Music Recommendation: Audio Neighbourhoods to Discover Music in the Long Tail

Susan Craw, Ben Horsburgh, and Stewart Massie

School of Computing Science & Digital Media
Robert Gordon University, Aberdeen, UK
s.craw@rgu.ac.uk and s.massie@rgu.ac.uk
http://www.rgu.ac.uk/dmstaff/lastname-firstname

Abstract. Millions of people use online music services every day and recommender systems are essential to browse these music collections. Users are looking for high quality recommendations, but also want to discover tracks and artists that they do not already know, newly released tracks, and the more niche music found in the 'long tail' of on-line music. Tag-based recommenders are not effective in this 'long tail' because relatively few people are listening to these tracks and so tagging tends to be sparse. However, similarity neighbourhoods in audio space can provide additional tag knowledge that is useful to augment sparse tagging. A new recommender exploits the combined knowledge, from audio and tagging, using a hybrid representation that extends the track's tag-based representation by adding semantic knowledge extracted from the tags of similar music tracks. A user evaluation and a larger experiment using Last.fm user data both show that the new hybrid recommender provides better quality recommendations than using only tags, together with a higher level of discovery of unknown and niche music. This approach of augmenting the representation for items that have missing information, with corresponding information from similar items in a complementary space, offers opportunities beyond content-based music recommendation.

Keywords: Recommender Systems · Novelty and Serendipity · Knowledge Extraction · CBR Similarity Assumption

1 Introduction

Long tail marketing techniques have a sales model based upon promoting less popular products in the 'long tail' as shown in Figure 1. It is most effectively employed by online retailers, so is very relevant for online music services. Therefore music recommenders should not overlook or ignore recommendations in this 'long tail'. A track that is not often listened to may be a niche recommendation that offers serendipity and an opportunity to discover new music. These recommendations encourage sales in this important area of the online music market.

Query-by-example music recommenders have access to different representations for items: audio representations like texture (timbre), harmony, rhythm; meta-data such as track title, artist, year, etc; and semantic information such as

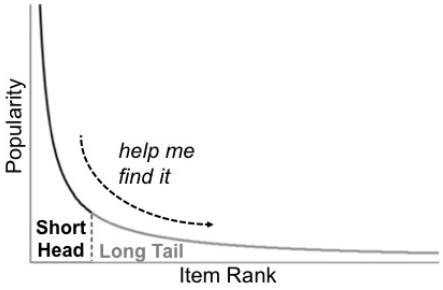

Fig. 1. Long Tail of Recommendation

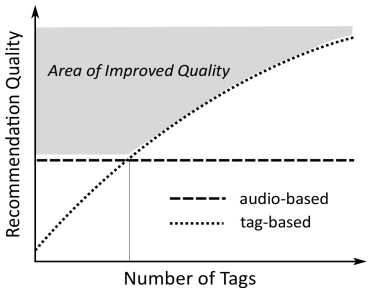

Fig. 2. Recommendation with Sparse Tags

social tagging from on-line music services. Many state-of-the-art recommender systems make use of social tagging [1]. These tags can provide useful semantic information for recommendation including genres, topics, opinions, together with social, contextual and cultural information. However, not all tracks within a collection are tagged equally: popular tracks tend to have more tags describing them, and niche tracks may have no tags at all.

Figure 2 illustrates the effect of tag sparseness on recommenders. When the query has few tags the tag-based recommendations have poor quality, and an audio recommender that is not affected by tags provides better recommendations. Conversely, tag-based recommenders cannot reliably identify good recommendations if they have few tags. This diagram motivates the idea for a hybrid recommender that combines tag and audio representations. Hybrid recommenders typically merge representations, or combine the processes of sub-recommenders. Our approach is different, because it augments existing tags when necessary. It exploits the similarity assumption of case-based reasoning to extract additional tag knowledge from audio neighbourhoods. It extends the notion of recommendation, that "similar tracks will be good recommendations", to "similar tracks will have useful tagging". It injects novelty and serendipity into its recommendations, since it is not biased against sparsely tagged tracks in the 'long tail'.

In the rest of this paper we first review relevant literature in music recommendation and serendipity. Section 3 introduces our music dataset and describes the tag and audio representation for tracks. Our new recommender that combines knowledge extracted from audio neighbourhoods with existing tagging is presented in Section 4. Its performance for accuracy and novelty in a user trial and a system-centric evaluation is discussed in Sections 5 & 6.

2 Related Work

One advantage that content-based recommendation has over collaborative filtering is that it does not suffer from 'cold start' where user data is not available, nor from the 'grey sheep' problem [2], where users with niche tastes are excluded because no similar users exist. Instead, content-based recommendation

has the potential to recommend any track within a collection to a user. The main disadvantage of content-based recommenders is that they rely entirely on the strength of each track's representation. If the representation is weak, then it is difficult to define meaningful similarity, and the quality of recommendations will be poor. The two core approaches used to represent music tracks are audio content and tags. However, audio content representation is weak, and does not provide high quality recommendations. When tag-based features are used, high quality recommendations can be made, and these have been shown to provide better quality recommendations than collaborative filtering methods [3].

