Characterisation of Explicit Feedback in an Online Music Recommendation Service

Gawesh Jawaheer
CeRC, City University London
Northampton Square
London, EC1V 0HB, UK
Gawesh.Jawaheer.1@city.ac.uk

Martin Szomszor
CeRC, City University London
Northampton Square
London, EC1V 0HB, UK
Martin.Szomszor.1@city.ac.uk

Patty Kostkova
CeRC, City University London
Northampton Square
London, EC1V 0HB, UK
Patty@soi.city.ac.uk

ABSTRACT

In this paper, we present our study and characterisation of explicit and implicit feedback on Last.fm, an online music station and recommender service. The dataset consisted of explicit positive feedback (through loved tracks) and implicit positive feedback (the number of times a track is played). As one would expect, our analysis shows that explicit feedback is very scarce. However, we also found that the rate at which a user provides explicit feedback decreases with time, and that overall leaving explicit feedback has a negative effect on the user's behaviour.

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

General Terms
Measurement, Performance, Experimentation.

Keywords
Explicit feedback, implicit feedback, recommender system, music recommendation.

1. INTRODUCTION AND MOTIVATION

Many recommender systems (RS) require a model of the users' interests in order to function properly. A common approach to building such a user model is through eliciting feedback from the user, either explicitly or implicitly. Explicit feedback such as rating scales provides users with a mechanism to unequivocally express their interests in items. On the other hand, implicit feedback is generated by the RS itself, through inferences it makes about the user's behaviour. What constitutes implicit feedback depends on the application domain. Typically, it will be one or multiple observable and measurable parameters that arise out of the user's interactions with the RS. Traditionally, RS have been built on one or the other type of feedback; only few have combined these two heterogeneous feedbacks. In this paper, we examine the explicit feedback provided by the user in relation to implicit feedback gathered by a music recommendation service, namely, Last.fm. The aim is to gain an insight into how the two types of feedback can complement each other. Last.fm is an online radio station and music recommender service which plays tracks to users based on their interests. It bases its recommendations on implicit feedback about the tracks played by a user. It also allows users to express explicit feedback through its 'Love a track' feature. Together this provided us with a rich dataset which we mined using the API provided by Last.fm. In the next section we talk about the properties and benefits of the two types of feedback. We then provide some notation and describe the dataset. Section 4 presents our analysis and Section 5 our discussion. We conclude with some related work in section 6.

2. EXPLICIT AND IMPLICIT FEEDBACK

Traditionally, it has been difficult to obtain sufficient and representative feedback from a population of users. This reluctance to provide explicit feedback can be partially explained by the cognitive effort it requires. However, this reluctance most likely has other reasons which researchers have tried to explain. On the other hand, implicit feedback is abundant. In terms of modelling the users' interests, it is generally accepted that explicit feedback is more accurate than implicit feedback. There are several domain independent tools, such as Likert scale, star rating or questionnaire for capturing and analysing explicit feedback. In contrast, an implicit feedback system relies on application dependent tools and methodologies for capturing and interpreting implicit feedback. Typically, the system will observe the user's actions and make inferences about the user's interests based on these actions. For example, in a music recommender system such as Last.fm, if a user listens to a track several times, the system may infer that the user has an interest in that track. Explicit feedback can be positive and negative whereas implicit feedback is only positive. Furthermore, both types of feedback suffer from noise [1,2,4] and are sensitive to the user's context, albeit not to the same extent.

To study the characteristics of explicit feedback, we used Last.fm, the online music station and recommender service. Last.fm provides its users the functionality to love (explicit positive feedback) and ban (explicit negative feedback) a track. Internally, Last.fm keeps a count, called playcount, of all the tracks played by a user (implicit feedback). This includes tracks played on the Last.fm website or media players on the user's computer or portable device. Last.fm provides an extensive API toolkit to mine its rich dataset. Unfortunately, as the API does not expose a user's banned tracks, we were only able to build a dataset which included positive explicit feedback (loved tracks) and implicit feedback (played tracks).

RecSys'10, September 26–30, 2010, Barcelona, Spain.
Copyright 2010 ACM 978-1-60558-906-0/10/09...$10.00.

