International Journal of Fuzzy System Applications
Volume 6 • Issue 2 • April-June 2017

Application of Fuzzy Logic for Adaptive Food Recommendation

Tousif Osman, North South University, Dhaka, Bangladesh

Maisha Mahjabeen, North South University, Dhaka, Bangladesh

Shahreen Shahjahan Psyche, North South University, Dhaka, Bangladesh

Afsana Imam Urmi, North South University, Dhaka, Bangladesh

J.M. Shafi Ferdous, North South University, Dhaka, Bangladesh

Rashedur M. Rahman, North South University, Dhaka, Bangladesh

ABSTRACT

The research introduces an adaptive food searching and recommending engine by taste and user preference using fuzzy logic. In contrast with existing system where food is searched by predefined keywords, this system searches food by its taste and users' preference which allows the system to provide better results. As food taste cannot be measured and user's preference is relative to each user, the authors have used concepts of artificial intelligence (AI) and fuzzy logic to better understand and deal the abstractness of these parameters. Along with food taste the authors have considered restaurant's environment, location, review and user's budget as searching parameters. The system includes a fuzzy database where food items of different restaurants with the specific parameters have been stored and gets updated by user feedback. System also maintains a user profile for individual user to adapt with individual user's choice of preference.

KEYWORDS

Adaptive Learning, Food Searching, Fuzzy Database, Fuzzy Logic, Searching, Soft Computing

1. INTRODUCTION

Since the dawn of technological boom, enormous amount of data is generated by different applications. The data being used in our day to day life could make our life simpler and easier. Today Amazon, E-bay, Facebook and many more popular tech giants alone have Exabyte of their own data and these data could be used in various ways. One of major issues of handling these big data is "searching". With the increment amount of data, we need more and more efficient solutions for searching. One of the widely used methods is searching by predefined keywords. When the searching is done on non-key attributes, this problem becomes more critical as there is no exact solution. Suppose, when a user is looking for someone in Facebook by their name, it applies some intelligence along with the keyword searching (user's name of interest) to present better result as there are thousands of users in

DOI: 10.4018/IJFSA.2017040106

that platform having same name over the world. Similar problem arises on other fields in the domain of searching. At present time most of the systems are focused on specific type of data. Due to the high volume of data, more precise and optimized algorithms are being developed for specific data types to provide results faster. However, in the domain of food searching and gaining intelligence over taste measurement, it is still not developed up to the mark as the key component of food is taste which is very abstract. Besides, taste has no exact scale of measurement. It usually depends on user's preference and it cannot be defined exactly. At present in our globalized and busy world, dining outdoors has become more regular act in our day to day life and hence the domain of searching food and taste has gained more importance. In this research we have modeled and constructed an adaptive searching engine to gain intelligence over abstractness of taste, user and other parameters and provide more accurate and better search results. We have constructed a fuzzy inference system (FIS) to solve this problem in the domain of fuzzy logic. Whenever a user searches food by our defined parameters (like taste, location, budget etc.), the engine looks up the fuzzy database according to each input parameter and makes a list of all food items that falls under certain threshold relative to the search query. Afterwards, the system assigns a fuzzy score to each food item using fuzzy inference system called fuzzy controller and returns the food list of preference in descending order sorted by the fuzzy score. Food item in higher order in the list means the item is more preferred for that specific user by the given search query. Another important feature of this system is that it adaptively puts higher weight to feedbacks given by more active users while searching. Moreover, the system scales itself according to users' preference when taking the feedbacks from users. There exists some excellent food searching systems like Yelp, Foursquare etc. but all of them use keywords and searches those keywords only but do not consider food taste and user preference. Hence our system can provide better and optimized results for each individual user. In this research we have built an adaptive food searching engine where food can be searched by taste, restaurant's environment, along with other conventional searching keys like location, user budget, and review by considering each user's preference. Users can search any food using different kind of tastes like "Something Spicy" or "Sweet and sour" etc. We have added two important features in our system which make the search results adapt with individual user and these two features have ability to better understand feedback by users' activity called "System Specific Adaptation" and scale the feedbacks by users' preference called "User Specific Adaptation".

2. RELATED WORK

There have been several researches conducted on searching, sorting and recommendation systems based on user preference. We have gained knowledge from those researches to build our system. In this section we will mention few of those research works that provide motivation in our research.

Ge, Elahi, Fernaández-Tobías, Ricci, & Massimo (2015) describe the essentiality and efficiency of using recommendation system using tags and latent factorials for sorting out the best solution in food searching. The research considers a menu card with proper details. This card has been considered as a tag. System uses these tags to search the suitable meal for a particular user. In this research, the system allows the users to provide tags and rating for various foods. When the data is being searched, system evaluates these tags and rating to find out the best choice. However, in this system, tags are pre-defined. In our system we have also considered different tags or meta-attributes. But in our system we have provided a generic food advice rather than tailored in user's point of view. We have gone for the approach where users will give their review and provide any kind of metadata about the food. Then these meta data will be used in future reference. In (Ramírez-García & García-Valdez, 2013), collaborative filtering and content based recommendation techniques have been used to build a hybrid fuzzy system for restaurants. The authors modeled the restaurants as vectors comprised of different fuzzy attributes. They calculated similarity between restaurants using cosine distance which is also known as cosine similarity. Here the content based method was based on a characteristic vector. This vector represents keywords for each item in the restaurant menus on which the users have given a high rating. While using collaborative filtering, the authors used previous behavior patterns or opinions of mass user group to suggest the item that the user will mostly prefer. In our research we have also used these two recommendation techniques together. In the previous research the authors have broken down the recommendation system into two parts which are expert recommendation and final recommendation. Here they have used 3 input variables to get their preferable fuzzy output where we have used 5 input variables to achieve more accuracy and user satisfaction. In this paper (Trevisiol, Chiarandini, & Baeza-Yates, 2014), the authors presented a recommendation system which prioritized the sentimental analysis to recommend a menu to the user. They have used User Generated Content (UGC) for the recommendation of menus and dishes to the users. To mine the information from the user review, the researchers have used Natural Language Processing (NLP). For the fuzzy system, they have used the extension of the Apriori algorithm. They also gave a weight against the dishes in the menu from negative to positive. For the recommendation part, the researchers used 3 different recommendation systems along with a simple base line system. Overall the research shows that the accuracy of recommendation system increases for the sentimental analysis of the reviews. In our paper we will also use user review for the enrichment of the database and increase the accuracy rate. Zhao, Wang, & Zhong (2013) proposed a fuzzy recommendation system based on customer online review. As we have implemented similar mechanism in our system this research work is a good source of knowledge to us. According to the paper, the system takes the user review as one of its input which is mostly in natural language. So after taking the inputs, the system processes it using POS (part of speech) and LTP (Language Technology Platform) and fuzzy inputs into the product database. Rules are included into the FIS (Fuzzy Inference System). Then the system calculates the approximate degree between the product list and the input given by the user. As a result, it will return a sorted list of products recommended for the user. In our paper we will also use a user review section, where user will give review about the existing taste. For several different reviews for a user, our system will change the scale of that particular taste.

Next paper that we are adapting in our research, is based on nutritional needs recommendation based on fuzzy logic by Priyono & Surendro, 2013. This paper, using fuzzy logic, calculates food nutrition and Body Mass Index (BMI that is a value calculated using height and weight) of a particular user and provides food recommendation. The authors used fuzzy logic, fuzzy inference model, calories assessment, food calories. They used Takagi Sugeno Kang(TSK) for daily calories need. They have also used other parameters like body weight, height, sex to provide suggestion. The authors determined the calorie of foods and it is different for same food in different conditions. For instance, food calorie is different for same chicken fry in different times. To solve this issue they have sorted ideal food values in the database and using fuzzy-logic and soft computing they have determined the food calorie. In our system the same problem arises as the parameters change for same food but in different situations and to solve we will be using similar approach.