Tags come directly from users and are most commonly generated socially, via user collaboration, and so a wealth of social and cultural knowledge is available to describe tracks. However, this also means that tagging is not evenly distributed, and a popularity bias in music listening habits further skews the distribution of tags [4]. New and niche tracks are in the popularity 'long tail' of Figure 1, so few people are listening to them, so few/no people are tagging them, so these tracks have few/no tags, so tag-based recommenders do not recommend them, so few people are listening to them, etc.

The Million Song Dataset [5] includes a Last.fm contribution containing a tagged dataset[1] and the tagging of this reference dataset is typical. It contains almost 950k tracks tagged with more than 500k unique tags, and on average each tagged track has 17 tags, but 46% of tracks do not have any tags at all. The 25k most tagged tracks each has 100 tags, but this number very quickly drops off in a 'long tail' similar to Figure 1. Halpin et al. found similar tagging 'long tails' in the various del.icio.us sites they investigated [6].

Auto-tagging is designed to overcome sparse social tagging [7]. A popular approach learns tags that are relevant to a track from a Gaussian Mixture Model of the audio content [8]. While this approach may guarantee a certain degree of tagging throughout a collection, humans are not involved with the association of tags with tracks, and thus it is likely that erroneous tags will be propagated to many tracks. It is also easy to learn common tags which co-occur often, but runs the risk of excluding more niche tags, which may be most appropriate for tracks with few tags. Track similarity has also been used for auto-tagging style and mood [9]. Here tag vectors of similar tracks are aggregated, and the most frequently occurring tags are propagated. The advantage of this method is that there is no attempt to correlate content directly with tags, or presume that tagging must fit any prior distribution. Instead it exploits consensus of human tagging. We take inspiration for our pseudo-tagging from this approach, but the way we use pseudo-tags recognises that they are not 'real' tags [10].

Hybrid representations that combine tag and audio representations can also cope with sparse tagging. Levy & Sandler [11] create a code-book from clustered audio content vectors, and these muswords are used as the audio equivalent of tags. Concepts are extracted using Latent Semantic Analysis (LSA) from the combined representation of tags and muswords. In previous work we concatenated tag and texture representations, before extracting latent concepts [12].

[1] http://labrosa.ee.columbia.edu/millionsong/lastfm

Taste in music is highly subjective, and so generating novel and serendipitous recommendations is particularly important, and challenging. Kaminskas & Bridge's [13] exploration of serendipity notes the trade-off in standard recommender approaches between quality and serendipity. The Auralist [14] and TRecS [15] hybrid recommenders address this trade-off by amalgamating sub-recommenders with differing priorities including quality, serendipity and novelty. In both systems, special novelty and serendipity recommenders influence choice.

3 Music Collection

Our music collection was created from a number of CDs that contain different genres, a range of years, and many compilation CDs to keep the collection diverse [16]. This dataset includes 3174 tracks by 764 separate artists. The average number of tracks per artist is 4, and the most common artist has 78 tracks. The tracks fall into 11 distinct super-genres: Alternative (29%), Pop (25%), Rock (21%), R&B (11%); and Dance, Metal, Folk, Rap, Easy Listening, Country and Classical make up the remaining 14% of the collection. We now describe two standard music representations applied to this dataset: one based on the tagging of Last.fm users; and a texture representation built from audio files.

Music tracks often have tag annotations on music services. Last.fm is used by millions of users, and their tagging can be extracted using the Last.fm API[2]. When a user listens to a track, they may decide to tag it as 'rock'. Each time a unique user tags the track as 'rock', the relationship of the tag to the track is strengthened. A track's tag vector $t = < t_1\ t_2\ \ldots,\ t_m >$ contains these tag frequencies t_i, and m is the size of the tag vocabulary. Last.fm provides normalised frequencies for the tags assigned to each track, with the most frequent tag for a track always having frequency 100. A total of $m = 5160$ unique tags are used for our music collection in the Last.fm tagging. On average each track has 34 tags with a standard deviation of 24.4, and the most-tagged track has 99 tags. The tagging is realistically sparse: 3% of the tracks have no tags at all; there are 24% with fewer than 10 tags; and 42% with fewer than 20 tag.

