Asian Journal of Information Technology 15 (24): 5019-5027, 2016
ISSN: 1682-3915

Quantitative Preference Model for Dynamic Query Personalization

N. Buvaneswari and S. Bose
Department of Computer Science and Engineering, Anna University, Chennai, India

Abstract: The emerging data science technologies in recent years has given rise to a new field of research consisting of context-aware query processing facilities in information systems. The extraction of timely actionable information from diverse data analysis is a real dilemma in data science. This study discusses a predictive analysis of personalization technique with quantitative user preference model. The first phase extracts personalized results from explicit learning. The second phase builds contextual preference rules form collection of personalized results using apriori algorithm. The view point of user interest retention and granular information processing examines the proposed personalization algorithm for user centric unification. Though many personalization algorithms have been proposed already they have limitations in terms of accuracy, user satisfaction and search time. The major advantages of the proposed system are reduced search time, improved customer satisfaction. Objective metrics, subjective user perception and behavioural measures are utilized to prove the quality of potentially effective result.

Key words: User preference model, personalization algorithm and preference rule, effective result, personalization algorithms, explict

INTRODUCTION

Data science is deep knowledge discovery through data inference and exploration. This discipline often involves analytical approach to handle business challenges that involve data, leveraging troves of raw information to figure out hidden insight that lies beneath the surface. Internet and its exploration causes information abundance which becomes a high priority challenging task to the users. Information rich world makes simple query search as complex because of the large result size. Information overloading problems are handled in information retrieval by filtering techniques and personalization techniques. Filtering techniques try to remove unwanted results from actual results. Personalization techniques ranks the actual results based on user preferences.

Personalization system enhances user query with preferences constructing personalized query and ranks the results according to user's interest. This process reduces the search time and identifies more favourite from actual result.

Preferences indicate great liking of an object. Generally, preferences are classified as explicit and implicit preferences. User directly states explicit preferences. Implicit preferences are learned preferences based on user activity (Qiu and Cho, 2006). User click, user log file and browser history are commonly used to derive implicit preferences (Shen *et al*., 2005). Existing personalization system works based on both the preferences. Many web based applications gathers preferences by user activity. Example: Amazon. They customize the products to individuals based on their need. Though there are several personalization algorithms exist in literature they have poor customer satisfaction and high personalization time. The proposed work presents system architecture for dynamic personalization with quantitative user preference model. Systematic work is still lacking in personalization, hence this research technically sounds for the strength of quantitative modelling. In addition, characteristics of preferences depends on individual's perception. Stating them as preference condition with numeric score is a challenging task that effectively carried out in this research.

Personalization technology serves information need of different users in different way satisfying their individual requirements. The main objective of personalization is user satisfaction (Koutrika and Ioannidis, 2004) and the major goals of personalization are accuracy and scalability. The proposed system efficiently handles personalization goals by user preference model. This model designs preference hierarchy based on the inference of movie selection aspect from IMDB site. The problem of preference finding with degree of interest (doi) is considered as a multi criteria decision making problem. The degree of interest (doi) is a numeric score that specifies interest level of the user about particular preference. It ranges from 0-1.

Corresponding Author: N. Buvaneswari , Department of Computer Science and Engineering, Anna University, Chennai, India

Personalization and recommendation system work for product customization. The differences in their working principles are mentioned by Buvaneswari *et al.* (2015). Recommendation system uses collaborative filtering technique to recommend product to the new user. MovieLens and Netflix are existing personalized recommendation systems that suggest movies to the new user with the help of active user rating dataset. MovieLens dataset is found to be incorrect in many cases. Though, several personalization systems are presented earlier, they suffers with incorrect data (Pham and Jung, 2013). Still there is a need to personalize movies efficiently in current internet based applications. For any personalization system user input plays important role. Thus, there is a need to provide accurate user input for processing and at the same time it is necessary to take steps to avoid inconsistent input for efficient personalization. In this scenario, the proposed preference model checks for preference consistency.

