Archives for the Philosophy and History of Soft Computing

INTERNATIONAL ONLINE JOURNAL OPEN ACCESS ISSN 2341-0183

© by the Authors. This Open Access article is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0) Use, distribution, and reproduction in any medium is allowed, provided the original work is cited.

Preferential Interpretation of Fuzzy Sets in Recommendation with Real E-shop Data Experiments

Ladislav Peska , Alan Eckhardt and Peter Vojtas*

Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic

Email: {peska,eckhardt,vojtas}@ksi.mff.cuni.cz;

Abstract

Our research aims to address the challenge of muti-user personalized recommendation. We hypothesize that data driven fuzzy technology can be used even in case of large, extremely sparse data and big number of users. In this paper we deal with this problem methodologically - technical details appear elsewhere.

To test our hypothesis, we first use fuzzy sets interpreted as user preference degree (rating). We used content-based learning of preferential fuzzy sets on each attribute and each user separately. To get the overall preference, we combine attribute preferences using tuning of parametric families of t-conorms . This way we estimate user's preferences for yet unvisited objects.

Our test subject was an e-shop operating in a crowded market that it did not dominate selling a product where individual purchases were highly infrequent, often only annual. User-item e-shop data were very sparse. The quality of the solution was measured on top-k in several order sensitive metrics.

Based on our experiments, we can conclude that using preferential fuzzy sets improved recommendations.

Submitted: 15/10/15. Accepted and Published: 31/12/15.

1 Introduction, Use-Case, Motivation ...

Today´s internet user faces a constantly changing and evolving environment, with new information about new products and services being continually made available. This ever-changing landscape requires automated web content processing, whether in the form of search engines, e-shops, aggregation shops and recommender systems. Our interest is the personalization of online information using preference mining and fuzzy (multiple value) as well as other (for example 5 star rating) models.

1.1 Our Use-Case

E-commerce is a rich field with various business models and variations of models available for study. One especially interesting area is the small to midsize e-shop (SME) with no repetition (nR). At a typical SMEnR, potential customers visit the site, spend a few moments clicking around and leave quickly when they do not immediately find something of interest.

Further complicating an SMEnR´s task is cookie volatility. Even when users must register prior to making a purchase decision, as these purchases are infrequent, often only once per annum, even if a post purchase is cookie placed, before the same user returns, that cookie has long since been deleted, making it impossible to associate a visitor with a past purchase even if the visitor is a returning customer. Thus, to be effective, an SMEnR recommender system must quickly identify a visitor´s need on the basis of very sparse, anonymous, user visit specific data primarily collected with scripts or other time volatile methods.

The SMEnR´s situation can be contrasted with sites such as Netflix. At this site, users pay a monthly fee after which they can stream any number of movies, yielding the site a rich collection of user-specific behavioral information. In addition, the site´s user rating system provides it with a robust source of explicit user preference data, which can be exploited to further fine-tune recommendations. Nevertheless, even with these advantages, when the site announced a million dollar competition to improve RMSE of its recommender system by 10% in 2006, it took until 2011 to reach this target.

This implies that, although a great deal of current recommender system research has been predicated on variants of winning method matrix factorization, continuing down this path will unlikely lead to fruitful results. For this reason, we decided on a different direction, moving forward with our own, data driven fuzzy technology.

1.2 Motivation and Related work

It is commonly accepted that personalized recommender systems increase user satisfaction and can lead to increased revenue for a site´s publisher. Our re-

search examines recommendation personalization in the context of an SMEnR´s competitive environment. As such, it runs parallel to research driven by companies such as Netflix with their different competitive environment. We are especially interested in user preference learning at a SMEnRs with non-registered users and the subsequent use of user preference learning in offering user´s top-k most preferred items.

In a wider context we have to go to the roots of fuzzy technology - there were always real data there. Origins of fuzzy sets are connected to seminal work of Lotfi A. Zadeh [1]. Original motivation was image/signal processing, where different grades of shade of image pixel were first practical examples of fuzzy sets.

