Predictability of Off-line to On-line Recommender Measures via Scaled Fuzzy Implicators

Ladislav Peska
Faculty of Mathematics and Physics
Charles University, Prague, Czechia
peska@ksi.mff.cuni.cz

Peter Vojtas
Faculty of Mathematics and Physics
Charles University, Prague, Czechia
vojtas@ksi.mff.cuni.cz

***Abstract*—This paper introduces fuzzy Challenge Response Framework, designed to understand the relationship between the model of a real-world situation and some real observations, based on scaled fuzzy Implicators between them. This general framework is applied to a particular case in recommender systems: the prediction of on-line performance given off-line evaluation results. We perform an empirical evaluation with real data from a Czech travel agency, comparing different recommender algorithms, different metrics for on-line and off-line evaluations, and different implication operators.**

***Index Terms*—fuzzy web intelligence, recommender systems, fuzzy decision support systems, on-line vs. off-line evaluation**

I. Introduction

Theoretical algorithmic models are required to be sound and complete. That is, computed results should be correct and all correct results should be computable. More challenging are scenarios, where models are connected to observable reality (either physical, e.g. weather forecast, or biological, e.g. diagnosis in medicine). At this point, the problem of how to measure soundness and completeness arises. However, the situation becomes even more challenging when human psychology is involved. As an example, one class of such real situations are users/customers aiming to buy some product in an e-shop and recommender systems (RS) aiming to model preferences of users via observing their behavior. Instead of correct answers RS responds with an ordered list of items, which correspond to the model of user's preferences. Soundness can be understood as a degree of user's satisfaction with this ordered response. Soundness becomes the only realistic goal (it is unrealistic to ask for completeness, if the human evaluation is involved, or e.g. while considering problems on the open web).

Jannach and Jugovac [1] critically discussed the value of algorithmic improvements in off-line recommender systems evaluation scenarios, which are common in academia. On the other hand, on-line evaluation in real-world scenarios has also certain drawbacks, such as high resource demands, temporal complexity, the lack of repeatability or potential negative impact on the user experience [2]. Nonetheless, the connection between off-line and on-line evaluation (and particular metrics utilized in each scenario) is often unclear and intensively researched. Therefore, we selected the problem of RS evaluation as a use-case for the proposed fuzzy Challenge Response Framework (fChRF).

We understand the relationship between a solution (model, algorithm) and relevance/satisfaction of the user as an fuzzy set inclusion/implication (e.g., computed → correct, model → reality, or off-line evaluation → on-line evaluation for our use-case). Many observed phenomena in RS are inherently fuzzy. This leads us to consider fuzzy implicators while measuring the success of the models.

Fuzzy logic has been used for flexible database querying for more than 30 years. As early as in the works of Zimmermann [3] and Fagin [4], [5], fuzzy sets were used as score interpreted as coding ordering of query results. In [6] Bordogna et al. reviewed the role of the inclusion operator in the interpretation of queries addressed to databases and Information Retrieval systems. Dubois and Prade [7] identified the role of fuzzy sets in answering queries with incomplete data and/or with ambiguity. Bosc and Pivert [8] analyzed trade-off non-commutative operators (e.g. convex combination of conjunctive and disjunctive ones), enabling merging positive and negative judgements.

In general, we follow the idea of Bellman et al. [9], where real world signal data and application needs contributed to the invention of fuzzy sets model. Likewise, we base our work also on a real world data and use-case.

The idea of Challenge Response Framework (ChRF) was motivated by the work of Galois [10]. Galois dealt with the problem of existence of formula for roots of higher degree polynomials. He constructed a correspondence between fields and groups acting on roots in such a way we can gather information about the group's structure from the field's structure and vice versa[1]. Motivated by Galois, in [11] we introduced Galois-Tukey (GT) connections using correspondence to gather information between structures of real line (e.g topology and measure). In [12] Blass interpreted GT connections as complexity reductions in computer science and illustrated it on the reduction of the 3SAT search problem to the 3COL search problem (vertex 3-colorability of graphs). Challenges are sets of formulas/graphs; responses are variable

This paper has been supported by Czech Science Foundation (GAČR) project Nr. 19-22071Y and by Charles University project Progres Q48. Source codes, evaluation data and complete results are available from github.com/lpeska/FUZZ-IEEE2020.

[1] see https://www.math3ma.com/blog/what-is-galois-theory-anyway

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

truth assignment and vertex 3 coloring resp. Finally, a binary relation determines whether the response meets the challenge. The Challenge Response terminology was introduced by Blass in [13]. Using this terminology, we introduced a Challenge Response Framework (ChRF) and used it to formalize the graphical support of querying [14].

In this paper, we utilize ChRF to understand the relationship between a (computed) model and real world observations in the context of small e-commerce recommender systems. Specifically, we aim on determining the usability of various off-line evaluation metrics in learning the true relevance of recommendations w.r.t. on-line observation of users responses. The base of this reduction is an implication securing that the model indeed meets needs of the on-line situation. For this task, we extend ChRF by fuzzy acceptance relations and introducing a class of "scaled fuzzy Implicators" (sfI) to evaluate these implications. We utilized fChRF in the on-line evaluation of recommending algorithms (i.e., evaluating the reduction from users behavior to algorithms predictions). Afterwards, we also utilized fChRF in evaluating usability of individual off-line metrics for the prediction of on-line evaluation results.

Specifically, the main contributions are as follows:

- We introduce fuzzy Challenge Response Framework (fChRF) as an alternative way of describing reduction of real situation to a model.
- We introduce scaled fuzzy Implicators (sfI), which serve as an alternative way of evaluating quality of models.
- By comparing on-line and off-line results in A/B experiments based on a small e-commerce enterprise, we identified off-line metrics, that imply the actual on-line results well.

II. Background Formal Models

A. Fuzzy Challenge Response Framework

First, let us introduce a formal framework (fuzzy Challenge Response Framework, fChRF) for the reduction of real world problems to computed solutions. Our use-case is to improve the on-line performance of recommendations based on the knowledge of off-line performance of these algorithms (model). In a sense we reduce the on-line problem to the off-line one.

