Negative Implicit Feedback in E-commerce Recommender Systems

Ladislav Peska
Department of Software Engineering
Charles University in Prague
Malostranske namesti 25, Prague, Czech Republic
peska@ksi.mff.cuni.cz

Peter Vojtas
Department of Software Engineering
Charles University in Prague
Malostranske namesti 25, Prague, Czech Republic
vojtas@ksi.mff.cuni.cz

ABSTRACT
In this paper, we imagine the situation of a typical e-commerce portal employing personalized recommendation. Such website typically receives user feedback from their implicit behavior such as time on page, scrolling etc. The implicit feedback is generally understood as positive only, however we present several methods how to identify some of the implicit feedback as negative user preference, how to aggregate various feedback types together and how to recommend based on it.

We have conducted several off-line experiments with real user data from travel agency website confirming that treating some implicit feedback as negative preference can significantly improve recommendation quality.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval - *Information Filtering*

General Terms
Measurement, Human Factors.

Keywords
Recommender systems, negative implicit feedback, fuzzy T-conorms, e-commerce success metrics

1. INTRODUCTION & RELATED WORK
Recommending on the web is both an important commercial application and popular research topic. The amount of data on the web grows continuously and it is impossible to process it directly by a human. The keyword search engines were adopted to fight information overload but despite their undoubted successes, they have certain limitations. Recommender systems can complement onsite search engines especially when the user does not know exactly what he/she wants. Many recommender systems, algorithms or methods have been presented so far. We can mention Amazon.com recommender [10] as one of the best commercial examples. Recommender systems varies in both type (Collaborative, Content-based, Context, hybrid etc.), input (user feedback types, object attributes etc.) or output (top-k objects, inferred rating of an objet etc.). We suggest the papers by Adomavicius and Tuzhilin [1] or Konstan and Riedl [8] papers for overview.

Traditionally most of the research time and effort in the recommending area was spent on the explicit user rating based experiments and single success metrics (RMSE). However the user ratings are often too rare to provide reasonable output alone and RMSE does not necessarily reflect real-world success metrics like increase in purchases or revenues for the e-commerce.

While moving to the implicit feedback, where user behavior is recorded without user cooperation, we may receive abundant amount of data, but the link between feedback and user preference becomes less clear.

Several authors studied various aspects of implicit feedback: quite common are comparisons of implicit and explicit feedback e.g. Claypool et al. [1] using adapted web browser or Jawaheer et al. [6] on an online music server. Using utility function based on implicit feedback is common approach while it is impossible to get explicit feedback [5].

The interpretation of the implicit feedback is also problematic, as the user does not have any direct way, how to state that he/she does not prefer an object. Such preference can however significantly improve our recommendations and so it may be valuable to infer it. The explicit negative feedback is taught to be easier for users to specify [3], but studies about implicit negative feedback are rare. We can mention e.g. Lee and Brusilovsky [9] and their work on job recommender system. Compared to [9], we did not specify directly which user behavior implies negative preference, but base this assumption on average user behavior. We have also used different success metrics due to the differences in our scenarios.

Our approach is based mainly on the work of Eckhardt [4] on the two-step content based recommending method.

The area of fuzzy systems is closely related to our work. Having multiple types of user feedback, resulting into the multiple local preferences, we need an aggregation function to create single value representing user preference on the given object. Such aggregation could be the weighted average, a fuzzy T-norm or T-conorm or similar functions. Zimmermann and Zysno[16] described human decision making process and suggested parameter for the level of compensations for aggregating functions. Yager [15] suggested using noble reinforced T-conorms to cope with the same problem. Our approach is different from the two described as we use local preferences from the [-1,1] interval. The resulting effect though is similar to [16] while using proper aggregations.

The rest of the paper is organized as follows: we outline our scenario and preference learning methods in section 2. Section 3 will describe our experiment settings and section 4 results of the experiment. Finally, we will conclude our paper in section 5 and point out some future work.

1.1 Example of problem domain

We have recorded the user behavior on one of the major Czech travel agencies. The website contains approx. 1000 tours which differ in type, destination, price, services etc. The site does not allow users to explicitly rate objects and only collects their behavior on each tour such as time on page, scrolling, mouse moves etc.

Usually the user buys at most one tour per year, so we can collect user data related to one intended purchase per user (the decision process is short enough that we can track the unique user, but the gap between two consecutive purchases of the same user is too large).

1.2 Main contribution

The main contributions of this paper are:

- Methods how to deal with negative implicit feedback
- Discussing various methods how to aggregate multiple types of feedback
- Evaluate negative implicit feedback recommenders and compare it to other recommending methods
- Gather data for future off-line experiments.

