Comparison of Implicit and Explicit Feedback from 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

Explicit and implicit feedback exhibits different characteristics of users' preferences with both pros and cons. However, a combination of these two types of feedback provides another paradigm for recommender systems (RS). Their combination in a user preference model presents a number of challenges but can also overcome the problems associated with each other. In order to build an effective RS on combination of both types of feedback, we need to have comparative data allowing an understanding of the computation of user preferences. In this paper, we provide an overview of the differentiating characteristics of explicit and implicit feedback using datasets mined from Last.fm, an online music station and recommender service. The datasets consisted of explicit positive feedback (by loving tracks) and implicit feedback which is inherently positive (the number of times a track is played). Rather than relying on just one type of feedback, we present techniques for extracting user preferences from both. In order to compare and contrast the performances of these techniques, we carried out experiments using the Taste recommender system engine and the Last.fm datasets. Our results show that implicit and explicit positive feedback complements each other, with similar performances despite their different characteristics.

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, combination of feedback, Taste recommender system.

1. INTRODUCTION AND MOTIVATION

All recommender systems (RS) require a model of the users' interests in order to function. A common approach to building such a user preference 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. Most of the research in RS has focussed on using one or the other type of feedback; only few have combined these two heterogeneous feedbacks.

In this paper, we present an overview of the differentiating characteristics of explicit feedback provided by the user in relation to implicit feedback gathered by a music recommendation service, namely, Last.fm[1] - an online radio station and music recommender service that recommends tracks to users based on their listening habits. It collects implicit feedback about the tracks played by a user, e.g. the number of times a track is played -commonly known as the playcount. It also allows users to express explicit feedback through its 'Love a track' or 'Ban a track' feature.

In a previous work [6] we presented a detailed examination of these two types of feedback. Explicit and implicit feedbacks provide different degrees of expressivity of the user's preferences. In order to build a more effective RS and maximise the potential of combining these two types of feedback, we compare the performances of each type of feedback on a RS.

For our experiment, we harvested the Last.fm profiles for 527 users, downloading metadata about all the tracks they listened to as well as the tracks they voted for using the 'Love track' feature. Together this provided us with a rich dataset that we used to experiment upon using the Taste[2] recommender engine. We used a collaborative filtering (CF) algorithm to generate the recommendations. However, the choice of the recommender algorithm is orthogonal to our concerns as we are not interested in the performance of the algorithms but rather the performance of the user preference models.

In the next section, we present an overview of the different characteristics of these two types of feedback. We then provide some notation and describe the datasets we used in Section 3. In Section 4 we present the techniques we used for extracting user preferences from our datasets. We present our experiments in Section 5, and a discussion of the results in Section 6. Finally, we conclude with some related work in Section 7.

HetRec '10, September 26, 2010, Barcelona, Spain.

[1] http://www.last.fm

[2] http://taste.sourceforge.net/

2. EXPLICIT AND IMPLICIT FEEDBACK

In order to develop an effective RS, user preferences need to be learned. However, it is 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, and it is likely that other factors as well serve as disincentives. 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 [2]. One possible reason may be because there are several domain-independent, objective, well researched and documented tools, such as Likert scale or questionnaires, for capturing and analysing explicit feedback. In contrast, an implicit feedback system relies on the application of domain-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 5 times, the system may infer that the user has an interest in that track.

There are similarities and differences between these two types of feedback. Both suffer from noise [1,3,5], and are sensitive to the user's context, albeit not to the same extent. In terms of differences, explicit feedback is scarce whereas implicit feedback is abundant. Explicit feedback is generally more accurate than implicit feedback in representing the user's interests (although this is dependent on the domain and the RS application). Also, explicit feedback can be positive or negative, whereas implicit feedback is only positive. Furthermore, explicit feedback tends to concentrate on either side of the rating scale, as users are more likely to express their preferences if they feel strongly for or against an item [2].

Explicit and implicit feedback provides different degrees of expressivity of the user's preferences. In typical explicit feedback RS, the user will provide ratings for items on a Likert scale. The rating scale will usually go from 'I like it a lot' to 'I do not like it'. Thus explicit feedback captures both positive and negative user preferences. On the other hand, implicit feedback can only be positive. For example, if a user did not listen to a track that does not imply he does not like the track. However, implicit feedback can be mapped to degree of preference analogous to going from the middle of a continuous scale to its positive extremity. For example, if a user listened to track A, 10 times and track B, 100 times, then we can infer that he has a higher preference for track B than track A. This leads to the point that implicit feedback tend to be relative where as explicit feedback is absolute. For example, a user listening to a track 10 times may still express high preference if typically the user tends to listen to each track once or twice. Another point is that implicit feedback is domain dependent. For example, in a movie recommender system, a user may watch an actor or actress 10 times, but that does not imply he has a relative high preference for that artist. It could be that the artist is a part of a series that the user watches regularly.

