

Evaluating Various Implicit Factors in E-commerce

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 focus on the situation of a typical e-commerce portal employing personalized recommendation. Such website could, in addition to the explicit feedback, monitor many different patterns of implicit user behavior – implicit factors. The problem arises while trying to infer connections between observed implicit behavior and user preferences - while some connections are obvious, others may not.

We have selected several often used implicit factors and conducted online experiment on travel agency web site to find out which implicit factors could replace explicit ratings and (if there are more of them) how to combine their values. As utility functions determining recommending efficiency was selected click through rate and conversions rate.

Our experiments corroborate importance of considering more implicit factors and their different weights. The best individual results were achieved by means of the *scrolling* factor, the best combination was *Prior_to* method (lexicographical ordering based on factor values).

Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Search and Retrieval - Information Filtering

General Terms

Measurement, Human Factors.

Keywords

Recommender systems, implicit factors, user feedback, e-commerce success metrics

1. INTRODUCTION

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 nearly impossible to process it directly by a human. The keyword search engines were adopted to cope with 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 [12] as one of the most popular commercial examples. Recommender systems varies in both type (Collaborative, Content-based, Context, hybrid, etc.), input (user feedback types, object attributes, etc.) or output. We suggest [17] for detailed recommender systems taxonomy.

The explicit feedback (given by the user consciously e.g. rating objects with stars) is often used in research and also in some commercial applications. Although it is quite easy to understand and refers very well to the user's preference, it also has drawbacks. The biggest ones are its scarcity and unwillingness of some users to provide any explicit feedback [7]. Contrary to the explicit feedback, the implicit feedback (events triggered by a user unconsciously) can provide abundant amount of data, but it is much more difficult to understand the true meaning of such feedback.

The rest of the paper is organized as follows: review of some related work is in section 2. In section 3 we describe our model of user preferences and in section 4 method how to learn it. Section 5 contains results of our online experiment on a travel agency website. Finally section 6 concludes our paper and points to our future work.

1.1 Motivation

In this paper we focus on an e-commerce website employing personalized object recommendation – e.g. travel agency. On such site we can record several types of user implicit feedback such as page-view, actions or time spent on page, purchasing related actions, click through or click stream, etc. Each of these factors is believed to be related to the user's preference on an object. However this relation can be non-trivial, dependant on other factors, etc. In this work, we focus on if and how such relations could be compared against each another. Our second aim is how to use or combine them in order to improve recommendations.

1.2 Contribution

The main contributions of this paper are:

- Evaluation of recommendation based on various implicit factors using typical e-commerce success metrics.
- A generic model that combines various types of user feedback.
- Experiments with several combining methods (average, weighted aggregation and prioritization).
- Gathered data for possible future off-line experiments.

2. RELATED WORK

The area of recommender systems has been extensively studied recently. Much effort has been made for creating different recommendation algorithms e.g. [3], [4], [5] and designing whole recommender systems e.g. [6], [15] and [16]. Our work is prependicular to some of those systems as we can supply them with a single-value object rating based on more implicit factors instead of using explicit user’s object rating or only single implicit factor.

A lot of recommendation algorithms aims to do decompose the user’s preference on the object into the preference of the object’s attributes [3], [4], [5] and [15], which can be a future extension to our work.

Some authors employ context information while deciding about true meaning of the user feedback e.g. Eckhardt et al. [2] proposes that good rating of an object is more relevant when the object appears among other good objects. Joachims et al. [8] proposes “Search Engine Trust Bias” while observing that the first result of a search engine search has higher click through rate than the second one, even if the results were swapped – so the less relevant result was shown at the first place.

Important for our research is the work of Kiessling et al. on the Preference SQL system e.g. [10]. The Preference SQL is an extension of SQL language allowing user to specify directly preferences (or so called “soft constraints”) and to combine them in order to receive best objects. We use three described combination operators: *Prior to (hierarchical)*, *Ranking* and *Pareto* in our model of user preference.

Several authors studied various aspects of implicit feedback: quite common are studies about comparing implicit and explicit feedback e.g. Claypool et al. [1] using adapted web browser or Jawaheer et al. [7] on an online music server. Using only an implicit feedback based utility function is a common approach when it is impossible to get explicit feedback [6], [14]. Lee and Brusilovsky proposed job recommender directly employing negative implicit feedback [11]. In our case we have focused on e-commerce recommenders, so we have used two typical e-commerce utility functions – *Click Through Rate* and user *Conversion Rate*. In contrast to several studies e.g. [1] who studied behavior of closed, small group of users (who installed special browser) on the open web, we have focused on the single website and all its users which in result let us to gather more feedback data and introduce more various feedback factors.

