Computer Science and Information Systems 17(3):983–1006 https://doi.org/10.2298/CSIS200115030K

Visual E-Commerce Values Filtering Framework with Spatial Database metric ⋆

M. Kopecky and P. Vojtas⋆⋆

Dpt. of Software Engineering, Faculty of Mathematics and Physics, Charles University, Prague
{kopecky|vojtas}@ksi.mff.cuni.cz

Abstract. Our customer preference model is based on aggregation of partly linear relaxations of value filters often used in e-commerce applications. Relaxation is motivated by the Analytic Hierarchy Processing method and combining fuzzy information in web accessible databases. In low dimensions our method is well suited also for data visualization.
The process of translating models (user behavior) to programs (learned recommendation) is formalized by Challenge-Response Framework ChRF. ChRF resembles remote process call and reduction in combinatorial search problems. In our case, the model is automatically translated to a program using spatial database features. This enables us to define new metrics with visual motivation.
We extend the conference paper with inductive ChRF, new representation of user and an additional method and metric. We provide experiments with synthetic data (items) and users.

Keywords: E-commerce values filtering, spatial database, recommender systems, user preference learning, experiments, synthetic data, spatial evaluation measures

1. Introduction, motivation, contributions

Our main motivation are recommender systems so far they point us to interesting items on e-commerce sites. Such a system has to be personalized to each user/customer preferences separately. We are modeling user by its behavior (rating) on visited items and expect (inductive) programs to be able to generalize this behavior to all items. We measure success of this generalization by several spatial database metrics.

In representation of customer preferences we restrict to Fagin-Lotem-Naor-class of models (FLN models). R. Fagin, A. Lotem and M. Naor in their paper [8] described a (middleware) top-k query system where each object in a database has m scores, one for each of m attributes (somewhere out in the web) that represent relevance degrees. To each object is then (on the middleware) assigned an overall score that is obtained by combining the attribute scores using a fixed monotone combining rule. This approach enables multicriterial ordering.

⋆ This is an extended version of a conference paper [13] Kopecky M., Vojtas P. (2019) Graphical E-Commerce Values Filtering Model in Spatial Database Framework. In: Welzer T. et al. (eds) New Trends in Databases and Information Systems. ADBIS 2019. Communications in Computer and Information Science, vol 1064. Springer, Cham. pp 210-220

⋆⋆ Supported by Czech grant Progres Q48,

We work on idea to use these types of models for e-commerce value filtering. Our goal is to present intuitiveness of visual features of these models and automated translation to programs. It is also suitable for implementation of ”best match” in case, when system has to respond ”we were not able to find any matching results, but we found these similar listings for you” (see e.g. [15]). This motivated softening, relaxing value filtering. Original motivation for [8] was multimedia search where attributes are inherently fuzzy (hence relaxed). Motivation comes also from combination of fuzzy information as developed along the IBM Almaden project Garlic ([7]) and top-k querying of web accessible databases ([15]). Degree of relaxation is motivated also by the AHP - Analytic Hierarchy Processing method ([20]). In AHP relaxation is often done by a domain expert - here we need fast automatic response.

The process of translating models to programs is formalized by Challenge-Response Framework ChRF. ChRF resembles many problem reduction scenario.

To have our models intuitive, we visualize them - the price we have to pay is we can depict only two or three dimensions. The idea is to use most important attributes and/or some aggregated ones. In our case, the model is automatically translated to a program using spatial database features.This enables us to define new metrics with spatial motivation.

We provide offline experiments with synthetic data (items) and users. This is an extended version of our conference paper [13]. This paper was extended with additional model, method, metric, experiments and lot of new comparisons.

Main contributions of this paper are those of the original conference paper and new ones (+ sign denotes additional extensions):

- Spatial representation of linear Fagin-Lotem-Naor model for most important attributes
- Challenge-Response Framework ChRF for translating models to programs - original conference paper version + is extended with inductive ChRF
- Visual aspects of our model enabling e-commerce customer user studies
- Spatial metrics - area based + new item size based
- Spatial methods of user preference learning - pivot based + new
- Representation of user using synthetic data or additionally a new representation using convex hull of most preferred objects
- Prototype and four types of experiments - two methods versus two metrics over multiple users
- Experiments with synthetic data (items) and users

Paper is organized as follows: Section 2. deals with visual, linear, multiuser, content based Fagin-Lotem-Naor class of models. Section 3. describes basic Challenge Response Framework ChRF and its inductive version. Section 4. is on data, methods, spatial metric and experiments. Here we describe practical challenge-response construction for our experiments. Detailed description of spatial SQL computing our metrics is also provided. We briefly mention related research and add conclusions and future work.

2. Visual, linear, multiuser, content based FLN models

First we describe Fagin-Lotem-Naor class of models. In [8] they assume, that each object o has assigned m-many attribute scores $x_i^o \in [0;1]$. A typical example is that this score is coding order of access when querying multiple web-accessible databases. Combination function $t : [0;1]^m \longrightarrow [0;1]$ is assumed to fulfill: $t(0,\dots,0) = 0$, $t(1,\dots,1) = 1$ and t preserves ordering, i.e. if $x_j \leq_{[0,1]} y_j$ for all $1 \leq j \leq m$ then

$$t(x_1,\dots,x_j,\dots,x_m) \leq_{[0,1]} t(y_1,\dots,y_j,\dots,y_m) \tag{1}$$

Because of this inequality we call this function monotone. The overall score of object o is $t(o) = t(x_1^o,\dots,x_j^o,\dots,x_m^o)$.

In [14] we have described a class of LT-linear triangular models which is a subclass of FLN models. Especially, such a model can be generated by domain preference functions $f_i : D_i \longrightarrow [0,1]$ and $x_i^o = f_i(o.A_i)$ and a combination function t. To be able to process such models by a spatial database it is suitable to have these functions linear or at least partially linear. Special case of such domain preference function are triangular (or trapezoidal) functions which can be considered as softening / relaxing of value filters in e-commerce.

In Figure 1 there are two such preference algorithms - the green one (user u) α (given by f_1, f_2 and $c^u_{2/3} = (t^u)^{-1}(2/3)$) and the red algorithm (user v) β (given by g_1, g_2 and $c^v_{2/3} = (t^v)^{-1}(2/3)$). We can think of α as being the correct model and of β as being the computed (learned) model. This should illustrate a decision maker (e-shop owner, customer) situation when comparing two decision alternatives. In this figure, attribute preferences are triangular - this is a softening of one element value filter. There are two aggregations $\frac{2*x_1+x_2}{3}$ and $\frac{2*x_2+x_1}{3}$ represented by $2/3$ contour line in the preference cube.

Main feature of our model is visualization of these contour lines in the data cube. Each contour line corresponds to the polygon in the data cube. Using attribute preferences can be endpoints of preference cube contour lines traced back (by respective horizontal and vertical lines and their intersections with attribute preferences) to respective attribute values in domains (in our figure there are always two of them). This should be intuitive for user / customer. Here we see areas in data cube (with preference bigger or equal to $2/3$) and area of their intersection.

Another aspect of illustration of Figure 1: - the green one (user u) α (given by f_1, f_2 and $(t^u)^{-1}(2/3)$) describes deduction and we can calculate polygon in data cube having preference at least 2/3.

$$P^u_{\alpha,2/3} = [t^u(f_1^u(o.A_1), f_2^u(o.A_2))]^{-1}\,(\geq 2/3) \tag{2}$$

Similarly to polygon for the red algorithm (user v) β describes the inductive procedure - we have only few (explicit ratings) of some visited objects (here A is rated 0.3 ... F is rated 0.7) and we try to specify either g_1^v, g_2^v and $(t^v)^{-1}(\geq 2/3)$ or directly polygon $P^v_{\beta,2/3}$. Instead of 2/3 we can consider arbitrary level of preference $h \in [0,1]$.

Let us stress here that this inductive task can not be considered as high dimensional regression, because we need ordering on each server separately (in typical situation of web accessible databases ([15])).

We will consider three metrics motivated by this spatial representation of relaxed value filters - the area metrics, number of data points metric and metric calculating with average distribution of produced items given by a measure. Having the "correct" model P^u_h given by data and computed (learned) $\hat{P}^u_{\alpha,h}$, it is natural to ask for precision and recall of such models at different levels.