In our next related paper (Sohail, Siddiqui, & Ali, 2015), the researchers suggested a book recommendation methodology which integrates fuzzy concepts with link mining. Initially the authors elected top 50 universities all over the world. They took the book list of one of their course for the initial research. Then they tabulated the books using Positional Aggregation Score (PAS) based technique. Next they assigned a score to each book using fuzzy quantifier. After that, they applied OWA (Ordered Weighted Averaging) aggregation operator over the initial list of the books. Finally, as a result they get a sorted list of recommended top books. The book that has the highest score was ranked first. Accordingly, the rest of the books were sorted. In our research, we have also used similar type of methods to score the list of menus and restaurants.

We found another paper (Nilashi, Ibrahim, Ithnin, & Zakaria, 2014) related to our work, which is Multi-criteria collaborative filtering (MC-CF) recommendation system that uses Neuro fuzzy techniques along with dimension reduction. Collaborative filtering (CF) is a recommendation system in which the recommendation is done on the basis of the users past actions. Multi-criteria collaborative filtering (MC-CF) is the recommendation system by which multiple features of the given values are considered. With this, the possibility of giving accurate recommendation is much higher. But the drawbacks of this system are known as sparsity, and scalability. The research of this paper was done to reduce those two drawbacks and enhance the accuracy of the system. There were two phases of the suggested system: offline and online. For the offline phase a model was produced by the researchers for MC-CF, which uses Higher-Order Singular Value Decomposition (HOSVD) and Adaptive Neuro Fuzzy Inference System (ANFIS) along with subtractive clustering. On the other hand, Support Vector Machine (SVM), K-nearest Neighbors (K-NN) & fuzzy backpropagation neural networks (FBNN) were used for predicting in the online phase. In this paper, the combination of subtractive clustering was used to mine out information from the user review. The mining part was done by Neuro-Fuzzy system. Over all the system shows more accuracy than the multi criteria technique. We will also use similar method to enhance the accuracy for the multi criteria dataset. There was another research based on personalized recommendation system (Ojokoh, Olatunji, Williams, & Ogunniyi, 2012) The authors have done this research mostly to provide suggestion to ecommerce portals related products. The research work used Fuzzy Near Compactness (FNC) concept to provide solution. In this method, related products are passed through a fuzzy controller to get compactness with other products and based on the similarity, product suggestion is provided.

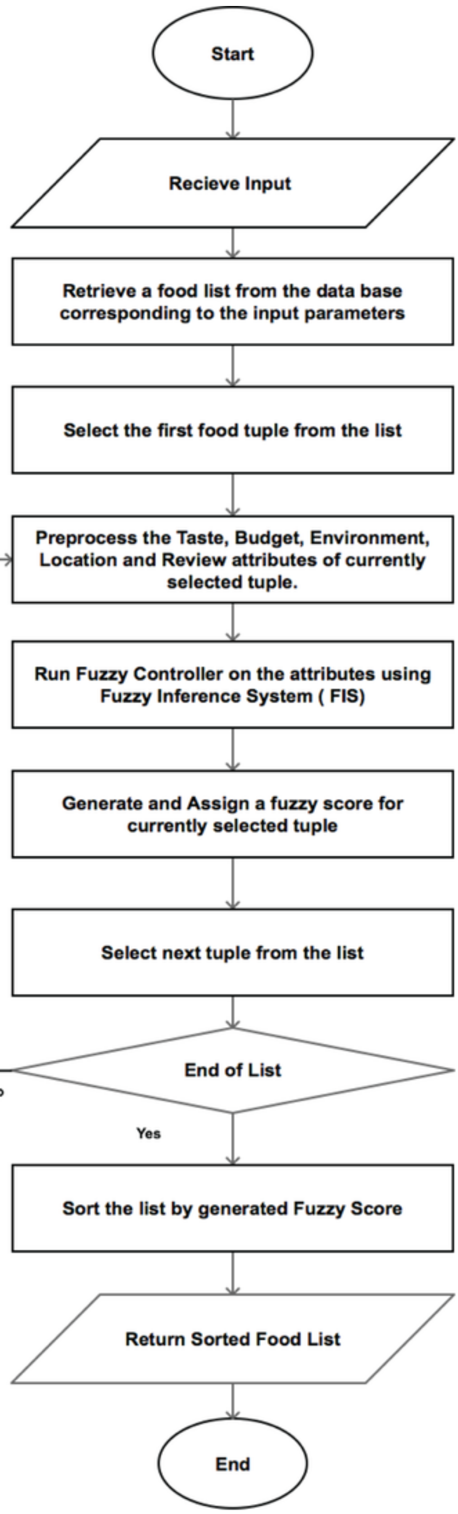
The next research work that we are using as our reference, is slightly different than ours. Considering the drawbacks and advantages of mobile phones, a content based recommendation system has been built by the researchers to improve user experience (Han, Schmidtke, Xie, & Woo, 2014). Here information retrieved from sensors and frequently used personal information have been used to build a context aware system. The system will provide contents according to users' needs. Through mapping mobile user context and properties of contents, they have modeled a modified reasoning search engine. The search engine will be able to compute semantic similarity between content and context to provide relevant contents to the users according to their need. In our recommender system, we are recommending restaurants with food type while they are recommending any type of data contents relevant to the mobile phone users. They are using granular data set obtained from mobile sensors,while we are using previously stored dataset of restaurant details and menus and also user reviews. They have used two non-fuzzy inputs and 1 non-fuzzy output in their system design, where we have used 5 fuzzy inputs and one fuzzy output in our system design. Finally, the last related research work to our research is on efficient advertisement recommendation system for TV program by Velusamy, Gopal, Bhatnagar, & Varadarajan, 2008. Though their goal is completely different than ours but their research methodology includes some concepts that helped in building our system. For providing TV program recommendation they have used fuzzy logic techniques and have taken into account various parameters like program content, users' interests, providers' preferences, program timing and so on. Based on those parameters they have provided keywords to each program and generated TV-program recommendation.

3. THEORY

The primary goal of this research is to build a fuzzy system that will take different input parameters as search query and as an output, the system will assign scores or precedence level to individual menu. More score means more preferred menu for a specific search query. Figure 1 shows the flowchart of the system.

During this research we have conducted a survey and based on that we have identified 5 input variables which are – taste, environment, budget, user review and location. First of all, the system

Figure 1. Flowchart of the system



takes input from the user and after that these inputs are categorized under 5 input parameters. After that the system will search food items from its fuzzy database according to the user inputs which has been categorized in five parameters. In the fuzzy database all the food items are also categorized under the same five parameters. Then, whenever it gets a result based on the search query it passes through some computational process which may vary from parameter to parameter. After that, the system has the final computed values of the 5 parameters for that particular food item which has been fetched from the database comparing with the given user input. Then, it will be passed through the fuzzy controller of the system which contains 1024 fuzzy conditions. The given input will pass through those rules and finally it will output a score which the system will level as the precedence level of that particular food item. As earlier it has been said that, the more score the food item gets, the more preferred is the food item. That is why, when all of the food items which have been fetched from the fuzzy database and passed through the fuzzy controller, they are arranged in descending order according to the score that has been given by the fuzzy controller. Finally, the system will show the user the names of the food items with their respected restaurants in that same order.

3.1. Input Variables

Our primary objective is to make a flexible food searching engine. We have considered 5 attributes as input variables, these are: taste, environment, budget, location and user review. For each input variable we have defined 3 membership levels.