Basic building blocks of fChRF are situations. A situation $S = (C, R, A)$ consists of a set of challenges C ($c \in C$ are instances of challenges), a set of possible responses R and a fuzzy acceptability relation $A : C \times R \longrightarrow [0, 1]$ (A need not be a crisp binary relation $A \subseteq C \times R$, a response can be acceptable in some degree).

The power of fChRF is based on reductions. Namely, challenges of real world (typically the ones that we can not solve, or would like to improve) can be translated to challenges of model (off-line evaluation in our use-case), where it is easier to find responses. Hopefully, one can translate these responses to the responses to original challenges. This appeared already in many-one reductions of combinatorial problems. Idea of this paper is to translate on-line evaluations to off-line evaluations and measure the degree of inherent implication.

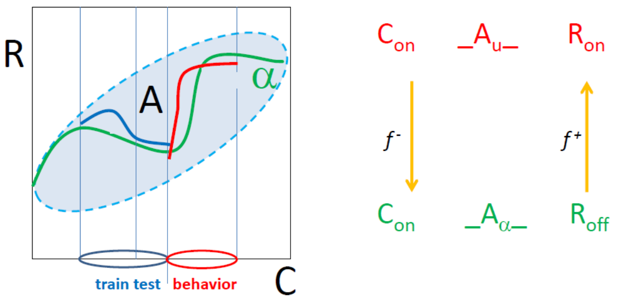

Fig. 1. fChRF illustration. Left: a situation with fuzzy acceptability with offline learning (blue line - train/test data) resulting in an algorithm α (green line), which extends calculated relevance to the whole domain. Online evaluation - actual user behavior (red line) has to be compared to α. Right: on-line fChRF - upper situation is real user activity and bottom is the model situation with recommendation of α (more on this in IV-B).

Assume, we have two fChRF situations, the real world situation $S_{real} = (C_{re}, R_{re}, A_{re})$ and the model situation $S_{model} = (C_{mo}, R_{mo}, A_{mo})$. A pair of functions $f^- : C_{re} \longrightarrow C_{mo}$, $f^+ : R_{mo} \longrightarrow R_{re}$, is said to form a fChR-reduction from S_{real} to S_{model} if the following condition holds $(\forall c_{re} \in C_{re})(\forall r_{mo} \in R_{mo})$:

$$A_{mo}(f^-(c_{re}), r_{mo}) \longrightarrow A_{re}(c_{re}, f^+(r_{mo})) \qquad (1)$$

So f^- and f^+ form a fChR reduction in some degree (see Figure 1 right). Here is the second place, where fuzziness can play a role. Namely, we have to find which fuzzy implicator between two fuzzy acceptability relations is a proper model to measure the quality of reduction.

Let us note that in our application we have $dom(A_{mo}) = C_{mo}$ and hence we do not need the machinery introduced in [14].

B. Scaled fuzzy connectives and residuation

For evaluation of implication (1) we need a larger scale of implicators to fit real needs. In the following section, we continue on our previous work [15]. Main motivation is to have sound and complete fuzzy rule semantics based on fuzzy modus ponens. Note that notation is mnemonic: B, b stands for body, H, h stands for head and R, r stands for rule. Based on Hajek's comparative notion of truth [16], our modus ponens is defined as follows:

$$\frac{(B, b), (H \leftarrow_I B, r)}{H, C_I(b, r)} \qquad (2)$$

Note that $\rightarrow_I$ is a connective (symbol in syntax), I denotes it's truth function. We say $C(b, r) = h$ is a conjunctor, if

- $C : [0, 1] \times [0, 1] \longrightarrow [0, 1]$
- (c1) C extends classical $\{0, 1\}$ conjunction
- (c2) C is nondecreasing in both coordinates
- (c3) C is left-continuous in b-coordinate

Note that left-continuous t-norms fulfill this, nevertheless our conjunctors need not be neither commutative nor associative.

We say that $I(b,h) = r$ is an implicator, if

- $I : [0,1] \times [0,1] \longrightarrow [0,1]$
- (i1) I extends classical $\{0, 1\}$ implication
- (i2) I is nonincreasing in b-variable and nondecreasing in h-variable
- (i3) I is right-continuous in h variable
- (i4) $(\forall h < 1) I(1,h) < 1$

Note that (i3) guarantees the pair (I, C_I) is a residuated pair and (i4) gives $C_I(1,1) = 1$.

Under these conditions we have soundness and (approximate) completeness of our fuzzy inference. Aggregation operators (see e.g. [5], usually inside body) are assumed to be left continuous in all variables.

For experiments we need a scaled fuzzy implicator to adapt both the b-coordinate contribution by a function $f(b)$ and the h-coordinate contribution by function $g(h)$. Assume both f and g are nondecreasing function from $[0,1]$ into $[0,1]$, $f(0) = g(0) = 0$, $f(1) = g(1) = 1$, f is left-continuous, g is right-continuous with $(\forall h < 1) g(h) < 1$ and I is an implicator. Then scaled fuzzy implicator $I^{f,g}$ is defined as follows (specifically for Łukasiewicz):

$$\begin{aligned} I^{f,g}(b,h) &= I(f(b), g(h)) \\ I_L^{f,g}(b,h) &= \min\{1, 1 - f(b) + g(h)\} \end{aligned} \tag{3}$$

Note that $I^{f,g}$ is an implicator.

III. Recommender Systems Use-Case

Now we would like to test these formal concepts in a real world production use-case. Let us describe in more details the considered recommender systems use-case. For the sake of space, we only describe the most significant properties. For more details, please refer to [2].