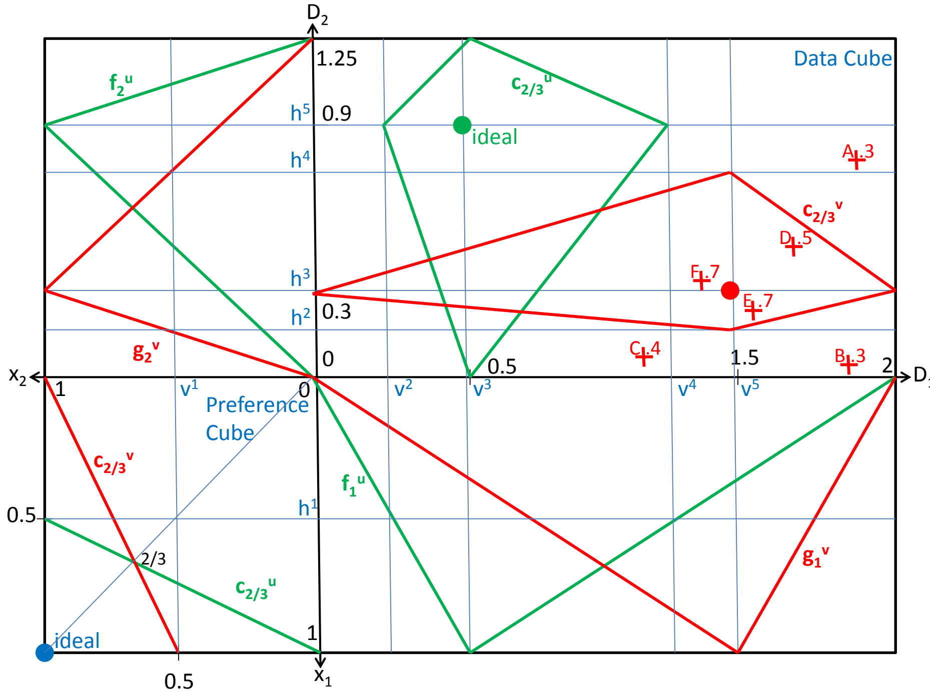

Fig. 1. We illustrate both deductive (in green) and inductive (in red) aspects of our linear FLN-class model. More description in the text.

We will provide experiments with two metrics, area based metric and item size based metric. Area precision and recall of algorithm α for user u at preference level h are expressed as

$$AP^u_{\alpha,h} = \frac{area(P^u_h \cap \hat{P}^u_{\alpha,h})}{area(\hat{P}^u_{\alpha,h})} \quad (3) \qquad AR^u_{\alpha,h} = \frac{area(P^u_h \cap \hat{P}^u_{\alpha,h})}{area(P^u_h)} \quad (4)$$

When having some items data D, instead of area metrics we can calculate these fractions by number of data points in each area and we get item size based precision and recall.

$$IP^u_{\alpha,h} = \frac{|D \cap P^u_h \cap \hat{P}^u_{\alpha,h}|}{|D \cap \hat{P}^u_{\alpha,h}|} \quad (5) \qquad IR^u_{\alpha,h} = \frac{|D \cap P^u_h \cap \hat{P}^u_{\alpha,h}|}{|D \cap P^u_h|} \quad (6)$$

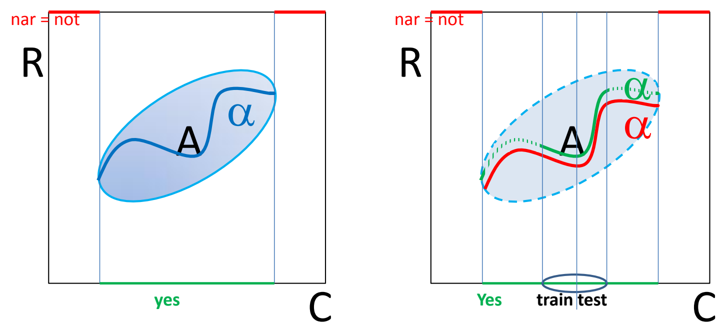

Fig. 2. Illustration of basic (left) and inductive (right) ChRF situation

Sometimes we know the measure μ of distribution of production of items (data points). In this case instead of area or number of data points we can use distribution based precision and recall.

$$DP^u_{\alpha,h} = \frac{\int_{P^u_h \cap \hat{P}^u_{\alpha,h}} x d\mu}{\int_{\hat{P}^u_{\alpha,h}} x d\mu} \quad (7) \qquad DR^u_{\alpha,h} = \frac{\int_{P^u_h \cap \hat{P}^u_{\alpha,h}} x d\mu}{\int_{P^u_h} x d\mu} \quad (8)$$

3. ChRF - Challenge Response Framework

In this chapter we will deal with a procedure known from many environments - as problem reduction, compilation based approach, many-one reduction, client server RPC, requester-helper in agent systems, question-answer - in all of these reduction converts instances of one decision/search problem into instances of a second decision/search problem. The solution of this second problem is then transformed to a solution of the starting one. We will adopt a general term for these - challenge-response framework.

In this chapter we will describe the Challenge Response Framework which will serve as a formal tool for our future activities. First we describe the case when the requester does not know anything about solution and takes whatever the solver answers. Second we will consider the case when we have already some partial information (usually examples data in supervised learning). This can help to choose the best helper/solver.

Specific form of Challenge Response Framework suitable for our multi-user data and spatial data metric will be introduced in experiments chapter.

3.1. Basic Challenge Response Framework

Motivated by an old mathematical idea from [21] we use the terminology arousing from [4] and define the Challenge Response Framework ChRF. Our goal is to use ChRF as a formal framework for description of translation of models to programs.

Challenge Response Situation $S = (C, R, A)$ consists of a set C of challenges, set of responses R and an acceptability relation $A \subseteq C \times R$ (can be preferential). For an

$c \in C, r \in R$ we read $A(c,r)$ as "r is an acceptable response (in some degree) for challenge c". We assume, that each set R contains also a special element nar representing "there is no acceptable response".

We assume: $A(c, nar)$ is equivalent to $(\forall r \in R\backslash\{nar\})(\neg A(c,r))$. The set $R\backslash\{nar\}$ are meaningful responses, nar is like logical "not" in combinatorial decision problems. See, Fig. 2 left.

Challenge Response Reduction of a situation $S_1 = (C_1, R_1, A_1)$ to a situation $S_2 = (C_2, R_2, A_2)$ consists of a pair of functions (f^-, f^+) such that $f^- : C_1 \longrightarrow C_2$, $f^+ : R_2 \longrightarrow R_1$, such that $f^+(nar_2) = nar_1$, $f^+(r_2) = nar_1$ implies $r_2 = nar_2$ and following holds:

$$(\forall c_1 \in C_1)(\forall r_2 \in R_2)(A_2(f^-(c_1), r_2) \longrightarrow A_1(c_1, f^+(r_2)) \qquad (9)$$

see Figure 3. Let us stress that we require here only implication. In decision problems we need equivalence, in search problems we have to prevent from fake reduction (when the implication can be true even when there is no acceptable response). This is the main reason we have introduced the "nar" element and we require that 9 holds also for "nars".

Note that $A_2(f^-(c_1), nar_2) \longrightarrow A_1(c_1, nar_1)$ is equivalent to $\neg A_1(c_1, nar_1) \longrightarrow \neg A_2(f^-(c_1), nar_2)$ and this to $(\forall c_1 \in C_1)(\exists r_1 \in R_1 \setminus \{nar_1\})(A_1(c_1, r_1)) \longrightarrow (\exists r_2 \in R_2 \setminus \{nar_2\})(A_2(f^-(c_1), r_2))$. See, Fig. 3

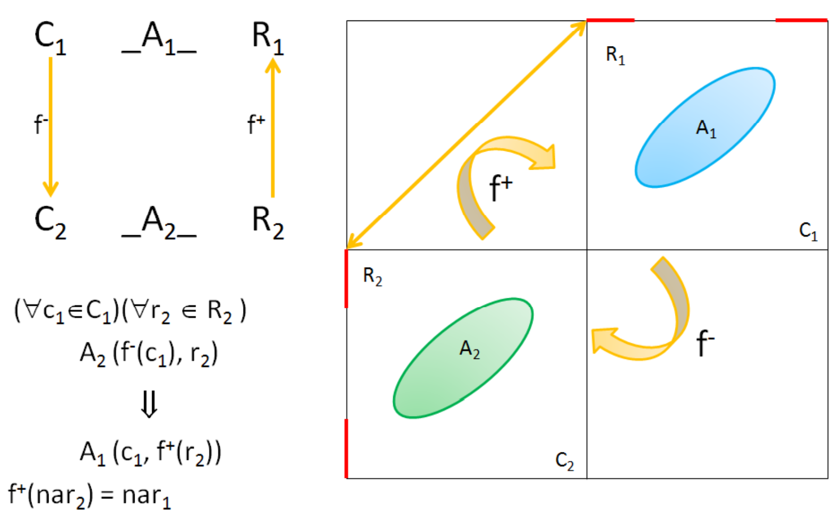

Fig. 3. Illustration of basic ChRF reduction.