Each user has unique identity. The user may not show similar interest for all type of query. Personalization approaches are discussed in five different ways (Buvaneswari *et al.*, 2015). In this scenario, the proposed personalization system is designed with user profile based approach followed by rule based approach. User preference model gathers preferences with doi. As the steps go up, the proposed system learns preference rules and checks for dependency. The truthful results help for efficient dynamic processing.

Koutrika and Lonnidis (2005, 2010) presents personalization framework for database queries with structured user profile. The problem of related preference selection is discussed as graph computation problem (Koutrika and Ioannidis, 2005). Further their research analyzes for personalization of composite preferences. They present two algorithms Exclude_Combine and Replicate_Diffuse for query personalization (Koutrika and Ioannidis, 2010). The proposed work discusses personalization problem as decision making problem. The problem of user profile construction is discussed as weight computation problem and solved by AHP (Analytic Hierarchy Process) method (Saaty, 1986). This method solves multi criteria decision making problems. Ali *et al.* (2012) propose rating and ranking criteria for island selection using decision making method (Ali *et al.*, 2012). User input validity is the main drawback of the existing work (Koutrika and Ioannidis, 2010). The strength of the proposed personalization system lies on the quantitative systematic procedure. The quantitative user preference model results robust preferences with accurate doi, good answer score and helps to reduce personalization time.

This research discusses a dynamic personalization system with quantitative user preferences model based on AHP and Weighted Personalization algorithm (WETPER). The preference query constructor constructs composite preferences from the atomic preferences present in the user profile for effective personalization. This work uses preference model both for searching and result ranking. The quantitative user preferences and contextual preference rules support for reliable effective personalization. The major advantage of combining user preference model with Feedback analysis module is that they analyze to make suitable decisions on preferences and personalized result. In short, to the best of our knowledge this research shows appreciable results in a quantitative way. The contributions are the following:

- Quantitative user preference model
- Personalization system architecture with the Weighted Personalization algorithm (WETPER)
- Contextual preference rules with complete evaluation

MATERIALS AND METHODS

This study presents the architecture and implementation of dynamic query personalization system. This architecture is flexible enough to address the reliability and effectiveness of the dynamic environment. The proposed system architecture presented in Fig. 1 consists of three major components namely user preference model, query personalization and feedback analysis. User preference model contains user interface, preference hierarchy and preference constructor and user profile. Query personalization consists of preference finder, preference query constructor and WETPER. Feedback analysis consists of analyser and rule miner. Each module coordinates together for efficient personalization.

IMDB dataset: The movie dataset used in this work is built from IMDB site. It consists of >2.5 million records of hollywood movies. The raw data dump of IMDB has movie list, voting list, month and year of release, genre, running time, actor list and director list, etc. The proposed research constructs movie database with the relational schema (movie database schema). Movie (mid, mname, year, duration) Genre (mid, gen) Acted_by (mid, aid) Actor (aid, name) Directed_by (mid, did) Director (did, dname).

