# Finding the Hidden Gems: Recommending Untagged Music

**Ben Horsburgh** and **Susan Craw** and **Stewart Massie** and **Robin Boswell**
IDEAS Research Institute
Robert Gordon University, Aberdeen, UK
{b.horsburgh, s.craw, s.massie, r.boswell}@rgu.ac.uk

## Abstract

We have developed a novel hybrid representation for Music Information Retrieval. Our representation is built by incorporating audio content into the tag space in a tag-track matrix, and then learning hybrid concepts using latent semantic analysis. We apply this representation to the task of music recommendation, using similarity-based retrieval from a query music track. We also develop a new approach to evaluating music recommender systems, which is based upon the relationship of users liking tracks. We are interested in measuring the recommendation quality, and the rate at which cold-start tracks are recommended. Our hybrid representation is able to outperform a tag-only representation, in terms of both recommendation quality and the rate that cold-start tracks are included as recommendations.

## 1 Introduction

Over recent years a vast number of online music services have appeared and grown. Unlike high-street music stores, listeners now have instant access to all recorded music. Traditional methods of finding music do not scale to large online music collections, and so an area of Music Information Retrieval which has been given much attention recently is recommendation. Every major online music service now has a music recommender system available to users, helping them navigate music collections. Such systems have been keenly adopted by users, to the extent that artists can now become massively popular solely based on a viral online interest.

Core to current state-of-the-art music recommender systems is social meta-data. This is typically in the form of free-text tags which describe any musical entity, such as an artist or track. Recommendations are then made based upon the similarity of social tags which are applied to each track.

Tag-based recommender systems have proven to be very powerful, but there are scenarios where they do not perform well. One such scenario is the well-known cold-start problem, where tracks in a collection do not have any tags applied to them. In these situations the untagged tracks will never be recommended. Often this means that the track will also never be tagged, since no one has been recommended the track, creating a Catch-22 style scenario. When the iTunes store opened in 2003, $200,000$ tracks were available; in 2010 $13,000,000$ tracks[1] were available. This illustrates the steep increase in volume of tracks available online, all of which must be tagged to be included in a recommender system. In this paper we show how the recommendation discovery of cold-start tracks can be increased by incorporating audio content-based data into a recommenders' representation.

Measuring the quality of a recommender system is typically achieved using genre, mood or artist classification accuracy. While these measures may be part of a good recommendation, they do not directly measure quality. We present a new evaluation measure which is not based on classification accuracy, but instead measures the level of positive association between two tracks based on user listening data.

The paper is structured as follows. In Section 2 we discuss recent related work on recommendation and evaluation. Our method for including content-based data in a tracks representation is presented in Section 3. Our evaluation measure is described in Section 4. Section 5 describes the experiments which we run to evaluate our representation, and presents our results. In Section 6 we draw some conclusions.

## 2 Related Work

Knowledge which can be gathered online has proven to be invaluable to music recommendation [Plaza and Baccigalupo, 2009]. Central to many state-of-the-art recommender systems are social tags [Nanopoulos *et al.*, 2010]. While these systems perform very well when data is available, tracks which do not have data available are left out. Reducing this cold-start problem, and enabling users to discover cold-start tracks, has been the focus of much recent work.

In general, there are two approaches to increasing discovery in tag-based recommenders. The first is to use audio similarity to propagate tags throughout a collection, therefore allowing every track to have a tag representation [Bertin-Mahieux *et al.*, 2008]. The limitation of this approach is that content similarity does not directly correlate to tag similarity, and therefore many erroneous tags are propagated. The second approach is to directly incorporate a content-based representation into the tag representation, therefore always presenting data on which to compute a similarity. Bu *et al.*

[1] http://www.apple.com/itunes/features/

[2010] present a hypergraph model which combines content with tags. This is an intuitive way to combine many different types of representations, but no new concepts are learned from this combination. Levy and Sandler [2009] construct a representation matrix which combines tags and clustered content representations, which they name *muswords*. They then employ probabilistic latent semantic analysis to learn hybrid concepts which generalize both tags and content. All notion of content-based similarity however is lost in their method, due to the *muswords* being treated as a bag of words.