Texture (timbre) is one of the most powerful audio-based representations for music recommendation [17]. We use the MFS Mel-Frequency Spectrum texture [18], available through the Vamp audio analysis plugin system[3]. MFS is a musical adaptation of the well-known Mel-Frequency-Cepstral-Coefficients (MFCC) texture [19]. Figure 3 illustrates the main stages in transforming audio tracks into MFS (and MFCC) vectors, and demonstrates the relationship between MFS and MFCC. Audio waveforms, encoded at 44.1kHz, are first split into windows of length 186ms, and each window is converted into the frequency domain using a Discrete Fourier Transform (DFT). Each frequency spectrum computed has a maximum frequency of 22.05kHz, and a bin resolution of 5.4Hz. Next, each window is discretised into a feature vector, based on the mel-scale [20]. We use 40 mel filters, the granularity found to be best for aggregation-based recommender

[2] www.last.fm/api
[3] www.vamp-plugins.org/download.html

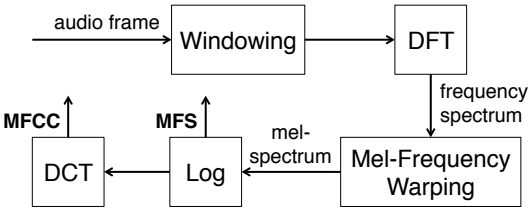

Fig. 3. Extraction of MFS and MFCC

models [18]. A mean feature vector MFS is computed for each track and these are used to construct a track-feature matrix. Latent Semantic Indexing (LSI) is used to discover musical texture concepts, and each track is projected into this texture space to create its MFS-LSI texture vector.

4 Hybrid Recommenders

Query-by-track recommender systems *Tag* and *Audio* may be defined using standard vector cosine similarity with these tag-based and texture vector representations. However, each individually can be problematic. Tag can give good recommendations but cannot recognise recommendations that have few or no tags, and cannot retrieve good recommendations for poorly tagged queries. Audio does not suffer this problem because all tracks have audio data, but does not offer the same performance as Tag with well-tagged tracks. Our two new hybrid recommenders are designed to reduce the semantic gap between audio content and tags, and allow recommendation quality to be improved when tracks are under-tagged. They take advantage of tagging, but also exploit similarity neighbourhoods in the audio space to learn pseudo-tags. These hybrid query-by-track recommenders are defined by standard tag-based representations and cosine similarity retrieval.

4.1 Learning Pseudo-Tags

Pseudo-tagging is different from other hybrid representations that combine tag- and audio-based representations. Instead, pseudo-tags are extracted from the tags of tracks that have similar audio content, and these pseudo-tags are used within a tag-based representation.

The first step to generating pseudo-tags for a track is to find tracks that are similar to this track. A k nearest-neighbour retrieval using cosine similarity in the musical texture MFS-LSI space identifies the K most similar tracks. A rank-based weighted sum of the tag vectors $t(1)$ ... $t(K)$ for these K retrieved tracks are used to learn the pseudo-tag vector $p = < p_1\ p_2\ \ldots\ p_m >$:

$$p_i = \sum_{k=1}^{K} (1 - \frac{k-1}{K}) t_i(k) \tag{1}$$

where $t_i(k)$ is the frequency of the ith tag in the tag vector of the kth nearest neighbour track[4]. Retrieved tracks from lower positions have less influence and so the retrieval list is restricted to K=40 neighbours for our experiments.

Our *Pseudo-Tag* recommender retrieves tracks using cosine similarity of these pseudo-tag vectors. The pseudo-tag representation reduces sparsity in tag-based representations because audio neighbourhoods of tracks are unlikely to be uniformly sparsely tagged. The advantage of using pseudo-tags over audio content directly is that factors such as context and opinions will also be present in the pseudo-tag representation, inherited from the neighbourhoods.

4.2 Augmenting Tags with Pseudo-Tags

Pseudo-tag vectors are useful when a track has few tags, but can influence the representation too much if the track is already well-tagged. In particular, the pseudo-tag vector has ignored any tag information that may be associated with the track itself, and includes all tags that are associated with any of the track's neighbours. Our *Hybrid* recommender uses a tag representation that augments any existing tags for a track by merging the track's learned pseudo-tag vector p. with its tag vector t.

A pseudo-tag vector p is much less sparse (fewer zero frequencies) than a tag vector t because p has been aggregated from tag vectors belonging to a number of tracks in the neighbourhood of t's track. The first step in creating the hybrid tag/pseudo-tag representation selects the number of pseudo-tags P to be included, so that it balances the number of existing tags T; i.e. non-zero frequencies in t. We experimented with different values of $P = 0, 10, 20, ...100$. The solid dark line in Figure 4 shows the best performing number of pseudo-tags P for tracks with different numbers of tags T grouped into tag buckets of size 10. Under-tagged tracks need higher numbers of pseudo-tags, and well-tagged tracks use fewer; this is consistent with intuition. The dark dashed line is the line-of-best-fit through these data points. We select the number of pseudo-tags retained based on an approximation of this line: $P = 100 - T$ for our dataset.

The vector of selected pseudo-tags $\tilde{p}$ is created by retaining the P highest frequencies in p and zeroing the rest. Next, an influence weighting α determines the influence of the selected pseudo-tags $\tilde{p}$ on the hybrid vector h[4]:

$$h_i = \alpha\tilde{p}_i + (1 - \alpha)t_i \tag{2}$$

Experiments similar to those for P, alter the weighting α from 0 to 0.5 in steps of 0.1. The grey lines and secondary axis in Figure 4 show the best weighting and dashed line-of-best-fit, estimated as $\alpha = 0.5 * (1 - T/100)$.