Fuzzy set theory has developed rapidly with main applications in control. In the case of fuzzy controllers, the system can be in a continuum of states which are usually a combination of all possible values from an interval. That is why rules for a fuzzy controller have to describe the control action for all possible combinations of values. This is a main difference in comparison with computer science applications - there is only a finite set of products and/or services and not all possible combination of values appear in production (as a result of marketing considerations).

Our current research to a large part drew on H. J. Zimmermann's fuzzy set theory [2] ideas. His classification scheme helped us design our factual models as well as devise tests for these models. Our first move in this direction is commented in [3]. Our contribution is that we collected a large set of data [4], which can be used to establish fuzzy sets coding preferences.

Methodologically the most appealing is the work of E. Trillas [5] and together with R. Seising [6] . They put empirical research in fuzzy set technology in the philosophical context starting from logical positivism of Vienna Circle.

Last but not least, there is an important use of fuzzy technology outside the fuzzy community. The first application to be mentioned is the IBM Almaden project Garlic ([7]) where the first use of several fuzzy factors was considered and implemented to process image queries. Our two-step model is very similar. Finally, in [8] an optimal algorithm was used to find the top-k most preferred object without the necessity of examining the entire domain (assuming indexing and access).

Our approach is quantitative. Preference relations are the most used and studied qualitative approach. There are several methods proposed for learning preference relations from user feedback (to name a few we cite [9, 10]).

Last but not least we have to mention linguistic variables. To mention at least one contribution, the work of Vilem Novak et al. [11] takes into account the human input of a furnace operator to produce a fuzzy controller. Nevertheless, our human input is implicit and for a great number of users cannot likely be represented by a linguistic variable.

Technical details for methods of this paper are published in our work [12].

Links to new datasets are made public in [10]. Basic dataset on which we run our experiments here are available under [4].

Concerning recommender systems, the approaches are usually divided by user's feedback type - either explicit or implicit. Experiments are divided to off-line and on-line A/B tests (of course on real data). We consider only off-line experiments here.

For explicit user feedback we refer to the work of Alan Eckhardt [13]. Different benchmarks for learning preference using the Choquet integral were used in [14]. Nevertheless those data are in a sense single user data. As for implicit feedback we refer for [12] and [10].

Rule based systems are important because of their intuitiveness for human users. Inductive logic programming is a method to obtain a (many valued) logic program rules. This program may be very general. In our case, it represents rules that the user uses in the decision process. After the application of these rules to an object, the preference of that object can be obtained. Initial work was done in[15] (sound and approximately complete model of fuzzy logic programming and GAP), [16] and [17] on fuzzy ILP. Our model based on preferential fuzzy sets can be described by a rule based system ([12]).

1.3 Main challenges.

The first challenge is the extreme sparsity of data, with a single direct preference indicator: the purchase. In the user-item matrix, less than 0.01 percent are purchases. Here we distinguish between direct (explicit rating, purchase) and indirect (implicit behavior) preference indicators.

The second challenge is using script-collected user implicit behavior to generate indirect preference indicators.

Finally, even if these two issues are resolved, the preference model must be extended to all objects in order to calculate top-k most preferred values for the entire family of objects that could be recommended to the user.

1.4 Major contributions.

As a starting point we use fuzzy logic and the interpretation of fuzzy values as preference degrees based on Hajek's comparative notion of truth [18]. As a secondary starting point we used H. J. Zimmermann´s chapter [2] on empirical research fuzzy set technology. We also considered IBM project [7] where fuzziness is evaluated on several attributes.

We were interested in content-based recommendations at real world SMEnR-shop systems. We presented models for both explicit and implicit user interaction. Finally, we evaluated our models and methods in off-line experiments.

Our main contributions are to data driven (empirical) fuzzy set technology. Specifically:

• The use of fuzzy sets as a representation of preference - we call them preferential fuzzy sets.

• Content-based methods of learning preference on all items (induced by a fuzzy set) from a user' rating of a small sample of objects.