## 3 Music Representation

We develop a new representation which combines tags and content, designed to increase the rate of discovering untagged tracks in a recommender system. The collection we are using consists of $3495$ tracks which span $951$ artists and $16$ genres, as defined by Gracenote[2]. For each track we collect tags and extract standard content-based representations. These representations are then used to create a new hybrid representation, designed to reduce the cold-start recommendation problem by increasing cold-start discovery.

### 3.1 Basic Representations

Our tag representation is built using data downloaded from Last.fm, using the API they provide[3]. We store the top $20$ tags for each track, along with their frequency when applied to the track. In total $3537$ distinct tags are collected, and on average a track has $18$ tags available. Last.fm normalizes each tag frequency relative to the most frequent tag, stored as a percentage value; the most frequent tag always has a frequency of $100$, and a tag occurring half as frequently is given a frequency of $50$. These frequency values are represented as a $3537$ wide vector. Each vector is extremely sparse, with an average of $0.51\%$ non-zeros.

To extract content features from audio we process each track in time slices of $750ms$. Beyond providing computational efficiency, this time interval was chosen based on evidence which suggests humans can distinguish genre upon hearing anything more than $475ms$ [Gjerdingen and Perrott, 2008]. This amount of time is not sufficient for the listener to distinguish any rhythmic features, and therefore we only include spatial features. For each time slice we first compute a Hamming windowed Fast-Fourier-Transform (FFT), and then extract our representations. The average over all time slices is then used as the final representation.

The inspiration behind our content features is to describe key spatial aspects of music: pitch, texture and harmony.

- A chroma representation describes the intensities of musical notes, and captures pitch [Kim and Narayanan, 2008]. To represent chroma we extract the intensity of each musical note found within the frequency domain of a time slice. These range from C0 at $8.176Hz$ to B7 at $3951Hz$. This provides a measure of the intensity of each musical note within an 8 octave range, described as a 96 bin vector.
- Mel Frequency Cepstral Coefficients (MFCC), originally used in speech recognition, describe the timbre of sound. MFCCs have proven to be very useful for genre, artist and mood classification, and so we are interested in how they perform for recommendation. We implement MFCCs as described by Sigurdsson *et al.* [2006], using $20$ equally spaced triangular Mel filters, and retaining the first 12 MFCC's in our representation.
- To represent harmony we include a discretized representation of each FFT, which we name the Discretized-Frequency-Domain (DFD). When two musical notes are played simultaneously many harmonics of each note interact to create a new harmonic sound, heard as a result of the complex distribution of frequencies. DFD aims to capture this distribution by discretizing the frequency domain of each time slice into 200 equally spaced buckets. The sum of all intensities within each bucket is used.

### 3.2 Tag-Concept Representation

Each of the basic representations capture specific facets of each track. However, they do not contain any knowledge of how those facets relate to the collection of tracks. This is well known throughout text retrieval, where synonymy is a problem. Two terms may be synonymous, but a direct comparison will always treat them as being different.

Latent Semantic Analysis (LSA) is a technique which generalizes the terms used into a conceptual representation, thus capturing synonyms. This is achieved by first constructing a term-document matrix, and then applying singular value decomposition (SVD). SVD decomposes a matrix, $M$, such that $M = U\Sigma V^T$; matrix $U$ consists of tags and track-concepts, and matrix $V^T$ consists of tag-concepts and tracks.

The tag-concept representation we use is $V^T$, obtained by decomposing the following matrix:

$$M = \begin{array}{c c} & \begin{array}{cccc} T_1 & T_2 & \phantom{\cdots} & T_N \end{array} \\ \begin{array}{c} t_1 \\ t_2 \\ \\ t_n \end{array} & \left( \begin{array}{cccc} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & f_{12} & \cdots & f_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nN} \end{array} \right) \end{array}$$

where $T_i$ denotes a track, and $t_j$ denotes a tag. $f_{ij}$ denotes the frequency of tag $t_i$ when applied to track $T_j$ as defined by Last.fm.