3.1.1. Taste

Taste is the primary component that distinguishes different foods. Our system majorly focuses on searching food by different tastes. User can give any kind of taste as input. For example: single tastes like sweet, sour etc. or even multiple tastes like sweet-sour, hot-spicy etc. We also do not want to limit the user to any specific number of taste attributes. So in the fuzzy controller we need N number of variables but it would not be a good design of choice. To solve this issue, we have defined only one variable for taste in the fuzzy system. This single variable is a generalized variable taste, which has 3 fuzzy levels with their membership functions. The levels are: Low, Medium & High which are represented as 1, 2 and 3 accordingly. This variable can represent any one taste of user's interest at a time. The system relates the user input with the database and sends taste level of single taste to the fuzzy system as an input. In the case of multiple tastes, for example, for sweet-sour the calculation will be done twice. So the server will call the fuzzy controller twice for a single food search and hence we will have two separate scores for two different tastes of the same set of menus. System will combine multiple scores to get a single score:

menu1_score_spicy = fuzzycontroller (spicy_level, … other_parameters)
menu1_score_sweet = fuzzycontroller (sweet_level, … other_parameters)
menu1_score = menu1_score_spicy + menu1_score_sweet
menu2_score_spicy = fuzzycontroller (spicy_level, … other_parameters)
menu2_score_sweet = fuzzycontroller (sweet_level, … other_parameters)
menu2_score = menu2_score_spicy + menu2_score_sweet
.
.
.
menuN_score_spicy = fuzzycontroller (spicy_level, …other_parameters)
menuN_score_sweet = fuzzycontroller (sweet_level, …other_parameters)
menuN_score = menuN_score_spicy + menuN_score_sweet

Let us assume a user is searching for food names that are "spicy and crispy". Initially system will derive all the food names from the database that have entries like spicy and crispy in the menu and taste table. After that system will assign fuzzy score of spicy and crispy to each menu items. The system will assign score to the foods' spiciness and crispiness separately. It will calculate the spiciness or the crispiness by taking the average of the user's reviews of that taste. For example, suppose, there are 10 user reviews about crispiness of KFC fried chicken. So the system takes those user reviews' crispiness level's value and takes the average of that 10 crispiness level's values. The resulting averaged value would be assigned as the crispiness value of KFC Fried Chicken. This calculation method is applicable for all of the taste parameters. Then, as a fuzzy controller cannot take two taste inputs at the same time, system will pass each menu's score of spiciness with other parameters and the score of crispiness and other parameters separately to the fuzzy controller. Then the system will combine resulting two scores for each menu to get the final score for individual menu item.

3.1.2. Environment

This input variable is just like the taste variable. The difference is that the input here will be environment type like formal, casual etc. and we do not allow the user to specify multiple environment at the same time. Except this slight difference the handling process and calculation are almost as same as taste and will have a separate fuzzy variable called environment.

Suppose, a user is searching for a casual environment restaurant. So the system will enlist all the restaurant names from the database table which has casual type of environment. After that the system will score each of the restaurants as "casual environment" value just as the same way the taste does. It will take the average of the all of the user reviews about the causality of the environment and that value would be the casual environment level of a particular restaurant. For example, if there are 10 user reviews about "KFC's" casual environment, it will take the casual level values from the user reviews and will average those value. The averaged value would be assigned as the casual environment value of KFC and after that this value will be passed in fuzzy controller as an environment parameter.

3.1.3. Budget

User's budget will be taken as an integer input. This input variable has 3 levels in the fuzzy system: Less Preferred, Preferred and More Preferred. Database has a budget for single person for each menu. When user specifies a budget system will find menu names in the range of \$4.5 more or less of the user defined budget. The system finds the absolute difference of users' budget and different foods' price. If a user's budget is 'x' and one of the food's price is 'a'. Then the difference 'D' is given below:

$$S = |x - a|$$

Hence the system creates a list of menus with the price in between x – 4.5 and x + 4.5, therefore, the difference will be in the range of \$0 to \$9. If the user difference between budgets is 0 ~ 3 it will be in the level of More Preferred, if the difference is between 3 ~ 6 then it will be in the preferred level and if the difference is between 6 ~ 9 then the menu will be in Less Preferred Level. The lower the difference is, the higher the preference.

For example, suppose, KFC fried chicken price is \$15 and a user gave his budget to the system which is \$20. Now the difference between the user input budget and the KFC Fried chicken's price is |20-15| or 5 which is in the range of preferred.

3.1.4. User Review

System also allows the user to search for food according to users' reviews. Users can assign 5 levels of satisfaction. Review is in the scale of 0 to 5. While making the query user can provide a score (R_{user}) from 0 to 5 and the system will find all the menus that are 1.5 more or less than the user defined score. Next system normalizes the review score ($R_{normalized}$) by subtracting the lowest review score from the actual review score (R_{actual}) like the below equation:

$R_{normalized} = R_{actual} - (R_{user} - 1.5)$

We have again defined 3 levels of fuzzy membership as Less Preferred, Preferred & More Preferred for the user review variable. User review will be handled similarly as Budget variable.

Suppose, user is searching for food with user review around 3. Again the user review value would be computed for a certain restaurant by taking all the user reviewed values averaged value with in the range defined above. For example, there are 10 user reviews about KFC and the system would average those 10 user reviews satisfaction level values and would assign the averaged value as the user review of KFC. Now, let us assume that the averaged value of user review for KFC is 3. System will get the normalized user review like below:

$R_{normalized} = 3 - (3 - 1.5) = 1.5$

So the final normalized value of user review is 2 and system uses this final normalized value to compute the score for that specific food item.

3.1.5. Location

Users can mention their preferred location. According to their preferred location we can calculate the distance of the restaurants and preferred location. User will give the name of the area of their choice. Using Google API, system gets the latitude and longitude of the preferred location. Restaurants will have their own location in the database as latitude longitude form. Using this information, we can calculate the distance.

Suppose we have a restaurant named 'A' which has latitude '$lat' and longitude '$lon'. User gives a preferred location 'B' which has latitude 'lat' and longitude 'lon'. So the distance between these two places are:

$$AB = \sqrt{\left(\left(lat - \$lat\right)^2 + \left(lon - \$lon\right)^2\right)}$$

System considers only the restaurants within that range of 3KM. Hence we have defined the location variable by 3 membership functions which are Close, Near and Far. When the difference is between 0 ~ 1 km then it will be on the close level, if the difference is between 1 ~ 2 km then it will be on near level and if the difference is between 2 ~ 3 km then it will be on the far level. The closer the distance, the higher the preference is.

Suppose, the difference between the user's location and KFC restaurant is 1.5km. So it would fall on the range of near level in fuzzy controller.

3.2. Input Curve

In the fuzzy system for input curve we have used the 'Gaussian Curve' because it is relatively more precise than the triangular, trapezoidal and other curves. For all of the five variables, there are 3 possible input level which are- Low, Medium and High. So we have defined the following 3 functions for 3 input levels respectively and used them for all the variables:

$$f_{low}\left(x\right) = e^{\frac{-200x^2}{81}}$$

$$f_{medium}\left(x\right) = e^{\frac{-200\left(x - \frac{3}{2}\right)^2}{81}}$$

$$f_{high}\left(x\right) = e^{\frac{-200(x\ -\ 3)^2}{81}}$$

Here f_{low} (x) represents the membership function. We have plotted these 3 functions in Figure 2 to visualize the input curves.

3.3. Output Variable

In the fuzzy system we have defined one output variable 'Preference' which we infer from the five variables using the fuzzy if-else logic. The preference is defined by 5 functions or levels. Higher preference level means -it is more similar to the users' requirements. The division of the preference level is with respect to the logic below.

We have defined all of the input variables in three levels. Each level can have maximum values up to 3. Let us assume that variables taste, environment, budget, location and user review are $X_{1,}$ X_2, X_3, X_4 and X_5. If all the variables are equals to highest value which is 3 we get:

$$\Sigma x_i = 15$$

If all the variables are equals to the lowest value which is 0, we get:

$$\Sigma x_i = 0$$

So if we declare 5 levels of preference, the range of each level is:

$$\Sigma \frac{x_i}{5} = \frac{15}{5} = 3$$

Therefore, we have defined 5 output membership function for 5 levels. The preference levels are given with their range in Table 1.