To study the characteristics of implicit and explicit feedbacks, we used data from Last.fm. The latter provides its users the functionality to love (explicit positive feedback) and ban (explicit negative feedback) a track. Last.fm also 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. It provides plug-in software that work with the media players to send the user information to the Last.fm servers in a process commonly referred to as scrobbling. Last.fm provides an extensive set of tools and APIs to harvest its rich dataset. Unfortunately, as the API does not expose a user's banned tracks, so we were only able to build datasets that included positive explicit feedback (loved tracks) and implicit feedback (played tracks). In Table 1 below, we summarise all the pertinent characteristics of implicit and explicit feedback.

Table 1. Characteristics of explicit and implicit feedback

	Implicit feedback	Explicit feedback
Accuracy	Low	High
Abundance	High	Low
Context-sensitive	Yes	Yes
Expressivity of user preference	Positive	Positive and Negative
Measurement reference	Relative	Absolute

In the next section, we introduce some notations used in the remainder of this paper and also describe the datasets we harvested 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: $S \subseteq U \times T \times A \times Z$, 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 random users of Last.fm. For each of these users, we collected information about all the tracks they loved. Removing the 6,382 users who did not love any tracks left us with $|U| = 10{,}012$ users and metadata about $|L| = 1{,}833{,}804$ tracks. We then queried Last.fm for metadata describing all the tracks played by a subset of users U' such that user $u' \in U'$, lovecount, $|L_{u'}| \geq 20$, playcount $|S_{u'}| \geq 20$ and $|S_{u'}| \leq 2000$. We used this restrictive subset for practical reasons, namely the time constraint within which we could practically mine the metadata while also ensuring that users had a sufficient amount of data to mine. In order to give an overview of the users' profiles in terms of the number of tracks loved and played, we reproduce below in Figure 1 and Figure 2, the cumulative frequency distribution (CFD) plots of the lovecount and placount respectively from our

previous work in [6]. Thus, we have $|U'| = 867$ users for whom we had the complete history of the tracks they played and the tracks they loved. As we are interested in the combination of feedbacks, our dataset only includes users which have both played and loved tracks.