### 3.3 Hybrid-Concept Representation

The tag-concept representation described suffers from the cold-start problem; no generalized concepts can be defined for untagged tracks, and therefore these tracks will not be recommended from tagged tracks. To reduce this problem we create a hybrid-concept representation, where content is included.

The aim for our hybrid representation is to include content in the matrix $M$ in such a way that SVD is able to generalize tag and content features into meaningful concepts. This generalization is extremely important when the cold-start problem exists. Suppose the query track is well tagged, and a frequent tag is happy, generalization will make it possible to

[2] http://www.gracenote.com/

[3] http://www.last.fm/api

recommend a non-tagged track which shares the same concept of happy, due to its content representation.

To include content in $M$ we extend the number of rows by the number of bins in the content representation being used. This extended matrix is constructed as follows:

$$M = \begin{array}{c|cccc} & T_1 & T_2 & & T_N \\ t_1 & f_{11} & f_{12} & \dots & f_{1N} \\ t_2 & f_{21} & f_{22} & \dots & f_{2N} \\ & \vdots & \vdots & \vdots & \vdots \\ t_n & f_{n1} & f_{n2} & \dots & f_{nN} \\ c_1 & i_{11} & i_{12} & \dots & i_{1N} \\ c_2 & i_{21} & i_{22} & \dots & i_{2N} \\ & \vdots & \vdots & \vdots & \vdots \\ c_m & i_{m1} & i_{m2} & \dots & i_{mN} \end{array}$$

where $c_i$ denotes a bin in our content representation, and $i_{ij}$ denotes the intensity of bin $c_i$ for track $T_j$. The values used in the matrix are the normalized intensity values of a content-based representation. These are normalized in the same way as with Last.fm tags

SVD generalizes well when sparse data is used. For this reason we only include the 20 most intense bins within each tracks' content representation. Our initial experiments did not introduce this sparsity, and obtained a significantly lower evaluation score. While intensity is different from frequency in tags, high valued bins are still the most reflective of a track. For example, a tag-based concept capturing "lots of Bass" may be present in a DFD based form, where specific bins for "bass" may be high, and bins for other frequency ranges may be low.

## 4 Recommendation Quality

Many evaluations throughout music retrieval use the classification accuracy of artists, genre or mood [Flexer *et al.*, 2010]. It would be much more desirable to be able to conduct a full user evaluation, as with [Firan *et al.*, 2007], but such evaluations however are impractical for iterative evaluations. Ellis *et al.* [2002] propose several methods of replicating a real-world user evaluation. Their first approach is based on using complex network-analysis to define a similarity measure. One drawback with this shortest-path based approach is that popular tracks are linked by one edge to many tracks, making it difficult to know what is truly a good recommendation. The second method they propose uses a peer-to-peer cultural similarity measure, based on listening habits. This measure defines a good recommendation as one which is similarly popular to the query, and occurs in a large percentage of users profiles with the query. Again, the bias introduced by popularity makes this measure unsuitable for evaluation when discovery is important.

In this section we describe our new evaluation measure for recommendation quality, and then provide a discussion of how this measure behaves.

### 4.1 Measuring Quality

We propose a new evaluation strategy which attempts to replicate a real-world user evaluation, by using available data from internet users. The inspiration for our measure of recommendation quality comes from conversations between two people:

P1: I've been listening to this awesome band called Klaxons.

P2: Me too. Do you like Neon Plastix?

P1: Never heard of them!

P2: You should get their CD, you'd like them.

The key point from this type of conversation is that Person P2 has made an association between Klaxons and Neon Plastix. This association is based upon P2 having listened to both artists, and having liked both artists. The concepts of listening to and liking both artists forms the basis for our measure.

We define a high quality recommendation as one for which Person 2 makes an association, and a low quality recommendation as one which Person 2 does not. We then extrapolate this definition to take account of $n$ peoples' opinions. The proportion of people who agree that there is an association between the two tracks quantifies the strength of association, shown in Eq. 1.

$$\text{association}(t_i, t_j) = \frac{\text{likes}(t_i, t_j)}{\text{listeners}(t_i, t_j)} \qquad (1)$$

where $t_i$ and $t_j$ are tracks, listeners$(t_i, t_j)$ is the number of people who have listened to both $t_i$ and $t_j$, and likes$(t_i, t_j)$ is the number of listeners who have liked both $t_i$ and $t_j$. This definition is similar to many offline user evaluations, where a set of people are asked to rate pairs of tracks. The average of all ratings, or associations, quantifies the strength of the recommendation. We are therefore simulating an offline user experiment where user ratings are binary.