Notice that we can consider two levels of understanding challenges and responses. First is on the level of each situation itself. Each situation has own challenge instances and responses. A level higher we can consider the situation S_1 as a sort of challenge (somebody requests help) and the situation S_2 provides help (response). This can form a second order structure $\mathcal{S}$ where challenges $C_{\mathcal{S}}$ are ChR situations, the same responses $R_{\mathcal{S}}$

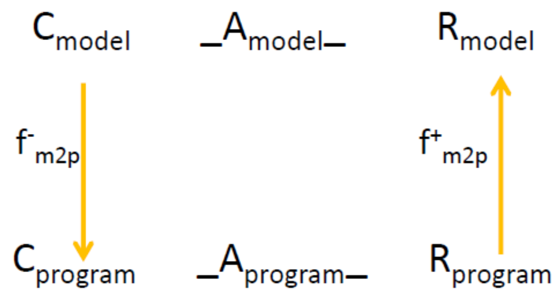

Fig. 4. Model to program transformation as ChRF reduction.

are ChR situations and acceptability relation A_S consists of pair (S_1, S_2) such that S_1 is reducible to S_2. We are not going deeper into this.

This ChRF reduction can be used to formally represent transformations of models to programs, see Figure 4. Notice that we cannot grasp the reality in full complexity. Here we understand model as a model of reality represented by available data, consisting of various signals, user behavior, etc.

In our model situation challenges can come from user interaction with visual interface (so far not implemented) were the user can slide ideal points, descent of relaxation, combination function etc. Transformation f^-_{m2p} has to be fully automated and sends these actions to program situation challenges (inputs). In what follows we describe implementation of f^-_{m2p} in Oracle Spatial. Challenges of the program situation are input and responses are output. So far we have implemented two user representations. First calculation of polygons of contour lines from user model in synthetic data set. Second, we represent user by a convex hull of most preferred items (after possible user changes behavior). Respective visualization sends f^+_{m2p} back to model response - a possibly user visual interface .

3.2. iChRF - inductive Challenge Response Framework

In this section we describe Challenge Response Framework with partial information. Our main motivation is to have a framework for handling supervised learning. We restrict ourselves to regression tasks, where we assume the example set in the form of a function $E(\bar{x}) = y$ with a vector of independent variables $\bar{x}$ and a dependent variable y. See Fig. 2 right.

A straightforward usage of ChRF is not fully satisfactory because we need a machinery for calculation error of learning (in contrast to basic ChRF where the truth of implication is understood as in mathematics).

We are not going to describe various aspects of the learning process. Here we assume that we have a class of machine learning algorithms/programs Π. Π can consist e.g. of linear regression, logistic regression, decision tree, SVM, ... Each $\alpha \in \Pi$ has a set of hyperparameters H^α and for each $h \in H^\alpha$ there is a program α^h which is a candidate for generalization of set of examples. This program generates a program situation

$$S_{\alpha^h} = \left(In_{\alpha^h}, Out_{\alpha^h}, \alpha^h\right). \tag{10}$$

iChR reduction will be used to calculate the quality of approximation of the example set. Reduction will reduce the challenge of generalizing the example data set (model) to program situation.

Assume we have an example set E, then the model situation looks like

$$S_E = (C_E, R_E, A_E) \tag{11}$$

where C_E is the set of independent variable vectors $\bar{x}$ from the domain of example set E. To define response set we need a metric ρ_i for calculating individual error (sometimes it depends on the algorithm ρ_i^α). R_E is the set of triples (usually of real numbers) $(y, \hat{y}, e))$. The acceptability relation is defined as follows:

$$A_{E,i}(\bar{x}, (y, \hat{y}, e)) \text{ iff } E(\bar{x}) = y \text{ and } e = \rho_i(y, \hat{y}). \tag{12}$$

An Inductive Challenge Response Reduction from S_E to S_{α^h} consists of two identity mappings f^-, f^+, where f^+ maps estimate $\alpha^h(\bar{x}) = \hat{y}$ to the second coordinate of R_E.

The trickier part is calculation of degree of validity of the reduction. The basic ChR reduction is universally quantified over challenges. Here it also makes sense to ask, how good is $p^{\alpha,h}$ in approximating the example set E? The truth value of the universally quantified statement ($\forall \bar{x} \in C_E$) will be calculated by an aggregate measure ρ_a (as usual in data mining, again maybe depending on α). So the quality of reduction is measured by

$$\rho_{a,i}^{\alpha,h}(E) = \rho_a\left(\left\{e : \bar{x} \in C_E, \text{ and } A_{E,i}\left(\bar{x}, \left(E\left(\bar{x}\right), \alpha^h(\bar{x}), e\right)\right)\right\}\right). \tag{13}$$

If measures depend on α we write $\rho^{\alpha,h}$.

We can imagine to let run this approximations in parallel (over all algorithms, parameters, Crossvalidation splits and tests) and the winner will be $argmin_{\alpha \in A} argmin_h \rho^{\alpha,h}$....

Example. If $\rho_i(y, \hat{y}) = |y - \hat{y}|$ then

$\rho_{ABS} = \sum_{C_E} |y - \hat{y}|$, $\rho_{AVG} = \frac{\sum_{C_E} |y-\hat{y}|}{|C_m|}$, similarly $\rho_{MAX} = \max_{C_E} |y - \hat{y}|$;

if $\rho_i(y, \hat{y}) = (y - \hat{y})^2$ then $\rho_{RMSE} = \sqrt{\sum \frac{(y-\hat{y})^2}{|C_E|}}$.

In this paper we will consider aggregated error on polygons (contour lines) in data cube.

4. Data, methods, spatial metric and experiments

In this section we will describe experiments with two metrics. First area based and the second based on the number of items in respective polygons.

4.1. Data

Our experiments are two fold. First are pure synthetic data where we know that data are generated by our model (hence inductive part measures true ability to find the model from global preferences). In the second part we mimic the situation that we have user's behavior, i.e. preferences of visited objects.

Data items are not evenly distributed, but form four distinct clusters near corners of the data cube.

Fully synthetic data for experiments In [14] we have studied a subclass of FLN models - Linear combination of Triangular attribute modes - LT-models. LT-models can be generalized to trapezoidal models.

We consider pivot based learning from [14] with some stochastic noise. Results are evaluated through new metrics calculating spatial data characteristics of LT-models.

Our experiments are using user and item data from [14] together with sparse preference matrix $M = \{< 0; 1 > \cup null\}^{|U| \times |I|}$ where U is set of users and I is set of items.

Training data for each user $u \in U$ contain only the corresponding row of preference matrix, i.e. set of ranked items with their preferences.

In our simulated environment, each user u, is fully represented by the triple $< i_1, i_2, w_1 >$, that can be understood as quadruple
$< i_1, i_2, w_1, w_2 = 1 - w_1 >$, where

- i_1 represents the ideal point in first data dimension.
- i_2 represents the ideal point in second data dimension.
- w_1 represents the weight (importance) of first data dimension.
- w_2 represents the weight (importance) of second data dimension.

This way we can know all preferences for all items (although we use only those on visited items, which are also generated randomly).

Data as from user behavior Here we use also synthetic user. We calculate behavior on visited objects and this is the only input for rest S_{model} situation. To follow our visualization strategy and to use spatial database metric we calculate the convex hull of these points.