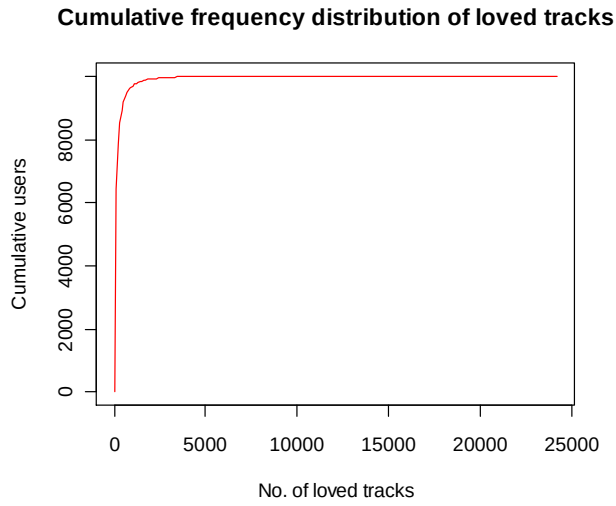

Figure 1. CFD plot of tracks loved by 10,012 users

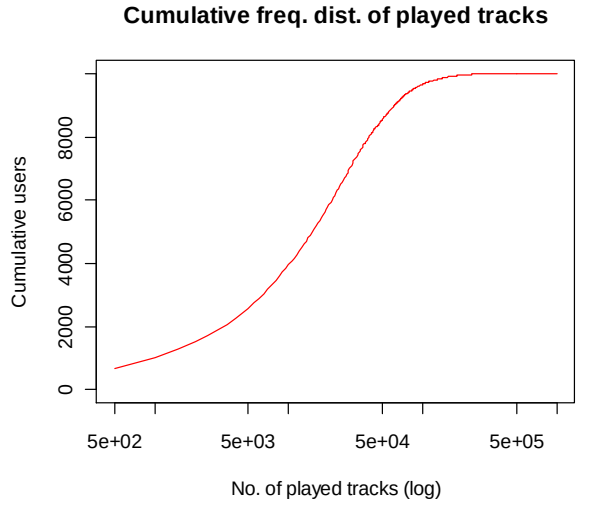

Figure 2. CFD plot of tracks played by 10,012 users

Typically in music recommender systems, user profile information is recorded on a track level whilst recommendations are made at the artist level [4]. Thus, we aggregated our dataset at the artist level. We then divided this dataset into two parts. The first part, which we called the Artist Playcount Dataset (initially 865 users and 29,094 unique artists) stores metadata about all the artists played by the various users and the playcount for each artist by each user. It represents the implicit feedback dataset. Similarly, the second part, called the Artist Lovecount Dataset (initially 865 users and 11,090 unique artists) stores metadata about all the artists loved by the users and the lovecount for each artist by each user. It represents the explicit feedback dataset. In order to avoid the few plays of an artist from affecting the overall performance, we removed from the Artist Playcount Dataset, all records where the user has played an artist less than 20 times. This shed a large part of the dataset such that the Artist Playcount Dataset now consisted of 527 users and 2167 unique artists. Similarly, we pruned down the Artist Lovecount Dataset to the same 527 users (8242 unique artists) although we did not remove records based on lovecount. Table 2 below summarises the various characteristics of these two datasets.

Table 2. Characteristics of the two datasets

Characteristics	Artist Playcount Dataset	Artist Lovecount Dataset
Type of feedback	Implicit (positive)	Explicit (positive)
No. of users (preprocessing)	865	865
No of artists (preprocessing)	29,094	11,090
Processing done	Removed the records where users had < 20 playcount per artist and match users in both datasets	Match users in both datasets
No. of users (postprocessing)	527	527
No. of artists (post-processing)	2,167	8,242

In the next section we discuss how we derived the user preferences for the artists from the above datasets.

4. CALCULATING USER PREFERENCES

We used the following three methods for calculating the user's preference for an artist. We tested the following three methods for calculating the user's preference for an artist (user-artist preference):

- Absolute: the preference is a count of the number of times a user has played or loved an artist
- Normalise: the preference is the ratio of counts of the number of times a user has played or loved an artist to the total number of artists played or loved by the user. Thus, we normalise the artist playcount or lovecount such to account for a user's usage of the system.
- Log: this is similar to the Absolute measure, except that preference is calculated as the log to base 10 of the artist playcount or lovecount.

In order to understand the user preference values obtained using the three above methods when applied to the Implicit dataset (Artist Playcount Dataset) and the Explicit dataset (Artist Lovecount Dataset), we show in Figure 3, the histograms of these preference values. For each set of preference values, we

divided the range in 5 bins and counted the number of values in each bin.

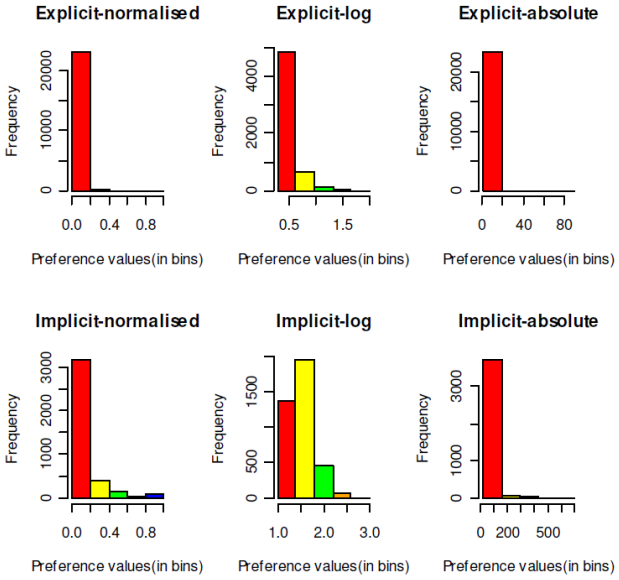

Figure 3. Histograms of user preferences

5. EXPERIMENT

In our experiments, we applied user preference data obtained using the above described techniques as input to the Taste Collaborative Filtering (CF) engine from the Apache Mahout project. We setup Taste for user-based CF, using a nearest neighbourhood value of 3 and Pearson Correlation as the measure of similarity. We used 90% of the user preference data to train the Taste engine and the evaluation was carried out the remaining 10%. We measured the outcome of each experiment in terms of the Root Mean Squared Error (RMSE) between the given user preferences and the predicted user preferences. For each experiment, we did five runs and averaged the results.

Table 3 shows the evaluation of the three methods for calculating the user preferences for an artist. Across both datasets, the normalised techniques produced the best results. Our better RMSE values than those traditionally seen in RS may be explained by the fact that we are only using positive user preferences – this is a limitation of our datasets and of any feedback dataset collected from Last.fm. Bearing in mind that our dataset is relatively small compared to others [10], it may also be the case that we were too aggressive in the pre-processing described in Section 3.1.