To apply this type of evaluation one must first have users available. Instead of offline users we use online users, the data for which is available using the Last.fm API. We collect data for over 175000 users, over a period of 2 months. For each user the tracks which they have clicked the "thumbs up" button on Last.fm are recorded, providing data on the tracks they like. On average a single user will have liked 5.4 tracks in our collection. Further information collected from Last.fm is the number of distinct listeners of each track.

To determine the level of association in Eq. (1) between two tracks we must first know two important facts; the number of users in our collection who have listened to a given pair of tracks, and the number of users who have liked both tracks. From our collected Last.fm data we know the number of users who have liked both tracks in any given pair. Unfortunately, data on who has listened to both tracks is unavailable from their API. However, the number of users who have listened to each track is available, allowing an estimate of listeners to both.

To estimate the number of users in our collection who have listened to a pair of tracks, we first assume that all tracks are listened to independently. We then estimate the number of users who have listen to each possible pairing of tracks, scaled to our collection size, using Eq. 2.

$$\text{listeners}(t_i, t_j) = \frac{C(t_i)}{|\text{Last.fm}|} \cdot \frac{C(t_j)}{|\text{Last.fm}|} \, |\text{Collection}| \qquad (2)$$

where $C(t_i)$ denotes the count of users who have listened to track $t_i$, $|\text{Last.fm}|$ denotes the number of users of Last.fm, and $|\text{Collection}|$ denotes the number of users in our collection. Using the estimate in Eq. (2), we are able to calculate the level of association between tracks as described in Eq. (1).

The measure of association defines quality on a scale of $0$ to $1$. A value of $0$ occurs when there is no evidence of users liking a pair of tracks. A value of $1$ occurs when the number of users who like a pair of tracks is equal to the number who have listened to the pair. It is possible to obtain a score of greater than $1$, since we are using an estimate to determine the number of listeners. As the actual level of association between a pair of tracks increases, so does the amount we under-estimate the number of listeners, due to our assumption of independence. To handle this we set a limit on the measure to $1$. When an under-estimate occurs, such that a value of more than $1$ is obtained, it is clear that the two tracks display a very high level of association, and therefore a score of $1$ reflects this.

### 4.2 Discussion of Measure

The evaluation measure we have described is based on associations between users having listened to and liked tracks. This approach may be compared to collaborative filtering, but a key difference exists. Collaborative filtering is an approach for finding similar users, and then recommending tracks which may be of interest. Our evaluation measure does not find similar users, but quantifies the global level of agreement between sets of users. Collaborative Filtering has been developed to make recommendations: our method has been developed to evaluate recommendations.

The mean level of association in our collection is $0.064$. An association score of $0$ is achieved by $87\%$ of all possible recommendation pairs, and a score between $0$ and $1$ is achieved by $10\%$ of all pairs. In our collection $3\%$ of possible query-recommendation pairs obtain an association score of greater than $1$, which we modify to be $1$. The distribution of association scores for all possible pairs in our collection is shown in Figure 1. We have excluded values for associations of $0$ and $1$ so that the meaningful part of the curve can be easily observed.

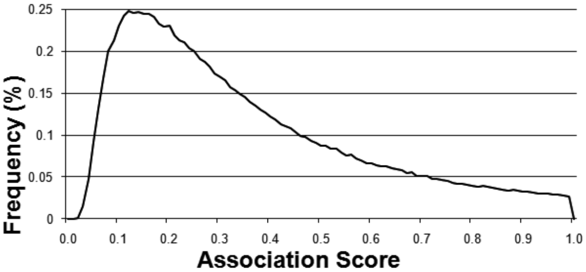

Figure 1: Distribution of association values

The distribution observed centers around the score $0.12$, which occurs for $0.25\%$ of all possible query-recommendation pairs. The tail of the curve decays less steeply than expected, due to our estimate of listeners. If an estimate was not used we would expect the tail to decay much faster, and reach a frequency of approximately $0$ much earlier, reflecting the level of true agreement. $13\%$ of all possible pairs in our collection obtain a score of greater than $0$, with $10\%$ lying in the distribution shown. This shows our measure is suitably discriminant of high and low quality recommendations.