In the next section, we introduce some notations used throughout the remainder of this paper and also describe the dataset we collected and mined for our analysis.

3. NOTATIONS AND DEFINITIONS

Our dataset is composed of a set of users *U*, artists *A*, tracks *T*, and timestamps Z (the set of integers). *S* is a relation such that $S \subseteq U \times T \times A \times Z$, that describes which users have played which tracks, by which artist at each particular timestamp. Similarly, *L* is a relation such that $L \subseteq U \times T \times A \times Z$, describing the tracks that users have loved, and when they expressed their affinity. A profile for a user *u*, is defined as a pair $P_u = (S_u , L_u)$ where $S_u = | (u,t,a,z) \in S |$ is the total number of tracks played, also known as the playcount, and $L_u = | (u,t,a,z) \in L |$ is the total number of tracks loved, which we call the lovecount.

3.1 Dataset

We first harvested the profiles for 16,394 users of Last.fm. Removing the 6382 users who did not love any tracks, we were left with $|U| = 10{,}012$ users. We fetched data describing all the tracks loved by all users in *U* such that $|L| = 1{,}833{,}804$. We then proceeded to download data describing all the tracks played by a user. For practical reasons we restricted the set of users, *U'* for which we harvested data about all the tracks played where $u' \in$

U', by fetching users whose lovecount, $| L_{u'} | \geq 20$ and

playcount, $| S_{u'} | \leq 2000$ such that $|U'| = 867$ users for whom

we had the complete history of the tracks they played and the tracks they loved. In the next section we analysed the dataset. We start by calculating basic descriptive statistics and plotting the cumulative frequency distributions and ending finally by analysing the characteristics of the explicit feedback provided.

4. ANALYSIS

Table 1 shows the basic descriptive statistics about the dataset. The highly skewed nature of the data is further illustrated in the cumulative frequency plots in Figure 1 and Figure 2.

Table 1 Descriptive statistics about dataset

Dataset	Mean	Median	σ
Played tracks	25910	15000	36826
Loved tracks	183.2	52	575.8

In Figure 3, we show a scatter plot of the number of tracks loved against the number of tracks played for each user *u*, where $u \in U$. The first observation is an intuitive one; the distribution is skewed towards more tracks being played than loved. Next, we