4.2. Practical challenge-response construction

Challenges C_{model} in model situation equal to rows of partially filled preference matrix M, i.e. known preferences y of each single user $u \in U$ for all visited items V^u.

$$C_{model} = \{\{(u, i, y) : i \in V^u\} : u \in U\} \tag{14}$$

or in content based notation item is represented by the vector of attribute values $\bar{x} = \{i.A_1, i.A_2, \ldots, i.A_m\}$. Then set of challenges looks like

$$C_{model} = \{\{(u, \bar{x}, y) : \bar{x} \in V^u\} : u \in U\}\,. \tag{15}$$

R_{model} consists again of triples, first coordinate is the correct value polygon P^u_h, the second is the polygon $\hat{P}^u_{\alpha,h}$ computed by $A_{program}$ and third is the measure of acceptance between first and second coordinate (description follows is subsection on metric). First coordinate of R_{model} will differ depending on type of data used for experiments. In case we have synthetic data R_{model} consists of data cube contour lines - polygons P^u_h for chosen levels of preference h (see deductive part of Figure 1). In case we have preference degrees of visited items, first coordinate of R_{model} consists of convex hulls K^u_h of rated data points in data cube with preference at least h. Acceptance relation on a preference degree h is defined as follows

$$A_{model,h} = \left\{ \left(\{(u, \bar{x}, y) : \bar{x} \in V^u\}, (P^u_h, \hat{P}^u_{\alpha,h}, \rho) \right) : u \in U \right\} \tag{16}$$

Mappings f^-_{m2p} and f^+_{m2p} are identities on respective domains.

Considering first part of experiments with synthetic data, $\_A_{program}\_$ finds an approximation of the user by choosing closest pivot m in the dataset [14] and computes polygons $\hat{P}^u_p$ from pivots deductive model.

In the second part of experiments, we have only user's preferences on visited objects and $\_A_{program}\_$ finds $\hat{i_1}, \hat{i_2}, \hat{w}_1$ (see inductive part of Figure 1) and computed respective polygon.

Evaluation of implication 9 is here quantified through all users $u \in U$ and done by computing precision and recall on polygons obtained by A_{model} and $A_{program}$.

4.3. Methods, programs, algorithms

We provided two different algorithms for modelling user u by an algorithm. First algorithm M_1 tries to find closest neighbor pivot from set of equidistantly spread set of pivots $P = \{p_k : p_k =< i^k_1, i^k_2, w^k_1 >\}$. The second algorithm M_2 is based on centre of mass computation.

Closest neighbor pivots In this method, further denoted as M_1, is each user $u =< i_1, i_2, w_1 >$ with the contour polygons P^u_h estimated by the closest neighbor pivot $p_k =< i^k_1, i^k_2, w^k_1 >$. There exist $11 \times 11 \times 11$ equidistantly distributed pivots that split the data cube to 10×10 tiles. For each of 11×11 possible positions there exist 11 pivots with different weights $w_1 \in \{0.0, 0.1, \ldots, 1.0\}$

This method represents supposedly more precise user preference estimation, but its computation is more time-consuming, because it needs to compare known information about each user with each pivot.

To find the closest neighbour pivot, we can use different metrics. Currently, the pivot with minimal average difference in preferences over all ranked items. The chosen pivot defines computed contour polygons $\hat{P}^u_{M_1,h}$.

Centre of mass This method, further denoted as M_2, induces ideal point of each user according to location of best rated items first. It is much faster to compute, and so it can be dynamically re-computed online during the user activity and adopt immediately any knowledge about user's changing preferences.

In each dimension, known rated items are split to sets of items rated in interval $(0.9; 1.0 >$, $(0.8; 0.9 >$ etc. The interval with at least three items (two or one if not exists) is taken and the average value is computed. We then take this average value as estimated ideal point $\hat{i}_1$, respectively $\hat{i}_2$ in corresponding dimension. Splitting of rated items and taking the highest rated item first gives priority to optimal items, that would probably have smaller variance, and thus allow us to estimate location of ideal point better.

The overall score depends not only on partial preferences in individual dimensions, but also on the weights, that user assigns to individual dimensions – item features. Having estimations of $\hat{i}_1$ and $\hat{i}_2$, for ranked item o we can compute partial estimated preferences

$\hat{x}_1^o$ and $\hat{x}_2^o$, and we also know the correct overall score $t(o)$. Thus we can estimate weight $\hat{w}_1^o$ estimation from equation

$$t(o) = \hat{w}_1 * \hat{x}_1^o + (1 - \hat{w}_1) * \hat{x}_2^o \tag{17}$$

Because resulting weights can be different for different items, we take the final estimation of the weight $\hat{w}_1$ as the average of computed values over all ranked items. Induced user model $< \hat{i}_1, \hat{i}_2, \hat{w}_1 >$ then defines computed contour polygons $\hat{P}^u_{M_2,h}$ for given h.

4.4. Metric

Metrics are implemented in spatial database. For given algorithm α (here algorithms M_1 and M_2) and for given level of preference h, we can compute corresponding polygon $P^u_h = [K^1, K^2, L^1, L^2, M^1, M^2, N^1, N^2]$ in the data cube, that represents the contour line $[A, B]$ at the level h in the preference cube (see Figure 5). First we compute the contour line itself. According to the chosen level of preference, some vertexes can merge together and the octagon can become the hexagon or even a tetragon.

$$A = [A_1, A_2] = \begin{cases} [1; h - w_1/(1 - w_1) * (1 - h)], & \text{if } A_2 \geq 0 \\ [h + (1 - w_1)/w_1 * h; 0] & \text{otherwise} \end{cases} \tag{18}$$

$$B = [B_1, B_2] = \begin{cases} [h - (1 - w_1)/w_1 * (1 - h); 1] & \text{if } B_1 \geq 0 \\ [0; h + w_1/(1 - w_1) * h] & \text{otherwise} \end{cases} \tag{19}$$

Next we can compute intersections X^L, X^H of horizontal line $[A_1; *]$ with (triangular-shaped) partial preference function x_1 and intersections Y^L, Y^H of vertical line $[*; B_2]$ with (triangular-shaped) partial preference function x_2.

$$X^L = [X_1^L, X_2^L] = [A_1; A_1 * i_1] \tag{20}$$

$$X^H = [X_1^H, X_2^H] = [A_1; i_1 + (MAX_1 - i_1) * (1 - A_1)] \tag{21}$$

$$Y^L = [Y_1^L, Y_2^L] = [B_2 * i_2; B_2] \tag{22}$$

$$Y^H = [Y_1^H, Y_2^H] = [i_2 + (MAX_2 - i_2) * (1 - B_2); B_2] \tag{23}$$

Third we can compute intersections C, D, E, F with (triangular-shaped) preference functions in both dimensions.

$$C = [A_2 * i_2; A_2] \tag{24}$$

$$D = [i_2 + (MAX_2 - i_2) * (1 - A_2); A_2] \tag{25}$$

$$E = [B_1; B_1 * i_1] \tag{26}$$

$$F = [B_1; i_1 + (MAX_1 - i_1) * (1 - B_1); \tag{27}$$

Finally, we can compute boundary of the polygon as follows:

$$K^1 = [E_2; Y_1^H]; \tag{28}$$

$$K^2 = [E_2; Y_1^L]; \tag{29}$$

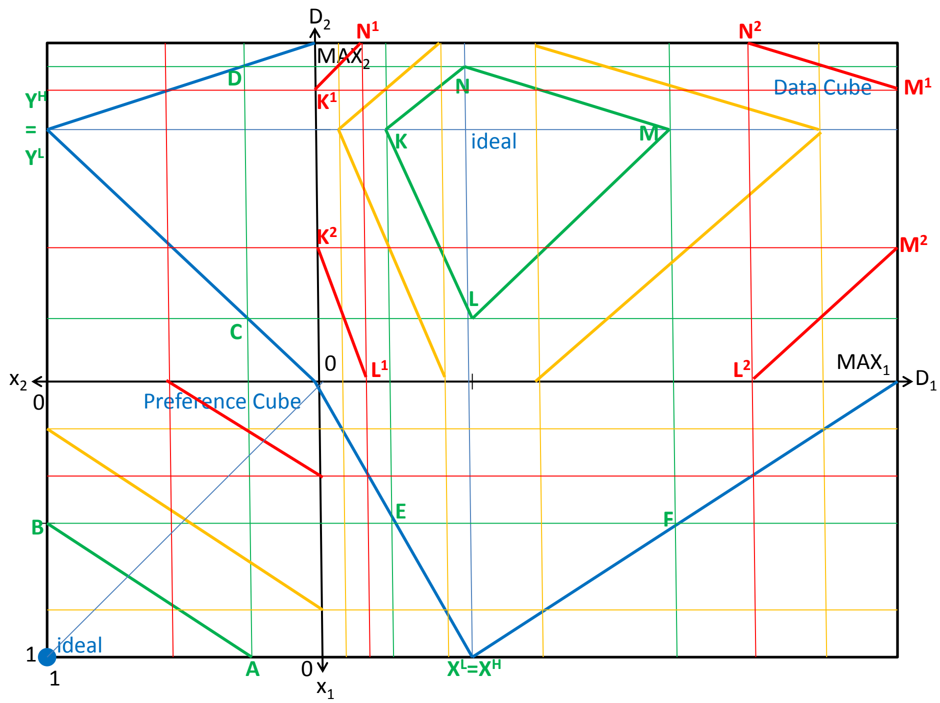

Fig. 5. Illustration of polygon computation for given levels of contour lines (green most preferred, yellow medium and red less preferred)

$$L^1 = [X_2^L; C_1] \tag{30}$$

$$L^2 = [X_2^H; C_1] \tag{31}$$

$$M^1 = [F_2; Y_1^L] \tag{32}$$

$$M^2 = [F_2; Y_1^H] \tag{33}$$

$$N^1 = [X_2^H; D_1] \tag{34}$$

$$N^2 = [X_2^L; D_1] \tag{35}$$

Correct ideal points $[i_1; i_2]$ and polygons $P_h^u = [K^1, K^2, L^1, L^2, M^1, M^2, N^1, N^2]$ for $h \in \{0.7, 0.8, 0.9\}$ were pre-computed and stored in *Oracle database* using *Oracle Spatial extension* as MDSYS.SDO_GEOMETRY points and polygon rings. Together with them all computed estimations provided by methods M_1 and M_2 were stored as well.