The Hybrid recommender uses representation h to retrieve tracks. For well-tagged tracks, the tag vector dominates h, and the Hybrid recommender benefits from the strengths of tag-based recommendation. Weakly-tagged tracks are augmented by the inclusion of pseudo-tags in h. The Pseudo-Tag representation is a variant of Hybrid, where the weighting α is 1, and all pseudo-tags are used.

[4] All tag-based vectors t, $t(k)$, p, $\tilde{p}$, and h are routinely normalised as unit vectors before use. For clarity, normalisation has been omitted from equations (1) & (2).

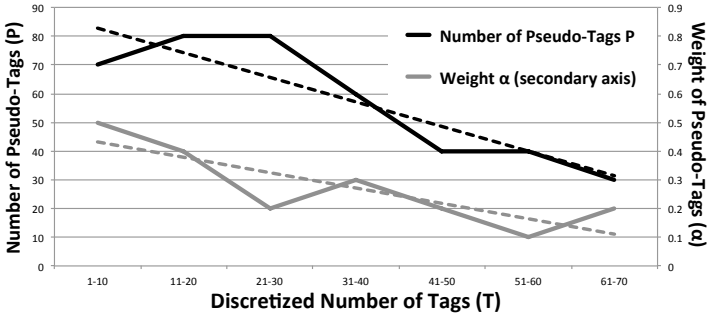

Fig. 4. Number of Pseudo-Tags and Weighting for Hybrid

5 User Evaluation

A user evaluation was undertaken to test the quality of recommendations with real users, but also to measure the level of discovery of new tracks in the recommendations. The two new hybrid recommenders Pseudo-Tag and Hybrid are included in the experiments to see the effect of replacing or augmenting tags with learned pseudo-tags. The Tag recommender is also included as a baseline.

5.1 Design of User Evaluation

The selection and presentation of the query and recommendations are designed to avoid bias, and the screen provides the same information for every query. The user is shown a query track and the top five recommended tracks from a single recommender. Each track has its title, artist, and a play button that allows the user to listen to a 30 second mid-track sample. The recommender is chosen randomly, and the top five recommendations are presented in a random order. Each query track is selected at random from either a fixed pool or the entire collection, with 50:50 chance. The pool contains 3 randomly selected tracks for each of the 11 genres in the collection. The 33 pool tracks will be repeated more frequently, whereas the other tracks are likely to be used at most once. Users evaluate as many queries as they choose, without repetition.

A user gives feedback on the quality of each of the recommendations by moving a slider on a scale between very bad (0) to very good (1). Each slider is positioned centrally on the scale initially, and records feedback in $\frac{1}{1000}$ths. To capture feedback on each track's novelty, the user also selects from 3 options: *knows artist & track*; *knows artist only*; or *knows neither*. When feedback for a query is complete, the user presses submit to save slider values and novelties for its 5 recommendations.

5.2 User Participation

The on-line user evaluation was publicised through social media and mailing lists. It was available for 30 days and a total of 132 users took part, evaluating a total

of 1444 queries. There were 386 queries where all 5 recommendations scored 500, suggesting that the user clicked submit without moving any of the sliders. These were discarded, and the remaining 1058 valid queries provide explicit feedback on their recommendations. On average users evaluated recommendations for 6.24 queries, and the most active user scored 29 queries.

Prior to providing feedback, each user completed a questionnaire to indicate their gender, age, daily listening hours, and musical knowledge: *none* for no particular interest; *basic* for lessons at school, reads music magazine/blogs, etc.; or *advanced* for play instrument, edit music on computer, professional musician, audio engineer, etc. Each user also selects any genres they typically listen to. Figure 5 contains a summary of the questionnaire data, showing there is a good spread across age, gender, and knowledge, and that the musical interests align well with the genres in the pool and collection overall.

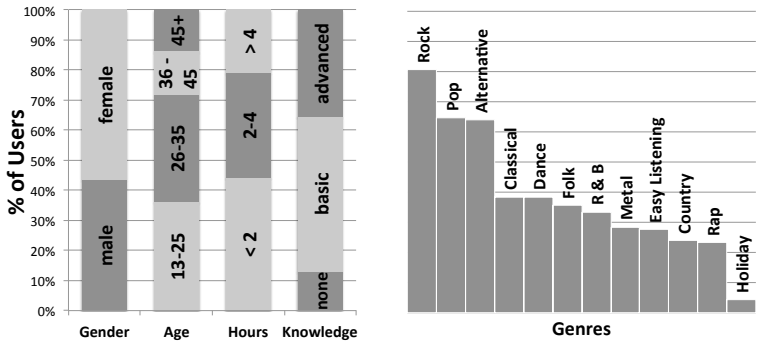

Fig. 5. Profile of User Group

5.3 Results for Recommendation Quality

We calculate recommendation quality Q for a query-recommendation pair q, r by aggregating the individual scores by a user u, across all users U providing feedback for this query's recommendations. We then use a $Q@N$ average of the top N recommendations r_n to evaluate the recommendations for query q.

