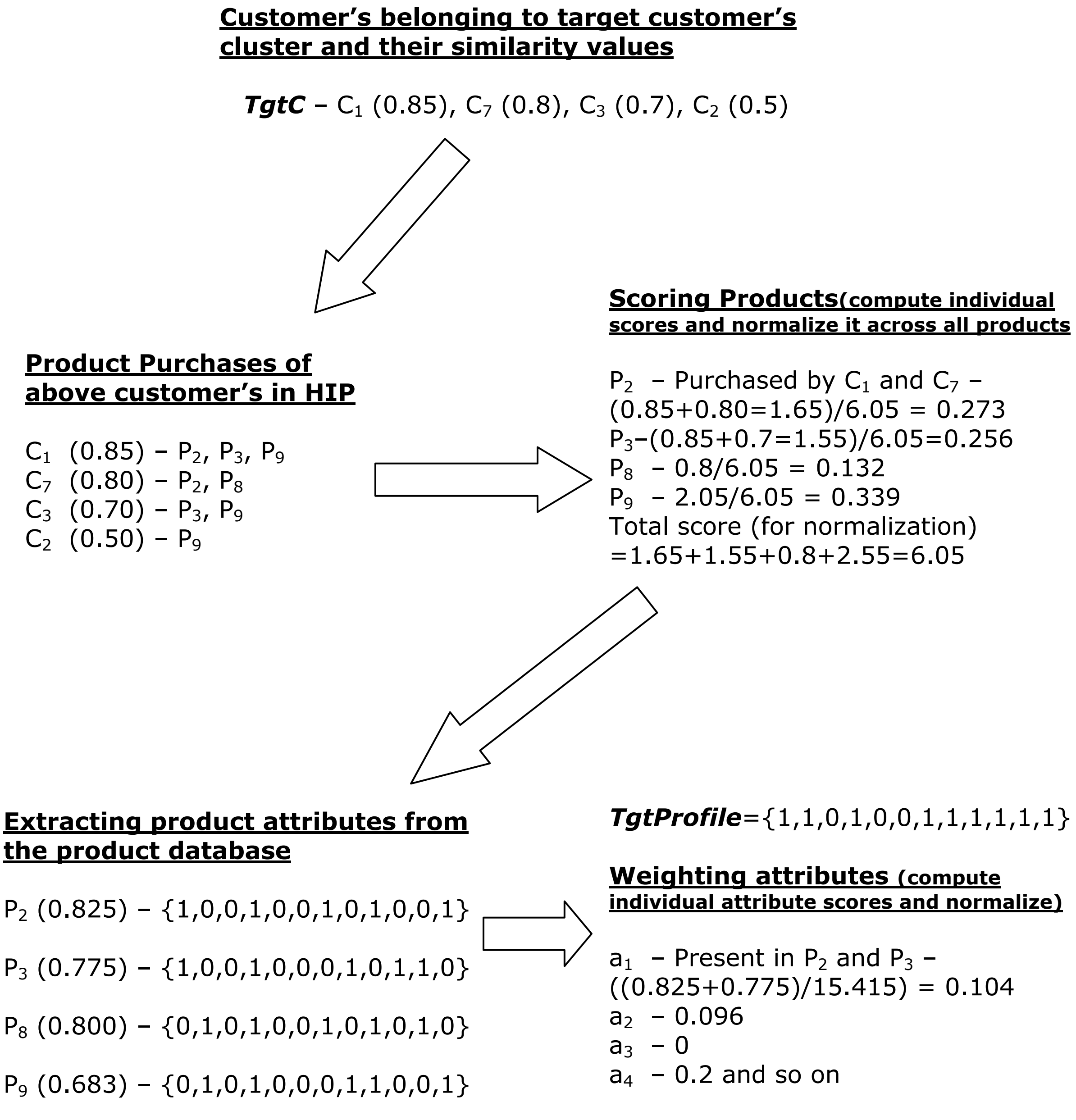

Target Customer
Pick high involvement products purchased by these customers & score them based on similarity values