For our experiments, we use the UPComp [13] recommender deployable into the running e-commerce applications. Compared to our previous work [14], we have conducted larger on-line experiment, revised utility functions in our learning method and introduced new model of user preference.

3. MODELS OF USER PREFERENCE

We assume that any feedback is in the form *Feedback(user, object, feedback type, value)*. At this stage of our research, we do not employ preference relations or feedback related to the object groups (e.g. categories) and object attributes.

We based our models on work of Kiessling et al. and their model of user preferences in Preference SQL [10]. The authors defined several patterns on how to express preferences (soft conditions) on a single attribute e.g. “prize *around* 2000” or “*Highest* distance”, etc. Each soft condition assigns to each object value

from [0, 1] interval. Then they defined three types of operators combining soft conditions together:

- *Preferring Operator*: preferring one (or more) condition against others.
- *Ranking Operator* to combine conditions by a ranking function. At this time we use weighted average as a ranking.
- *Pareto Operator* for combining equally important conditions, or conditions where their relation is unknown. We plan to use this operator in our future work.

In our research, we have replaced the soft conditions by the implicit factors forming the *Preference algebra model*. Each implicit factor value has assigned preference value from [0, 1] interval – currently we simply linearly normalize the space between highest and lowest factor values. Those preference values can be then freely combined with the operators e.g.:

Scrolling PRIOR TO Avg(Time, MouseClicks)

We will demonstrate behavior of our model on a small two-dimensional example: Table 1 contains four sample objects and their scrolling and time on page feedback for fixed user (data already normalized into [0, 1]). They are visualized on Figure 1: as it can be seen, we will receive different top-k for their various combinations.

Table 1: example objects and their scrolling and time on page implicit factor values.

Object	Amount of scrolling	Time on page
Object1	1.0 (e.g 10 times)	0.4 (e.g. 200sec)
Object2	0.7 (e.g 7 times)	1.0 (e.g. 500sec)
Object3	0.8 (e.g 8 times)	0.6 (e.g. 300sec)
Object4	0.4 (e.g 4 times)	0.3 (e.g. 150sec)

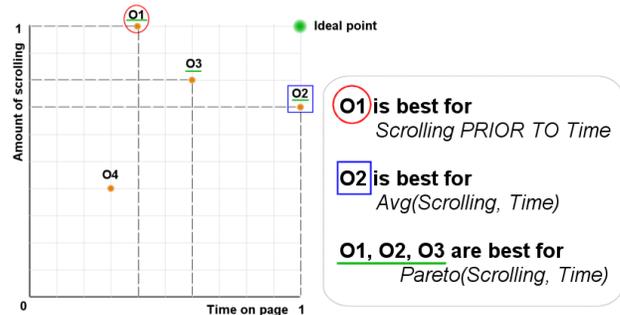


Figure 1: Combining single implicit factor values into the preference for objects from Table 1.

4. LEARNING PREFERENCE MODEL

The idea behind our learning model is following: If we use a fixed recommendation methods supplied with various implicit factor data and then compare the effectivity of the recommendations, we can estimate how successful each implicit factor is.

For the purpose of our experiment, we have divided our learning model into two phases: in the first phase, we have learned successfulness of the considered implicit factors (see Table 2 for their list and description). In the second phase we have implemented several methods combining various implicit factors together based on the *Preference algebra model*.

Table 2: Description of the considered implicit factors for arbitrary fixed user and object

Factor	Description
PageView	Count(<i>OnLoad()</i> event on object detail page)
MouseActions	Count(<i>OnMouseOver()</i> events on object detail page)
Scroll	Count(<i>OnScroll()</i> 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)

In both phases we have measured success of the recommendations according to the two widely used e-commerce success metrics:

- Conversion rate - #buyers / #users
- Click through rate (CTR) - #click through / #shown objects by the recommending method

As we stand on the side of the e-shop owner, we determine that the main task for the recommender system is to increase the shop owner’s profit. It is possible to measure the profit directly as an utility function, however we did reject this method for now and use only conversion rate measuring overall goal (purchase) achievements. In this stage of our work we mainly focus on convincing user to buy any product rather than convince him/her to buy product B instead of A (see table 3 – the overall conversion rates are rather low and need to be improved prior to the other goals).