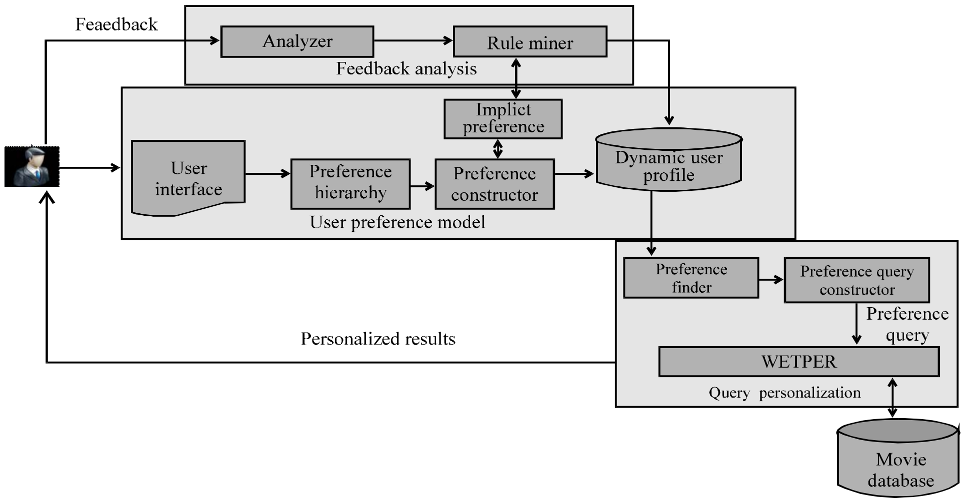

Fig. 1: System architecture

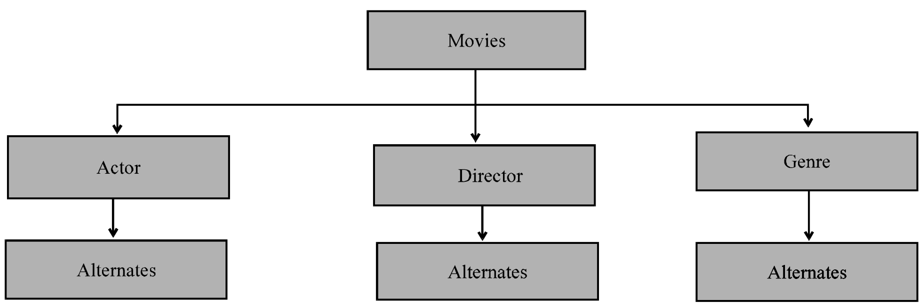

Fig. 2: Preference hierarchy

The user preference model represents preferences as query condition and find doi of preferences by decision making method. AHP captures subjective and objective evaluation measures providing a useful mechanism for checking the consistency of evaluation. When making complex decisions involving multiple criteria, the preference hierarchy is designed such a way that the main goals are decomposed into sub goals. The choice of director, actor, actress and genre has much higher influence on the movie ratings. Since the proposed research constructs the preference hierarchy as shown in Fig. 2.

User preference model: User preference model consists of user interface, preference hierarchy and preference constructor. The user interface allows user to enter positive and negative preferences. Positive preferences begin with "I like" phrase and negative preferences begin with "I don't like" phrase. The positive preferences are processed to find alternates for the constructed preference hierarchy. As stated earlier, actor, director and genre features are considered as major criteria. The corresponding alternates are added in next level in the hierarchy. User interface gathers relative comparisons as input for the criteria and alternates in the ratio scale ranging from 1-9. The values 1-9 stands for linguistic variables like equal, strong and extreme. Each node in the preference hierarchy stands for the preference condition. The preference constructor makes this information as user preferences and stores them in user profile. Sample of user profile is given in Table 1. Implicit preferences are derived from two levels. The first level finds composite preferences from preference hierarchy. The second level learns preference rules from personalized results in feedback analysis. The doi of implicit preferences are derived by Eq. 1:

$$\text{doi}(p) = \sum_{j=1}^{n} \text{doi}(Cj)\ \text{doi}(Aj) \quad (1)$$

Query personalization: The preference finder in query personalization module finds related preferences. The two

Table 1: Sample of user preferences in user profile

Preferences	doi
Movie.mid = Cast.mid and Cast.actor.aid	0.0740
Movie.mid = Directed_by.mid and directed_by.did = director.did	0.2830
Movie.mid = Category.mid	0.6430
Movie.mid = Category.mid and Category.cat='sci-fic' and moive.mid = Directed_by.did and moive.year >2000	0.0484
Movie.mid = Category.mid and category.cat = 'family' and movie.mid = cast.mid and cast.aid = actor.aid and actor.name = 'charles bronson'	0.3821

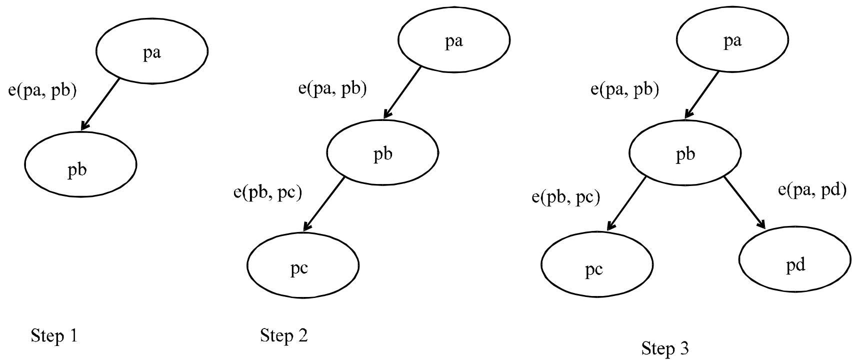

Fig. 3: Algorithm for personalization

preferences pa and pb with corresponding query conditions qa and qb are said to be related if they satisfy any one of the condition as follows:

∀ qa of pa → ∃ ti which satisfies qb of pb refers tuple generating dependency ∀qa of pa → ∃ ti which equals to tj of qb of pb equality generating dependency

Preference finder finds top k related preferences (tuple generating dependencies) from user profile. The algorithm to find top k related preference. In further iterations, preference rules help to find related preferences easily. These preferences are arranged in decreasing order of their interest since the doi of implicit preference is measured by applying a non increasing function on the doi of its constituent preferences. The preference that gets minimum doi shows maximum interest. When the size of personalized answer is less, the maximum interested preference can alone decide the personalized answer. The personalization logic of proposed WETPER Algorithm 1 says the personalized answer should satisfy maximum number of related preferences.

Algorithm 1:

Input: User profile U, User query Q
Output: Set of top k related preferences
Step 1: RP={}, S={}, P={}
Setp2: For each p∈U
    S= atomic condition of p related Q
  End For
 Step 3: While (S≠ϕ) {
    For each pi ∈ S {
    RP=Execute (pi and Q)
    If RP not null
    add p to P

The personalized query is executed with maximum weighted preference. This personalized result is checked for the satisfaction of other related preferences. The personalized result that satisfies maximum number of preferences is ranked with higher priority and given as personalized result. The personalization Algorithm 2 is given in Fig. 3. The preference query constructor groups all preferences as preference network as shown in Fig. 4.

Algorithm 2:

Input: User profile U, Query Q, related preferences P
Output: Personalized Result PRS
Begin
VH={}, EH={}, Preference_Network G={}
For each p from P
{   Qp={p, P} where P has set of related preferences for Q
While Qp ? {}
{   Remove (p, P) add to G
If (root = {}) {make pi as root}
else
Find dependency between nodes of G and p
{
    If (p not related to nodes in G) {set as individual node}
Else {add p in respective place with overriding property}
    }
}
}
}
Find maximum connected node in G

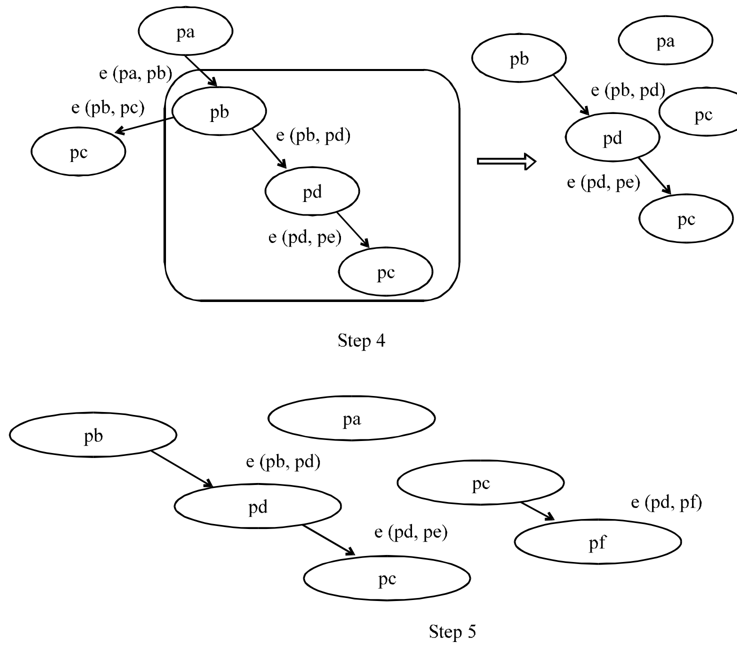

Fig. 4: Steps of WETPER algorithm

```
RS= Execute (query with maximum weighted preference)
For each ti in RS
                                                              {
    PRS= tuples satisfying maximum node in G
    If PRS < 5 {move to next maximum connected node in G}

    Else exit
    }
End
```