## 5 Experiments

We evaluate several recommender systems, each using a different representation for tracks. The representations evaluated are as follows:

- Tag is the LSA representation from the tag-track matrix (as described in Section 3.3)
- DFD, MFCC and Chroma use LSA representations from the corresponding matrix where the tags are replaced with the appropriate content-based representation
- Hybrid is the LSA representation from the extended matrix for tags with content features
- Random does not have a representation, but instead randomly selects tracks to recommend

Euclidean distance is used to retrieve and rank retrieved tracks for all representations. This is a metric which is commonly used in content-based retrieval. It is standard that a cosine similarity measure is used for text, because it defines similarity based on terms co-occurring with the same relative frequency. For tags however, the value of frequency values used is important. Euclidean distance is able to capture differences in these values, and is therefore best suited to each of our representations.

We follow a standard experimental design for each representation. We hold-out $30\%$ of tracks as test queries, and the remaining $70\%$ make the set of possible recommendations. For each track in our test set we obtain the top $20$ recommendations. We evaluate each recommendation subset in this top $20$, that is, we calculate the association score for only $1$ recommendation, $2$ recommendations, and so on. For each recommender system we repeat the experiment procedure four times. The results presented are an average of the results obtained over all runs. All error bars shown are at a $95\%$ confidence interval.

### 5.1 Recommender Results

Initially we evaluate each content-based representation to understand which performs best at the task of music recommendation. The expectation was that MFCCs would perform best, due to their strength in other MIR tasks. Figure 2 shows the results for recommender systems built using the DFD, MFCC and Chroma representations. The x-axis is the number of recommendations that were made, and the y-axis is the association score for this number of recommendations. For a recommendation set of up to $8$ recommendations, the DFD representation outperforms MFCC's. The first recommendations presented are far more important than those further down the list, and therefore we use the DFD representation as the content component in our hybrid representation.

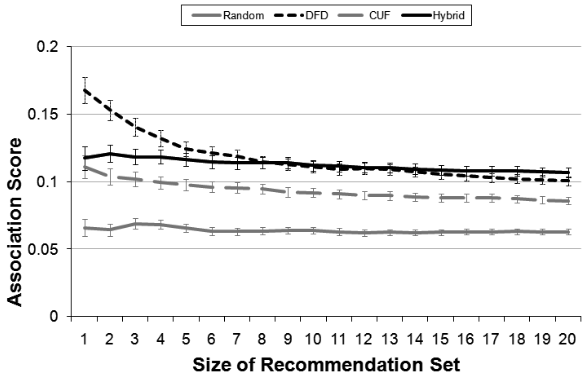

Figure 2: Content-based representations

The second experiment (Figure 3) is intended to understand the performance of the Tag, DFD and Hybrid representations. We construct our Hybrid representation using DFD as the content component, since it outperforms MFCCs and Chroma. It is interesting to note the difference between the Tag and Hybrid association score, and the DFD association score. When DFD only is used, the quality of recommendations made is extremely low. When DFD is integrated into our Hybrid representation however, the quality of recommendations is competitive with the Tag recommender. The reason for this is that the process of SVD is taking advantage of the discriminant power of tags far more than it does for DFD.