6. DISCUSSIONS

As shown in Table 3, calculating user preferences in terms of absolute counts produced the worst results. This is because it does not account for the usage patterns of the user in contrast to normalised figures which produced the best results. If we exclude the esults from the experiment with absolute counts, we notice that both the implicit and explicit datasets produced similar results for the Normalised and Log experiments. Despite the different characteristics of these two datasets, they produced similar performances. This is counter intuitive as implicit feedback is seen as less accurate than explicit feedback [1][2]. The histograms in Figure 3, show that the calculated user preferences are all skewed to the lower end of the scale, except in the case of the user preference values calculated using the Log method on the Implicit dataset (i.e Artist Playcount Dataset). This shows a lack in diversity of the preference values. The extremely good RMSE values and the lack in diversity in user preference values may be due to the limited size of the dataset and a consequence of the pre-processing we carried out. Another possible explanation we will explore as part of our future work is the suitability of RMSE as the evaluation metric for comparing the performances of the methods for calculating user preferences.

7. RELATED WORK

Feedback has been studied extensively in Information Retrieval. [7] 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. [9] 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. There is a disproportionate amount of literature studying implicit feedback for use in web search engines, personalisation and recommender systems. This is probably due to the fact that 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. They propose a system of re-feedback as a solution and suggest that removing noise in explicit feedback can be more beneficial in improving RS accuracy rather than gathering explicit feedback on unseen items. This natural noise in explicit feedback bears similarity to the differences in relevance judgements that [8] found in their work on personalisation. [10] proposed a method of learning multiple matrices over common items in order to improve overall predictive performance. Although the authors used datasets from Last.fm to illustrate their techniques, their work is

Table 3. Evaluation of three ways of computing user-artist preference using RMSE

User-Preference Method	RMSE for Artist Playcount Dataset						RMSE for Artist Lovecount Dataset					
	R1	R2	R3	R4	R5	AVG	R1	R2	R3	R4	R5	AVG
Absolute	50.53	65.82	99.58	86.06	109.20	82.24	1.76	2.82	3.38	3.20	3.78	2.99
Log	0.30	0.33	0.30	0.30	0.35	0.32	0.33	0.33	0.30	0.37	0.42	0.35
Normalised	0.09	0.11	0.07	0.04	0.10	0.08	0.04	0.04	0.07	0.04	0.06	0.05

on combination of metadata about played tracks and user generated tags. Their combination technique can be applied to our explicit and implicit datasets. The researchers in [5] studied the use of collaborative filtering on implicit feedback datasets. They discuss the properties of such datasets and proposed the notion of applying confidence levels to interpret the implicit feedback measures as positive and negative preference values. They test their algorithm for calculating user preferences using Latent factor models rather than CF as we did. In contrast to our work, they do not have any comparative performance between explicit and implicit feedback as the combination of these two types of feedback was not the aim of their study.

8. CONCLUSIONS

In this paper we focussed on comparing implicit feedback and explicit feedback, two types of feedback with different characteristics. We built implicit and explicit feedback datasets out of the tracks played and tracks loved, respectively, for a random sample of users on Last.fm. We compared and contrasted three techniques for extracting user preferences from these datasets. Explicit and implicit feedbacks provide different degrees of expressivity of the user's preferences. In order to build more effective RS and maximising the potential of combining these two types of feedback, we compared the performances of each type of feedback on a RS. Our experiments show that although they have different characteristics, the two datasets produced similar performances. Our aim in studying explicit and implicit feedback is to better understand their characteristics in order to combine them effectively in a RS. Thus, in our future work, we will be experimenting with different ways of combining these two types of feedback in a user preference model and finding better evaluation measures that work across datasets.

9. ACKNOWLEDGMENTS

Our thanks to Last.fm for making a rich dataset available to the research community and the public in general through their extensive API.

10. 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] Amatriain, X., Pujol, J., and Oliver, N. I like it... I like it not: Evaluating User Ratings Noise in Recommender Systems. In G. Houben, G. McCalla, F. Pianesi and M. Zancanaro, User Modeling, Adaptation, and Personalization. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, 247-258.

[3] 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.

[4] Herrada, C. Music recommendation and discovery in the long tail. 2008.

[5] Hu, Y., Koren, Y., and Volinsky, C. Collaborative Filtering for Implicit Feedback Datasets. 2008 Eighth IEEE International Conference on Data Mining, (2008), 263-272.

[6] Jawaheer, G., Szomszor, M., and Kostkova, P. Characterisation of explicit feedback in an online music recommendation service. ACM Recommender Systems Conference 2010, Barcelona (in press), (2010).

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

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

[9] 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.

[10] Williamson, S. and Ghahramani, Z. Probabilistic models for data combination in recommender systems. NIPS 2008 Workshop:, (2008), 1-4.