As the conversion rate should evaluate the overall success of the whole system, the CTR refers directly to the success of the recommendation itself.

5. EXPERIMENT

We have conducted an online experiment on the SLAN tour travel agency website¹ to confirm our ideas. We have exchanged the previous random recommendations on the category pages for our methods. The experiment lasted for 2 months in February and March 2012. We have collected data from in total 15610 unique users (over 200 000 feedback events). We first describe in Figure 2 the simplified diagram of the travel agency e-shop. We recognize four important states of user interaction with the e-shop:

- User is creating conjunctive query *Q* (either implicitly e.g. by viewing category pages or explicitly via search interface).
- The (possibly very large) set of objects *OQ* is response to *Q*. The objects are recommended at this state. We recommend some objects from *OQ* to the user (membership in *OQ* set is necessary condition, each recommended object from *OR* has to fulfill).
- User is viewing detail of the selected object *o*. We believe that most of the interesting user feedback should be recorded in this phase.
- User purchased the object *o*, which is the criterion of success for us.

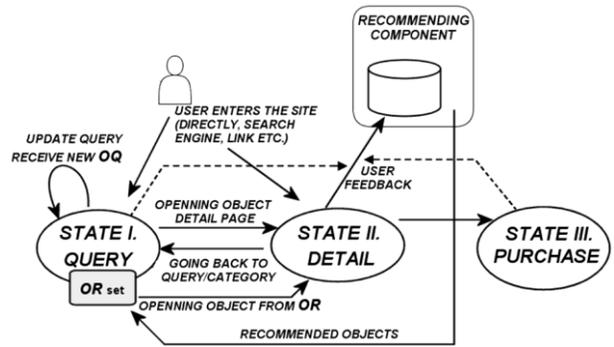


Figure 2: The simplified state diagram of an e-commerce site: User enters the site in *STATE I.* or *II.* He/she can either navigate through category or search result pages – updating query *Q*, receiving new recommended objects *OQ* and *OR* (*STATE I.*) or proceeds to the detail of an object (*STATE II.*). The object can be eventually purchased (*STATE III.*).

The Figure 3 depicts the schema of our experiment.

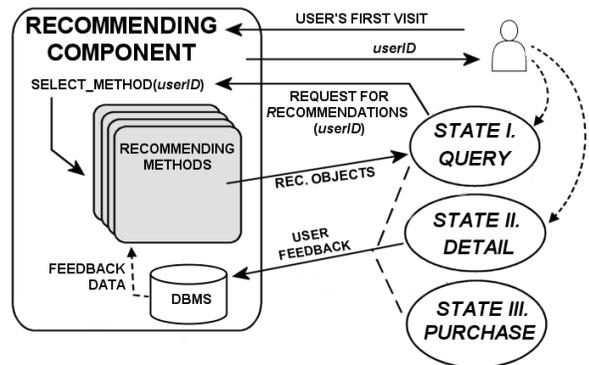


Figure 3: General schema of our experiment. When user visits the website for the first time, he receives *userID*, whenever he access page with recommendations, the component selects the recommending method according to the *userID*. The experiment results for each method are computed from user feedback (Click throughs, purchases).

5.1 UPComp recommender

The UPComp (user preference component) is an independent e-commerce recommender. It consists of a database layer storing user feedback, server-side computing user preference and recommendations and client-side which captures the user events and shows recommended objects. Among UPComp main advantages belong:

- Easy deployable to a various e-commerce systems regardless to the domain of objects.
- Large (extendible) set of recorded user behavior.
- Several recommending methods which can be combined together.

In the current experimental setting, we have used only a small portion of UPComp capabilities (*ObjectRating* and *Collaborative* methods, recommending objects for known category). For more complex description see [13].

¹ <http://www.slantour.cz>

5.2 Single implicit factors

For the first learning phase we have created a total of seven variants of *ObjectRating* recommending method, each based on one implicit factor (*PageView()*, *MouseActions()*, *Scrolling()*, *TimeOnPage()*, *Purchases()*, *ClickThrough()* and *ClickThrough()/Shown()* rate). Each variant of *ObjectRating* method used the same recommendation algorithm, but based on only one feedback type data. We have also added *Random()* method recommending random objects from the current category as a baseline. Each unique user received recommendations based only on one of these methods all the time he visited the website. The method is determined as $userID \bmod K$, where K is number of possible methods.