Figure 2. Plot for input membership functions

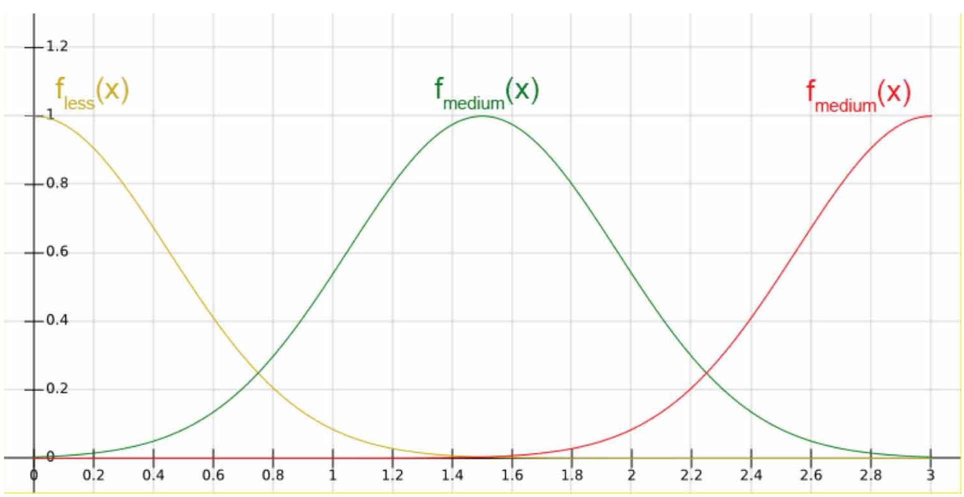


Table 1. Rules separation table

Range	Preference Levels	Membership Function
0 – 3	Level 1	μ_1
3 – 6	Level 2	μ_2
6 – 9	Level 3	μ_3
9 – 12	Level 4	μ_4
12 –15	Level 5	μ_5

3.4. Output Curve

This system has single variable 'Preference' as output. We have considered 5 output levels. Hence we have defined 5 output membership function μ_1, μ_2, μ_3, μ_4, μ_5. They represent 5 preference levels. These functions are triangular function defined by the below piecewise defined functions:

$$\mu_1(x)=\begin{cases}0; & x\le 0\\ \dfrac{2x}{3}; & 0\le x\le \dfrac{3}{2}\\ \dfrac{6-2x}{3}; & \dfrac{3}{2}\le x\le 3\\ 0; & 3\le x\end{cases}$$

$$\mu_2(x)=\begin{cases}0; & x\le \dfrac{3}{2}\\ \dfrac{2x-3}{3}; & \dfrac{3}{2}\le x\le 3\\ \dfrac{9-2x}{3}; & 3\le x\le \dfrac{9}{2}\\ 0; & \dfrac{9}{2}\le x\end{cases}$$

$$\mu_3(x)=\begin{cases}0; & x\le 3\\ \dfrac{2x-6}{3}; & 3\le x\le \dfrac{9}{2}\\ \dfrac{12-2x}{3}; & \dfrac{9}{2}\le x\le 6\\ 0; & 6\le x\end{cases}$$

$$\mu_4(x)=\begin{cases}0; & x\le \dfrac{9}{2}\\ \dfrac{2x-9}{3}; & \dfrac{9}{2}\le x\le 6\\ \dfrac{15-2x}{3}; & 6\le x\le \dfrac{15}{2}\\ 0; & \dfrac{15}{2}\le x\end{cases}$$

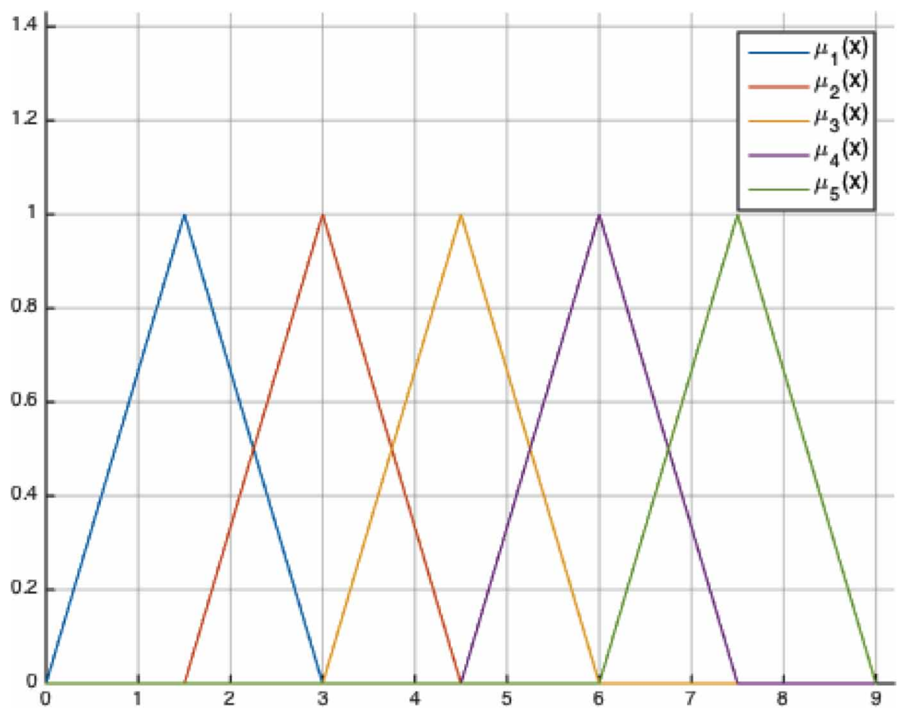
$$\mu_5(x) = \begin{cases} 0; & x \le 6 \\ \dfrac{2x-12}{3}; & 6 \le x \le \dfrac{15}{2} \\ \dfrac{18-2x}{3}; & \dfrac{15}{2} \le x \le 9 \\ 0; & 9 \le x \end{cases}$$

We have plotted the curves for each membership function in the Figure 3. Triangular portion of each curve overlaps on the midpoint of the adjustment curve and the span of 3 in the X-axis. Hence we have the output in the range of 0 to 9. Therefore, output score lays on the range of 0 to 9.

3.5. Fuzzy Conditioned Rules

Each variable has 3 membership functions and 3 levels. However, some variable may not be present on the search query. For instance, a user may not be interested in the budget so he or she may not provide budget in the search query. If a variable is not present we use ‘none’ or 0 as the input. Therefore, each variable in the system can have at most 4 values including the option ‘none’. Therefore 5 input variable can have 4 values. So there are 4^5 or 1024 combinations possible. For assigning rule we have taken the arithmetic sum of each combinations and used Table 1 to assign rules by correlating the summed values with range column and preference level. Let us consider an example where a combination of input parameters are as follows:

Figure 3. Output membership function plot



Taste = high = 3
Environment = Medium = 2
Budget = Medium = 2
Location = Low = 1
User Review = High = 3

So, Total = 3 + 2 + 2 + 1 + 3 = 11.