• A new model of aggregation based on tuning t-conorms from several parametric families.

• Off-line experiments on real world SMEnR e-shop data.

1.4.1 The paper is organized as follows.

In Section 1, we present our use-case and the motivation for our research. In Section 2, we present our two-step content-based model of learning preferences from a rating of a (small) set of items. Section 3 presents experiments on real production data from a SMEnR-shop. The paper concludes in Section 4.

2 Models and Methods

In this section we introduce preferential fuzzy sets. We distinguish two types of preferential fuzzy sets - acting on object and acting on domains of attributes.

We present a user preference model consisting of two steps - the user's preference on attributes and a fuzzy aggregation of those. These two steps combined together give the overal preference.

To learn user's preferences, we assume we have the user's rating of a small sample of visited objects. This can be obtained either by user's explicit rating or by interpreting implicit behavior on visited items.

2.1 Preferential fuzzy sets

We assume a set of objects $\mathcal{O}$ and data in an object - attribute form. Objects of user interest can have attributes $A_1, \ldots, A_N$ taking values in domains $D_1, \ldots, D_N$. An object $o \in \mathcal{O}$ has assigned attribute values $o.A_1, \ldots, o.A_N$.

We represent user's u preference both on items and attribute domains. We assume that preference of user u is a quasi linear ordering $<^u$.

Preference of user u on items $\mathcal{O}$ is represented as a **preferential fuzzy set** $r^u : \mathcal{O} \longrightarrow [0,1]$. The item o_1 is more preferred by the user u than the item o_2 iff $r^u(o_1) > r^u(o_2)$. In symbols

$$o_1 >^u o_2 \text{ iff } r^u(o_1) > r^u(o_2)$$

Preference of user u on values of attribute A_i is represented as a **preferential fuzzy set** $f_i^u : D_i \longrightarrow [0,1]$. The value $o_1.A_i$ is more preferred by the user u than the value $o_2.A_i$ iff $f_i^u(o_1.A_i) > f_i^u(o_2.A_i)$. In symbols

$$o_1.A_i \ >_i^u \ o_2.A_i \text{ iff } f_i^u(o_1.A_i) > f_i^u(o_2.A_i)$$

We call such fuzzy sets **preferential fuzzy sets** and we switch freely between the ordering and function.

The initial thought for a two-step model came from [7], [19] and [20]. Our user preference model is thus based on the idea of a content-based user preference model. User preferences depend on the value of attributes of objects, some of the values are more preferred and some are less. Some attributes are more important while others less. The overall preference of an object is a combination of preference degrees of the attribute values. Because of the division of preference into two levels, we will call this a "two-step model" preference model, as described in detail in [21].

In the first step, called *local preferences*, every attribute domain value of an object o is ordered using a fuzzy set f_i^u. These preferential fuzzy sets are also called objectives in economy, operational research and decision support. With these fuzzy sets, the original range of objects' attributes values (also occasionally called data–cube) $\prod_{i=1}^{N} D_{A_i}$ is transformed into a preference cube $[0,1]^N$. Moreover, we know that the object with preference values equal to $[1,...,1]$ is the most preferred object, though it probably does not exist in the real world. On the other hand, the object with values $[0,...,0]$ is the least preferred, and is more likely to exist in reality. For more details and examples see [22].

In the second step, called *global preference*, the attribute preferences are aggregated into the overall object score using an aggregation function $@^u : [0,1]^N \to [0,1]$.

Then the overall preference (by our model) will be equal to

$$r^u(o) = @^u(f^u_{A_1}(o.A_1), \ldots, f^u_{A_N}(o.A_N)).$$

Function r^u is also called utility function in economy, operational research and decision support.

An important part of this preference model is an assumption based on engineering practice. For a user u, we can assume only a small set of samples (visited objects) $\mathcal{S}^u \subseteq \mathcal{O}$ and a rating (preferential fuzzy set) $r^u : \mathcal{S}^u \longrightarrow [0,1]$.