The *ObjectRating* method calculates for each object (o) the object rating as the sum of feedback values of given type (f) from all users U . The score is then normalized into $[0, 1]$ (see pseudo SQL code below).

```
SELECT (SUM(value) / MAX(SUM(value))) as ObjectRating
FROM Feedback
WHERE Object = o and FeedbackType = f
```

We have selected this simple method, because we wanted to avoid the problems suffered by more complex methods (e.g. *Cold Start Problem*). On the other hand, this decision decreases variability of recommendations, so we want to use also other methods in our future work.

Table 3. shows results of the first phase of our experiment. Anova test proves statistically significant differences in Click through rate (p -value < 0.001), but the differences in the Conversion rate were not statistically significant (probably due to relatively small number of purchases – 106 buyers in total).

Rather surprising is the supreme position of the *Scrolling()* method comparing to the e.g. Claypool et al. [1]. However in contrast to the Claypool et al. the most of our object detail pages overflows typical browser visible area. However important controls like purchase button are visible in top of the page, scrolling is necessary to see some additional information like accommodation details, all hotel pictures, trip program, etc. On sites with bookmark-style design with no or a little scrolling needs, opening an in-page bookmark should be considered as a similar action to our scrolling event. Also time spent on page seems to improve recommendations (despite the results of e.g. Kelly and Belkin [9]).

Table 3. Results of the experiment’s first phase. * significant improvement over *Random()* (TukeyHSD, 95% confidence).

Method	Conversion rate	Click through rate (CTR)
<i>Random()</i> (baseline)	0.97%	3.02%
<i>PageView()</i>	1.34%	4.11%*
<i>MouseActions()</i>	0.96%	4.15%*
<i>TimeOnPage()</i>	1.71%	4.50%*
<i>Scrolling()</i>	1.98%	4.94%*
<i>Purchases()</i>	1.39%	4.06%
<i>ClickThrough()</i>	0.84%	4.32%*
<i>ClickThrough/Shown()</i>	1.70%	4.38%*

5.3 Combining implicit factors

Following to the first phase, we have defined our three main tasks and perform experiments to receive at least initial answers/results for them:

- T1. Measure whether combined methods produce better recommendations than the single-factor ones.
- T2. Measure whether various combination functions affect recommendation effectivity.
- T3. How to use our results in more complex recommending methods.

Table 4: Results of combined methods: AVG stands for average, in *Weighted_AVG* we use the factor’s placement in the CTR results as weight, similarly *Prior_to* prioritize first factor against second, etc. * significant improvement over *Random()* (TukeyHSD). ** significant impr. over *AVG(best 3 factors)* (TukeyHSD). *** significant impr. over *Scrolling()* (t-test).

Method	Conversion rate	CTR
<i>Random()</i> (baseline1)	0.97%	3.19%
<i>Scrolling()</i> (baseline2)	1.07%	4.36% *
<i>AVG(all factors)</i>	1.41%	4.54% *
<i>AVG(best 3 factors)</i>	1.35%	3.95%
<i>Weighted_AVG(best 3 factors)</i>	1.49%	4.95% **, **
<i>Prior_to(best 3 factors)</i>	1.05%	5.12% *, **, ***
<i>Collaborative+ Weighted_AVG (all factors)</i>	0.95%	4.64% *

Again Conversion rate unfortunately did not provide us with any significant results, so we have focused on the CTR. The combined methods overall achieved better results than the *Scrolling()*, but only the *Prior_to()* was significantly better. Almost every method outperforms *Random()* recommendation.

For the Task 2, we have compared Weighted average, Priorization and Average methods on the best three implicit factors, where both *Weighted average* and *Priorization* methods receives significantly better results than *Average* in Click through rate. Both *Prior_to* and *Weighted_AVG* significantly outperformed *AVG* method, from which can be concluded that there are important differences in various single implicit factors performance and that combination function should weight somehow the single factors performance. However even though the *Prior_to* CTR results were better than *Weighted_AVG*, the difference was not significant enough, so we can not yet make a conclusion about which combination method is the best.