It lies in the range of 9 to 12 which has a preference level 4. So, the 4th membership function will be triggered for generating the score. The if/then rule for the given example is given below:

```
If Taste = high
      and environment = medium
      and budget = medium
      and location = low
      and user_review = high
then output level 4
```

As there are 1024 rules, we have used the following pseudocode to generate the if/then rules for MATLAB fuzzy toolbox:

```
for (taste = 0; taste ≤ 3; taste++)
  for(environment = 0; environment ≤3; environment ++)
    for(budget = 0; budget ≤ 3; budget ++)
      for(user_review = 0; user_review ≤ 3; user_review ++)
        for(location = 0; location ≤ 3; location ++)
          sum = taste + environment + budget + user_review + location
          if (0 ≤ sum ≤ 3)
            rule = 1
          else if (3 < sum ≤ 6)
            rule = 2
          else if(6 < sum ≤ 9)
            rule = 3
          else if(9 < sum ≤ 12)
            rule = 4
          else if(12 < sum ≤ 15)
            rule = 5
          end_if
          print: taste ` ` environment ` ` budget ` ` user_review ` ` location ` ` rule
          end_for
        end_for
      end_for
    end_for
```

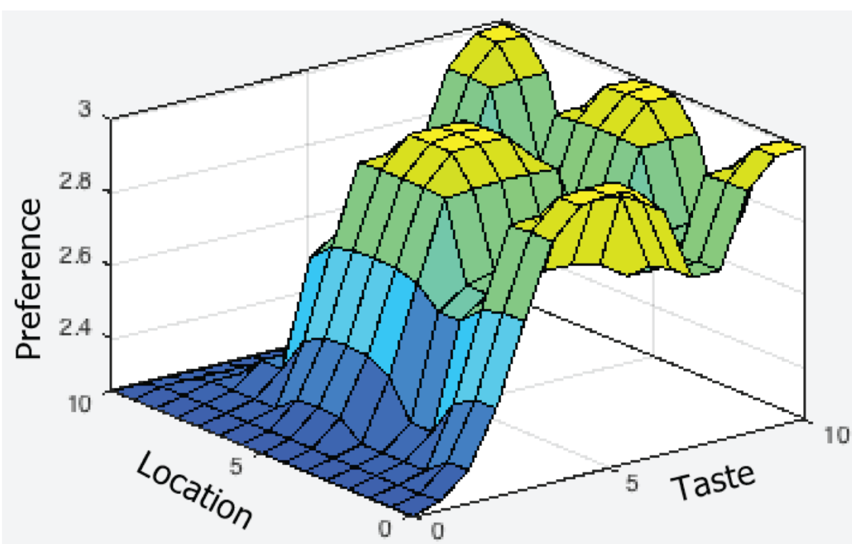
In Table 2, we have listed the input variables and corresponding membership function. Here None, Low, Medium, High correspond to following values: 0, 1, 2, 3.

As there are 5 input variables, all the variables cannot be visually represented in a single plot. So, as a demonstration of accuracy, Figure 4 maps two variables as a graphical representation. It is

Table 2. Rule generation table

Rule	Taste	Environment	Budget	User Review	Location	Preference (Membership Function)
R_0	None	None	None	None	None	Level 1
R_1	Low	None	None	None	None	Level 1
R_2	Medium	None	None	None	None	Level 1
R_3	High	None	None	None	None	Level 1
R_4	None	Low	None	None	None	Level 1
R_5	Low	Low	None	None	None	Level 1
.	.	.	.	.	.	.
.	.	.	.	.	.	.
R_{512}.	None	None	None	None	Medium	Level 1
R_{513}	Low	None	None	None	Medium	Level 1
R_{514}	Medium	None	None	None	Medium	Level 2
R_{515}	High	None	None	None	Medium	Level 2
.	.	.	.	.	.	.
.	.	.	.	.	.	.
R_{1020}	None	High	High	High	High	Level 4
R_{1021}	Low	High	High	High	High	Level 5
R_{1022}	Medium	High	High	High	High	Level 5
R_{1023}	High	High	High	High	High	Level 5

Figure 4. Graphical representation or surface view of taste and location



the surface view of the input variable taste and location. From the graph we can see that, more the taste and location is closer to the desired taste and location, higher the score it gets.

3.6. Adaptive System Based on User Weight

In our system Taste, User review and Environment scores are given by the user. In the Input Variable section (3.A), we have taken the average of scores to get the scores for the input parameters. We can further enhance this scoring by neural network's adaptation techniques. We have assigned each user a weight W_i. This score will be calculated by the frequency f_i where f_i stands for number of reviews that users have given on the system. To get individual user score for a given parameter (i.e. taste) we will multiply user given score by the user's weight. To get the final score, we will take the mean of all the scores for that given parameter. We will compute the adaptive score with the following equations:

$$W_i = \frac{2\arctan\left(fi\right)}{\pi}\text{; fi} > 0$$

$$Total\ Score = \frac{\sum_{i=1}^{n} XiWi}{n}$$

Here X_i stands for individual user, W_i stands for weight of that user and n stands for total number of user. User frequency is an integer. To generate the weight, we need to convert it to a range of 0 ~ 1. Hence we have formulated the above function that generates our desired output. Figure 5 shows the plot of the weight function.

In Figure 6, we have demonstrated our neural networks logic that we have used to make the system adaptive.

Now at the very beginning, we will demonstrate an example for the case of taste. Let us assume that there are 5 user reviews in the system and the users are A, B, C, D and E. Now we know the f_i is the frequency of giving review in a system by a user and let us assume that this frequency f_i for user

Figure 5. Plot of weight function

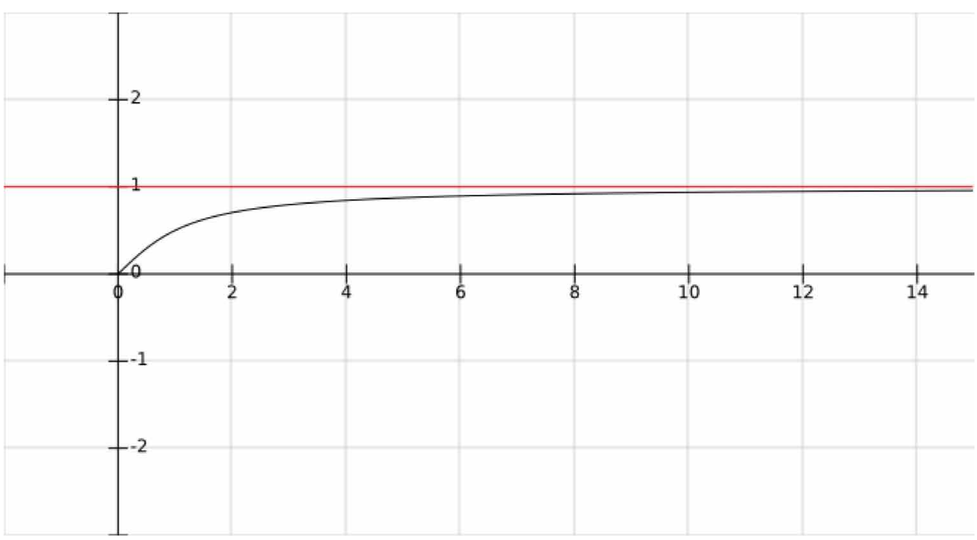
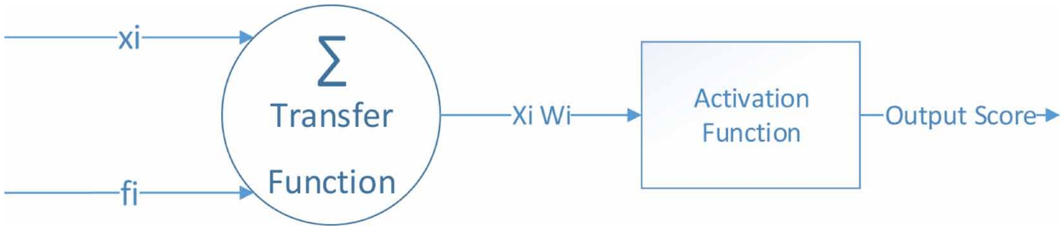