Papers by A. Eckhardt ([13]) provide an additional view on this scenario. In our case, we only have implicit new user feedback data r^u, which is obtained by applying their preference relevance to the set of visited objects. Because of our assumption on SMEnR, this is the limit of what is known about the user.

2.2 Methods for learning preferential fuzzy sets

Our main goal is to find $\hat{r^u} : \mathcal{O} \longrightarrow [0,1]$, which will represent user's preference extended to all objects.

In our two step model, this means, given r^u to find $\hat{f}^u_{A_i}$ and $\hat{@}^u$ such that

$$\hat{r}^u(o) = \hat{@}^u(\hat{f}^u_{A_1}(o.A_1), \ldots, \hat{f}^u_{A_N}(o.A_N))$$

will be the best estimate of a user's preference for every o in the domain of $\mathcal{O}$.

From the ponit of view of economy, operational research and decision support these are optimization models. In economy, operational research and decision support we have usually only few alternatives, models are usually very complex (e.g. AHP) and are composed with human assistance. Our models have big number of alternatives (objects), models are much simpler and are composed automatically.

As to learning preferential fuzzy sets $\hat{f}^u_{A_i}$ and learning $\hat{@}^u$ (and consequently $\hat{r}^u$), in [12] and [22] there are described many possible methods and metrics that can be used for successful learning.

These consist of:

Methods for content based learning local preference M^c_l which projects ratings to attribute domains, splits to handling nominal and numerical domains and uses mainly Linear regression and Peak regression.

Content based learning global preference methods M^c_g use mainly learning of weighted average, Geometrical method and tuning a parametric family od t-conorms (this is described in detail in the next subsection).

Similar methods are used to interpreting implicit behavior as preference: behavioral analysis methods M^b_l and M^b_l (see [12, 22, 10]).

2.2.1 T-conorm aggregation tuning

As indicated by Zimermann and Zysno [23], users tend to aggregate good values OR-like and bad values AND-like. Because we are interested in ranking items at top-k, we concentrate on OR-like part and consider t-conorms for the aggregation. Since the aggregation depends on the user, we need to have a wide range of different t-conorms (also called s-norms). Therefore, we adopt parametrized families of fuzzy t-conorms. The idea is to have a parameter λ, which determines the exact form of t-conorm, which best fits the user. In general, the family of t-conorms is defined by an equation parametrized by the λ. For example, the Sugeno-Weber [24] t-conorms are defined as:

$$S^{SW}_\lambda(x,y) = min(x + y + \lambda xy, 1), \lambda \in (-1, \infty)$$

Having the preference of the attributes and the overall preference of the object, we can fine-tune the λ so that the t-conorm gives the best approximation of the preference.

We use associativity and commutativity for processing more than two attribute values. We have implemented several non-parametric t-conorms: basic t-conorms max, bounded sum and algebraic sum, parametrized t-conorm families Acel-Alsina, Dombi, Frank, Hamacher, Mayor-Torrens, Schweizer-Sklar, Sugeno-Weber and Yager.

Finding the right λ. Thanks to the monotonicity of t-conorms, we can use binary search when fine-tuning the λ parameter. We maintain three numbers:

the actual value of λ, the maximal value of λ and the minimal value of λ. The search starts by setting the maximal λ_{max} and the minimal value of λ_{min}, which depends on a particular family, in the case of Sugeno-Weber it is -1 and ∞ respectively. We used the empirically defined maximal value instead of ∞ e.g. 1000 in the case of Sugeno-Weber. During computation, we set a ratio of correctly estimated records $threshold = 0.3$. The maximum iteration number and the threshold were set heuristically. The reason is purely pragmatic, as it enhances the effectiveness of the computation in our framework. Then, the actual value of λ is set to an average of maximal and minimal. The pseudo-code is presented in Algorithm 1.