Feedback analysis: Feedback analysis module interactively works on interdependencies and perceived consequences of the personalized results. Feedback analysis analyses two performance metric such as answer score and degree of difficulty of the personalized result. Answer score is a user satisfaction score that denotes level of user satisfaction about the personalized answer ranges from 1-10. The degree of difficulty stands for the difficulty to choose the personalized top k answer from the personalized answer. The analyzer finds movies that scores positive z-score values. The z-score values of answer scores of the movies compare the different query results. The movies with positive z-score selects movies as good as for further processing. These movies are considered as transactions and attributes of the movies are treated as individual items. The association among these attributes are translated as association rules by apriori algorithm. The rule which secure maximum support and confidence are stored in user profile as contextual preference rules. Dynamic personalization is achieved by preference rules. User preference model stores explicit preferences as preferred preferences and learns implicit preferences as contextual preference rules from feedback analysis.

RESULTS AND DISCUSSION

This study presents a systematic approach to evaluate the proposed system by making explicit domain knowledge. The movie dataset used in this work is built from IMDB site. It consists of >2.5 million records Hollywood movies. An API is generated to aggregate the information available in IMDB as relational movie database. The entire experiment is written as a separate API in Java language. The movie database has information around 80,000 movies and it is searched with personalized queries with the proposed algorithm. The empirical evaluation of the system is done with 400 users. 357 out of 400 people are started to enter their own preferences with single condition. Remaining 43 users are started to enter multiple conditions as their choice. As the system proceeds further, the users shift their preferences as single condition. This observation seems to indicate that users who gave fine grained choice have more knowledge about the domain. From analysis it is found that users do not want to mix their preference as the system can make more number of combinations:

$$\text{Number of combinations for n preference} = \sum_{r=1}^{n} {}_{n}C_{r}$$

The proposed system is evaluated in three different levels (Yang and Padmanabhan, 2005). The first level checks for valid doi of preferences. All personalization systems are designed based on user input data. This valid input makes successful personalization system. User's interest information is gathered based on preference hierarchy. The relative comparisons of same level of attributes in preference hierarchy are used to find normalized relative scores. These attributes are stored as preferences with doi. Procedure to find doi of actor, director and genre is given.

Preference finding with doi: Pair wise comparison matrix normalized relative weights:

$$\begin{bmatrix} 1 & 1/5 & 1/7 \\ 5 & 1 & 1/3 \\ 7 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1/13 & 1/21 & 3/31 \\ 5/13 & 5/21 & 7/321 \\ 7/13 & 15/21 & 21/31 \end{bmatrix}$$

$$\text{Sum} \quad 13 \quad 21/5 \quad 31/21$$

Each element of the matrix is divided by the sum of its column to get normalized relative weight. The normalized principal eigen vector is obtained by averaging across the rows. This gives doi of actor, director and genre:

$$w = \frac{1}{3}\begin{bmatrix} 1/13+ & 1/21+ & 3/31 \\ 5/13+ & 5/21+ & 7/31 \\ 7/13+ & 15/21+ & 21/31 \end{bmatrix} = \begin{bmatrix} 0.074 \\ 0.283 \\ 0.643 \end{bmatrix}$$

The doi specifies that user gives 7.4% importance for Actor, 28.3% importance for Director and 64.3% importance for genre in movie search. This method finds genre as user's first choice, director as second choice and actor as third choice.