$$\mathrm{Q}(q,r) = \frac{1}{|U|}\sum_{u\in U}\mathrm{score}_u(q,r) \qquad\qquad Q@N(q) = \frac{1}{N}\sum_{n=1}^{N}\mathrm{Q}(q,r_n) \qquad (3)$$

Figure 6 shows the $Q@N$ values averaged across all pool queries in the user evaluation. We focus on pool queries, approximately 47% of all queries, since non-pool queries are typically evaluated by only a single user. The error bars indicate 95% confidence and are included to give a sense of separation for the graphs. They show quite high variability because of the following: a user provides feedback on the recommendations of a single recommender for each of

their queries, there are a relatively small number of queries, and each user gives feedback on only a subset of these.

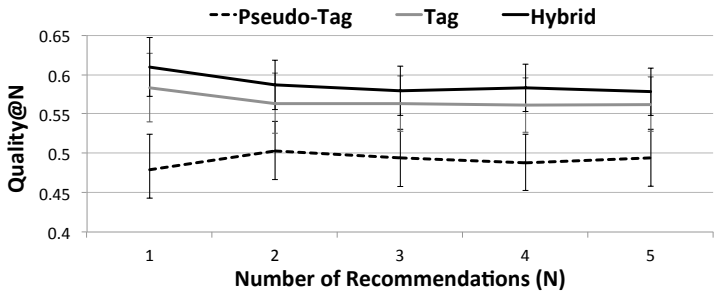

Fig. 6. Recommendation Quality from User Evaluation

The Hybrid recommender provides higher quality recommendations than Tag and Pseudo-Tag by augmenting existing tags used by Tag with some of the pseudo-tags from Pseudo-Tag. The small improvement of Hybrid over Tag shows that augmenting the tags with pseudo-tags does not damage recommendation. Sparsely tagged tracks gain from additional pseudo-tags, although pseudo-tags on their own are not so good for recommending. It appears that the adaptive balancing of existing tags with pseudo-tags from the audio neighbourhood is helpful. The equivalent figure for all queries is similar, but it has the Hybrid and Tag graphs slightly closer together, and a larger separation from Pseudo-Tags.

In general, the $Q@N$ drops slowly as N increases, so tracks later in the recommender's list gradually decrease in quality as expected. Remember that the recommendations are presented in a random order so there is no user bias towards tracks higher up the list. It is not clear why the top recommendation by Pseudo-Tag is poorer than those that are ranked lower, but possibly users are not good at ranking accurately recommendations that are generally poor.

5.4 Results for Discovery with Quality

We are interested in recommenders that offer new and niche tracks as serendipitous recommendations whilst retaining the all-important quality of recommendation. Here we explore the novelty of recommendations in the user evaluation by analysing the user replies about knowing the track.

One interesting observation from the user evaluation is the confirmation that users give higher feedback to recommendations that they know, and slightly higher ratings to tracks where they know the artist. Figure 7 shows the average score for recommendations from users according to the user's knowledge of the artist and track.

Figure 8 captures the quality and novelty of all recommendations in the user evaluation. The location and spread of the clusters for Hybrid (black), Tag

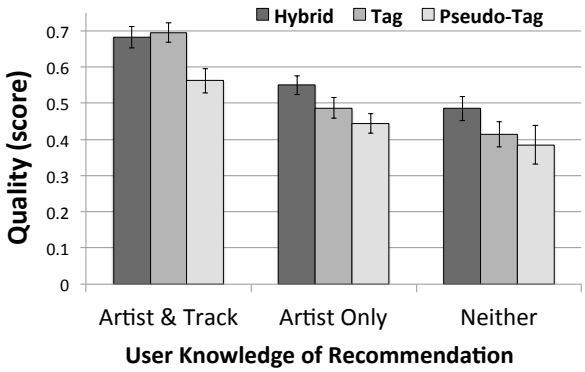

Fig. 7. Quality by What is Known

(grey) and Pseudo-Tag (white) demonstrate well the trade-off between quality and novelty. Good quality recommenders are higher; and those suggesting more recommendations that are unknown are towards the right, so best recommenders that combine novelty with quality are towards the top right. The individual points in the clusters show the score@N and % unknown tracks for different $N = 1..5$. For hybrid the top point with highest quality is $N = 1$; larger Ns have increasingly lower quality. For Tag and Pseudo-Tag the isolated point to the left is $N = 1$; the other 4 Ns are very tightly clustered.