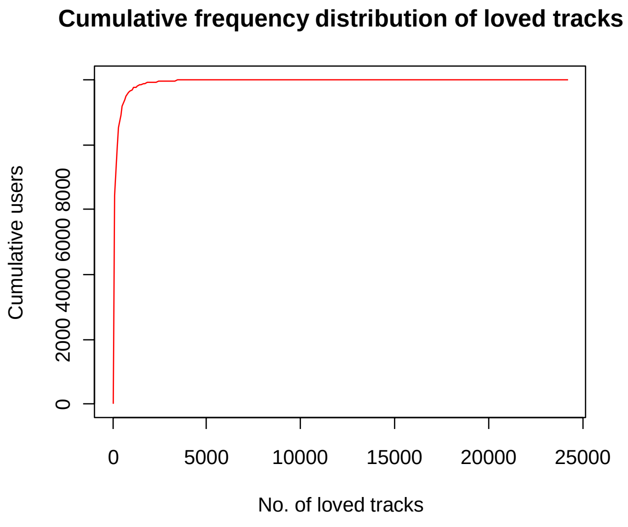

Figure 1 Cumulative frequency distribution of loved tracks

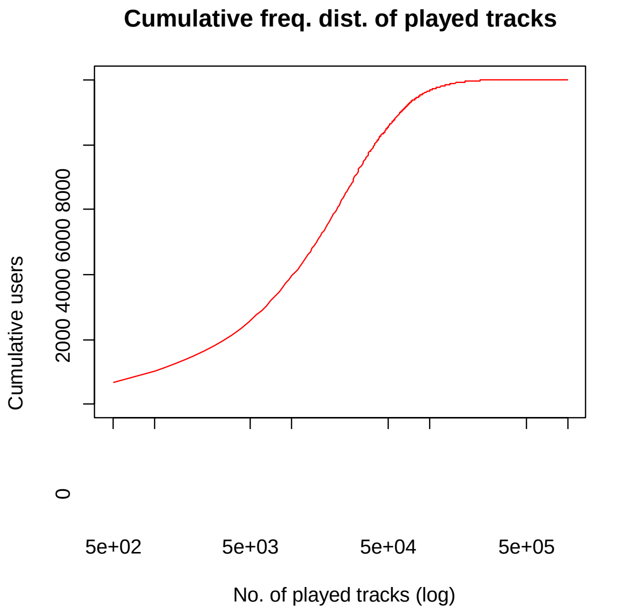

Figure 2 Cumulative frequency distribution of played tracks

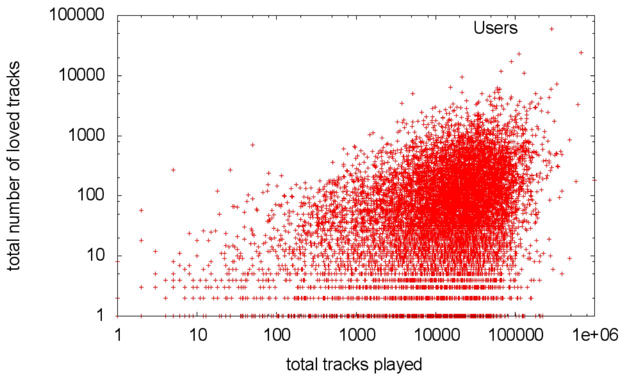

Figure 3 Scatter plot of loved tracks v/s played tracks for all users in *U*

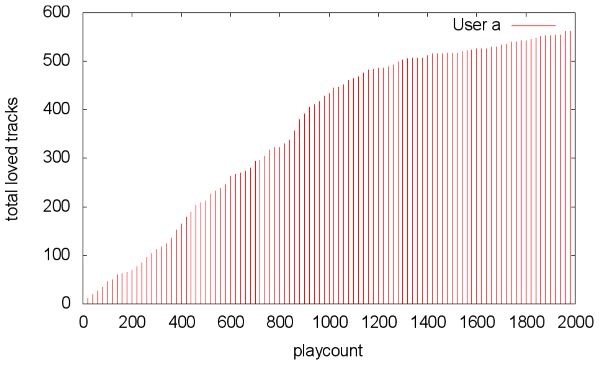

Figure 4 Histogram of loved tracks v/s played tracks for user *u'*

show a histogram of the number of loved tracks against the number of played tracks, for a particular user, $u' \in U'$. Note that playcount ordered in ascending order is synonymous to timestamp in chronological order. Observe that as the number of tracks played increases, the number of loved tracks increases as well. As mentioned, this is intuitive. Interestingly, the plot also shows that the number of loved tracks tends to plateau as the number of played tracks increases. And the question we ask is whether other users in U' exhibit a similar behaviour? In order to answer this question, we need to look at a distribution of loved tracks for all users as time increases. But it is not possible to represent such an analysis using time as an axis on a graph as each user will have different times at which they played their tracks. Instead, we used playcount, which represents a uniform metric for representing time across all the users. We used a normalised value for the number of loved tracks during a playcount interval by representing it as a percentage of all loved tracks by the user. And we averaged this normalised value across all the users in U'. Figure 5, shows a plot of percentage of loved tracks against playcount, where:

The percentage of loved tracks over time interval x,

$$Y_x = \left(\frac{|L_{u',s_{z+x}}| - |L_{u',s_z}|}{|L_{u'}|} \right)$$

As seen in Figure 5, the percentage of loved tracks, decreases with increasing playcount (and time). Figure 6, shows a similar plot but with error bars denoting the standard deviation.