Recommender systems (RS) belong to the class of automated content-processing tools, aiming to provide users with unknown, surprising, yet relevant objects without the necessity of explicitly query for them. The core of recommender systems are machine learning algorithms applied on the matrix of user to object preferences. Throughout the decades of recommender systems research, there was a discrepancy between industry and academia in evaluation of proposed recommending models. While academic researchers often focused on off-line evaluation scenarios based on recorded past data, industry practitioners value more the results of on-line experiments on live systems, e.g., via A/B testing. While off-line evaluation is easier to conduct, repeatable, fast and can incorporate arbitrary many recommending models, it is often argued that it does not reflect well the true utility of recommender systems as seen in on-line experiments [17]. On-line evaluation is able to naturally incorporate current context, tasks or search needs of the user, appropriateness of recommendation presentation as well as causality of user behavior. On the other hand, A/B testing on live systems is time consuming and it can even harm retailer's reputation if bad recommendations are shown to users.

A. Dataset and Evaluation Domain

We focused on recommendation tasks for small e-commerce vendors. Specifically, our test domain was a medium-sized Czech travel agency. The agency sells tours of various types to several dozens of countries. Each object (tour) is available in selected dates. All tours contain a textual description accompanied with a range of content-based (CB) attributes, e.g., tour type, meal plan, type of accommodation, length of stay, prices, destination country/ies, points of interest etc.

The agency's website contains simple attribute and keyword search GUI as well as extensive browsing and sorting options. Recommendations are displayed on a main page, browsed categories, search results and opened tours. However, due to the importance of other GUI elements, recommendations are usually placed below the initially visible content.

B. Recommending Algorithms

A wide range of item-to-item recommending models were defined for the experiment. We specifically considered three branches of algorithms corresponding with the three principal sources of data: object's content based attributes (CB attributes), their textual description and the history of users' visits (collaborative filtering).

– Skip-gram **word2vec** models [18] were utilized for the stream of user's visits, i.e., the sequence of visited objects was used instead of a sentence of words. Individual trained models considered different embedding sizes $e \in \{32, 64, 128\}$ and context window sizes $w \in \{1, 3, 5\}$.

– **Doc2vec** models [19] were utilized for the textual description of objects. Doc2vec model, in addition to the word embeddings, calculates also embeddings of the document itself. Therefore the output of the algorithm are embeddings of a given size for each object (document). Textual description of objects was preprocessed by a Czech stemmer[2] and stop-words removal. Individual trained models considered different embedding sizes $e \in \{32, 64, 128\}$ and context window sizes $w \in \{1, 3, 5\}$.

– Finally, **cosine similarity** models were utilized for CB attributes. Nominal attributes were binarized, while numeric attributes were standardized before the similarity calculation. We evaluated two variants of the approach differing in whether to allow evaluating similarity on self ($s = True/False$). In this way, we may promote/restrict recommendations of already visited objects, which belongs to some of the commonly used strategies.

Given a query of a single object, the recommendations would be a list of top-k objects most similar to the query object (or its embeddings vector). For each considered item-to-item recommending algorithm, we utilized several variants of aggregation for individual items from user's visits history. Aggregations are composed as follows. The parameter $k \in \{1, 3, 5, 10\}$ denotes the maximal length of the input list of visited objects, e.g., considering only 5 most recently visited objects.[3] The keyword from {"mean", "max", "positional"

[2] https://github.com/UFAL-DSG/alex/tree/master/alex/utils

[3] Full user profile is used if no maximal length was specified.

TABLE I

OFF-LINE TRAINED ALGORITHMS. Div. and Nov: stands for diversity and novelty enhancements; parameter e stands for embeddings size, w denotes context window size and s denotes whether calculating similarity on self is allowed.

ID	Algorithm	Parameters	History	Nov.	Div.
1	doc2vec	e: 128, w: 1	positional-1	yes	no
2	doc2vec	e: 128, w: 1	temporal	no	yes
3	doc2vec	e: 32, w: 5	mean	no	no
4	doc2vec	e: 32, w: 5	mean	no	yes
5	doc2vec	e: 128, w: 5	max	yes	no
6	cosine	s:False	temporal	yes	no
7	cosine	s:True	mean	yes	no
8	cosine	s:True	positional-10	no	no
9	word2vec	e: 64, w: 5	mean	no	yes
10	word2vec	e: 32, w: 5	temporal	no	yes
11	word2vec	e: 128, w: 3	positional-1	no	no
12	word2vec	e: 32, w: 3	positional-10	no	no

and ”temporal”} denotes the aggregation strategy, i.e., what algorithm is utilized to aggregate individual results for items from the input list. Mean and max denote average and maximal values for each candidate object respectively. Positional and temporal denote weighted average, where weights of less recently visited / lower positioned objects are reduced. For positional aggregation, the weight w_p decrease with the position k_p of the object in the input list: $w_p = 1-(rank/k_p)$. For temporal aggregation, the weight w_t is based on the timespan between the visit's date t and present:

$$w_t = 1/(log(timespan(t).days) + \epsilon) \tag{4}$$

As certain types of algorithms may provide recommendations that lacks sufficient novelty or diversity, we utilized strategies enhancing temporal novelty as well as diversity. For diversity enhancements, we adopted the Maximal Margin Relevance approach [20] with λ parameter held constant at 0.8. For enhancing temporal novelty, we applied a similar approach and re-ranked the list of objects based on a weighted average of their original relevance r and their temporal novelty $n = w_t$: $\bar{r}(o) = \lambda * r(o) + (1 - \lambda) * n(o)$. The λ parameter was held constant at 0.8.

In general, a recommending algorithm α gets user behavior from session $t-1$ and outputs ordered list of objects for session t by assigning their position $\alpha(o)$.

In total, 800 variants of recommender systems were evaluated off-line. Based on the results of off-line metrics, 12 algorithms depicted in Table I were selected for on-line evaluation.

IV. Recommender Systems Evaluation

Due to the differences in evaluation processes, measures/metrics used off-line and on-line evaluation often differs.

A. *Off-line Evaluation metrics*

In off-line evaluation, we focused on four types of metrics, commonly used in recommender system's evaluation: rating prediction, ranking prediction, novelty and diversity.