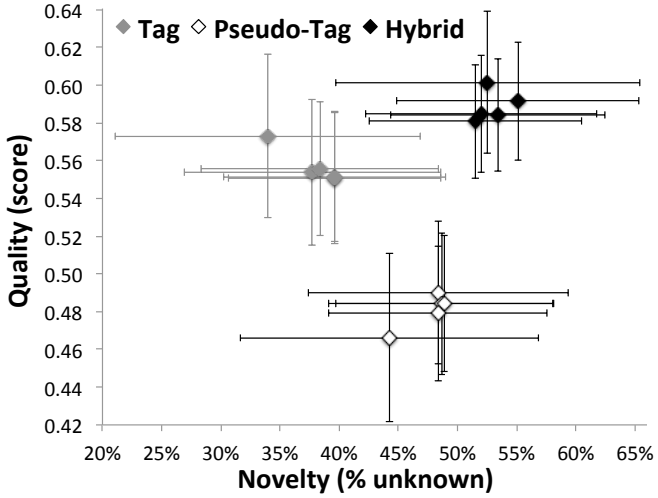

Fig. 8. Balance Between Quality and Novelty from User Evaluation

Hybrid achieves quality recommendations and is able to suggest unknown tracks; it recommends novel tracks 50-55% of the time. Although Tag has comparable quality it is significantly poorer for novel recommendations. Only 30-40% of its recommendations are unknown because Tag tends to recommend well-tagged tracks and these are also often well-known. Hybrid and Pseudo-Tag are comparable for novelty since they each exploit the tags inherited from neighbouring tracks. However, quality is also important, and Hybrid gives significantly better recommendations – despite the users' quality bias towards known tracks!

6 Evaluation Using Last.fm User Data

A larger system centric evaluation has also been undertaken using leave-one-out testing on the whole music collection. We use the socialSim score that defines the recommendation quality Q as the association between the numbers liking and listening to tracks q and r:

$$Q(q,r) = \text{socialSim}(q,r) = \frac{\text{likers}(q,r)}{\text{listeners}(q,r)} =_{est} \frac{\text{likers}(q,r)}{\text{listeners}(q) \cdot \text{listeners}(r)} \quad (4)$$

where likers(q,r) and listeners(t) are available through the Last.fm API (see [21] for details). This evaluation uses socialSim@N averaged over all tracks q in the collection. Notice that tag data used in the recommenders is distinct from user data underpinning socialSim, although both are extracted from Last.fm.

Figure 9 contains the quality results for Hybrid, Tag and Pseudo-Tag as in the user evaluation, now for $N = 1..10$ recommendations. Results for an Audio recommender based on MFS-LSI texture are also included as a purely audio-based baseline; it was omitted from the user evaluation, to reduce the number of very poor recommendations presented to users for feedback. The 95% error bars are much more compressed now because of the very large set of queries from leave-one-out testing, and the combined opinions of very many Last.fm users.

The overall findings confirm those from the user evaluation: Hybrid and Tag are comparable, with Hybrid having a tendency to give higher quality recommendations. Pseudo-Tag is significantly poorer and, as expected, Audio is much poorer still. Compared to Figure 6, the recommendation quality drops more quickly for all three recommenders, and continues decreasing as N increases. With users, later recommendations did not dilute the quality of earlier ones, but since a user rated all 5 recommendations at the same time perhaps less variation between a query's recommendations is natural. Also, we have seen that whether a track is known or not affects a user's score, and the system-centric evaluation does not suffer the effect of individual subjectivity. The placing of the Hybrid and Tag graphs is slightly higher than with users, and there is a significantly increased gap between Pseudo-Tag and Tag. However, exact values are not really comparable. There is a prevalence of zeros in the socialSim score when there is no evidence of likers in user data, but users may give less pessimistic ratings for poor recommendations when responding to real queries, and Q is unlikely to generate 0; i.e. all users scoring 0 for a recommendation.

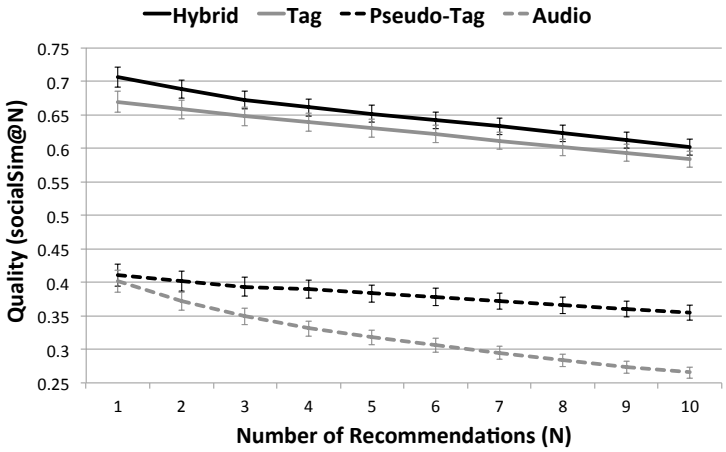

Fig. 9. Recommendation Quality from Last.fm User Data

The notion of novelty is difficult to capture from user data. Instead we introduce an artificial measure that exploits the link between tracks that are well-known and the level of tagging. Recommendations with few tags will be classed as novel and the % of novel recommendations will measure novelty. Figure 10 has quality replicated from Figure 9, and novelty is the % of recommendations with fewer than 30 tags; i.e. those whose tags have been augmented with 70-100 pseudo-tags and 35-50% weighting with tags. Again the advantage from combined quality and novelty for Hybrid over Tag is clear. The points in each cluster,