Consistency checking of doi: The Principal Eigen value λ_{max} is calculated by the summation of products between each element of doi with the sum of columns of the reciprocal matrix:

$$\lambda_{max} = 13\ (0.074) + 21/5\ (0.283) + 31/21\ (0.643)$$

$$CI = \frac{3.0998-3}{2} = 0.0499$$

$$CI = 0.0499\ RI = 0.58,\ CR = \frac{0.0499}{0.58} = 0.8603$$

$$CR = 8.60\% < 1\%\ \text{(acceptable)}$$

The doi of preferences are accepted when CR is <15%. This is the first level evaluation of the proposed system. The second level of evaluation is done with controlled experiments. This is the classical approach for the evaluation of goodness of personalization. Precision, recall and F-measure are the basic measures used in evaluating search strategies. The results are given in Table 1. Obviously personalized results may relevant or sometimes irrelevant. The user interface is designed to search for movies based on actor, director, genre and year of the movie. All the users are asked to give the five compulsory queries with their own search queries:

- Movies based on actor with year
- Movies with actor, actress combinations
- Aactor with director combinations
- Actor with category combinations
- Category with year

The personalized results are asked for user satisfaction to evaluate the proposed system. User can set personalized result size ranges from 5-20. Though the query conditions are explicitly specified by the user, the system personalizes the query with related preferences. The problem of evaluating personalization system is done by user itself and depends on individual perception.

In addition to the standard measures, the proposed approach is also evaluated with two additional measures such as answer score and degree of difficulty. Answer score stands for user satisfaction score about the personalized answer. Degree of difficulty stands for the difficulty associated to find exact personalized result from the proposed personalized result. The evaluation reports of the two scores are given in Fig. 5 and 6.

Figure 5 indicates, both Exclude_combine and Replicate_Diffuse algorithms show comparatively less answer score while comparing with the proposed PERSONA algorithm, since it works on valid input.

Figure 6 indicates WETPER algorithm shows higher degree of difficulty while comparing with Replicate_Diffuse algorithm because the personalized results are more relevant for the proposed algorithm. The entire evaluation of the system is done by controlled experiments. The five compulsory queries are executed continuously and without indicating which algorithm is

Table 2: Metrics to evaluate goodness of personalization

Algorithm	Metrics (%)	Q1	Q2	Q3	Q4	Q5
Without personalization	Precision	19.5	21.6	35.3	34.2	25.6
	Recall	20.3	22.4	32.3	21.4	23.5
	F-measure	17.5	24.2	24.6	21.5	23.5
Exclude_combine	Precision	57.5	57.3	61.4	53.6	61.4
	Recall	59.4	56.3	62.3	53.7	64.5
	F-measure	58.2	57.3	62.4	64.3	67.4
Replicate_diffuse	Precision	74.6	68.4	73.2	67.4	66.4
	Recall	74.7	67.8	72.4	68.4	64.6
	F-measure	74.8	68.3	72.1	63.6	67.3
WETPER	Precision	89.3	89.4	85.6	85.4	85.4
	Recall	85.8	84.5	87.7	86.4	85.7
	F-measure	84.5	67.4	85.4	87.5	84.3