Algorithm 1 λ tuning

1: $\lambda_{min} = -1, \lambda_{max} = 1000$;
2: $i = 0, threshold \in [0,1], u \in \texttt{Users}, T = \texttt{TrainSetOfObjects} \subseteq \texttt{VisitedBy}(u)$;
3: **for** $i < 100$ or $err > threshold$ **do**
4: $\lambda = (\lambda_{min} + \lambda_{max})/2$;
5: $\text{getErrorRatio}(\lambda) = \frac{\left|\left\{o \in T : S_\lambda^{SW}(o_1,\ldots,o_n) < r^u(o)\right\}\right|}{|T|}$
6: $err = \text{getErrorRatio}(\lambda)$;
7: **if** $err \leq threshold$ **then**
8: $\lambda_{min} = \lambda$;
9: **end if**
10: **if** $err > threshold$ **then**
11: $\lambda_{max} = \lambda$;
12: **end if**
13: $i++$;;
14: **end for**

After all iterations on line 3 of the algorithm, we also have to check if the error is less than the required threshold. The for-cycle finishes only after finding the highest lambda with an error less than the threshold.

The function *getErrorRatio* on line 5 returns the ratio of the objects with incorrectly estimated preference. This means the objects, where the value of the t-conorm of preferences of the attributes is greater than or equal to the rating, i.e. $@^S(o) \geq r^u(o)$. For creating the set of incorrectly estimated objects, we apply the contrary $@^S(o) < r^u(o)$.

For each iteration of the algorithm, the following steps are executed:

1. The number of objects with incorrectly estimated preference $err(\lambda)$ is counted. This means the objects, where the value of the t-conorm of preferences of the attributes is lower than the rating, i.e., $@^S(o) < r^u(o)$, relative to the size of the training set.

2. If $err(\lambda) < threshold$, then this λ is okay and we can try to increase it. We set $\lambda_{min} = \lambda$ and then the new $\lambda = (\lambda_{min} + \lambda_{max})/2$, effectively increasing the value of λ.

3. Else, the λ performs badly, so we would need to decrease it. We set $\lambda_{max} = \lambda$ and then the new $\lambda = (\lambda_{min} + \lambda_{max})/2$, effectively decreasing the value of λ.

For some of the families of fuzzy functions, we have to use the value $1/\lambda$ instead of λ to maintain the increasing property with regards to λ. Some of the families have different boundaries than 0, ∞ - the boundaries in the algorithm have to be set accordingly.

3 Experiments

3.1 Source of real data

We will demonstrate properties of a typical SMEnR-shop on our experimental site: a Czech travel agency. The website contains approximately 1000 tours which differ in type, destination, price, services, etc. The site does not enable users to explicitly rate objects. Using a web server application, we collect user behavior as they search for a tour. The site does not require users to register in order to search tours and identifies them with a cookie.

We were interested in users that actively searched for items, so we only used data on users (identified with a cookie) that visited at least four objects. All of the visits in the data collection period numbered 10^5. Those actively searching (visiting at minimum 4 objects) numbered 10^3, while purchases numbered 10^2 users (hence the items-user matrix size of 10^6). The usual user purchases no more than a single tour per year, so we can collect user data related to one intended purchase per user (the decision process is short enough that a discrete purchase can be associated with a unique user, but the time lag between consecutive purchases is so long that a unique user usually cannot be associated with multiple purchases (except for the occasional exception where multiple purchases occur during a short span of time)). The resulting dataset [4] contains records about user behavior (feedback) on each object.

Purchases in these behavioral data are very sparse - less than 0.01% of user $\times$ item (two orders of magnitude below the Netflix direct preference indicator). The remainder of implicit feedback has a density more or less equal to the Netflix density.

In the SMEnR-shops without an explicit feedback (such as ours) we do not have any direct information about user preference (prior to the purchase decision) and so this information must be inferred (empirically). We have adopted a common sense approach, assuming that the user positively prefers purchased object(s). Because of the sparsity of data, we decided to use the behavior data of all users to derive one joint indicator of preference.

Table 1: Attributes of objects (tours).