4.5. Experiments

We have all spatial data stored in the database and indexed by spatial index. This allowed us to effectively compute (not only) areas of both user's correct and computed polygons P_h^u, $\hat{P}_{M_1,h}^u$ and $\hat{P}_{M_2,h}^u$ and their intersections.

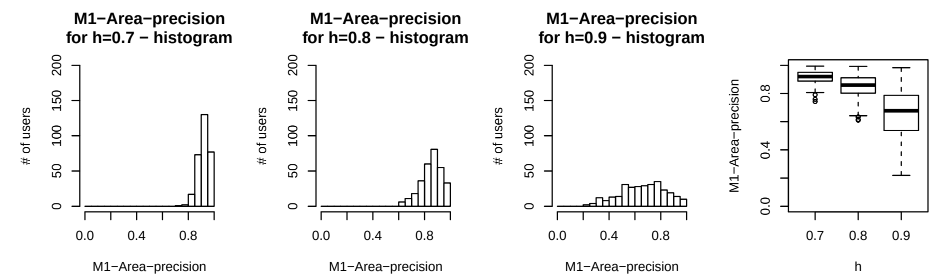

Fig. 6. From left to right - histograms of area based precision of algorithm M_1 for levels 0.7 , 0.8, and 0.9 and corresponding box-plots.

Experiments based on the area sizes of polygons and their intersections

Having given level of preference h we can compute both area based precision AP_h and recall AR_h for each user estimation. Formulae, presented above can be computed using SQL expressions

AP_h^u =
SDO_GEOM.SDO_AREA(
  SDO_GEOM.SDO_INTERSECTION(P_h^u,$\hat{P}_{M_k,h}^u$),0.001),
  0.001
)
/ SDO_GEOM.SDO_AREA($\hat{P}_{M_k,h}^u$),0.001)

$AR_{M_k,h}^u$=
SDO_GEOM.SDO_AREA(
  SDO_GEOM.SDO_INTERSECTION(P_h^u,$\hat{P}_{M_k,h}^u$),0.001),
  0.001
)
/ SDO_GEOM.SDO_AREA(P_h^u),0.001)

Figure 6 represents distribution of area based precision of algorithm M_1 for levels 0.7, 0.8 and 0.9 over 300 randomly generated users with different rating frequencies. As the h goes higher, contour polygons become smaller while ideal points (centers) stay on the same places. As a result the area of the polygon intersection decreases faster than the areas of polygons themselves. The more distant are real and estimated ideal points the faster. This results in lower median and higher variance of observed area-precision values.

Figure 7 represents distribution of area based recall of algorithm M_1 for levels 0.7, 0.8 and 0.9 over 300 randomly generated users with different rating frequencies. The same reasoning as for area-precision measure leads us to the assumption, that also area-based recall will show lower median and higher variance for higher levels of h. This assumption was confirmed by our tests.

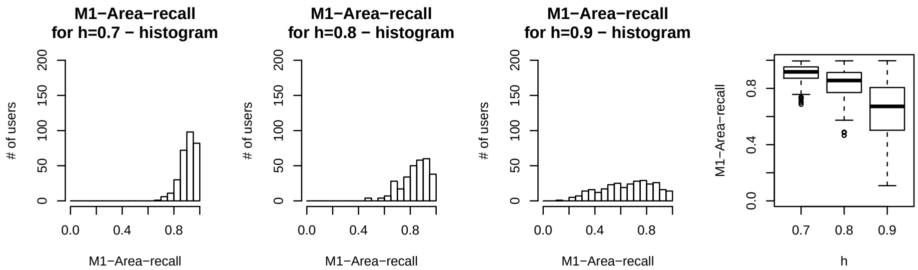

Fig. 7. From left to right - histograms of area based recall of algorithm M_1 for levels 0.7, 0.8 and 0.9 and corresponding box-plots

Figure 8 on the right shows both area precisions and recalls for individual users for level h=0.7 (red), 0.8 (orange) and 0.9 (green). Every symbol represents one user. We can see, that precision-recall pairs are located near the diagonal. From the computation of both measures we can deduce, that areas of both contour polygons - one belonging to the user and the other to its model are approximately the same. Thus the area of their intersection divided by the area of any of them results in close numbers.

Experiments based on the number of items in polygons and their intersections

Because area size need not to reflect the successfullness of the method well (some areas of data cube can be empty, while other can contain lot of items, we were interested not only in the area size, but also in number of items in corresponding areas. We can imagine the situation where substantial reduction of intersection area will lead to a small or no change in number of items, located in the intersection because almost all items are located in the remaining area. While the area-recall will be much smaller, the item-recall will remain the same. On the other hand, if almost all existing items would be located out of remaining area, we could notice substantial reduction of item-recall together with marginal reduction of area-recall. Similar considerations apply also to the precision measure. Item based metrics are also computed using *Oracle Spatial extension*, this time as

IP^u_h =
(SELECT COUNT(o) FROM $Item$ WHERE SDO_INSIDE(o,
  SDO_GEOM.SDO_INTERSECTION(P^u_h, $\hat{P}^u_{M_k,h}$, 0.001))='TRUE')
/
(SELECT COUNT(o) FROM $Item$ WHERE SDO_INSIDE(o,
  SDO_GEOM.SDO_AREA($\hat{P}^u_{M_k,h}$), 0.001)='TRUE')

$IR^u_{M_k,h}$=
(SELECT COUNT(o) FROM $Item$ WHERE SDO_INSIDE(o,
  SDO_GEOM.SDO_INTERSECTION(P^u_h, $\hat{P}^u_{M_k,h}$, 0.001))='TRUE')

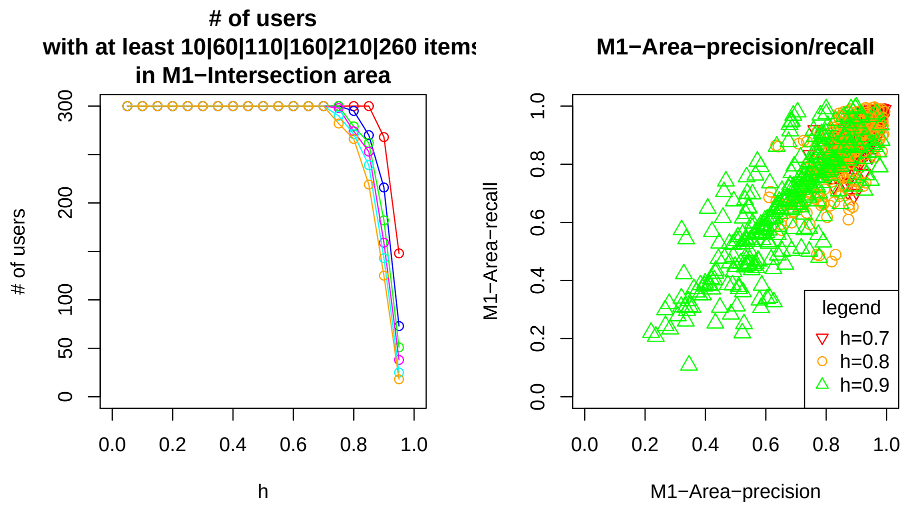

Fig. 8. Visualisation of precision and recall values for different users (right). Number of users having at least n items in the intersection of correct and computed contour polygon (left)

```
/
(SELECT COUNT(o) FROM Item WHERE SDO_INSIDE(o,
  SDO_GEOM.SDO_AREA(P_h^u),0.001)='TRUE')
```