2. RECOMMENDING SCENARIO

In our previous work e.g. [12, 13], we have evaluated recommendation based on various implicit factors over the business success metrics. We have examined each factor separately then various aggregation functions combining them together. After these experiments, we have designed a framework for combining various implicit feedback types into the user preference and recommending based on it [14]. The Figure 1 illustrates our approach: to create recommendations we first evaluate each feedback value separately for arbitrary fixed object and user with the *PREF()* method and receive list of local user to object preferences based only on the single feedback factor value. Then we combine local preferences together into the global preference via *@()* method (we may also specify importance *Imp()* of each feedback factor). Afterwards we can use any recommending algorithm suitable for user rating (content based, k-nearest neighbors, matrix factorization etc.).

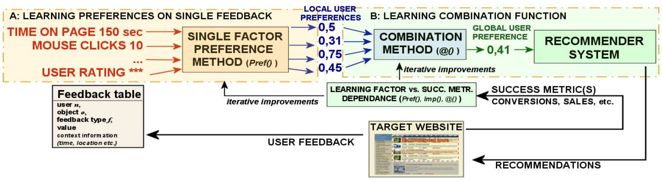

Figure 1: recommending scenario for multiple implicit feedback types.

The considered implicit factors are listed in Table 1, learning of local preference function *PREF()* is described in section 2.1, aggregating various feedback types (*@()* method) in section 2.2.

Table 1: Description of the considered implicit factors for arbitrary fixed user and object.

Factor	Description
PageView	Count(*OnLoad()* event on object detail page)
MouseMoves	Count(*OnMouseOver()* events on object detail page)
Scroll	Count(OnScroll() events on object detail page)
TimeOnPage	Sum(time spent on object detail page)
Purchase	Count(Object was purchased)
Open	Count(Object detail page accessed via link from recommending area)
Shown	Count(Object shown in recommending area)

2.1 Learning local preference

In domains without explicit feedback (such is ours) we do not have any direct information about what is user preference and so it needs to be inferred. The major approach in e-commerce systems is to use business-like view and state that user positively prefers the object(s) which he/she has *purchased*. We will then receive user preference on object as binary function *pref(u,o)*:

- 1 for object(s) purchased by the user
- 0 for all other objects

The problem of *pref(u,o)* is that the purchase actions are very sparse. The vast majority of users did not purchase any object, so *pref(u,o)* is useless to create any personalized recommendations. However we can use other feedback factors to predict probability, that user will purchase objects he/she has already visited (and we have some feedback from that visit).

The user preference *PREF(u,o)* will be defined as probability, that the user *u* will purchase (and like) the object *o*. However because of the insufficient amount of data about each user (only 8 visited objects per user in average and at best single purchase in our dataset), no tested method was capable to reliably learn preference of distinct users. Due to that the methods were set to ignore the individual users and treat the whole data as if it was from a single user.

The local preferences can be computed either by regression, or we can discretize the feedback factor domain into intervals and compute preference independently for each interval. The local preference for each interval *i* of feedback factor *f* is defined as:

$$PREF(f,i) = \sum purchases \Big/ \sum (user, object)\ pairs$$

This approach is motivated by the observation, that user preference of some factors may have quite complex dependence, which is difficult to be approximated by standard regression methods.

2.2 Using negative implicit feedback

So far, we have considered every implicit feedback item to carry some positive preference. This is given by using local preferences on feedback factors from [0,1] interval. However, the implicit feedback can also indicate that the user do not prefer the object (e.g. user opened an object, but leaves within few seconds).

In order to be able to model such situations and to test whether negative feedback is an important feature, we have adjusted the local preference learning method in a following way. Let the *AvgPREF(f)* be the average local preference for the feedback factor *f.* Now we can suppose, that if the value of *PREF(f,i)* is below average, it indicates negative preference: for each interval of each feedback factor, the negative local preference is:

$$NegPREF(f,i) = PREF(f,i) - AvgPREF(f)$$

The preference was then for technical reasons linearly normalized into the [-1,1] interval.

2.3 Aggregating various feedback types

In our previous work, we have used mostly weighted average with various algorithms for computing factor weights. However the results of tested methods were not satisfactory enough. The problem is that weighted average is not compensatory, so only single low value among the feedback factors can significantly decrease the resulting user preference. This problem is known in the area of decision making and fuzzy systems, where several T-conorms are suggested to cope with this problem. Having *x* and *y* local user preferences, their T-conorm aggregations are e.g.:

$$S^{BoundedSum}(x, y) = \min(x + y, 1)$$

$$S^{ProbabilisticSum}(x, y) = x + y - xy$$

$$S^{Sugeno-Weber}(x, y) = \min(x + y + \lambda xy, 1)$$

$$S^{Yager}(x, y) = (x^{\lambda} + y^{\lambda})^{1/\lambda}$$

However some T-conorms tends to be too compensatory even for low local preferences: for example if we have five local preferences PREF=0.2 and bounded sum, then the global user preference is 1 (fully preferred object), which is not a desired result. Zimmermann and Zysno[16] described human decision making process as very compensatory, if the local preferences are high (the aggregation function is similar to OR) and not compensatory if the local preferences are low (the aggregation function is similar to AND). Our approach – using local preferences on the [-1,1] interval can simulate such behavior, while using proper aggregation method. We have implemented two aggregation functions: *bounded sum*, where no changes in formula were necessary and *Sugeno-Weber* changed as follows:

$$S^{negSW}(x, y) = \min(x + y + \lambda xy, 1) \quad IF\ x, y \geq 0$$

$$S^{negSW}(x, y) = x + y \qquad OTHERWISE$$

So negative Sugeno Weber aggregation compensates only positive local preferences. The lambda parameter was set to minimize Mean Absolut Error (MAE) of predicted preferences against the user's actual purchases.

3. EXPERIMENT SETTINGS

Our key goal in the experiments was to corroborate, that negative implicit preference can improve recommendation quality. In order to do that, we have selected several other methods to create user rating based on implicit feedback. Some of them designed according to our recommending model (Figure 1) using only positive feedback and also some well known machine learning methods (SMOreg support vector machine, M5P decision tree). As the recommender system (whom we have supplied user ratings) was selected Content based system "Statistical" described in [4]

As the experiment domain, we have selected the travel agency data. We have collected data from one of the main Czech travel agencies. The site contains about 1000 objects – tours. We have monitored the site usage for one month period during November and December 2012. The original dataset contains data from about 40000 users purchasing 316 objects. We have filtered only users who visited four or more objects and bought at least one. The final dataset contains 62 users, who actually bought a tour at the website, totaling to 72 purchases. The average number of visited objects is 9.35 and the average number of purchased tours is 1.16.

The examined methods for learning ratings from implicit feedback were trained on the whole dataset and created rating of all visited objects by each user resulting into 580 ratings.

The ratings of each method were then supplied into the recommending system as follows:

We have selected train set sizes from 3 up to 10. For each train set size *K* and each user (with sufficient amount of visited objects), we have randomly selected *K* of his ratings, omitting the object user actually bought (cross validation applied here). The recommender system learned content based user model based on these ratings. Then we let recommender to rate each object and order the objects according to their rating. We followed the business-like scenario where the system can display only a limited amount of objects to the user, so we slice the list to top-10 and top-5 best rated objects.

We denote as success if the method recommends object, the user has actually bought. However as the position of the object is also important, we have selected the normalized discounted cumulated gain (nDCG) as our success measure. The nDCG measure represents the degree of usefulness of objects (whether object was purchased in our case) weighted by logarithm of their position in the top-k list. The formula for computing DCG is the following:

$$\mathrm{DCG_k} = purchased_1 + \sum_{i=2}^{k} \frac{purchased_i}{\log_2(i)}$$

The *k* is the size of the top-k list, $purchased_i$ is 1 if object on the i-th position was purchased by user and 0 otherwise. The normalization is done by dividing the DCG of the list by the DCG of the ideal ordering scaling results into [0,1].

4. EXPERIMENT RESULTS

The results for top-5 are shown in Figure 2 and for top-10 in Figure 3. In both cases the negative local preferences combined with Sugeno-Weber T-conorm was outperforming other methods for larger train sets. In general the negative Sugeno-Weber method performs best in the top-5 scenario and on 88% of the best method in top-10 scenario suggesting, that negative preferences can become important while trying to sort already good objects. The negative bounded sum did not perform very well and was mostly worse than original bounded sum with positive local preferences. The reason for this might be, that simple *x*+*y* is not sufficiently compensatory for the negative preferences.

In both scenarios all tested methods outperformed significantly random recommendations and at least some methods generated according to our model provided better recommendations then SMOreg and M5P baselines.

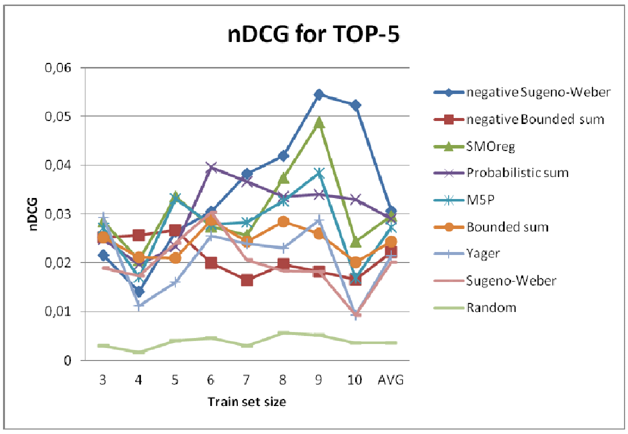

Figure 2: results of tested method for the top-5 list. The best resulting methods were: negative Sugeno-Weber, SMOreg, Probabilistic sum and M5P.