Another relationship that we wanted to investigate was the number of times a track was played before and after it was marked as being a loved track. For user u', we define the ‘Effect’ of track t, as:

$$E_{tu'} = \frac{|S_{tu'z_{after}}| - |S_{tu'z_{before}}|}{|S_{tu'}|}$$

where

$|S_{tu'z_{after}}|$ = playcount of track t by user u' after it was loved

$|S_{tu'z_{before}}|$ = playcount of track t by user u' before it was loved

$|S_{tu'}|$ = total playcount of track t by user u'

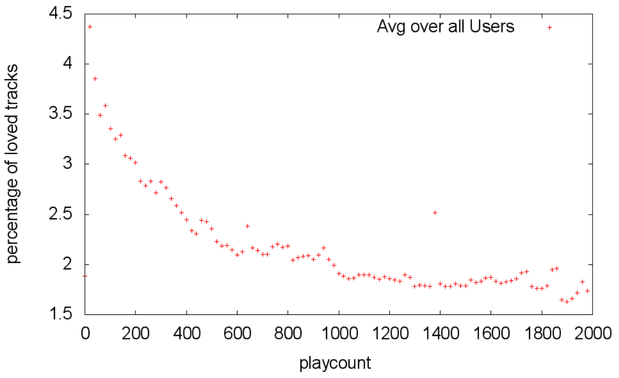

Figure 5 Percentage of loved tracks as playcount increases

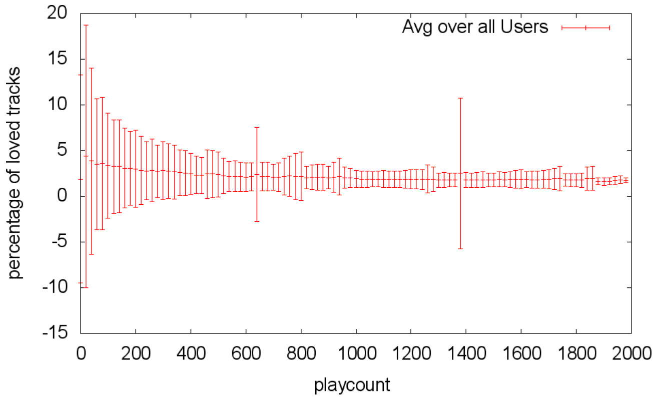

Figure 6 Standard deviation of percentage of loved tracks as playcount increases

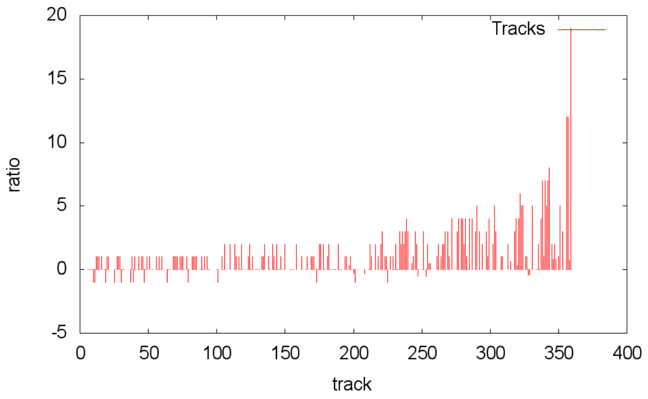

Figure 7 Effect of loved track on user

Figure 7, shows a plot of $E_{tu'}$ for a particular user u', for each of the track loved by that user. We observe that the effect is mostly positive; i.e., user u' played more of the tracks he loved after he loved them rather than before. But there are some tracks which exhibited a negative ‘Effect’. Once again, we try to answer the question whether all users in U' exhibited the same behaviour by plotting a normalised distribution of the ratio for all users in U' as shown in Figure 8. And here we find that the positive effect seen by user u', is not representative of the sample, as this latest plot shows that for almost two third of the sample, the overall effect of providing explicit feedback (loving a track) is negative.