For rating and ranking based metrics, we suppose that visited objects have the rating (relevance) $r = 1$ and all others $r = 0$. Mean absolute error (MAE) was evaluated as a member of rating-based metrics. Following ranking-based metrics were evaluated: area under ROC curve (AUC), mean average precision (MAP), mean reciprocal rank (MRR), precision and recall at top-5 and top-10 recommendations (p5, p10, r5, r10) and normalized discounted cumulative gain at top-10, top-100 and a full list of recommendations (nDCG10, nDCG100, nDCG). The choice of ranking metrics reflects the importance of the head of the recommended list (e.g., MRR) as only a short list of recommendations can be displayed to the user. However, as the list of recommendable objects may be restricted due to the current context of the user (e.g., currently browsed category), we also included metrics evaluating longer portions of the recommended lists (e.g., AUC, nDCG).

Two types of novelty in recommendations were considered: recommending recently created or updated objects (temporal novelty, nov_t) and recommending objects not seen by the user in the past (user novelty, nov_u). For temporal novelty, we utilized the score equal to w_t (4). For user novelty, a fraction of already seen vs. all recommended objects was used. Finally, the intra-list diversity (ILD) [21] was utilized as a diversity metric. Novelty and diversity metrics were evaluated on top-5 and top-10 recommended objects.

All off-line metrics were evaluated for each pair of test set user and recommender. Mean values for each recommender are reported.

B. *On-line Evaluation measures*

In on-line evaluation, our primary concern is to evaluate the extent to which the provided recommendations imply ”positive” user actions (i.e., actions that can be considered as an evidence of user's preference). Due to the sparsity of high-level preference data, such as purchases, we focused mainly on low-level evidences of user's preference. Specifically, we focused on collecting two types of users' actions: clicks and visits. First, we consider a click on a recommended object as (strong) evidence of positive user preference. Second, if a recommended object was later on displayed by the user, we also consider it as a (weaker) evidence of positive preference (i.e., the recommendation presentation was not so persuasive or noticed by the user, however the recommended object itself was probably relevant). In cases, where a visited item was recommended multiple times in the past, we attribute the visit to the most recent recommendation. Formally, we define **user's click response** $h_u^c(o)$ to the recommended object o, as 1 if the user u clicked on recommended object o and 0 otherwise. Similarly, **user's visit response** $h_u^v(o)$ is defined as 1 if the user u visited object o after it was recommended to him/her and 0 otherwise.

While the utilized user feedback is binary, the relevance of recommended objects can be considered as a graded set (i.e., the relevance of objects decrease with the position within the list of recommendations).

To be more formal, assume X is a set of objects, α is an algorithm assigning each object $o \in X$ a position in a list of recommended objects (e.g. $\alpha(o) = 1$ is the best object).

Assume we are interested only in top-k objects, $k \leq |X|$. Positions $p \leq k$ have assigned a fuzzy score $0 < s_k(p)$ such that $s_k(1) = 1$ and

$$\alpha(o_1) < \alpha(o_2) \text{ iff } s_k(\alpha(o_1)) > s_k(\alpha(o_2)) \tag{5}$$

Note that $s_k(\alpha) : X \longrightarrow [0, 1]$ is a fuzzy set representing the ordering of top-k elements of X assigned by α.

The fuzzy score we use is a linear one

$$s_k(p) = \frac{1}{k} * (k + 1 - p) \text{ for } p = 1, \dots, k \tag{6}$$

The online evaluation is based on an idea that the lack of user activity on an object (no-click, no-view) has to be discounted according to position of that object in the list of recommended objects. The higher the object in list (hence more visible) the higher the penalization. Instead of defining discount on position we define it on the respective score $s(p)$.

Motivated by the logarithmic relevance discount introduced in nDCG metric, we define $f_k^{DG} : [0, 1] \longrightarrow [0, 1]$ by $f_k^{DG}(0) = 0$,

$$f_k^{DG}(s_k(p)) = \frac{1}{\log_2(p + 1)} \text{ for } p = 1, \dots, k \tag{7}$$

and we extend it to $[0, 1]$ by piece-wise constant left-continuous interpolation. Notice e.g. that $f_{20}^{DG}(0.9) = f_{20}^{DG}(s_{20}(3)) = \frac{1}{2}$.

1) Fuzzy Challenge Response Framework for on-line measures: Let us now describe the fuzzy Challenge Response Framework for on-line evaluation measures (for an overview, see Figure 1, right). First, due to the nature of the underlined domain, we consider two ChRF reductions, depending on the underlined evidence type (clicks or visits). The reductions differ in the considered volume of objects: for clicks, challenges C_{on}^c contain top-k_c and for visits, challenges C_{on}^v contain top-k_v objects recommended for user u. Analogically, we differentiate two online user behavior situations (see Figure 1, top-right), namely "click" situation $S_u^c = (C_{on}^c, h_u^c, A_u^c)$ and "visit" situation $S_u^v = (C_{on}^v, h_u^v, A_u^v)$.

In both cases, we record user's feedback u (clicks h_u^c, or visits h_u^v) on objects from C_{on}^c and C_{on}^v resp. Acceptability relations A_u equals the user's responses, e.g., $A_u^c(o, h_u^c(o)) = h_u^c(o)$.

We further consider two analogical model situations (see Figure 1 bottom, right) S_α^c and S_α^v. They contain the same challenges as in behavior situations, and record actions of a recommender α, i.e., positions of objects in the list of recommendations $\alpha(o)$. Note that as usually $k_c \leq k_v$ and therefore a response for C_{on}^v is an extension of response for C_{on}^c. A_α^v is an extension of A_α^c. Acceptability relations have the same definition for both model situations, up to the usage of k_c and k_v parameters and equals the linear fuzzy score (6) of objects' position w.r.t. algorithm α.

$$A_\alpha^c(o, \alpha(o)) = f_k^{DG}(s_k(\alpha(o))) \tag{8}$$

To sum up, model situations S_α^c and S_α^v can be denoted as $(C_{on}^c, \{1, 2, \dots, k_c\}, A_\alpha^c)$ and $(C_{on}^v, \{1, 2, \dots, k_v\}, A_\alpha^v)$ respectively. Reductions f^-, f^+ are in these cases identities as users and objects are same (and similarly responses are numbers from $[0, 1]$). Correctness of reduction (1) turns to fuzzy implicator $I_{L/G}^{f,g}(b, h)$, where $b = A_\alpha(o, \alpha(o)) = s(\alpha(o))$ and $h = A_u$.