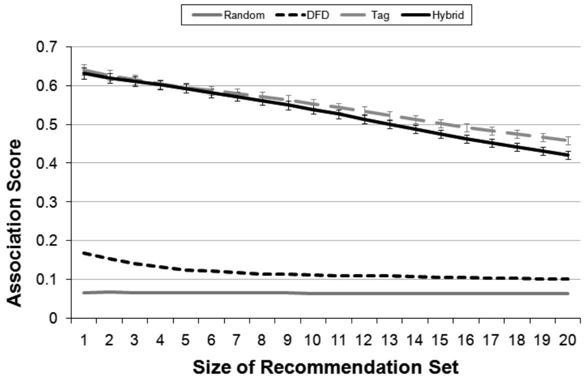

Figure 3: Recommendation quality

Our final experiment shows how each of the representations performs on our collection with injected cold-start. Our collection has only 2% natural cold-start within it. To simulate the cold-start problem we inject the problem into a randomly selected 25% of our collection by removing their tags, allowing us to clearly observe the problems effects.

When the cold-start problem has been injected into our collection, our Hybrid representation outperforms the Tag representation at all recommendation list sizes (Figure 4). The SVD of the original track-representation matrix generalizes the representation, and takes advantage of discriminant concepts. In the case of our Tag representation, 25% of our data is identical; it has no tags. For these data there is no discriminant concepts which can be used, and therefore no meaningful similarity measure can be defined.

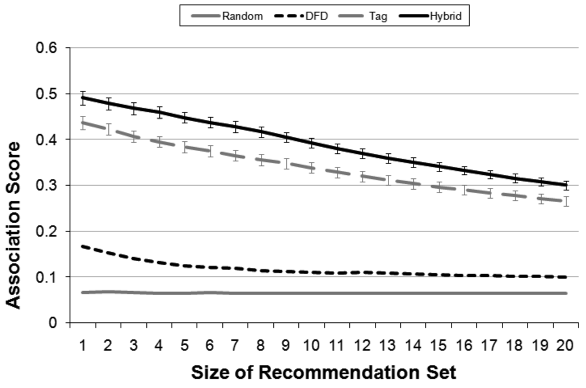

Figure 4: Recommendation quality in cold-start scenario

It should be noted that the injection of cold-start tracks has nevertheless lowered the performance of our Hybrid representation. This is directly caused by the untagged tracks, since tags naturally outperform content representations. In this scenario however, our Hybrid representation still has a representation for the untagged tracks. Further, the representation available for these tracks is always available for the whole collection. This in turn means that SVD is again able to take advantage of discriminant concepts within the entire representation. It is for this reason that our Hybrid representation is able to outperform the Tag representation, and provide a recommender system which can produce higher quality recommendations when the cold-start problem exists.

## 5.2 Discovery Results

A complementary measure we use is the discovery rate of cold-start items. This is the ratio of untagged items which are recommended, when the query track is tagged.

Figure 5 shows the discovery rate for each representation. The random and DFD based recommenders achieve a consistent discovery rate of approximately 25%. While this at first appears to be good, these results must be interpretted with Figure 4 in mind. The random recommender obtains a high discovery rate, but also obtains a very low association score. DFD obtains the same discovery rate since it is unbiased towards tag representation. The association score is again low however, meaning that the tracks discovered are not strong recommendations.

The Tag representation has a discovery rate of approximately 0 for the first 4 recommendations. This is expected, since the Tag representation will naturally rank tagged tracks as most similar. As the size of the recommendation set increases the tag representation gradually makes more discoveries. The reason for this is that for some query tracks there are no more recommendation tracks which share tag-concepts, and therefore everything obtains a similarity of 0.

The Hybrid representation has a steeper discovery rate than Tags, starting at a rate of 0.03 for 1 recommendation. Un-

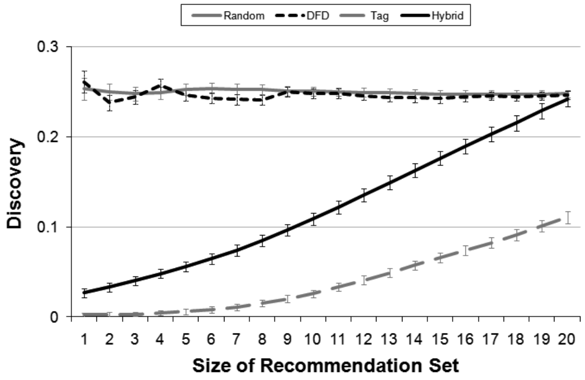

Figure 5: Recommendation of untagged Tracks

like with the random and DFD recommenders, DFD achieves this increase in discovery while maintaining a high association score. This shows that the tracks being discovered are relevant, and good recommendations.