Figure 9 represents distribution of item based precision of algorithm M_1 for levels 0.7, 0.8 and 0.9 over 300 randomly generated users with different rating frequencies. We can see, that while area-based precision histogram corresponds to Gaussian distribution, where the maximal number of users achieve the average value of the measure, here the distribution is substantially skewed to the right. It means, that while the area of intersection of contour polygons is in average smaller than contour polygons themselves, the intersection contains most of items with almost none located outside the intersection. As the

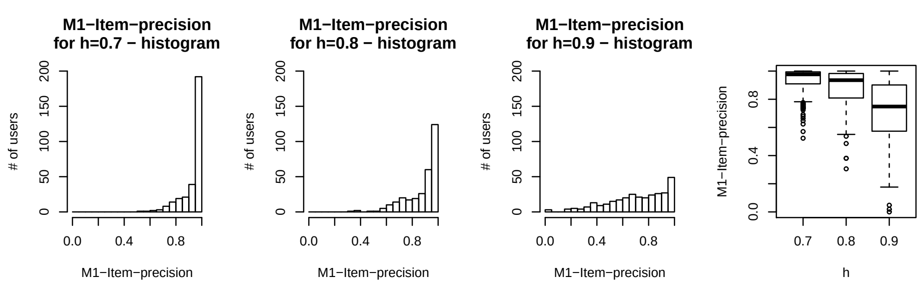

Fig. 9. From left to right - histograms of item based precision of algorithm M_1 for levels 0.7, 0.8, 0.9 and corresponding box-plots.

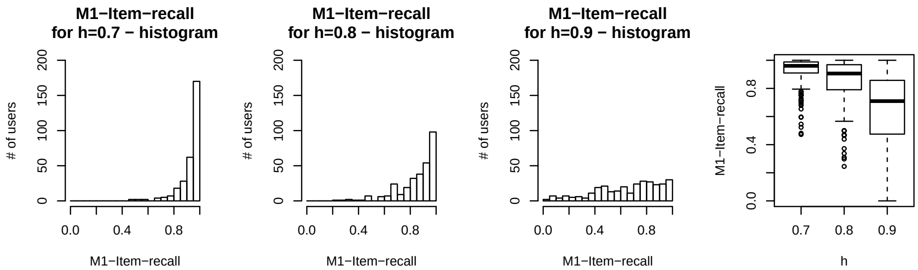

Fig. 10. From left to right - histograms of item based recall of algorithm M_1 for levels 0.7, 0.8, 0.9 and corresponding box-plots

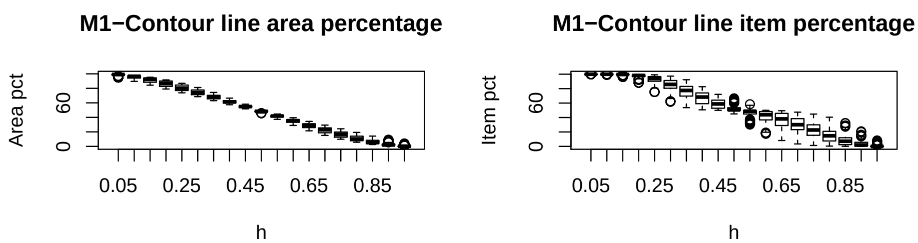

Fig. 11. Distribution of percentage of both area size (left) and item number (right) for different levels h

h grows the intersection area decreases, and some items become outside the intersection area. It causes gradual decreasing of the item-precision.

Figure 10 represents distribution of item based recall of algorithm M_1 for levels 0.7, 0.8 and 0.9 over 300 randomly generated users with different rating frequencies. We can see very similar behaviour as in case of item-based precision. As h increases, some items leave decreasing intersections of contour polygons while remaining in user contour polygon. Again, the item recall for average user will gradually decrease.

We were also interested in how many interesting items with a preference exceeding a certain limit h exist for a given user. Figure 8 on the left shows that only approximately 25 users have less than 10 items in the intersection of correct and computed contour polygon for $h = 0.95$. Approximately one half of users still has 250 and more items in this intersection. The absolute number of items depends on the density of items near the user's ideal point. With less items in the data cube this number will be proportionally smaller.

Comparison of area- and item- based metrics

Figure 11 shows how the distribution of contour polygon area size and item number within it are affected by increasing required level h. It is possible to see, that with higher required level h the average percentage of contour polygon size decreases. With level h close to zero the contour polygon covers almost whole data cube. With level h close to one, the area containing sufficiently preferred items decreases to zero size. For any preference level the box-plot is very small. From this we can see, that contour polygons for all users have almost the same size for given value of preference level h. The distribution of object numbers within contour polygons shows higher variances. Depending on whether the user has the ideal point in an area with more or fewer objects the number of items within polygons decreases at some levels faster or slower, than the area size itself. Average area size as well as average item size decrease almost linearly.

Other algorithms and metrics

So far we have discussed results based on the M_1 algorithm and metrics, that need correct contour polygon for each user, preset mesh of pivots, or both. Algorithm M_2 doesn't rely on the presence of pivots in the database, and induces all user parameters solely from ranked items. In real environment, exact data about users will be completely unavailable, and thus correct contour polygons will be unreachable. To replace unknown correct contour polygon P_h^u, we tried to take convex hull H_h^u of all items o in data cube, that have known user preference h or higher. This approach provided four different combinations of examined algorithm and metric evaluation, as it is shown in Table 1.

Table 1. Algorithms and metrics combinations, always need users actions on visited items (either from synthetic data or any other user behavior)

Computed polygon	Correct user's polygon	Notation in figures
M_1	Contour polygon P_h^u	M*-P-* needs whole synthetic data
M_2	Contour polygon P_h^u	M2-*-P-* needs user's synthetic model
M_1	Convex hull H_h^u	M1-*-H-* needs pivots
M_2	Convex hull H_h^u	M2-*-H-* can be used without synthetic data

In general, we can expect that metrics computed using user's correct contour polygon will show better results than corresponding results computed using convex hulls. It is because convex hulls are smaller and represent only subsets of items with required level of preference h. I.e. $H_h^u \subseteq P_h^u$. While the user's correct contour polygon can contain tens of sufficiently preferred items, it could happen that the user visited (and rated) for example only three of them. In this case the convex hull H_h^u will be a triangle somewhere inside the contour polygon P_h^u. Its size depends on diversity between visited objects. If all of them are located near each other, the triangle will be very small. Moreover, if the level h exceeds rating of the worst of these three items, the convex hull will be constructed by two remaining items, and will have zero area (with current algorithm, some geometric extrapolation is possible to avoid this ([22])). Thus we can expect in general worse results of both area and item-based measures in our tests.

The results obtained by closest neighbour pivot algorithm M_1 would be typically better than corresponding results for induced algorithm using centre of mass M_2. While the

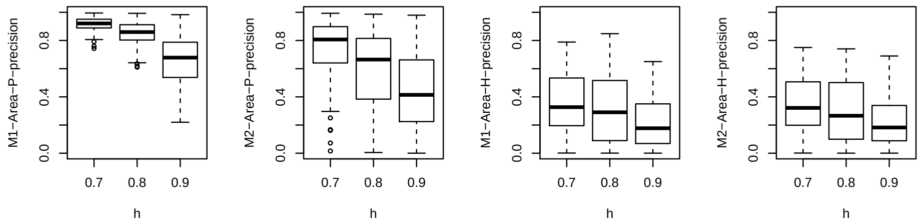

Fig. 12. Comparison of Area precisions achieved by different combinations of algorithm and user polygon As we expected, results where the correct contour line were approximated using convex hulls achieved worse values. Compare the third box with the first or the fourth with the second. Also replacing algorithm M_1 by M_2 show worse results as expected. Compare the second box with the first or the fourth with the third.