Let us show an example. Assume $k_c = 6$, $k_v = 20$, $\alpha(o_1) = 1$ and $\alpha(o_2) = 6$ (i.e. o_1 is the best object by recommender α and o_2 is on the tail of the "click" list and in the first third on the "visit" list). Furthermore assume f to be f_k^{DG} and g to be an identity, α and u are fixed. Then the online performance of algorithm α if object o_2 was not "clicked" or "visited" is computed as follows:

Acceptability degree in user behavior situation is

$$g(h) = h = A_u^{c/v}(o_2, h_u^{c/v}(o_2)) = A_u^{c/v}(o_2, 0) = 0 \tag{9}$$

For the model situation, acceptability is as follows:

$$A_\alpha^{c/v}(o_2, 6) = f_6^{DG}(1/6) = f_{20}^{DG}(0.75) \approx 0.356 \tag{10}$$

Note that we use an approximation of $f_k^{DG}(s_k(6)) = 1/\log_2(6+1) \approx 0.356$. Corresponding Łukasiewicz implicator evaluates as

$$I_L^{f_6^{DG},g}(1/6, 0) = I_L^{f_{20}^{DG},g}(0.75, 0) \approx 0.644 \tag{11}$$

If o_2 was clicked/visited, the result is 1 for both implicators. For o_1, the result is 1 if clicked/visited and 0 otherwise. For Goedel implicator, fuzzy scaled implicator for this scenario is just the classical binary implication.

2) Aggregations of on-line measures: In [22] Hajek and Havranek introduced implicational quantifiers. These were various combinations of number (sums) of true positives, false positives, true negatives and false negatives. They used it in cases where data could be expressed in a four-fold table (4ft). Implicational quantifiers were used to measure relevance of implication $\varphi \rightarrow \psi$ between crisp classifications $\varphi/\neg\varphi$ and $\psi/\neg\psi$ forming axes of 4ft. Motivated by [22] we interpret quantifiers in implication (1) by aggregation. It makes good sense because we would like to have overall evaluation on "how good is α (trained off-line with respect to some metric) in predicting users on-line behavior".

Lets assume the previously described fuzzy Challenge Response Framework and furthermore f to be f_k^{DG} and g to be an identity. Then we can define several variants of scaled fuzzy implicators as a function of item o recommended by algorithm α to the user u and his/her on-line response:[4]

$$\begin{aligned}
\mathfrak{I}_L^c(\alpha, o, u) &:= \min\left\{1, 1 - f_{k_c}^{DG}(s_{k_c}(\alpha(o))) + h_u^c(o)\right\} \\
\mathfrak{I}_L^v(\alpha, o, u) &:= \min\left\{1, 1 - f_{k_v}^{DG}(s_{k_v}(\alpha(o))) + h_u^v(o)\right\} \\
\mathfrak{I}_G^c(\alpha, o, u) &:= h_u^c(o) \\
\mathfrak{I}_G^v(\alpha, o, u) &:= h_u^v(o)
\end{aligned} \tag{12}$$

Having the scaled fuzzy implicators defined on the individual pairs of recommended objects and user's responses, we yet need to define their aggregations in order to obtain some

[4] Note that Goedel and Product implications are equal here and reduced to the binary form due to the binary user's response h and non-zero object's relevance w.r.t. α.

relevant performance statistics for individual recommending algorithms. While in theory, it is possible to consider e.g. minimal or median values, in our use-case it is impractical, as the user's positive response is quite scarce in recommender systems.[5] Instead, we focused on several aggregation metrics based on (weighted) average of implicator values for individual recommending algorithms. First, let $\mathfrak{I}_L^c(\alpha)$ denote the mean value of $\mathfrak{I}_L^c(\alpha, o, u)$ for all applicable combinations of object o and user u. Other implicator variants will be aggregated analogically. Note that $\mathfrak{I}_G^c(\alpha)$ is identical to the click through rate (CTR) commonly utilized in RS evaluation.

Two additional aggregation metrics were evaluated: position discounted and user's novelty based. In user's novelty based approach, we address the problem that the evaluation may be infested by a small volume of users with excessive interaction records. This is a specific consideration for travel agencies, where it is extremely rare if a user buys more than a single tour at once. Therefore, individual users have similar value for the agency and it should be reflected in the RS evaluation metrics - specifically it is important to assess, how the system performs for the novel users. Therefore, $\mathfrak{I}_{G,nov}^c(\alpha)$ and $\mathfrak{I}_{G,nov}^v(\alpha)$ are defined as weighted averages with weight $w = 1/|X_u|$, where X_u denotes all objects visited by the user u. In position discounted weighting, the idea is that recommending algorithms should provide ordering as consistent with the user's evaluation as possible (beyond a simple presence in top-k) and so the positively evaluated objects recommended on the lower positions should receive less credit. Therefore, $\mathfrak{I}_{G,pos}^c(\alpha)$ and $\mathfrak{I}_{G,pos}^v(\alpha)$ are defined as weighted averages with weight $w = 1/log_2(\alpha(o) + 1)$. Please note that novelty and position based weighting could be introduced analogically also for Łukasiewicz implications, but we omit them here for the sake of space and clarity.