The increased discovery rate of the Hybrid representation is due to the influence of the DFD component. Where no tags exist, a track still has a concept representation, and is therefore able to obtain meaningful recommendations, avoiding the 0 similarity problem found in the tag recommender.

## 6 Conclusions

We have developed a novel way of incorporating content-based representations into a tag-based recommender system. This has been achieved by extending the tag-track matrix to include content, and then learning hybrid concepts using LSA. Through generalization, the concepts learned make associations between tags and content in such a way that recommendations can still be made when tags are absent.

A new evaluation measure simulating real-world user evaluations of recommender systems has also been developed. This evaluation defines associations between pairs of tracks based upon users having listened to both tracks, and having liked both tracks. In developing this measure, we have shown how recommender systems can be evaluated in terms which are directly relevant to recommendation, not classification.

Our results show that by including content an increased rate of discovering cold-start items can be achieved. Further, this discovery of cold-start improves the overall performance of the recommender system, and the Hybrid representation is able to outperform the Tag representation when the cold-start problem exists. When cold-start tracks do not exist, our recommender is competitive with a tag-only recommender.

The merging of two representations in the way we have developed has promising directions. We have shown how content can be of benefit to tag-based recommenders. Using our approach it may even be possible to include many different sources of knowledge, such as play lists and mined web data, creating a truly all-encompassing representation. Humans make recommendations based on a wealth of information, the next challenge is to expand representations to include this knowledge.

## References

[Bertin-Mahieux *et al.*, 2008] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2):115–135, 2008.

[Bu *et al.*, 2010] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X He. Music recommendation by unified hypergraph: combining social media information and music content. In *Proc. International Conference on Multimedia*, pages 391–400. ACM, 2010.

[Ellis *et al.*, 2002] D. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proc. International Society for Music Information Retrieval*, pages 170–177. Ircam, 2002.

[Firan *et al.*, 2007] C.S. Firan, W. Nejdl, and R. Paiu. The benefit of using tag-based profiles. In *LA-WEB 2007. Latin American*, pages 32 –41, 2007.

[Flexer *et al.*, 2010] A. Flexer, M. Gasser, and D. Schnitzer. Limitations of interactive music recommendation based on audio content. In *Proc. 5th Audio Mostly Conference: A Conference on Interaction with Sound*, pages 1–7. ACM, 2010.

[Gjerdingen and Perrott, 2008] R.O. Gjerdingen and D. Perrott. Scanning the dial: The rapid recognition of music genres. *Journal of New Music Research*, 37(2):93–100, 2008.

[Kim and Narayanan, 2008] S. Kim and S. Narayanan. Dynamic chroma feature vectors with applications to cover song identification. *In Proc. IEEE 10th Workshop on Multimedia Signal Processing*, 984–987, 2008.

[Lee *et al.*, 2010] S.K. Lee, Y.H. Cho, and S.H. Kim. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. *Information Sciences*, 180(11):2142–2155, 2010.

[Levy and Sandler, 2009] M. Levy and M. Sandler. Music information retrieval using social tags and audio. *IEEE Transactions on Multimedia*, 11(3):383–395, 2009.

[Nanopoulos *et al.*, 2010] A. Nanopoulos, D. Rafailidis, P. Symeonidis, and Y. Manolopoulos. Musicbox: Personalized music recommendation based on cubic analysis of social tags. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):407 –412, 2010.

[Plaza and Baccigalupo, 2009] E. Plaza and C. Baccigalupo. Principle and praxis in the experience web: A case study in social music. In *The Proc. 8th International Conference on Case-Based Reasoning*, pages 55–63. Springer, 2009.

[Sigurdsson *et al.*, 2006] S. Sigurdsson, K.B Petersen, and T. Lehn-Schiler. Mel frequency cepstral coefficients: An evaluation of robustness of MP3 encoded music. In *Proc. International Society for Music Information Retrieval*. University of Victoria, 2006.