Figure 4: Customer Segments based on customer's related product purchase history

Figure 5. That is, ***TgtProfile*** = {1,1,0,1,0,0,1,1,1,1,1,1} with the following weights, ***w*** = {0.104,0.096,0,0.2,0,0,0.105,0.095,0.150,0.05,0.102,0.098}. The customer has initially specified his requirement ***TgtP*** as {0,0,0,1,0,0,0,1,0,0,0,0}. But, our system tries to enhance the customer's requirement by using his past purchases in the related product category and derive a ***TgtProfile*** with weights for each of the attributes. In this illustration, ***TgtP*** has attributes, $\boldsymbol{a_4}$ and $\boldsymbol{a_8}$ value as 1. Similarly, the target profile derived also has $\boldsymbol{a_4}$ and $\boldsymbol{a_8}$ value as 1 (in addition to other attributes). This signifies that percentage of common requested attributes between ***TgtP*** and ***TgtProfile*** is 100%. If this value falls below 60% (say), then we can provide the user with the generated profile as well as its weights and request him to revise his/her requirements.

Now, the product selection agent matches the target profile with set of other products in the database. Similar products are identified by using tanimoto coefficient discussed in section 2. More specifically, ***TgtProfile*** is matched with each product in the database (i.e. 72 products in Table 1) and similarity values are computed using tanimoto



Customer's belonging to target customer's cluster and their similarity values

TgtC – C_1 (0.85), C_7 (0.8), C_3 (0.7), C_2 (0.5)

Product Purchases of above customer's in HIP

C_1 (0.85) – P_2, P_3, P_9
C_7 (0.80) – P_2, P_8
C_3 (0.70) – P_3, P_9
C_2 (0.50) – P_9

Scoring Products(compute individual scores and normalize it across all products

P_2 – Purchased by C_1 and C_7 – (0.85+0.80=1.65)/6.05 = 0.273
P_3–(0.85+0.7=1.55)/6.05=0.256
P_8 – 0.8/6.05 = 0.132
P_9 – 2.05/6.05 = 0.339
Total score (for normalization) =1.65+1.55+0.8+2.55=6.05

Extracting product attributes from the product database

P_2 (0.825) – {1,0,0,1,0,0,1,0,1,0,0,1}

P_3 (0.775) – {1,0,0,1,0,0,0,1,0,1,1,0}

P_8 (0.800) – {0,1,0,1,0,0,1,0,1,0,1,0}

P_9 (0.683) – {0,1,0,1,0,0,0,1,1,0,0,1}

TgtProfile={1,1,0,1,0,0,1,1,1,1,1,1}

Weighting attributes (compute individual attribute scores and normalize)

a_1 – Present in P_2 and P_3 – ((0.825+0.775)/15.415) = 0.104
a_2 – 0.096
a_3 – 0
a_4 – 0.2 and so on

Figure 5: Sample computation of TgtProfile and Weights

coefficient. The products are then offered in ranked order of their similarity scores. If ***N*** is set as 4, then four products are generated as recommendation. For example, let us say the four products and their weights generated are: P_3 (0.276), P_2 (0.274), P_9 (0.25) and P_6 (0.24). Then, we can infer that the customer who specified his initial requirements as 350ltrs capacity and High energy consumption will have more liking for product P_3 made of Samsung Make, 350ltr capacity, High Energy Consumption, Bottom freezer type and Left door opening followed by product P_2 made of Samsung Make, 350ltrs capacity, Medium Energy Consumption, Top freezer type and Right door opening and so on. Hence, our system tries to enhance the customer's intentional requirement (that are incomplete) by building customer profile based on their past purchases in the related product category.

So, our methodology is configured to build customer profile and offer personalized recommendations.

4. Experimental Results

The complete agent based architecture of our system discussed in section 3 is built using C++ on a 450MHz Pentium-III PC with 64MB of memory running Red Hat Linux 7.2. We gathered a real-life data from one of the leading online retailers in India for the experimental evaluation of the suggested system.

4.1 Experimental Design and Metrics

For the experimentation, we chose Earrings (a product under the category of Jewellery) as high involvement product. Our motivation for choosing this product is due to the fact that customer's typically follow a think-feel-do model [Vakratsas et al, 1999] while purchasing a Earring. The characteristics of this product category are described in Table 2. This product has 7 features viz. stone, seller, weight, number of stones, carat, features and price. The features that have continuous values (e.g. weight, carat etc) are discretized with the help of equal frequency method [Dougherty et al, 1995]. Thus the resulting number of attributes for the product, Earring, were 27 (refer to Table 2).