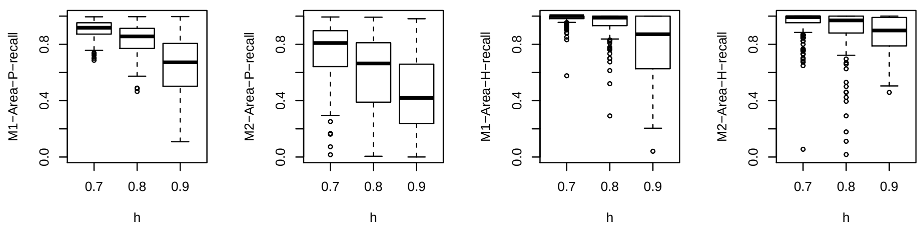

Fig. 13. Comparison of Area recalls achieved by different combinations of algorithm and user polygon. Similar comments as in Figure 12 apply.

first algorithm takes the closest available user approximation, the second one can be mistaken, if the user have highly rated only – or almost exclusively – items on one side of his/her ideal point. If user's ideal point i_k is in the half of the dimension, he/she can assign ratings 0.5 to items in the first or third quarter of the dimension. If only items from one group are rated (which is possible because each user visits and rates only small amount of items), their centre of mass and thus the ideal point estimation $\hat{i}_k$ can be significantly shifted to one side. Nevertheless, using M_2 is more realistic, in practice it would be difficult to build a set of pivots and compare each user with all of them.

Figure 12 compares precisions achieved by different versions of algorithms and user polygons. Results shown in this figure follows our expectation.

On the other hand similar Figure 13 that compares recalls show much better (higher) values for variants that compute recall using convex hulls H_h^u in comparison with variants that compute the recall using correct contour polygons P_h^u. It is caused by the fact, that convex hulls, created with the only knowledge about ranked objects, are much smaller than correct contour polygon. Typically it is fully inside estimated contour line polygon $P_{M_k,h}^u$. Thus the intersection $H_h^u \cap \hat{P}_{M_k,h}^u = H_h^u$ and the recall is equal to 1. If the user rated very small number of items, respectively there are no his/her ratings higher

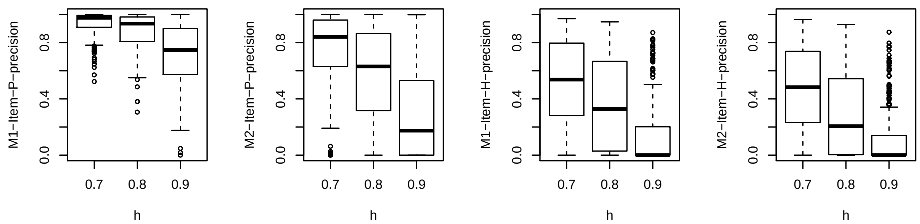

Fig. 14. Comparison of Item precisions achieved by different combinations of algorithm and user polygon.

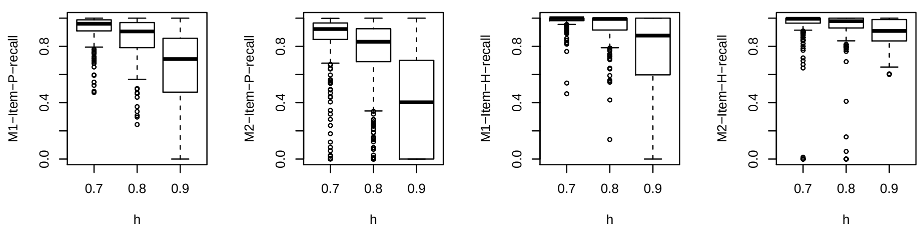

Fig. 15. Comparison of Item recalls achieved by different combinations of algorithm and user polygon.

than expected preference level h, the convex hull H_h^u cannot be computed at all and both precision and recall for this user cannot be evaluated.

Figures 14 and 15 show similar behaviour. As number of items in given area decreases quickly than the area itself, item-measure results have usually slightly lower median value, greater variance and so among others substantially lower first quartile and minimal values than corresponding area-measures.

Last two figures 16 and 17 compare both achieved Area precision / recall distribution and Item precision / recall distribution for all users. Second plot in Figure 16 shows, that areas of estimated contour polygons $\hat{P}_{M_2,h}^u$ are mostly of the same size as correct contour polygons P_h^u, even if they are not perfectly aligned. Thus both M_2 Area precision and M_2 Area recall are almost the same, and the visualization shows all results on the diagonal. The revision showed, that only 4 percent of users differs in areas of correct and computed contour polygons by more than 5 percent. This shows limitations of our synthetic data and methods. And remains a challenge for future work.

In first two subsections we have described experiments by algorithms. Then we commented results. Much more experiments and comparisons can be made. We hope this has shed light to the nature of proof of concept.

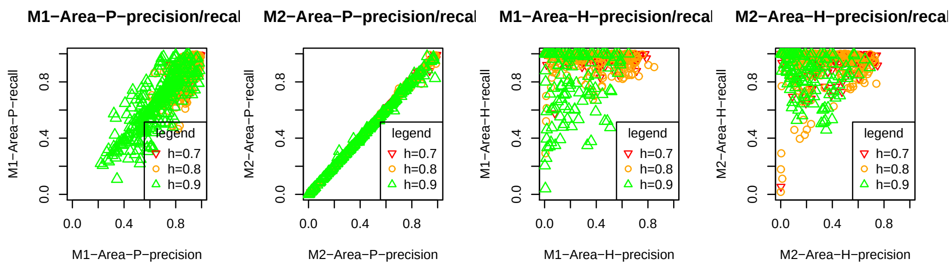

Fig. 16. Comparison of Area precisions / Area recalls for individual users achieved by different combinations of algorithm and user polygon. The second figure seems suspicious. It is still correct, as only 4 percent of users differs in areas of correct and computed contour polygons by more than 5 percent. This shows limitations of our synthetic data and methods. And creates a challenge for future work.

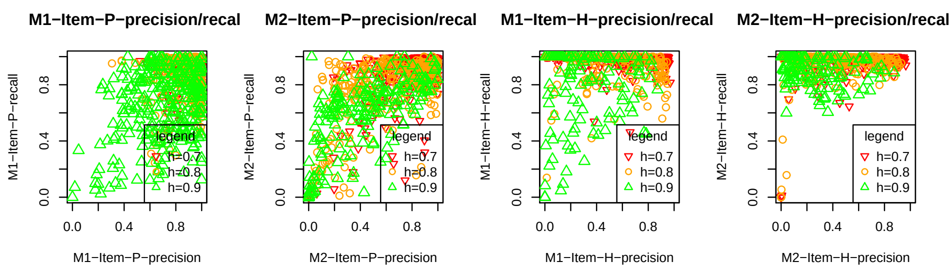

Fig. 17. Comparison of Item precisions / Item recalls for individual users achieved by different combinations of algorithm and user polygon.

5. Related research

Although our paper presents a formal model and experiments are only on synthetic data, still our main motivation comes from e-commerce.

E-commerce sites try hard to find out what is a user looking for. Various aspects are taken into account - design, user experience, usability etc. To achieve this, various search and filtering techniques are used (see e.g. [10] and many others). Although it is more involved in areas where attributes are not so easy to describe (e.g. clothing), for our motivation value filtering in ordered (numeric) domains is sufficient. Best practices consider filtering by category, by theme, multiple values of the same type, several displays, truncation of more than ten value ... They are advised to review how customers use filters, improve user experience, never return “no results”, care about speed ... Others concentrate on NLP, voice, image, context (attribute), personalized filtering techniques.

In our approach relaxation guarantees, that there are never ”no results”. The linearity of our components increases speed of reply. More values of the same type can be modeled by partly linear preferences consisting of e.g. two triangular or trapezoidal shapes which was not considered here.

Output of e-commerce applications usually use list or grid view. We offer an additional spatial view. Using human ability to grasp overall information is usually used in information visualization of large data according to a fixed metrics. Our visualization is personalized as it uses the closeness notion derived from overall user's preference and corresponding contour line.

Intuitiveness as one of attributes of user experience was also original idea of QBE - querying by example. Namely, an untrained user should be able to specify query without any knowledge of programming. Formal representation of QBE are tableaux queries, see [1]. Our model induces also a form of tableaux model with inequalities (above the contour line). These were introduced in [12] and further studied mainly from complexity theoretic point of view [16]. Once it influenced UX design ([19]), nevertheless today this connection to QBE is no more visible. Maybe our approach can revive this point of view.

One possibility of representing relaxation are fuzzy sets. The "fuzzy world" is not black-and-white, it recognizes degrees of shade. Hence it admits also relaxation of preference, if preference is interpreted as fuzzy score, see [17]. Fuzzy systems were initiated by paper [3]. Fuzzy systems as a tool for combining information (needed for multicriterial querying) were used in [7]. Original motivation was multimedia search in IBM Almaden project Garlic [9], where we can see a graphical querying interface without spatial nor personalized visualization. R. Fagin, A. Lotem and M. Naor in [8] introduced a formal model (FLN) in a more general use case, namely for querying web accessible databases. FLN considers only deductive problem (top-k) and does not touch induction for multi-user personalisation. For us it is most important that [8] introduces preferences for each attribute separately and hence implicitly defines a FLN-class of functions. We simplify these only to linear ones and hopefully win the speed. We extend this FLN approach by considering learning of both attribute preferences and combination function from linear FLN-class of functions.