Figure 6. Neural networks logic for adaptive system



A, B, C, D and E are respectively 1, 5, 2, 8 and 1. So at the very beginning the system will calculate W_i for each of the user and the calculation goes by this:

$$W_i = \frac{2\arctan(1)}{\pi} = 0.50 \text{ ; For A}$$

$$W_i = \frac{2\arctan(5)}{\pi} = 0.87 \text{ ; For B}$$

$$W_i = \frac{2\arctan(2)}{\pi} = 0.70 \text{ ; For C}$$

$$W_i = \frac{2\arctan(8)}{\pi} = 0.92 \text{ ; For D}$$

$$W_i = \frac{2\arctan(1)}{\pi} = 0.50 \text{ ; For E}$$

So we get the weighted value for each of the user A, B, C, D and E which are 0.50, 0.87, 0.70, 0.92, 0.50 respectively. Now there is a food item KFC fried chicken in the database which has a taste type which is crispy. Now when a user searches for crispy food, it would trigger the weighted average formula on user given taste reviews (Given that user A, B, C, D and E rated crispiness of KFC fried chicken's 2,1,2,2 and 1 respectively where 2 represents crispy and 1 represents less crispy) which is like the following:

$$\frac{2*0.50+1*0.87+2*.70+2*092+1*0.50}{5} = 1.12$$

Therefore, for KFC Fried Chicken, the system finds the level of crispiness is 1.12 which would then passed through the fuzzy controller with other parameters. After that, let us focus on environment and user review also. The calculation of weighted value for the 5 users would be same as before. Let us assume, Users A, B, C, D and E gave the environment type of causality of KFC restaurants as follows: A gave 2, B gave 3, C gave 3, D gave 3 and finally E gave 3. Similarly here, 2 represents casual and 3 represents more casual. So whenever a user wants a casual type of restaurants the output of KFC restaurant would be as following:

$$\frac{2*0.50+3*0.87+3*.70+3*92+3*0.50}{5} = 1.99$$

So, for the restaurant KFC, the casual environment level is 1.99 and it would be passed through the fuzzy controller. Finally, for the user review parameter, user A, B, C, D and E gave the following feedback respectively 3, 4, 5, 5, 3. Provided that user review scaling ranges from 0 to 5. So after calculation using the weighted value of each user the result is the following:

$$\frac{3*0.50+4*0.87+5*.70+5*92+3*0.50}{5}=3.00$$

The value we get here is the average value of the user review of KFC restaurants. While passing to the fuzzy controller the system normalizes this value explained in the User Review section (3.A.4).

3.7. User Specific Adaption

One of the major problem of taste is that it is some extent relative to individual user. For example, a food can be spicy for a person while the same food can be less spicy for another person. To resolve this issue, we have constructed the system to adapt with the user specific preference. The general solution is to assign a user preference scale for each attribute based on the user's previous feedbacks. This preference scale will be updated every time a user provides a review. System keeps track of the user's profile and consider last 5 user feedback. Only the taste and environment input variables are dependent on user preference and hence the system considers only these two parameters while scaling the system based on user preference. The system scales the user's feedback when they provide it and then stores their scaled feedback (f_{scaled}) in the database. On receiving a new feedback (f_{new}), the system will look for maximum last 5 feedbacks $\left(fi\right)$ for similar taste or environment preference by the same user. For each of those feedback system will subtract user's feedback from the average of that specific food's particular taste or environment feedback $\left(fui\right)$ given by other the users. Finally, the system will consider the average of the subtracted values for maximum 5 feedbacks as the user specific preference scaling constant (λ) and when saving the new review of the user in the database, system will subtract the scaling constant from the given review. Below equations are used to scale the user feedback:

$$\lambda=\sum_{i=1}^{n}\frac{fi-fui}{n}\ ;\ n\leq 5$$

$$f_{scaled}=f_{new}-\lambda$$

For instance, let us assume a user has given a feedback for food as "less spicy" out of 3 options less spicy, spicy and more spicy. System will list previous feedbacks where he or she has provided spicy as taste feedback. System considers at most 5 recent feedbacks from the list. System will subtract user's feedback with average feedback given by other users of each item in the list. Next system will average the differentiated value and use it as the scaling constant.

As an instance, in the database, there are foods with medium spicy scoring as previous feedbacks like below.

Other foods where user has provided "spicy" with different magnitude as feedback:

Fried Chicken, KFC = 2 (spicy)
Beef & Bacon, Take Out = 2 (spicy)
Chicken Cheese Burger, Preetom = 3 (More Spicy)

Average Feedback given by other users for the same foods:

Fried Chicken, KFC = 1.3
Beef & Bacon, Take Out = 1.7
Chicken Cheese Burger, Preetom = 2.3

New scaled feedback has been calculated in the below section:

$$\lambda = \frac{2-1.3}{3} + \frac{2-1.7}{3} + \frac{3-2.3}{3} = 0.567$$

$f_{scaled} = 2 - 0.567 = 1.433$

Now this score will be saved in the user account. Next time whenever the user will search for "something spicy", taste score 2, it will be scaled. In this case its (2 – 0.567) = 1.433 which means system should search for less spicy food.

4. METHODOLOGIES

Our goal is to develop a system that will allow the users to assign variable number of taste parameters with a magnitude. Also it will allow few other defined parameters and magnitude. System will allow the users to search a fuzzy database and provide a list of preferred menu-restaurant pairs. System also maintains a user profile.

The system consists of 3 major components which are server, database & fuzzy controller. System performs two major tasks- search for preferred menus and take user review. While searching, the server receives data from the user interface(UI) and fetches relevant data from database. Then, it matches and calculates parameters and passes it to the fuzzy system. Fuzzy system uses the attributes level to calculate a preference score. Combining the preferences score with the database, system generates a sorted menu list as output. Figure 7 shows the three components of the system.

4.1. Server

This part of the system is responsible for data fetching from the database, preprocessing the data for calculating fuzzy scores for each parameter and communications between the user and the fuzzy system. We have used Apache web server as the server software and PHP as the scripting language.

One of the primary tasks of the server is to serve user with web interface. There are two separate forms, one for searching menus and another for taking feedback about specific menus. The server also keeps a profile of each user.

Figure 7. Block diagram of the system

For searching, user can give different taste input using their natural language for example something spicy. For multiple tastes search, they can provide the input as comma separated taste parameters like "sweet, sour". For location parameter, user provide location address just using strings. Using Google Map Application Program Interface (API), the system will derive the latitude and longitude values for that location. For the budget user has to give estimation of their budget for each parson. User can provide their preferred environment but this input takes only one environment. Users can also choose a level of satisfaction from other users on a scale of 0 to 5 which is called user review. In this form, users are presented with 5 options- "Very Poor", "Poor", "Average", "Good", "Very Good" having scores 1, 2, 3, 4, 5 for each level of satisfaction and allowed to select one as input for user review. If no option is selected during the query it means the score is 0 and the user is not interested to choose previous users' review as his search criteria.

The other form is intended for User Feedback. Users will be asked to give the feedback under a specific restaurant's menu. The user is asked several questions like 'was the food too spicy?' or 'was the environment as expected?' based on the previous feedback of other users and also users can provide new suggestions of their preference. At the end user is asked to rate the food by selecting 1 out of 5 options like in searching form as feedback for overall user's review. Finally, system stores this information in the database.

One of the major task of the server is to preprocess the data before passing it to the fuzzy controller. For preprocessing, server uses the methods described in the theory section. There may be multiple reviews available for each menu in the database. Converting those multiple entries to a single score is the primary task of preprocessing.

4.2. Database

Initially every menu in the database will have scores for each attribute. This score is provided by system administrators or volunteers (i.e. restaurants). The system uses MySQL database, as it is highly integral with the apache server. Fuzzy controlled System does not directly communicate with the database rather, the server acts as a mediator and communicates with the fuzzy controller and database. Server fetches the data from the database and fuzzifies the data, hence the database acts as a fuzzy database.

Design of the database is shown in Figure 8. To maintain a simple relationship in the database we have defined six tables in the database. These tables are, (i) Restaurants, for holding the necessary information of restaurants, (ii) Menu holds menu items and its parameters paired with the primary key of restaurant, (iii) Taste, for holding several user feedbacks about the taste of the food (iv)Review stores the user reviews against each menu, (v) Environment stores the environment suggestion of a restaurant, (vi) User, for maintaining user profile.