We selected the HIP (i.e. Earrings) in such a way that customers who have purchased products under this category also purchased a few products in the related product category. The related products chosen belong to categories of other jewellery items viz. Necklace, Pendant, Ring, Bracelet, Nosepin etc. The total number of related products chosen were 128. The resulting HIP dataset had 47 products (with 7 features and 27 attributes as in table 2) and 126 customers. That is, 126 customers who had purchased an Earring also had purchased some other items in the related product category.

Two datasets are prepared for analysis (1) HIP dataset having customer purchases in the HIP category of Earrings. This dataset had 126 customers, 47 products and 159 product purchases. For every customer, their purchases in the HIP are stored in the HIP dataset. (2) Related product dataset having customer purchases in the related product category. This dataset also has the same 126 customer's (i.e. 126 customers who had purchased products in HIP) purchases but purchases are on their related products. Total number of product purchases in related product dataset were 256. These two datasets form the customer database (refer to the architecture in Figure 1).

Table 2: Feature and Attribute Space for Earrings

S.No.	Features	Number of Attributes
1	Stone	3 (e.g. diamond, silver etc)
2	Seller	4 (e.g. surat, siddhi, sparkles etc.)
3	Weight	3 (e.g. low, medium etc)
4	Number of Stones	3 (e.g. low, medium etc)
5	Carat	3 (e.g. <0.01gms, 0.01-0.13gms etc)
6	Features	6 (e.g.drop type, stud, designer etc)
7	Price	5 (e.g. very low, low, med, etc)

The product database comprises of product representation in terms of attributes as in Table 1 for Refrigerators. In this case, the product is Earring represented on a set of 27 attributes given in Table 2.

The qualities of recommendations are evaluated by dividing the dataset (both HIP dataset and related product dataset) randomly into training and test sets (80% training and 20% test). In our case, the training data had 101 instances (i.e. 126 * 0.8) and test data had 25 instances. Recommendations are generated by the system using training data. The remaining data, which is the test data, is divided vertically into two parts, i.e. each test data has two parts. The first part contains one product (referred as input item), and the second part contains all-but-1 products (referred as hidden items). The first part (taken from HIP dataset) is used as user input for generating recommendations. If the generated recommendation contain at least one of the items present in the hidden items (taken from HIP dataset), it is regarded as a hit. The recommendation quality is evaluated on information retrieval metric – widely used in the literature [Deshpande & Karypis, 2004; Srikumar & Bhasker, 2004] – such as recall. The recall is expressed as follows: If x is the total number of test cases, H is the total number of hits, then recall = H/x.

4.2 Performance Results

Table 3 give the performance results for our system. The results are given for different parameter conditions, i.e. changes in similarity threshold values and number of items recommended. A cursory look at table 3 reveal that, decrease in similarity threshold values result in improved recall values. However, using very low threshold values would result in generating recommendations that are less useful for the user. Secondly, reducing the total number of recommendations generated results in reduction in recall values. This is quite expected, as some of the useful recommendations may not get generated when we decrease the value of ***N***.

Table 3: Performance Results

Sim_threshold	***N***	**Recall (values in %)**	**Average number of recommendations generated**
0.25	4	40	4
0.25	6	52	6
0.25	8	60	8
0.25	10	64	9.96
0.25	20	88	15.32
0.20	20	92	19.60
0.20	30	92	21.48

We also made a performance comparison of our approach with standard product selection based approach. In the standard product selection approach, the user's input is taken and matched with set of other products in the database to identify a set of similar products. The similarity between products are assessed with the help of tanimoto coefficient discussed in section 2.