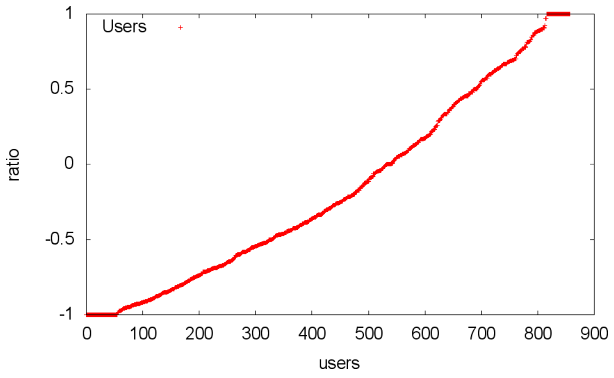

Figure 8 Overall effect of explicit feedback across all users in *U'*

5. DISCUSSIONS

Initial analysis showed that the distribution of the dataset is heavily skewed towards playcount (implicit feedback) as shown in Table 1 and Figure 1, 2 and 3. We showed through a histogram of the loved tracks v/s playcount and by way of the plot of the percentage of loved tracks in Figure 5 that the rate at which a user provides explicit feedback decreases with time, i.e, explicit feedback is more forthcoming at the beginning of the user's interactions with Last.fm than in the later stages. There are a number of plausible explanations for this observation. Firstly, it is possible that Last.fm as a RS becomes so good over time that the user does not see the need to provide explicit feedback anymore. Another reason may be that the user does not see any benefit to providing explicit feedback. It is impossible to find a definite answer without further analysis. Next we analysed the effect,

$E_{tu'}$ of loving a track *t* on a user *u'*. For each user we divided the

number of tracks played and marked as loved into two parts; those played before being marked as loved and those played after being marked as loved. Overall, as shown in Figure 8, almost two third of the users experienced a negative effect, i.e., the user played less of tracks after they have been marked than before they have been marked as loved. This observation is partly explained by users marking a track as loved only after listening substantially. Also, users may be using the 'Love a track' feature for different purposes (e.g bookmarking).

6. RELATED WORK

Feedback has been studied extensively in Information Retrieval. [3] provides an extensive overview of the literature on implicit feedback in IR and RS. Most previous works on this topic have studied either implicit or explicit feedback. [6] compared explicit and implicit feedback for online information retrieval, namely investigating the extent to which the two types of feedback are interchangeable. They found that some degree of substitution does exist. It is generally accepted that there is room for improvement in implicit feedback. But [1] recently showed that explicit feedback still needs to be improved. They found that user variability and inconsistency in providing explicit feedback, which they referred as natural noise negatively affects the accuracy of RS. This natural noise in explicit feedback bears similarity to the differences in relevance judgements that [5] found in their work on personalisation. Natural noise is a possible explanation we will explore when investigating the negative effect of explicit feedback that we observed in our sample.

7. CONCLUSIONS

In this paper we focussed on the characteristics of explicit feedback in an online radio station and recommender service. We looked at histories of tracks loved and tracks played for a sample of users. Our analysis showed that the rate of providing explicit feedback by a user decreases over time and that overall providing explicit feedback has a negative effect on the user's behaviour. In our future work we would like to scale up the sample size and further the analysis carried out here; namely we would like to further investigate this negative effect of explicit feedback. We would also like to experiment with ways in which these two types of feedback can complement each other in a RS.

8. REFERENCES

[1] Amatriain, X., Pujol, J., Tintarev, N., and Oliver, N. Rate it again: increasing recommendation accuracy by user re-rating. *Proceedings of the third ACM conference on Recommender systems*, ACM (2009), 173–180.

[2] Anand, S.S., Kearney, P., and Shapcott, M. Generating semantically enriched user profiles for Web personalization.

ACM Transactions on Internet Technology 7, 4 (2007), 22-es.

[3] Kelly, D. and Teevan, J. Implicit feedback for inferring user preference: a bibliography. *ACM SIGIR Forum*, (2003).

[4] O'Mahony, M.P., Hurley, N.J., and Silvestre, G.C. Detecting noise in recommender system databases. *Proceedings of the 11th international conference on Intelligent user interfaces - IUI '06*, (2006), 109.

[5] Teevan, J., Dumais, S., Horvitz, E., and others. Potential for Personalization. *ACM Transactions on Computer-Human Interaction 1*, 212 (2008), 1-35.

[6] White, R., Jose, J., and Ruthven, I. Comparing explicit and implicit feedback techniques for web retrieval: Trec-10 interactive track report. *NIST SPECIAL PUBLICATION SP*, (2002), 534–538.