We used fuzzy sets to model preferences in [17]. We considered learning user preferences for recommender systems in [18].

Another model of relaxing selects appears as acceptable violation of ideal values in AHP - analytic hierarchy process method of T. L. Saaty [20]. Visualization to support AHP process is widely studied, see e.g. [2]. It uses various models like treemaps, parallel coordinates, scatterplot matrices and the tabular visualization ([5]). With a certain degree of simplification we can say that AHP is used for decision support, usually assisted by a human, on low number of objects with very complex (hierarchical) description. In our case we need fully automated fast recommendation on large number of objects with relatively simple (shallow) description.

Challenge response framework was originally motivated in set theoretic study in real analysis (calculus) in [21]. We coined it Galois-Tukey connection. A. Blass ([4]) showed that this idea has appeared in different settings. Namely, it can be used also in computer science reduction of combinatorial problems. He later coined it "challenge-response". We have already mentioned that ChRF facilitates a much more general principle which appear in various types of human endeavor. Notice, that ChRF also appeared as a (http) authentication scheme in [6]. We will call these authentication procedures ChRF in narrow sense. Our approach will be called ChRF in broad sense.

Stochastic data creation with various types of distribution is described in [15]. Extensive experiments with different versions of top-k algorithms (also those from [8]) are provided. We created synthetic (stochastically generated) data set with additional user model with overall preference known and use it for inductive task.

We are aware of a gap between academia and industry, as described e.g. in [11]. Nevertheless we try to avoid proprietary e-commerce solutions. Our formal model based on Fagin-Lotem-Naor [8] class of functions, Challenge-Response-Framework (in broader sense) and visual (spatial) presentation is up to our knowledge unique. Although it is based, so far, only on proof-of-concept experiments on synthetic data, it is a promising candidate worth of future research. Especially when we look for generic models based on sound formal models.

6. Conclusions, future work

Main theme of this paper are e-commerce systems and we present an alternative proposal of filtering. Our proposal consists of a formal model integrating some aspects of Fagin-Lotem-Naor approach, Challenge Response Framework (in broad sense) and visual (spatial) aspects. Our main result is a candidate for new generic solutions.

Our proposal was tested on one collection of synthetic data. Of course it does not imply anything for practical applications. On the other hand, as observed by [11] online evaluation in real-world scenarios can be risky for e-commerce. There can appear several problems, such as high resource demands, temporal complexity and the lack of repeatability or potential negative impact on the user experience.

Moreover, companies usually do not share their data, even not historic ones and anonymized for off-line testing. Next step in improving business value of our solutions probably could be creating more synthetic data with distribution (at least statistically) similar to real world data. Especially, it would be interesting to provide tests with large, sparse data and large number of users.

There is another aspect of data we are working with. Our data contain information about user linear FLN-model - ideal points and weights. Let us consider situations where we have only user-item matrix with ranking. Instead of pivots we can use other users behavior in some ”spatial” version of collaborative filtering and models estimated from k-NN users. Our acquaintance with using convex hull in place of ”correct” user model shows that this is usually a subset of ”computed” polygon. Here probably AHP can us to help to consider triangle (trapezoidal, generalized partially linear) preference only in acceptable distance from ideal point or interval. Something like deviation tolerance saying how precipitous relaxation should be. Some initial experiments with ceteris paribus, categorical and nominal data were provided - this is a challenge for future work.

We are convinced that users' preferences usually depend on low number of attributes. Nevertheless, the two are really the minimum. In computer graphics and information visualization there are techniques for visualization of more complex data. Nevertheless we need to have our visualization personalized and this is based mainly on ability to visualize contour lines (areas or items in those areas). For instance it is difficult to visualize contour lines in parallel coordinates in higher dimensions. We have some preliminary work done already on visualization of four (up to eight) dimensions.

What is also unknown is how our visualization can improve user experience and usability. We did not consider here any aspects of design. To provide user studies in this direction is definitely an interesting task for future.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. T. Asahi, D. Turo, Ben Shneiderman. Using Treemaps to Visualize the Analytic Hierarchy Process, Information Systems Research 6,4 (1995) 357–375
3. Bellman, R. E., Kalaba, R. E., Zadeh, L. A.: Abstraction and Pattern Classification, RAND memorandum RM-4307-PR, October 1964, https://www.rand.org/pubs/research_memoranda/RM4307.html
4. Blass, A.: Questions and Answers - A Category Arising in Linear Logic, Complexity Theory, and Set Theory. In: J.-Y. Girard et al. (eds.) Advances in Linear Logic, London Math. Soc. Lecture Notes, vol. 222 pp. 61-81 London Math. Soc. (1995)
5. E. Dimara, A. Bezerianos, P. Dragicevic. Conceptual and Methodological Issues in Evaluating Multidimensional Visualizations for Decision Support. IEEE Transactions on Visualization and Computer Graphics, Institute of Electrical and Electronics Engineers, 2018, 24
6. Encyclopedia of Cryptography and Security. Editors: van Tilborg, Henk C.A., Jajodia, Sushil (Eds.) Springer Science and Business Media 2011
7. Fagin, Ronald, Combining Fuzzy Information from Multiple Systems, J. Comput. Syst. Sci. 58,1 (1999) 83–99
8. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. JCSS 66,4 (2003) 614–656
9. L. M. Haas, R. Fagin, M. D. Flickner, P. Schwarz et all. Querying Multimedia Data from Multiple Repositories by Content: the Garlic https://www.researchgate.net/publication/2764550, downloaded pdf accessed May 29, 2020
10. Holst, Ch.: The Current State Of E-Commerce Filtering, https://www.smashingmagazine.com/2015/04/the-current-state-of-e-commerce-filtering/. Last accessed 8 May 2019
11. D. Jannach, M. Jugovac. Measuring the Business Value of Recommender Systems. ACM TMIS 10,4, Article 16 (December 2019), 23 pages
12. A. Klug. On Conjunctive Queries Containing Inequalities. Journal of the Association for Computing Machinery 35,I (1988) 146-160
13. Kopecky M., Vojtas P. (2019) Graphical E-Commerce Values Filtering Model in Spatial Database Framework. In: Welzer T. et al. (eds) New Trends in Databases and Information Systems. ADBIS 2019. Communications in Computer and Information Science, vol 1064. Springer, Cham, pp 210-220
14. Kopecky, M., Vomlelova, M., Vojtas, P.: Basis Functions as Pivots in Space of Users Preferences. In: Ivanovic, M. et al. (eds.) ADBIS 2016, CCIS, vol. 637, pp. 45–53. Springer (2016).
15. A. Marian, N. Bruno, L. Gravano. , Evaluating top-k queries over web-accessible databases, ACM Transactions on Database Systems 29,2 (2004) 319-362
16. Ron van der Meyden. The Complexity of Querying Indefinite Data about Linearly Ordered Domains. Journal of Computer and System Sciences 54, 113-135 (1997)
17. Peska, L., Eckhardt, A., Vojtas, P.: Preferential Interpretation of Fuzzy Sets in Recommendation with Real E-shop Data Experiments. Archives for the Philosophy and History of Soft Computing No 2 (2015) http://aphsc.org/index.php/aphsc/article/view/32/2
18. Peska, L., Vojtas, P.: Using Implicit Preference Relations to Improve Recommender Systems, J. Data Semantics, 6,1 (2017) 15–30
19. Tony Russell-Rose. Designing Search: Entering the Query. UX Magazine Article No :804, March 14, 2012, https://uxmag.com/articles/designing-search-entering-the-query , last accessed May 25th, 2020
20. Saaty, T. L.: Decision making with the analytic hierarchy process. Int. J. Services Sciences 1,1 (2008) 83–98

21. Vojtas, P.: Generalized Galois-Tukey connections between explicit relations on classical objects of real analysis. Israel Math. Conf. Proc. 6 (1993) 619–643
22. P. Vojtáš and A. Eckhardt, ”Considering Data-Mining Techniques in User Preference Learning,” 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, NSW, 2008, pp. 33-36

Received: January 15, 2020; Accepted: September 1, 2020.