4.3. The Fuzzy System

The fuzzy system is the key factor of the system as it calculates the preference level of the restaurants with foods. Fuzzy System acts as a fuzzy controller in our system. The one and only job of the fuzzy system is to take different score of input parameters and provide a score for that given set of input. We have used the set of instructions described in the Theory section (3) to construct the fuzzy controller. MATLAB fuzzy toolbox has been used to build the fuzzy system.

Figure 9 shows the overview of the fuzzy system which takes taste, environment, budget, location and user review as fuzzy input variable and gives the output of preference level. We have used Mamdani inference rule to calculate the preference level.

Server fetches data and passes data to Fuzzy Controller. On receiving the input data, fuzzy system passes those data through the fuzzy conditions and calculates the preference level for each item. MATLAB fuzzy system has been wrapped as an external executable Command-Line (CLI) program. The program can take multiple set of input parameters from the CLI.

Figure 8. Database tables schema

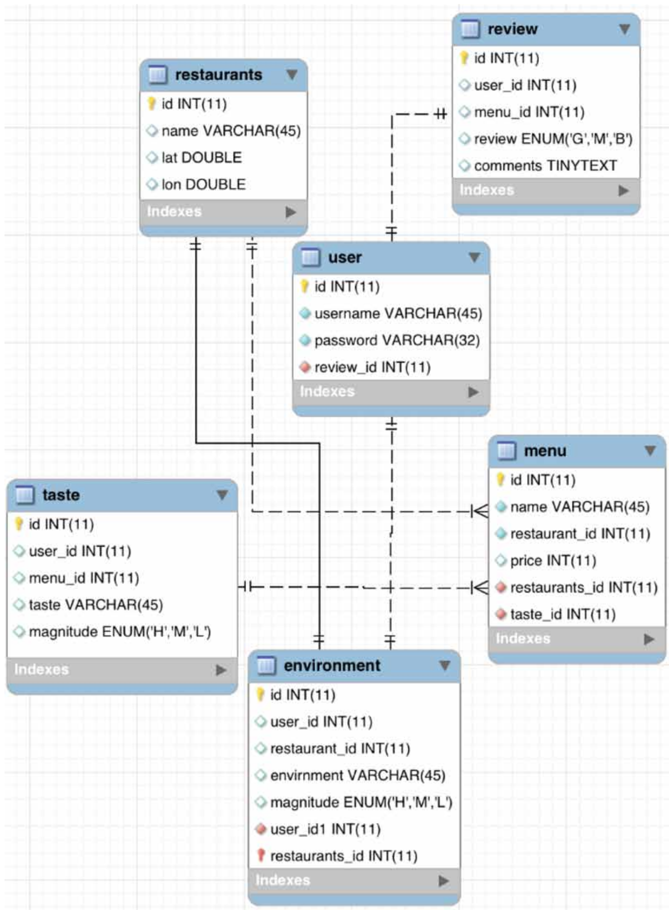
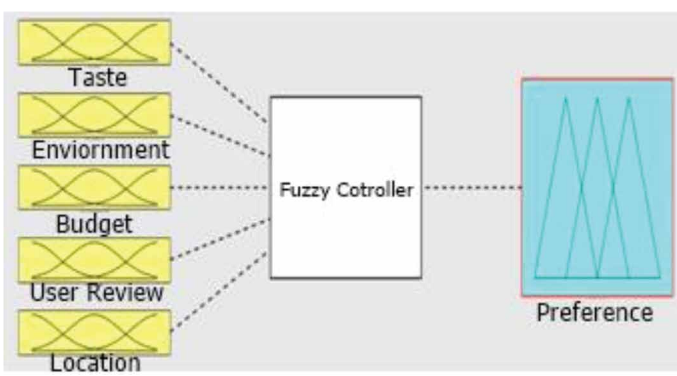


Figure 9. The Fuzzy System



When the system will get the preference levels of various restaurants menu's from the fuzzy system, it will order menus by scores from the highest to the lowest.

5. RESULTS AND DISCUSSION

For testing the system, we have inserted 20 restaurant and 10 menu items for each of the restaurant. We have done 200 survey on the university campus to get reviews on the inserted menus. We have run the system on simulated data to verify the results.

For example, user searched for a food using the keyword 'something spicy and crispy', '$20 budget', 'Gulshan 1 Area', 'casual environment' and 'user review scale preferred 4'.

Firstly, system will find all the menus within the ranges as mentioned in the theory section and make a list. Now let us go back to the example of KFC which we had discussed in the theory (3) section. Using the all of the calculation methods from the theory section for each parameter we get the following values: for taste crispiness 1.12 and spiciness 2.04; for casual environment 1.99; for user review scale 3.00; for location 1.5. As here is a multiple taste to search, the system would pass the value of spiciness which is 2.04 along with the other four parameters once and after that keeping the other four parameters constant the system will again pass the value of crispiness through the fuzzy controller. After that it will get 2 preference level for two tastes of KFC Fried Chicken and the system will sum the two values and this value will be the final preference level of KFC Fried Chicken. Now using these input parameters, we can calculate the preference level using the Mamdani Min Implication.

Table 3 lists the sample input of this example and Table 4 shows the corresponding outputs of this sample table. Using this example, we get the preference level 3.32 for the above situation. This is also enlisted as the first sample output of Table 4.

Table 3. Sample input

Taste	Spicy, Crispy
Environment	Casual
Location	Gulshan 1 Area
Budget	$20
User Review	4

Table 4. Resulting output score of sample input

Food	Price	Environment	User Review	Location	Taste	Preference
Fried Chicken, KFC	5	1.2	2.01	3.0	1.12	3.80
Burger, KFC	2	1.6	1.22	7.69	1.44	3.52
Pizza, Pizza Hut	9	0.4	1.04	5.46	2.04	3.41
Pizza, Pizza Inn	8	1.50	1.50	4.92	1.50	3.36
Calamari Fry, Fish & Co.	8.1	1.12	1.50	4.75	1.88	3.10
Tempura, Fish & Co.	8.5	1.10	1.20	7.92	2.05	2.99
Nachos, Smoke Cafe	8	1.12	1.21	6.42	0.80	2.76

The system will have several menu names and score for given input parameters. After preparing the list with processed parameters server will call the fuzzy controller and pass the list. As a response fuzzy controller will return score for each menu.

In Table 5, we have showed a quantitative comparison of the outcomes of our search engine based on the different kind of adaptation methods of our search engine. Here, first we have applied both user specific adaptation and system adaptation, followed by only using system adaptation, after that only using user specific adaptation and finally excluding both user specific adaptation and system adaptation. If we give a closer look to the table, we can see that there are slight changes in values and results in each case. For example, in the first case where we have accounted both user specific adaptation and system adaptation there "Burger, KFC" gets the second choice preference while on the other hand when we only apply the system adaptation, there "Pizza, Pizza Hut" gets the second choice preference. One thing here should be noted that, in the case of taste and environment, the results may deflect slightly from the actual case as the taste of each food or environment of each restaurant is based on the users' feedbacks and whenever a prominent user (who gives higher number of reviews) gives review about any of the two parameters that gets biased on his review. Let us say, a restaurant says, their environment is more casual while one or two prominent users give their feedback that it is actually less casual. So, in this case, our search engine is getting the inputs for the restaurant's environment from the user and that is why our system will show that the environment of this particular restaurant is more likely to be less casual.

In case of multiple taste search the server will call the fuzzy controller multiple times for each test keywords and finally combing multiple results system constructs the final score for each menu and sorts the menus by score and provide the user with order list of menus according to preference.

There are two similar services, e.g., Yelp, Foursquare are available that suggest food and restaurants like our service. The food searching features are compared with those two services in Table 6 in terms of five searching features. One of the key advantages of our system is that it supports user specific adaptation that means it suggests food according to user's previous taste and taste is adaptive based on the user's taste review of the food. Besides, the system itself is adaptive which means it assigns more weight to the user who visits frequently to a restaurant and comments on the foods compared to others who rarely visit a restaurant.