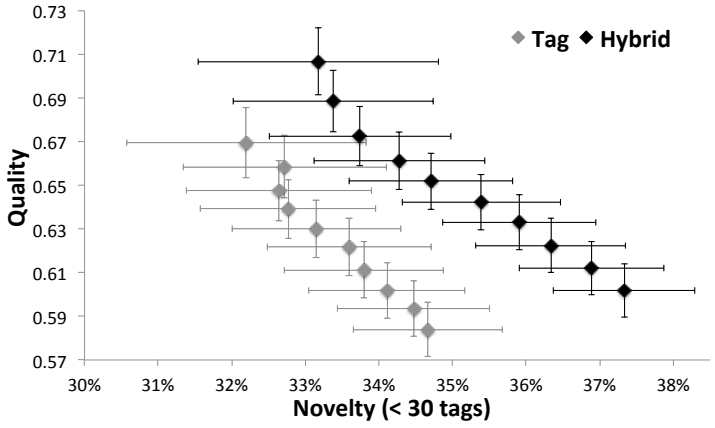

Fig. 10. Quality and Discovery from System-Centric Evaluation (30 tags)

showing quality and novelty with differing numbers of recommendations, are for $N = 1..10$ with $N = 1$ being the top point, with larger N strictly in order below. Hybrid gives a better level of discovery of tracks with relatively few tags although this tendency is not significant for $N \leq 5$.

What happens with a more demanding criterion for novelty than 30 tags? The quality-novelty scatter for discovery involving fewer tags is a little more overlapping on the novelty axis, as shown in Figure 11 for the discovery of tracks with fewer than 20 tags. Hybrid and Tag are now comparable for novelty, with a tendency for Hybrid to be better for $N > 5$.

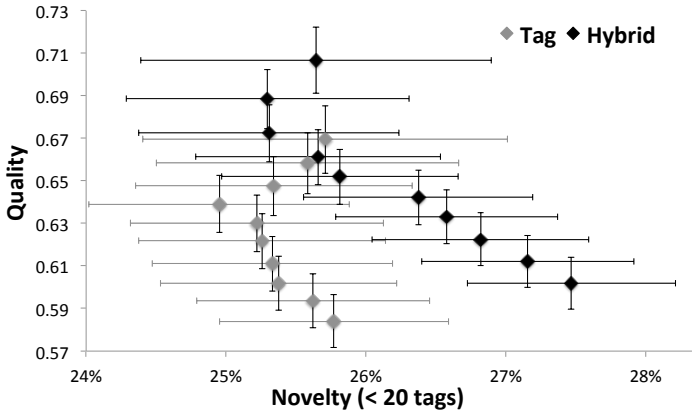

Fig. 11. Quality and Discovery from System-Centric Evaluation (20 tags)

In Figures 10 & 11, only the novelty values change for different levels of tagging, and the heights of the points on the quality scale are identical in both figures. Hybrid and Tag are clearly recommending different tracks because of the significant quality gains for Hybrid's use of pseudo-tags in these figures. With an overall frequency in the dataset of 54% for < 30 tags, and 42% for < 20 tags, the discovery rates in these figures indicate fair treatment of the 'long tail'.

7 Conclusions

Tags introduce a blur between the classical notion of a recommender being either content-based or collaborative filtering. In the strict sense tags are content, but because they are created collaboratively, recommendations made based on tags are influenced by other users, which classical content-based systems are not. As a result tag-based recommenders offer some of the advantages of collaborative filtering, but also suffer some of their disadvantages. Collaborative user tagging provides semantics including contextual, social and cultural information that allow tag-based recommenders to take advantage of this information when making recommendations. However, this also means that tag-based recommenders are

affected by under-tagging of new and niche tracks that are in the 'long tail' of music tracks. There is a reinforcement loop whereby tracks that are not listened to often, do not get tagged frequently, so have relatively few tags, and so do not get recommended. The combination of the 'long tail' of music tracks and the popularity bias of tagging means that there is also a 'long tail' of tagging of tracks. These are precisely the tracks that one wishes to include in a recommender that will introduce serendipity and novelty for users.

We have developed a Hybrid recommender that learns pseudo-tags for tracks with fewer tags so that the tagging 'long tail' is removed, and a tag-based recommender does not face the sparseness of user tagging. Pseudo-tags are related to audio since they are learned from tracks that are similar in the audio space, but they also capture semantics that users have given to neighbouring tracks. Further the weighting and selection of pseudo-tags allows only the most popular tags to be inherited from the musical neighbourhood. Finally the balancing of tags with pseudo-tags ensures that user-generated tags are used, and are most influential, whenever they are available.