V. Results and Discussion

A. Off-line Evaluation

For the off-line evaluation we used a dataset from [2] and evaluated it w.r.t. metrics described in Section IV-A. After some cleansing steps, the evaluation dataset contained 260K records of 72K users. We split the dataset into a train set and a test set based on a fixed time-point, where test set contained one and half month of most recent interactions. After limiting to users, who have at least one record in the train set as well, the resulting test set contained 3400 records of 970 users. Based on the results w.r.t. individual off-line metrics, 12 algorithms (see Table I) were selected for on-line A/B testing. Selected off-line and on-line metrics of these algorithms are displayed in Table II.

B. fChRF for On-line Evaluation

The on-line A/B testing was conducted on the travel agency's production server for a period of approximately one month. One recommending algorithm was assigned to each user based on his/her ID. During the on-line evaluation, we monitored which objects were recommended to the user, whether (s)he clicked on some of them and which objects (s)he visited.

A total of 4238 users participated in the on-line evaluation and in total over 26000 recommending sessions were observed. The total volume of click events was 830 and the total volume of visits after recommendation was 2824.

In order to retrieve the implication scores, we yet need to set the size of considered top-k lists. Although the selection can be arbitrary in general, we followed the properties of the background application. The application displays top-6 recommended items to the user, while it logs top-20 most relevant items according to the algorithm α. Therefore we set $k_c = 6$ for the clicks-based response h_c (as lower ranked items were not displayed to the user and therefore (s)he could not click on them) and $k_v = 20$ for the visits-based response h_v (as lower-ranked items could be still visited).

Table II contains results of on-line A/B testing. One interesting observation is that although the evaluation metrics are based on the same response, achieved results are quite diverse. Mean and median of Kendall's Tau coefficient for pairwise comparison of on-line metrics is 0.41 and 0.36 respectively. Similar values were achieved also while comparing results of the same implicators applied either on clicks or visits, so we may conclude that both action types, although correlated, measure inherently different aspects of user's preference.

Furthermore, there is an observable increase in performance between $\mathfrak{I}_G$ and $\mathfrak{I}_{G,pos}$, so we may conclude that even within the top-k recommended items, the ordering tends to be consistent with user's response. However, it is yet to clarify, whether the difference can be attributed to the correctly learned user model, or a position bias known, e.g., from search engines [23]. Similarly, we also observed an increase of performance between $\mathfrak{I}_G$ and $\mathfrak{I}_{G,nov}$, meaning that evaluated algorithms are in average better at recommending for novice users. As for the individual recommending algorithms, there is no single best algorithm. Nonetheless, while observing mean ranks of algorithms w.r.t. individual on-line metrics, the top three algorithms are IDs 10, 3 and 8 in Table I. Surprisingly, these algorithms are members of all three branches (word2vec, doc2vec and cosine), however, they all utilize longer users history profiles. From the practical point of view, word2vec models can be recommended over the other two branches, because the evaluated word2vec models achieved most consistent results, while both cosine and doc2vec branches contained some badly performing models (IDs 6 and 4 in Table I).

C. fChRF for Off-line to On-line Reductions

In this subsection we would like to evaluate the contribution of off-line metrics' results to the on-line ones.

First, we consider on-line situations (i.e., results of individual algorithms w.r.t. some on-line metric) as real world situation and off-line situations (results of algorithms w.r.t. individual off-line metrics) as models. Hence, we are going to use once again fChRF reduction. Because each of off-line metrics express a different quality of an algorithm and

[5] I.e., in majority of cases, $h_c^u(o)$ and $h_v^u(o)$ equals to zero.

TABLE II
OFF-LINE AND ON-LINE RESULTS OF RECOMMENDERS SELECTED FOR A/B TESTING. BEST RESULTS W.R.T. EACH METRIC ARE IN BOLD.

alg.	AUC	MRR	nDCG100	nov10$_t$	nov10$_u$	ild10	$\mathfrak{I}^c_G$	$\mathfrak{I}^c_{G,pos}$	$\mathfrak{I}^c_{G,nov}$	$\mathfrak{I}^c_L$	$\mathfrak{I}^v_G$	$\mathfrak{I}^v_{G,pos}$	$\mathfrak{I}^v_{G,nov}$	$\mathfrak{I}^v_L$
1	0.617	0.031	0.057	0.239	0.905	0.797	.00385	.00415	.0074	0.5531	.00520	.0142	.00544	0.3554
2	0.679	0.031	0.075	0.224	0.742	0.838	.00408	.00475	.0074	0.5534	.00516	.0102	.00691	0.3553
3	0.555	0.028	0.050	0.213	0.818	0.786	.00741	**.00948**	.0087	**0.5560**	.00570	.0116	.00616	0.3560
4	0.555	0.025	0.046	0.216	0.841	**0.859**	.00417	.00504	.0050	0.5536	.00428	.0090	.00540	0.3548
5	0.526	0.012	0.031	0.233	0.569	0.740	.00581	.00707	.0070	0.5547	.00552	.0163	.00595	0.3558
6	0.796	0.142	0.211	**0.263**	**0.959**	0.277	.00324	.00321	.0040	0.5525	.00338	.0069	.00342	0.3537
7	0.795	**0.148**	0.214	0.232	0.804	0.223	.00554	.00590	**.0111**	0.5540	.00510	.0125	**.00748**	0.3551
8	0.783	0.128	0.205	0.220	0.801	0.208	**.00772**	.00851	.0106	0.5555	.00600	.0100	.00600	0.3555
9	0.833	0.106	0.204	0.218	0.721	0.666	.00527	.00539	.0066	0.5537	.00558	.0107	.00746	0.3555
10	0.838	0.136	0.217	0.253	0.782	0.479	.00633	.00758	.0094	0.5550	.00612	**.0174**	.00640	**0.3561**
11	0.755	0.096	0.174	0.218	0.852	0.513	.00563	.00592	.0056	0.5540	**.00630**	.0117	.00595	**0.3561**
12	**0.842**	0.124	**0.234**	0.217	0.746	0.421	.00512	.00595	.0072	0.5541	.00584	.0097	.00676	0.3554