For the third task, we have slightly changed our experiment schema (see Figure 2), where we have exchanged the *ObjectRating()* method for *UserObjectRating(User, Object, Feedback type)* calculating object rating separately for each relevant user (see pseudo SQL code below).

```
SELECT (SUM(value) / MAX(SUM(value))) as ObjectRating
FROM Feedback
WHERE User = u and Object = o and FeedbackType = f
```

UPComp then calculated standard user-to-user collaborative filtering. The method results (see Table 4, *Collaborative+ Weighted_AVG*) were though rather moderate. The method outperforms *AVG*, *Scrolling* and *Random* in CTR, however the difference was not significant enough and other simple methods

(e.g. *Prior_to*) achieved better results. One of the possible problems was the higher computational complexity of this method resulting in higher response time which could reduce the user's interest in the objects presented in recommending area. This method can be in future compared / replaced with e.g. object-to-object collaborative filtering with precomputed similarity as described in [12].

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have discussed the problem of using more various implicit factors and how to formulate user's preference from them. We have adapted the *Preference algebra model* to this task, selected several possibly good implicit factors and organized a small online experiment to verify our ideas. The experiment results showed that the most of our proposed factors outperforms baseline recommendation and that it is important to use more various implicit factors combined accordingly to their performance.

The usage of e-commerce success metrics (especially CTR) to determine success of recommendations provided us with interesting results, so we plan to continue using Click through rate as a success metrics (conversions due to the relatively small number of purchases only in large scale experiments).

Our research on this field is in its early stage, so there is both space for more experiments (e.g. with negative implicit feedback, dependencies between various factors, temporal aspect of user's preference and behavior, etc.) and for possible improvements in our experimental settings (e.g. replacing recommending methods, extend the implicit factors set, etc.).

However our main task should be to move from such experiments into a working recommender system based on implicit preferences with various (dynamic) importances.

7. ACKNOWLEDGMENTS

The work on this paper was supported by Czech projects SVV-2012-265312, MSM 0021620838 and GACR 202-10-0761.

REFERENCES

- [1] 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.
- [2] Eckhardt A., Horváth T., Vojtáš P.: PHASES: A User Profile Learning Approach for Web Search. *In Proc. of WI 2007*, Silicon Valley, CA, IEEE Computer Society, pp. 780-783
- [3] Alan Eckhardt, Peter Vojtáš: Combining Various Methods of Automated User Decision and Preferences Modelling. *MDAI '09* 172-181. Springer-Verlag Berlin, Heidelberg, 2009.
- [4] Alan Eckhardt, Peter Vojtáš. 2009. How to learn fuzzy user preferences with variable objectives. In *proc. of IFSA/EUSFLAT Conf. 2009*: 938-943.
- [5] Jill Freyne, Shlomo Berkovsky, and Gregory Smith. 2011. Recipe recommendation: accuracy and reasoning. *In Proceedings of the 19th international conference on User modeling, adaptation, and personalization (UMAP'11)*. Springer-Verlag, Berlin, Heidelberg, 99-110.
- [6] 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.
- [7] 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.
- [8] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. *ACM Trans. Inf. Syst.* 25, 2, Article 7 (April 2007). DOI=10.1145/1229179.1229181
- [9] 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
- [10] Kießling, W.; Endres, M. & Wenzel, F. The Preference SQL System - An Overview. *IEEE Data Eng. Bull.*, 2011, 34, 11-18
- [11] Lee, D. H. & Brusilovsky, P. Reinforcing Recommendation Using Implicit Negative Feedback *In Proc. of UMAP 2009*, Springer, LNCS, 2009, 422-427
- [12] Linden, G.; Smith, B. & York, J. Amazon.com recommendations: item-to-item collaborative filtering *Internet Computing*, IEEE, 2003, 7, 76 - 80
- [13] 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.
- [14] Ladislav Peska and Peter Vojtas. 2012. Estimating Importance of Implicit Factors in E-commerce. *To appear on WIMS 2012*, <http://ksi.mff.cuni.cz/~peska/wims12.pdf>
- [15] Pizzato, L.; Rej, T.; Chung, T.; Koprinska, I. & Kay, J. RECON: a reciprocal recommender for online dating *Proc. of RecSys'10*, ACM, 2010, 207-214
- [16] Symeonidis, P.; Tiakas, E. & Manolopoulos, Y. Product recommendation and rating prediction based on multi-modal social network. *Proc. of RecSys'11*, ACM, 2011, 61-68
- [17] Bo Xiao and Izak Benbasat. 2007. E-commerce product recommendation agents: use, characteristics, and impact. *MIS Q.* 31, 1 (March 2007), 137-209.