Table 4: Performance Comparisons

System	**Recall (values in %)**	**AvgN***	**Parameters**
Proposed system	60	8	***N*** =8; ***Sim_threshold***=0.25;
	64	9.96	***N*** =10; ***Sim_threshold***=0.25;
	64	10	***N*** =10; ***Sim_threshold***=0.2;
	88	15.32	***N*** =20; ***Sim_threshold***=0.25;
	92	19.6	***N*** =20; ***Sim_threshold***=0.2;
	92	21.48	***N*** =30; ***Sim_threshold***=0.2;
Pure Product Selection	52	7.04	***N*** =8; ***Sim_threshold***=0.25;
	56	8.12	***N*** =10; ***Sim_threshold***=0.25;
	64	9.96	***N*** =10; ***Sim_threshold***=0.2;
	60	9.04	***N*** =20; ***Sim_threshold***=0.25;
	88	15.32	***N*** =20; ***Sim_threshold***=0.2;
	88	15.36	***N*** =30; ***Sim_threshold***=0.2;

* **AvgN** – Average number of recommendations generated

The results of performance comparison of our approach with that of standard product selection approach are provided in Table 4. Some of the key trends that can be observed from the results are as follows:

Firstly, for each of the parameter conditions studied, the proposed system shows a marginal improvement over pure product selection approach (and in case 3, i.e. ***N***=10, ***Sim_threshold***=0.2, the results were same, i.e. Recall=64%). Furthermore, average number of recommendations generated was found to be higher for our methodology in all the cases studied. So, more experiments on other datasets are needed to clearly understand the performance of the suggested approach. Secondly, increase in the value of similarity threshold resulted in reduced recall values, as expected, for both the systems studied.

4.3 Discussion

From the foregoing analysis, it is evident that variations in similarity threshold values and total number of products offered for recommendation can affect the overall quality of recommendation (refer to Table 3). For instance, setting a very high value for similarity threshold may result in no recommendation being generated. On the other hand, when the similarity threshold values are much lower, the system may throw up large number of products that may not be of interest to the user. Similar phenomenon can occur in the case of selection of ***N*** value as well. So, a judicious choice of these values is of paramount importance for the system to generate useful recommendations.

We also showed experimental comparisons of our system against pure product selection approach under varying experimental conditions. However, the improvements observed were very marginal. So, more experiments (on other real-life datasets) are needed to substantiate that the suggested approach performs better compared to pure product selection approach.

A pure product selection system retrieve similar products by matching customer specified attributes with the products available in the database. But, the suggested system is distinct from such an approach in the following ways: (1) Our system cluster customers based on their related product purchases, (2) Unlike in other system, a customer need not specify precisely all the attributes he/she desires and also his preference for each of the attributes. Our system tries to identify the target profile and customer's preference for attributes (in terms of weights) that are of interest to the customer, (3) As the suggested methodology first identifies similar users and then perform similar product search, the system can be treated as a combination of collaborative and reclusive methods [Yager, 2003]. It is to be noted that Yager, 2003 has also pointed out that optimal recommender system should be based on both collaborative and reclusive methods.

The suggested methodology can be useful for e-commerce managers in providing personalized services to their customers. It also aids the customer in rejecting the products that may not be of interest to him and zero-in on the best product (the one that best satisfy the customer's desire).

As illustrated in the methodology, our system presumes that customers who wish to purchase a product in the HIP category, has purchased a set of products in the related product category, in the past. Although this assumption is reasonable, the customer need not purchase all products from the same on-retailer. That is, a customer may purchase products from multiple stores. Such a scenario would be a limitation (in terms of data availability) for the suggested methodology. However, even in such cases, the customer profiles can be captured with other set of features like demographic and psycho-graphic details of customers.

5. Summary and Future Research Directions

In this paper, we introduced a methodology for product selection in Internet business. The suggested approach is especially applicable for high involvement products such as refrigerators, cars etc. Our methodology handles the nebulous nature of customer product specification effectively by building customer profile based on customers past purchases in the related product category. We also experimentally evaluated and demonstrated the benefits of the suggested methodology on a real life dataset.

In future, it would be interesting to extend this work in many ways and some of the possible extensions are:

a) In addition to purchase histories in the related product category, it may be useful to consider other features such as customer demographics, psychographics etc. It would also attenuate the problem of lack of data for customers who have not purchased any product in the related product category.

b) The concepts of fuzzy logic can also be integrated with the approach to strengthen the procedure. More specifically, the concepts introduced by Yager, 2003 may be integrated with the suggested methodology.

c) For the current experimentation, we have used a set of products purchased by the customers in the related product category (irrespective of whether it belongs to LIP or HIP). However, it may be useful to investigate the effect of using related products in the LIP and HIP category and assess if there any differences. That is, while recommending products for the category of desktop computers (HIP), does the use of customer purchase history for the related HIP category of "laptop computers" produce more useful results than the use of the related LIP category of "cheap electronic goods"?

d) In this paper, not specified and not preferred attributes are treated equally. It may be useful to treat them differently in future work. For example, a value between 0 & 1 and zero can be given for not specified and not preferred attributes respectively.

e) Studying the system in real-time environments to assess its performance is another interesting and useful area of work.