Column	Description
tourID	Tour identifier
tourType	Type of the tour
country	Destination country
region	Destination more detailed
typeOfBoard	Enum of board types
typeOfTransport	Enum of transport types
typeOfAccomodation	Enum of accommodation types
listOfInformation	IDs list of additional information
price	Base price for the tour

Table 1 contains the data structure of the e-shop at which these methods were tested. Some domains can be described by following examples: Type of the tour e.g. sightseeing, spa holiday, winter skiing etc.; More detailed specification of the destination e.g. Barcelona or Makarska Riviera; Enum of board types (e.g. breakfast, half board); Enum of transport types (e.g. bus, plane); List of IDs of additional information about e.g. visited places.

Our deep conviction is that theoretical results must be validated with empirical data. On this point, we are in full agreement with H. J. Zimmermann in [2].

3.2 Experiment Settings and Success Metrics

We have conducted a series of experiments with the travel agency dataset [4] described in Table 1. Various instances of our two-step model

$$\mathbf{m} = [m_1, m_2, m_3, m_4] \in \ M_l^b \times M_g^b \times M_l^c \times M_g^c$$

were compared with several baselines where both methods were replaced with standard machine learning methods. In Table 2 we use following abbreviations: SVM-Support vector machines, ML.Perc.-Multi layer perceptron and M5P-Decision trees or simple baseline methods (e.g. single binary or numeric feedback, random ordering). The methods were compared in terms of nDCG, the presence at top-k and the average position of preferred objects - i.e. we can measure improvement.

The experiments can be divided into three parts. In the *first*, different approaches for implicit behavior analysis were evaluated. In the *second*, we evaluated methods for content-based learning using input from implicit behavior analysis. In the *final* part our aim was to find the best combined approach.

Based on our testing we can state that the testing/evaluating portion of our model (behavioral or content based) is valid only if the other portion is

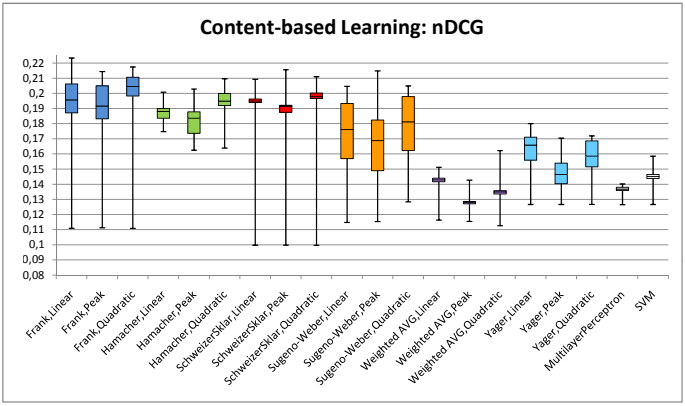


Figure 1: Overall results of combined local and global content based preference learning methods (M_l^c, M_g^c): nDCG (e.g. Frank $\in M_g^c$, Peak $\in M_l^c$).

averaged over all considered methods (alternatively only the best method is taken into account).

We let each recommendation method rate each object and order the objects according to their rating. In our scenario of preferential fuzzy sets, the exact computed rating is irrelevant, as the value itself does not bring any useful information to the user and we do not plan to present it to him/her. The recommended objects and their presentation is only affected by the ordering deduced from the predicted rating. Therefore we did not use predictive accuracy metrics such as RMSE, MAE or correlation coefficient, but focused on ranking prediction metrics.

Our primary success metric was a normalized discounted cumulated gain (nDCG). The nDCG measure represents the degree of usefulness of a recommendation weighted by its position in the top-k list. The formula for computing DCG is:

$$DCG_k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{log_2(i)}$$

where k is the size of the top-k list, rel_i is the relevance of i^{th} element on the list. In our situation, the relevance is only 1 for a purchased item and 0 for a non-purchased item. The normalization is done by dividing the DCG of the result by the DCG of the ideal ordering in order to scale the results to $[0, 1]$.