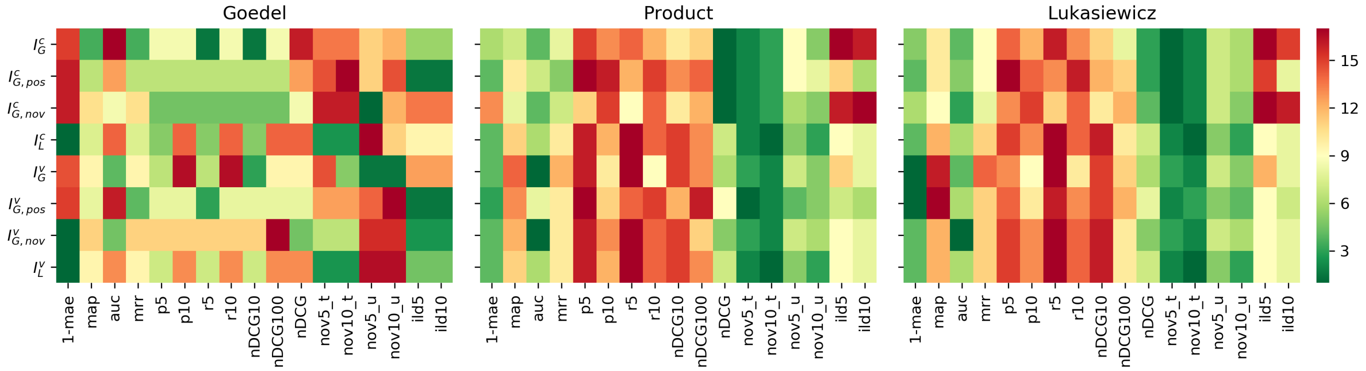

Fig. 2. Heatmap of ranked results for off-line to on-line implicators. For each on-line metric, the off-line metrics are ranked according to the corresponding mean fuzzy implication value. Dark green represents the best results, light yellow corresponds to the mid-field and dark red represents the worst results.

algorithms decide the ordering of objects in on-line sessions, metrics should be a part of challenges. It is impossible to incorporate users into challenges, because the set of users significantly differs for both off-line and on-line evaluations. Objects in the considered domain also considerably vary in time, hence individual objects can not be part of challenge set as well. So, we have to utilize aggregations over users and objects instead and apply fChRF on per-algorithm's results w.r.t. each combination of off-line and on-line metrics.

To be more specific, assume $\rho^{on}_1, \dots, \rho^{on}_m$ are evaluated on-line metrics (all mentioned in Table II) and $\rho^{off}_1, \dots, \rho^{off}_n$ are evaluated off-line metrics. Entries in Table II are results aggregated along users and objects. Notice however, that columns in Table II are not commensurable - we have to normalize them.

For each off-line metrics $\rho^{off}_p, p = 1, \dots, n$ let $x^p_i : i = 1, \dots, 12$ be the performance of algorithm α_i measured under ρ^{off}_p and let $b^p_i : i = 1, \dots, 12$ be its unit vector normalization in ℓ_2 norm. Similarly for $\rho^{on}_q, q = 1, \dots, m$ and performance $y^q_i : i = 1, \dots, 12$ let $h^q_i : i = 1, \dots, 12$ be its unit vector normalization.

While constructing the fChRF scenario, our approach is following. We fix an on-line metric ρ^{on}_q and evaluate contribution of each off-line metrics to ρ^{on}_q as (13), i.e., an instance of implication (1) for all algorithms.

To be more formal, let $q \in \{1, \dots, m\}$ be fixed. Let real behavior fChRF situation $S^{on}_q = (C^{on}_q, R^{on}_q, A^{on}_q)$ consists of

- $C^{on}_q = \{\alpha_i : i = 1, \dots, 12\}$
- $R^{on}_q = \{p : p = 1, \dots, n\}$
- $A^{on}_q(\alpha_i, p) = h^q_i$

Model fChRF situation $S^{off} = (C^{off}, R^{off}, A^{off})$ is constructed similarly:

- $C^{off} = C^{on}_q$
- $R^{off} = R^{on}_q$
- $A^{off}(\alpha_i, p) = b^p_i$

Let f^- and f^+ be identities on respective domains.

Then implication (1) has a form of

$$A^{off}(\alpha_i, p) \longrightarrow A^{on}_q(\alpha_i, p) \qquad (13)$$

and this evaluates to $I(b^p_i, h^q_i)$ for q fixed, $i = 1, \dots, 12$ and $p = 1, \dots, n$. For evaluation of implication (13) we use three classical fuzzy implications (Łukasiewicz, product and Goedel). For the sake of comprehensibility, we show the heatmaps of ranking results w.r.t. each on-line metric (see Figure 2, which is in fact a visualization of three fuzzy sets).

Heatmaps are constructed as follows. For fixed metrics ρ^{off}_p and ρ^{on}_q, c^q_p is an average of respective per-algorithm implication values:

$$c^q_p = \frac{\sum_{i=1}^{12} I(b^p_i, h^q_i)}{12} \qquad (14)$$

Each row in Figure 2 represents the ranking of c^q_p values for a fixed q, i.e., which off-line metric implies the selected on-line metric more (in average).

While observing the results in Figure 2, we may note that results w.r.t. Łukasiewicz and product implications are highly similar. In both cases, ranking-based metrics that focus on the whole list of recommendations (AUC, nDCG) provided fairly good results, while other ranking-based metrics were clearly inferior. Also 1-MAE and user-based novelty metrics performed well. Nonetheless, the highest values were achieved by temporal novelty metrics. On the other hand, results of Goedel implications are almost complementary to the previous cases. The best ranks are achieved by ranking-based metrics focused on a short list of top-k results (r5, nDCG10, p5, MRR) and also diversity metrics (ILD5, ILD10), which performed especially well on visits-based on-line metrics.