Table 5. Comparison with adaptive and non-adaptive system

Suggestion with System and User Specific Adaptation	Score	Suggestion with System Adaptation but without User Specific Adaptation	Score	Suggestion with User Specific Adaptation But without System Adaptation	Score	Suggestion without System and User Specific Adaptation	Score
Fried Chicken, KFC	3.80	Fried Chicken, KFC	3.82	Fried Chicken, KFC	2.80	Fried Chicken, KFC	3.76
Burger, KFC	3.52	Pizza, Pizza Hut	3.59	Burger, KFC	2.51	Burger, KFC	3.54
Pizza, Pizza Hut	3.41	Burger, KFC	3.51	Pizza, Pizza Hut	2.43	Pizza, Pizza Inn	3.38
Pizza, Pizza Inn	3.36	Pizza, Pizza Inn	3.32	Pizza, Pizza Inn	2.35	Pizza, Pizza Hut	3.36
Calamari Fry, Fish & Co.	3.10	Calamari Fry, Fish & Co.	3.00	Calamari Fry, Fish & Co.	2.10	Calamari Fry, Fish & Co.	2.99
Tempura, Fish & Co.	2.99	Nachos, Smoke Cafe	2.98	Tempura, Fish & Co.	1.92	Nachos, Smoke Cafe	2.86
Nachos, Smoke Cafe	2.76	Tempura, Fish & Co.	2.97	Nachos, Smoke Cafe	1.86	Tempura, Fish & Co.	2.54

Table 6. Comparison with existing food search engines

Features	Yelp	Foursquare	Our System
Search by taste	Not Available	Not Available	Taste based search is available
Search by price	Available	Available	Available
Search by distance	Available	Available	Available
Search by environment	It searches using the keywords, pub, good for kids, has TV etc.	It uses the key words Coffee, Nightlife, Fun etc.	It uses the key words, formal, romantic, casual
Search by User Review	Uses ratings or review numbers to sort the food list	Uses ratings to sort the list of the food items	Uses user feedback by accounting the keywords chosen by the user like good, bad, excellent etc. as a review
User Specific Adaptation	Not Available	Not Available	Available
System Specific Adaptation	Not Available	Not Available	Available

6. FUTURE WORK

In future we will integrate machine learning technique in our system to make more precise reviews. Currently, we have implemented our fuzzy recommender system as a web application. In future we want to build an android application for this fuzzy recommender system which will be more precise and easily accessible. We would also like to modify the triangular output in future to get more optimized output.

REFERENCES

Ge, M., Elahi, M., Fernaández-Tobías, I., Ricci, F., & Massimo, D. (2015). Using Tags and Latent Factors in a Food Recommender System.*Proceedings of the 5th International Conference on Digital Health DH '15*. doi:10.1145/2750511.2750528

Han, J., Schmidtke, H. R., Xie, X., & Woo, W. (2014). Adaptive content recommendation for mobile users: Ordering recommendations using a hierarchical context model with granularity. *Pervasive and Mobile Computing*, *13*, 85–98. doi:10.1016/j.pmcj.2013.11.002

Nilashi, M., Ibrahim, O. B., Ithnin, N., & Zakaria, R. (2014). A multi-criteria recommendation system using dimensionality reduction and Neuro-Fuzzy techniques. *Soft Computing*, *19*(11), 3173–3207. doi:10.1007/s00500-014-1475-6

Ojokoh, B. A., Olatunji, O. M., Williams, S. O., & Ogunniyi, T. (2012, October). A Fuzzy Logic Based Personalized Recommender System. *International Journal of Computer Science and Information Technology & Security*, *2*(5), 1007–1015.

Priyono, R. A., & Surendro, K. (2013). Nutritional needs recommendation based on fuzzy logic. *Procedia Technology*, *11*, 1244–1251. doi:10.1016/j.protcy.2013.12.320

Ramírez-García, X., & García-Valdez, M. (2013). Restaurant recommendations based on a domain model and fuzzy rules. In Recent Advances on Hybrid Intelligent Systems (pp. 533–546). doi:10.1007/978-3-642-33021-6_40

Sohail, S. S., Siddiqui, J., & Ali, R. (2015). OWA based book recommendation technique. *Procedia Computer Science*, *62*, 126–133. doi:10.1016/j.procs.2015.08.425

Trevisiol, M., Chiarandini, L., & Baeza-Yates, R. (2014). Buon appetito.*Proceedings of the 25th ACM conference on Hypertext and social media HT '14*. doi:10.1145/2631775.2631784

Velusamy, S., Gopal, L., Bhatnagar, S., & Varadarajan, S. (2008). An efficient ad recommendation system for TV programs. *Multimedia Systems*, *14*(2), 73–87. doi:10.1007/s00530-008-0117-1

Zhao, N., Wang, Q., & Zhong, J. (2013). Research on Fuzzy Intelligent Recommendation System Based on Consumer Online Reviews. In Knowledge Science, Engineering and Management, LNCS (Vol. 8041, pp. 173-183). doi:10.1007/978-3-642-39787-5_14

Tousif Osman's primary research fields include visual data analysis, data mining, machine learning, pattern recognition, fuzzy logic and soft computing. He is also interested in cluster computing, computer network architecture and internet of things. One of his major research projects is traffic management using visual data analysis. It was initially his Bachelor's final year project and has been showcased in ICT-EXPO 2016, Bangladesh. Two of his other research works based on fuzzy logic and data mining have been published in 2016. He has also carried out research on online learning system for students and face detection using computer vision. He is currently working as a lab officer at North South University. He has completed his Bachelor in Computer Science & Engineering from the same university in 2016.

Maisha Mahjabeen is currently working as an Intern in iCreation Ltd. In her Bachelor's final project work she has worked on "Trailing Detection of Human Activity: GPS Tracking". One of her research work based on fuzzy logic got published in 2016. Her research interests are soft computing and fuzzy logic. She has also worked on vision based human face detection system in her undergrad. She has completed her Bachelor in Computer Science & Engineering in 2016.

Shahreen Shahjahan Psyche is currently working as a Lab Officer in North South University. She has worked on a vision based traffic jam management system in her Bachelor's project work which has also been showcased in ICT-EXPO 2016 in Bangladesh. Two of her research work based on fuzzy logic and data mining got published in 2016. She has also worked on a face detection system which is based on computer vision- in her undergrad. Her research interest focuses on computer vision, machine learning, pattern matching, fuzzy logic and data mining. She has completed her Bachelor in Computer Science & Engineering in 2016 with "Magna Cum Laude".

Afsana Imam is a current under grad student of Northsouth university in Dhaka, Bangladesh. She recently working on her final project before completing her undergrad. Her recent work is to develop a mobile application for track someone's own travel on abroad and a website which can plan someone's travel plan and review them for places. This work is known as "Trailing Detection of Human Activity: GPS Tracking" Her basic interest is in the field of Web Development, Mis and Fuzzy Logic. Afsana Imam published a conference paper based on fuzzy logic in ICIS 2016. She is going to complete her undergrad in Computer Science and Engineering in Fall 2016.

J.M. Shafi Ferdous is a computer science and engineering graduate from North South University, Bangladesh. He is currently working as a lab instructor for North South University. His research interest involves the area of fuzzy logic, embedded systems and computer architectures.

Rashedur M. Rahman is working as an Associate Professor in Electrical Engineering and Computer Science Department in North South University, Dhaka, Bangladesh. He received his Ph.D. in Computer Science from University of Calgary, Canada and Masters from University of Manitoba, Canada in 2007 and 2003 respectively. He has authored more than 100 peer reviewed journals and conference proceedings in the area of parallel, distributed, grid and cloud computing, knowledge and data engineering. His current research interest is in data mining particularly on financial, medical and educational data, data replication on grid, cloud load characterization, optimization of cloud resource placements and computational finance. He has been serving in the editorial board of a number of journals in the knowledge and data engineering filed. He also serves as a member of organizing committee of different international conferences.