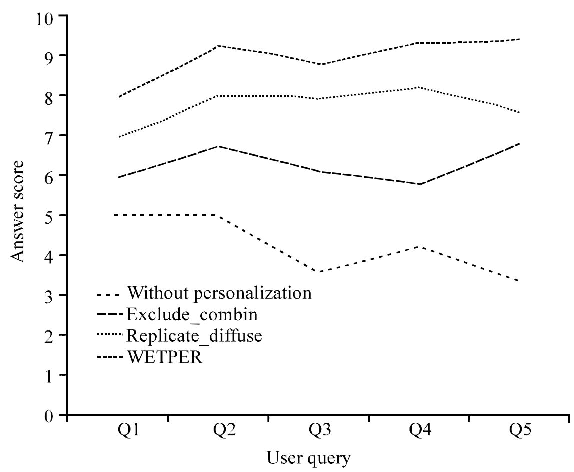

Fig. 5: Answer score of personalization algorithms

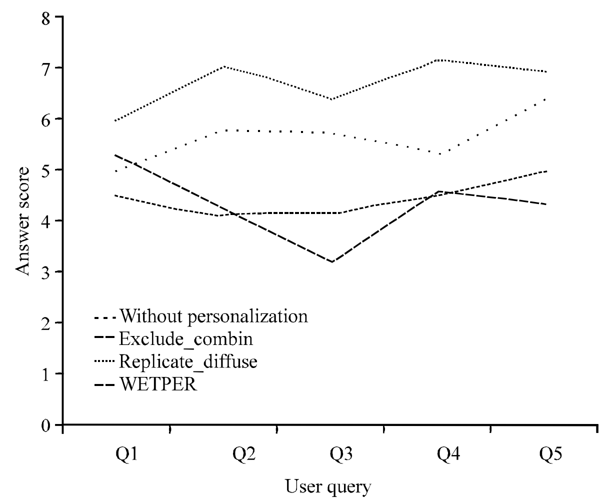

Fig. 6: Degree of difficulty of personalization algorithms

processed currently indicating the effectiveness of personalization. The user gives their answer score measure unaware of which algorithm is processed. This scenario makes the evaluation valid and the results are given in Fig. 7.

The third level of evaluation is done by knowledge driven evaluation method. In this method the proposed personalized results with their answer score are compared. Z-score method checks whether the algorithm works as good as other case also. Each personalized result whose answer score is positive is considered to construct association rules. The finding of Z-score value of sample data is given in Table 2. The movies with positive Z-score are considered as a transaction and their attributes are considered as items. Apriori algorithm is used to build preference rules of the form x_y. The dependency between x and y is verified by chi square test. The threshold of support and confidence measures is set to be 100%. The sample of preference rules is given in Fig. 8 and Table 3. This is knowledge base extracted from

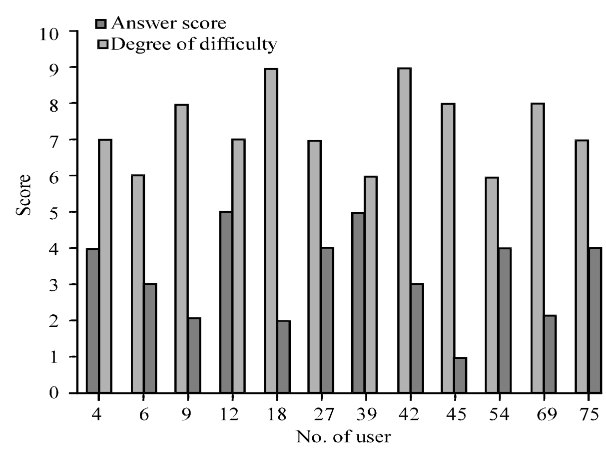

Fig. 7: Answer score vs degree of difficulty

personalized answer and stored in user profile. This result is further useful to reduce personalization time. The proposed algorithm reduces personalization time along with number of executions. The results are given in Fig. 9. Shortly, it is interesting that the user has

Table 3: Z score values of personalized movies

Query	Movies	U1	U2	U3	U4	U5
Q1	m1	0.8125	-1.0625	-0.4375	-0.4375	0.8125
	m2	0.1875	-0.4375	0.1875	0.1875	1.4375
	m3	-1.6875	0.1875	0.8125	0.8125	0.8125
	m4	0.1875	0.8125	-1.6875	1.4375	-0.4375
	m5	1.4375	-1.6875	-0.4375	0.1875	0.1875
Q2	m1	-1.6875	0.1875	0.8125	-0.4375	0.8125
	m2	-0.4375	0.8125	-1.6875	0.8125	-1.6875
	m3	0.8125	1.4375	0.1875	1.4375	0.8125
	m4	0.1875	-1.6875	0.1875	0.1875	-0.4375
	m5	0.1875	0.1875	-1.6875	0.8125	0.1875

```
Apriori
=======

Minimum support: 0.1 (5 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 27

Size of set of large itemsets L(2): 7

Best rules found:

 1. attribute_3=HenryJones 6 ==> attribute_0=MartinScorsese 6    <conf:(1)> lift:(1.74) lev:(0.05) [2] conv:(2.56)
 2. attribute_2=JohnMcIntire 5 ==> attribute_0=MartinScorsese 5    <conf:(1)> lift:(1.74) lev:(0.04) [2] conv:(2.13)
 3. attribute_3=TomHelmore 5 ==> attribute_0=MartinScorsese 5    <conf:(1)> lift:(1.74) lev:(0.04) [2] conv:(2.13)
```