We assume that this discrepancy is caused by the very nature of each implicator. Note that Goedel implicator does not consider the truth value of $f(b)$ if it is larger than $g(h)$. On the other hand, both Łukasiewicz and product implicators incorporate $f(b)$ and roughly aim on quantifying the difference of both truth value levels. As such, Goedel implicator tends to be "less forgiving", if an off-line metric over-estimates the on-line performance. This can be illustrated by the mean values for each implicator, which were 0.63, 0.89 and 0.96 for Goedel, product and Łukasiewicz respectively.

As the metrics like nDCG10, p5 and r5 are highly biased towards the performance of top results, it is unlikely that they will over-estimate the on-line performance, especially in the case of clicks behavior (especially $\mathfrak{I}^{c}_{G,pos}$), which is also tied with the short list of top recommended objects. Therefore, results of Goedel implicator can be utilized to determine, which metric performs best for the "exclusion task", i.e., remove as many bad-performing algorithms based on a threshold on some off-line metric. This may be relevant in cases, where we suppose to have multiple well-performing algorithms and therefore it is not too harmful to inadvertently exclude one of these together with the bad-performing ones.

On the other hand, product and Łukasiewicz implications provide a better view on the actual amount of over-estimation and therefore may be more suitable for "selection tasks", i.e., ensuring that there are well-performing algorithms w.r.t. an on-line metric in the selected top algorithms w.r.t. some off-line metric.

VI. Conclusions and Future Work

This paper presented a fuzzy Challenge Response Framework (fChRF) to model the relationship between real situations and their (trained) models. We further introduced a class of scaled fuzzy Implicators (sfI) to measure the quality of real world to model situations reduction. These formal concepts were evaluated in a real-world use-case from the domain of recommender systems and small e-commerce enterprises. We utilized fChRF both in the on-line evaluation as well as for evaluating connections between off-line and on-line metrics. Experiments identified several off-line metrics suitable for assessing the relevance of RS during online deployment.

One of the main limitations of this paper is the binary form of user's response $h_u^{c/v}(o)$. This is a common case in contemporary RS evaluation scenarios. However, one can imagine that an analysis of finer-grained feedback (e.g. time spent on page or additional actions performed by the user [24], [25]) could result into a graded relevance scores and more interesting applications for fChRF. This problem is the main direction of our future work. Additional directions incorporate evaluation on more domains and additional use-cases for fChRF. We would also like to conduct experiments measuring the business value of off-line metrics.

References

[1] D. Jannach and M. Jugovac, "Measuring the business value of recommender systems," *ACM Trans. Manage. Inf. Syst.*, vol. 10, no. 4, 2019.

[2] L. Peska and P. Vojtás, "Off-line vs. on-line evaluation of recommender systems in small e-commerce," *CoRR*, 2018.

[3] H. Zimmerman, *Fuzzy Set Theory and Its Applications*. Springer, 1996.

[4] R. Fagin, "Combining fuzzy information from multiple systems," *J. Comput. Syst. Sci.*, vol. 58, no. 1, p. 83–99, 1999.

[5] R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," in *PODS'01*. ACM, 2001, p. 102–113.

[6] G. Bordogna, P. Bosc, and G. Pasi, "Fuzzy inclusion in database and information retrieval query interpretation," in *SAC'96*. ACM, 1996, p. 547–551.

[7] D. Dubois and H. Prade, *Using Fuzzy Sets in Flexible Querying: Why and How?* Springer, 1997, pp. 45–60.

[8] P. Bosc and O. Pivert, "On four noncommutative fuzzy connectives and their axiomatization," *Fuzzy Sets Syst.*, vol. 202, p. 42–60, 2012.

[9] R. Bellman, R. E. Kalaba, and L. A. Zadeh, "Abstraction and pattern classification." RAND memorandum RM-4307-PR, 1966.

[10] E. Galois, *Ecrits et mémoires mathématiques*. Gauthier-Villars, 1962.

[11] P. Vojtáš, "Generalized Galois-Tukey connections between objects of real analysis," *Israel Math. Conf. Proc.*, vol. 6, pp. 619–643, 1993.

[12] A. Blass, "Questions and answers: A category arising in linear logic complexity theory, and set theory," in *Adv. Lin. Logic*. USA: Cambridge University Press, 1995, p. 61–81.

[13] ——, *Combinatorial Cardinal Characteristics of the Continuum*. Springer, 2010, pp. 395–489.

[14] M. Kopecky and P. Vojtas, "Graphical e-commerce values filtering model in spatial database framework," in *ADBIS'19*. Springer, 2019, pp. 210–220.

[15] P. Vojtáš, "Fuzzy logic programming," *Fuzzy Sets and Systems*, vol. 124, no. 3, pp. 361 – 370, 2001.

[16] P. Hajek, *The Metamathematics of Fuzzy Logic*. Springer, 1998.

[17] A. Gilotte, C. Calauzènes, T. Nedelec, A. Abraham, and S. Dollé, "Offline a/b testing for recommender systems," in *WSDM '18*. ACM, 2018, pp. 198–206.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS '13*. Curran Associates Inc., 2013, pp. 3111–3119.

[19] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, 2014.

[20] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *SIGIR '98*. ACM, 1998, pp. 335–336.

[21] T. Di Noia, I. Cantador, and V. C. Ostuni, "Eswc 2014 challenge on book recommendation," in *ESWC'14*. Springer, 2014, pp. 129–143.

[22] P. Hajek and T. Havranek, *Mechanizing Hypothesis Formation*. Springer, 1978.

[23] T. Joachims and F. Radlinski, "Search engines that learn from implicit feedback," *Computer*, vol. 40, pp. 34 – 40, 09 2007.

[24] L. Peska, "Using the context of user feedback in recommender systems," *Electronic Proceedings in Theoretical Computer Science*, vol. 233, p. 1–12, 2016.

[25] P. Vojtas, A. Eckhardt, and L. Peska, "Preferential interpretation of fuzzy sets in e-shop recommendation with real data experiments," *Archives for the Philosophy and History of Soft Computing*, vol. 2, 2015.