Results depicted in Figure 1 show a combination of both-content based methods. In general, the best global methods are Frank, Hamacher and Schweizer-Sklar (which is less stable against M^b). These three methods are not comparable in general. Frank is better in nDCG (it means it will be better for smaller top-k, which is crucial for our e-shop) and Hamacher and Schweizer-Sklar are

better in Average position (Hamacher in average of M^b, Schweizer-Sklar for the best M^b). The latter are better suited for bigger top-k. All the t-conorms are better than the baselines (SVM a Multilayer Perceptron).

Table 2: Overall results for combination of best behavior and content-based methods, nDCG. Best result for each column is underlined, best for each row is in bold (bfBeh. stands for best for Behavioral Analysis and bfCont. stands for best for Content based methods. For other abbreviations see section 3.2

		Behavioral Analyzis					
	nDCG	Ml.Perc.	SVM	M5P	Sug.-W.	Hamach	bfCont.
Content based	Ml. Perc.	0,127	0,137	0,140	0,137	0,139	0,140
	SVM	0,127	0,149	**0,159**	0,143	0,144	0,159
	WAVG	0,116	0,146	0,151	0,143	0,144	0,151
	Yager	0,127	0,135	0,167	**0,175**	0,175	0,175
	Sug.-W.	0,128	0,198	**0,206**	0,190	0,183	0,206
	Schw.-Skl.	0,100	0,200	**0,215**	0,197	0,197	0,215
	Hamacher	0,175	0,200	**0,210**	0,195	0,194	0,210
	Frank	0,111	0,198	0,214	**0,223**	0,217	0,223
	bfBeh.	0,175	0,200	0,215	**0,223**	0,217	**0,223**

The last part of the experiments section is dedicated to finding the overall best combination of M_l^b, M_g^b, M_l^c, M_g^c methods. The results are summarized in Table 2 (nDCG). To save space, method names are abbreviated: Ml. Perc. = Multilayer Perceptron, Binary = Single binary feedback, Numeric = Single numeric feedback, Sug.-W. = Sugeno-Weber, WAVG = Weighted Average and Schw.-Skl. = Schweizer-Sklar. Best for Content(bfCont.) column and Best for Behavior(bfBeh.) row show results of the best method with the content (behavior) method fixed to the one named in the current row (column). Note that such a method might not be presented in the rest of the table.

4 Conclusions and Future Work

We theorized that a content-based recommender system would improve web site performance even if implicit user data were sparse. We tested this hypothesis on real world production data from an e-shop. Our testing yielded a significant result, which confirmed our hypothesis. Based on our testing, we can conclude that data driven preferential fuzzy set technology can improve performance even if input data are not robust.

Our main contributions are:

• interpretation of fuzzy sets as preference indicators

• methods for learning preferential fuzzy sets from a small sample of user's rating

• real offline experimentation with real data with measurement of the improvements and comparison with common data mining methods

• evidence that fuzzy technology can be used to produce meaningful recommendations even in an environment with extremely sparse data

We hope this can be a contribution to much deeper discussion started by E. Trillas and R. Seising in [5] and [6].

Acknowledgment

This work was supported by Czech grant P46. Authors would like to thank the e-shop for providing data [4].