Fig. 8: Preference rules

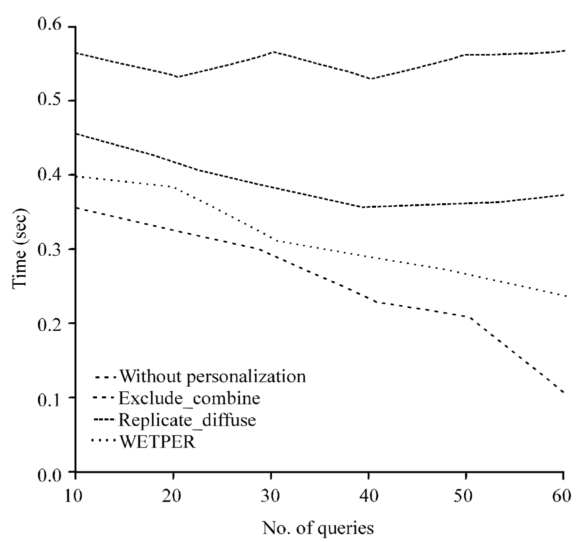

Fig. 9: Personalization time analysis

preferences for specific actors and he had given very high preferences for the groups of actors participating in the same movie as his wish list. Such fine grained taste made possible to discover not to miss movies with high answer score.

CONCLUSION

In this researcher, we propose a new personalization system with the algorithm WETPER. For this purpose, we propose a user preference model which is developed by multi criteria decision making approach for efficient personalization. Expressive preferences from user, robust performance and simple procedure for quantitative measure are the benefits of the proposed research. The movie selection objective is normally interdependent and sometimes conflicting. The conflicting scenario is not considered in the proposed systems. In this research, content based and rule based personalization approaches are applied to increase quality of dynamic personalization. This research can be extended with personalized movie recommendations for scalable problems applying similarity measures.

ACKNOWLEDGEMENTS

This research is based on the support of anna university under the grant of UGC. Any opinions findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of Anna Univeristy.

REFERENCES

Ali, N.H., I.A. Sabri, N.M.M. Noor and F. Ismail, 2012. Rating and ranking criteria for selected islands using Fuzzy Analytic Hierarchy Process (FAHP). Int. J. Applied Math. Inform., 1: 57-65.

Buvaneswari, N., D.S. Bose and D.A. Kannan, 2015. Personalization trends and issues in database perspective. Int. J. Appl. Eng. Res., 10: 13265-13271.

Koutrika, G. and Y. Ioannidis, 2004. Personalization of queries in database systems. Proceedings of the 20th International Conference on Data Engineering, March 30-April 2, 2004, IEEE, Greece, ISBN: 0-7695-2065-0, pp: 597-608.

Koutrika, G. and Y. Ioannidis, 2005. Personalized queries under a generalized preference model. Proceedings of the 21st International Conference on Data Engineering (ICDE'05), April 5-8, 2005, IEEE, Greece, ISBN: 0-7695-2285-8, pp: 841-852.

Koutrika, G. and Y. Ioannidis, 2010. Personalizing queries based on networks of composite preferences. ACM. Trans. Database Syst. TODS., Vol. 35, 10.1145/1735886.1735892

Pham, H.X. and J.J. Jung, 2013. Preference-based user rating correction process for interactive recommendation systems. Multimedia Tools Appl., 65: 119-132.

Qiu, F. and J. Cho, 2006. Automatic identification of user interest for personalized search. Proceedings of the 15th International Conference on World Wide Web, May 22-26, 2006, ACM, Edinburgh, Scotland, ISBN: 1-59593-323-9, pp: 727-736.

Saaty, T.L., 1986. Absolute and relative measurement with the AHP. The most livable cities in the United States. Socio Econ. Plann. Sci., 20: 327-331.

Shen, X., B. Tan and C. Zhai, 2005. Implicit user modeling for personalized search. Proceedings of the 14th International Conference on Information and Knowledge Management, Oct. 31-Nov. 05, ACM, New York, USA., pp: 824-831.

Yang, Y. and B. Padmanabhan, 2005. Evaluation of online personalization systems: A survey of evaluation schemes and a knowledge-based approach. J. Electr. Commerce Res., 6: 112-122.