REFERENCES

Cho, Y.H., J.K. Kim, and S.H. Kim, "A personalized recommendation system based on web usage mining and decision tree induction", *Expert Syst. With Applications*, Vol. 23, 329-342, 2002.

Deshpande, M., and G. Karypis, "Item-based Top-N Recommendation Algorithms", *ACM Trans. On Inf. Systems*, Vol. 22, No. 1, pp. 143-177, 2004.

Dougherty, J., R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features", *In Proc. Twelfth Int'l Conf. Machine Learning (ICML)*, pp.194–202, 1995.

Dubois, D., and H. Prade, "Addition of interactive fuzzy numbers", *IEEE Trans. Automat. Contr. AC*, Vol. 26, 926-936, 1981.

Dubois, D., and H. Prade, "Operations on fuzzy numbers", *Intl. J. Syst. Sci.,* Vol. 9, 613, 1978.

Edwards, W., "How to use multiattribute utility measurement for social decision making", *IEEE Trans. On Syst. Man Cybern.,* Vol. 7, 326-340, 1977.

Haveliwala, T.H., A. Gionis, D. Klein, P. Indyk, "Evaluating strategies for similarity search on the web", *In Proc. Of WWW Conference,* Honolulu, 2002.

Lee, R.M., and G.R. Widmeyer, "Shopping in the electronic market place", *J. Management Inf. Syst.,* Vol. 2, 21-35, 1986.

Lee, J., H.S. Lee, and P. Wang, "Analytical Product Selection Using a Highly-Dense Interface for Online Product Catalogs", *IBM Institute of Advanced Commerce Technical Report*, 2001.

Lee, W.-P., C.-H. Liu, and C.-C. Lu, "Intelligent agent-based systems for personalized recommendations in Internet Commerce", *Expert Syst. With Applications*, Vol. 22, 275-284, 2002.

Kacprzyk, J., and R.R. Yager, "Linguistic quantifiers and belief qualifications in fuzzy multi-criteria and multi-stage decision making", *Control Cybern.*, Vol. 13, 155-173, 1984.

Mohanty, B.K., and B. Bhasker, "Product classification in the Internet business – a fuzzy approach", *Decision Support Syst.* (forthcoming).

Mohanty, B.K., "Assigning weights to multiple criteria – a fuzzy approach", *J. Fuzzy Math.*, Vol. 6, 209-222, 1998.

Ryu, Y.U., "A hierarchical constraint satisfaction approach to product selection for electronic shopping support", *IEEE Trans. Syst. Man Cybern., Part A, Syst. Humans,* Vol. 29, 525-532, 1999.

Saaty, T.L., *The Analytical Hierarchy Process*, Mc-Graw Hill Inc., 1980.

Saward, G., and T. O'Dell, "Micro and macro applications of case-based reasoning to feature-based product selection", *Proc. of Conf. on Expert Syst.*, Cambridge, 2000.

Srikumar, K., and B. Bhasker, "Personalized recommendations in E-Commerce", *In Proc. of 5th World Congress on E-Business in 25th McMaster World Congress,* Canada, 2004.

Strehl, A., J. Ghosh, and R. Mooney, "Impact of Similarity Measures on Web Page Clustering", *Proc. of AAAI Workshop on Artificial Intelligence for Web Search*, 2000.

Vakratsas, D., and T. Ambler, "How Advertising Works: What Do We Really Know?", *Journal of Marketing*, Vol. 63, No. 1, 26-43, 1999.

Von Winterfeldt, D., and W. Edwards, Decision *Analysis and Behavioral Research*, Cambridge University Press, 1986.

Yager, R.R., "Fuzzy logic methods in recommender systems", *Fuzzy Sets and Systems*, Vol. 136, 133-149, 2003.