References

1. Zadeh LA: **Fuzzy sets**. *Information and Control* 1965, **8**(3):338353.
2. Zimmermann HJ: **Chapter 16 - Empirical Research in Fuzzy Set Theory**. In *Fuzzy Set Theory and Its Applications, 4th edition*, Kluwer Academic Publishers 2001:443–475.
3. Vojtas P: **Fuzzy logic as Hajek's comparative notion of truth and applications in preference modeling**. In *In Witnessed Years: Essays in Honour of Petr Hajek*. Edited by Cintula P, College Publications 2009:345–360.
4. Peska L: **Basic travel agency data** 2012, : 21 MB, [http://www.ksi.mff.cuni.cz/~vojtas/hw/res/aphsc2016/basic_user_behavior.sql.zip].
5. Trillas E: **How I would like to foresee the future of theoretic fuzzy logic**. In *Proc. IFSA-EUSFLAT 2015*, Atlantis Press 2015:10 pages.
6. Trillas E, Seising R: **On Meaning and Measuring: A philosophical and historical View**. *Agora: papeles de Filosofa* 2015, **34**(2):1–23, [http://www.usc.es/revistas/index.php/agora/article/view/2096].
7. Cody W, al: **Querying Multimedia Data from Multiple Repositories by Content: the Garlic Project**. In *Proc. IFIP 2.6 VDB-3* 1995:17–35.
8. Fagin R, Lotem A, Naor M: **Optimal aggregation algorithms for middleware**. *J. Computer System Sciences* 2003, **66**:614 – 656.
9. Hüllermeier E, Fürnkranz J: **Learning Preference Models from Data: On the Problem of Label Ranking and Its Variants**. In *Preferences and Similarities*. Edited by Riccia GD, Dubois D, Kruse R, Lenz H, Springer-Verlag 2008:283–304.
10. Peska L, Vojtas P: **Using Implicit Preference Relations to Improve Recommender Systems**. *to appear Special Issue on Recommender Systems of the Journal on Data Semantics* 2016, :16 pages.
11. Belohlavek R, Novak V: **Learning rule base in linguistic expert systems**. *Soft Comput.* 2002, **7**:79–88.
12. Peska L, Vojtas P: **Interpreting Web Shop User's Behavioral Patterns as Fictitious Explicit Rating for Preference Learning**. In *In Proc. RuleML 2014*, LNCS 8620. Edited by Bikakis A, Springer 2014:251–265.
13. Eckhardt A: **Similarity of users' (content-based) preference models for Collaborative filtering in few ratings scenario**. *Expert Systems with Applications* 2012, **39**:11511–11516.

14. Tehrani AF, Cheng W, Hüllermeier E: **Preference Learning Using the Choquet Integral: The Case of Multipartite Ranking**. *IEEE T. Fuzzy Systems* 2012, **20**(6):1102–1113.

15. Krajci S, Lencses R, Vojtáš P: **A comparison of fuzzy and annotated logic programming**. *Fuzzy Sets Syst.* 2004, **144**:173–192.

16. Horváth T, Vojtáš P: **Induction of Fuzzy and Annotated Logic Programs**. In *ILP06* 2007:260–274.

17. Horváth T: **A Model of User Preference Learning for Content-Based Recommender Systems**. In *Computing and Informatics*, *Volume 28* 2009:453–481.

18. Hájek P: *Metamathematics of Fuzzy Logic (Trends in Logic)*. Springer 2001.

19. Nepal S, Ramakrishna M: **Query processing issues in image (multimedia) databases**. In *ICDE 1999*, Sydney, Australia: IEEE Computer Society 1999:22–29.

20. Güntzer U, Balke WT, Kießling W: **Optimizing Multi-Feature Queries for Image Databases**. In *VLDB '00: Proceedings of the Twenty Sixth Conference on Very Large Databases*, Cairo, Egypt 2001:419–428.

21. Eckhardt A, Pokorný J, Vojtáš P: **A system recommending top-k objects for multiple users preferences**. In *2007 IEEE Conference on Fuzzy Systems*, London, United Kingdom 2007:1101–1106.

22. Peska L, Eckhardt A, Vojtas P: **Interpreting Implicit User Behavior for E-shop Recommendation with Families of Fuzzy T-conorms**. *Technical Report* 2013, :36 pages, [http://www.ksi.mff.cuni.cz/~vojtas/hw/res/aphsc2016/FuzzyTransactions0114.pdf].

23. Zimmermann HJ, Zysno P: **Latent connectives in human decision making**. *Fuzzy Sets Syst.* 1980, **4**:37 – 51.

24. Weber S: **A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms**. *Fuzzy Sets Syst.* 1983, **11**:103–113.