The user trial and larger off-line evaluation demonstrate that Hybrid is effective in bridging the semantic gap between user tagging and audio. The semantic knowledge extracted from audio neighbourhoods is useful in improving the quality of Hybrid recommendations over those from the Tag recommender. These evaluations also explored the novelty of recommendations. Importantly the user trial results were based on responses about whether the track, or track and artist, were unknown, and there is a significant separation between the Hybrid cluster for 'novelty with quality' compared to Tag. The off-line evaluation gave consistent findings, but its tagging criterion for novelty is artificial, and for more sparsely tagged tracks, Hybrid's novelty advantage is less.

Our interest is in recommenders that offer serendipity whilst maintaining good recommendations. Hybrid indeed achieves this, without introducing a specialised serendipity recommender. Augmenting pseudo-tags has even increased recommendation quality. This approach of augmenting a weak representation with equivalent knowledge from neighbourhoods in a complete representation may be useful in related recommendation tasks; e.g. extracting pseudo-captions to improve image retrieval, learning pseudo-ratings for collaborative filtering. We have focused on pseudo-tags to improve recommendation but it could be interesting to understand inconsistencies between tags and pseudo-tags that indicate possible malicious tagging and shilling attacks.

References

1. Nanopoulos, A., Rafailidis, D., Symeonidis, P., Manolopoulos, Y.: MusicBox: Personalized music recommendation based on cubic analysis of social tags. Audio, Speech, and Language Processing, IEEE Transactions on **18**(2) (2010) 407–412
2. Barragáns-Martínez, A.B., Costa-Montenegro, E., Burguillo, J.C., Rey-López, M., Mikic-Fonte, F.A., Peleteiro, A.: A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. Information Sciences **180**(22) (2010) 4290–4311

3. Firan, C.S., Nejdl, W., Paiu, R.: The benefit of using tag-based profiles. In: Proc. Latin American Web Conference. (2007) 32–41
4. Celma, O., Cano, P.: From hits to niches?: Or how popular artists can bias music recommendation and discovery. In: Proc. 2nd Netflix-KDD Workshop, (2008) 1–8
5. Bertin-Mahieux, T., Ellis, D.P., Whitman, B., Lamere, P.: The million song dataset. In: Proc. 12th International Society for Music Information Retrieval Conference. (2011) 591–596
6. Halpin, H., Robu, V., Shepherd, H.: The complex dynamics of collaborative tagging. In: Proc. 16th International Conference on World Wide Web. (2007) 211–220
7. Bertin-Mahieux, T., Eck, D., Mandel, M.: Automatic tagging of audio: The state-of-the-art. In: Machine Audition: Principles, Algorithms and Systems. IGI Global (2010) 334–352.
8. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Semantic annotation and retrieval of music and sound effects. Audio, Speech, and Language Processing, IEEE Transactions on **16**(2) (2008) 467–476
9. Sordo, M., Laurier, C., Celma, O.: Annotating music collections: How content-based similarity helps to propagate labels. In: Proc. 8th International Conference on Music Information Retrieval (ISMIR). (2007).
10. Horsburgh, B., Craw, S., Massie, S.: Learning pseudo-tags to augment sparse tagging in hybrid music recommender systems. Artificial Intelligence **219** (2015) 25–39.
11. Levy, M., Sandler, M.: Music information retrieval using social tags and audio. IEEE Transactions on Multimedia **11**(3) (2009) 383–395.
12. Horsburgh, B., Craw, S., Massie, S., Boswell, R.: Finding the hidden gems: Recommending untagged music. In: Proc. 22nd International Joint Conference in Artificial Intelligence, AAAI Press (2011) 2256–2261.
13. Kaminskas, M., Bridge, D.: Measuring surprise in recommender systems. In: Proc. ACM RecSys Workshop on Recommender Systems Evaluation: Dimensions and Design, (2014).
14. Zhang, Y.C., Séaghdha, D.Ó., Quercia, D., Jambor, T.: Auralist: Introducing serendipity into music recommendation. In: Proc. 5th ACM International Conference on Web Search and Data Mining, (2012) 13–22.
15. Hornung, T., Ziegler, C.N., Franz, S., Przyjaciel-Zablocki, M., Schatzle, A., Lausen, G.: Evaluating hybrid music recommender systems. In: Proc. IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology. Volume 1., IEEE (2013) 57–64.
16. Horsburgh, B.: Integrating content and semantic representations for music recommendation. PhD thesis, Robert Gordon University (2013).
17. Celma, O.: Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space. Springer (2010).
18. Horsburgh, B., Craw, S., Massie, S.: Music-inspired texture representation. In: Proc. 26th AAAI Conference on Artificial Intelligence, AAAI Press (2012) 52–58.
19. Mermelstein, P.: Distance measures for speech recognition, psychological and instrumental. Pattern Recognition and Artificial Intelligence **116** (1976) 91–103.
20. Stevens, S., Volkmann, J., Newman, E.: A scale for the measurement of the psychological magnitude pitch. Journal of the Acoustical Society of America **8** (1937) 185–190.
21. Craw, S., Horsburgh, B., Massie, S.: Music recommenders: User evaluation without real users? In: Proc. 24th International Joint Conference in Artificial Intelligence, AAAI Press (2015) 1749–1755