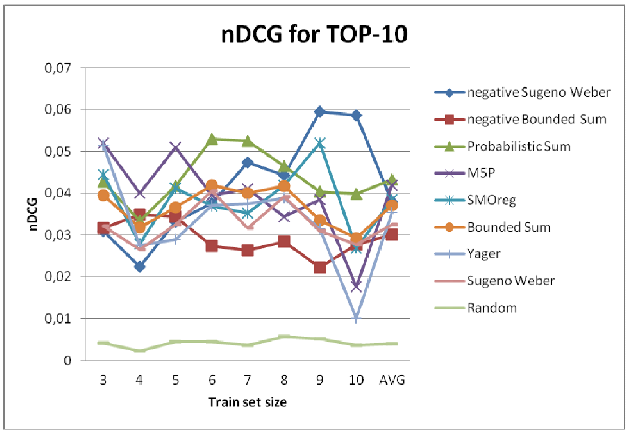

Figure 3: results of tested methods for top-10 list. The best resulting methods were: Probabilistic sum, SMOreg, M5P and negative Sugeno-Weber.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have discussed the problem of using implicit feedback as indicators of negative user preference. We have adapted a two step model to create user preferences from implicit feedback, describe steps to gain negative implicit feedback and how to use it and conducted off-line experiments on the dataset from e-commerce domain.

The experiment results showed that negative implicit feedback can be a valuable addition for the recommender system and improves recommendation quality (measured according to nDCG).

Our research should continue both in discovering other possible ways how to infer negative implicit preference and how to properly combine it. Online experiments on other e-commerce websites should be planned to confirm our ideas.

6. ACKNOWLEDGMENTS

The work on this paper was supported by the grant SVV-2013-267312.

REFERENCES

[1] Adomavicius G ; Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. Volume: 17 Issue: 6 Pages: 734-749

[2] Mark Claypool, Phong Le, Makoto Wased, and David Brown. 2001. Implicit interest indicators. *In Proceedings of the 6th international conference on Intelligent user interfaces* (IUI '01). ACM, New York, NY, USA, 33-40.

[3] Chao, D. L.; Balthrop, J. & Forrest, S. Adaptive radio: achieving consensus using negative preferences. Proc. of SIGGROUP 2005 - Supporting group work, ACM, 2005, 120-123

[4] Eckhardt, A.: Similarity of users' (content-based) preference models for Collaborative filtering in few ratings scenario. Expert Syst. Appl. 39(14): 11511-11516 (2012).

[5] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proc. of ICDM '08*. IEEE Computer Society, Washington, DC, USA, 263-272.

[6] Gawesh Jawaheer, Martin Szomszor, and Patty Kostkova. 2010. Comparison of implicit and explicit feedback from an online music recommendation service. In *Proc. of* HetRec'10. ACM, New York, NY, USA, 47-51.

[7] Kelly, D. & Belkin, N. J. Display time as implicit feedback: understanding task effects *Proceedings of the 27th ACM SIGIR conference on Research and development in information retrieval*, ACM, 2004, 377-384

[8] Konstan, J. & Riedl, J. Recommender systems: from algorithms to user experience. User Modeling and User-Adapted Interaction, 2012, 22, 101-123

[9] Lee, D. H. & Brusilovsky, P. Reinforcing Recommendation Using Implicit Negative Feedback *In Proc. of UMAP 2009*, Springer, LNCS, 2009, 422-427

[10] Linden, G.; Smith, B. & York, J. Amazon.com recommendations: item-to-item collaborative filtering *Internet Computing*, IEEE, 2003, 7, 76 - 80

[11] Ladislav Peska, Alan Eckhardt, and Peter Vojtas. 2011. UPComp - A PHP Component for Recommendation Based on User Behaviour. *In Proceedings of WI-IAT '11*, IEEE Computer Society, Washington, DC, USA, 306-309.

[12] Ladislav Peska and Peter Vojtas. 2012. Estimating Importance of Implicit Factors in E-commerce. In *WIMS 2012*, ACM, 2012, 62:1-62:4

[13] Peska, L.; Vojtas, P.: Evaluating Various Implicit Factors in E-commerce. In *RUE 2012*,CEUR vol. 910, 51-55

[14] Peska, L.; Vojtas, P.: Implicit feedback in Recommender Systems. In *WIKT 2012*, STU, 2012, 79-82

[15] Yager, R. R. Noble Reinforcement in Disjunctive Aggregation Operators. IEEE Transactions on Fuzzy Systems, 2003, 11, 754-767

[16] Zimmermann, H. J. & Zysno, P. Latent connectives in human decision making. Fuzzy Sets and Systems, 1980, 4, 37-51