If You Like Radiohead, You Might Like This Article

Òscar Celma and Paul Lamere

■ *With the recent dramatic transformations in the world of digital music, a music listener is now just a couple of clicks away from being able to listen to nearly any song that has ever been recorded. With so much music readily available, tools that help a user find new, interesting music that matches his or her taste become increasingly important. In this article we explore one such tool: music recommendation. We describe common music recommendation use cases such as finding new artists, finding others with similar listening tastes, and generating interesting music playlists. We describe the various approaches currently being explored by practitioners to satisfy these use cases. Finally, we show how results of three different music recommendation technologies compare when applied to the task of finding similar artists to a seed artist.*

In the last 15 years we have seen a major transformation in the world of music. Musicians use inexpensive personal computers instead of expensive recording studios to record, mix, and engineer music. Musicians use the Internet to distribute their music for free instead of spending large amounts of money creating CDs, hiring trucks, and shipping them to hundreds of record stores. As the cost to create and distribute recorded music has dropped, the amount of available music has grown dramatically. Twenty years ago a typical record store would have music by fewer than ten thousand artists, while today online music stores have music catalogs by nearly a million artists. With so much more music available, listeners are increasingly relying on tools such as automatic music recommenders to help them find, organize, and experience their music.

While the amount of new music has grown, some of the traditional ways of finding music have diminished. Thirty years ago, the local radio DJ was a music tastemaker, finding new and interesting music for the local radio audience. Now radio shows are programmed by large corporations that create playlists drawn from a limited pool of tracks. Similarly, record stores have been replaced by big box retailers that have ever-shrinking music departments. In the past, you could always ask the owner of the record store for music recommendations. You would learn what was new, what was good, and what was selling. Now, however, you can no longer expect that the teenager behind the cash register will be an expert in new music or even be someone who listens to music at all. To fill the gap left by the departing radio DJs and record store clerks, listeners are turning to machines to guide them to new music.

As we rely more and more on automatic music recommendation it is important for us to understand what makes a good music recommender and how a recommender can affect the world of music. With this knowledge we can build systems that offer novel, relevant, and interesting music recommendations drawn from the entire world of available music.

In this article we describe some of the primary ways music

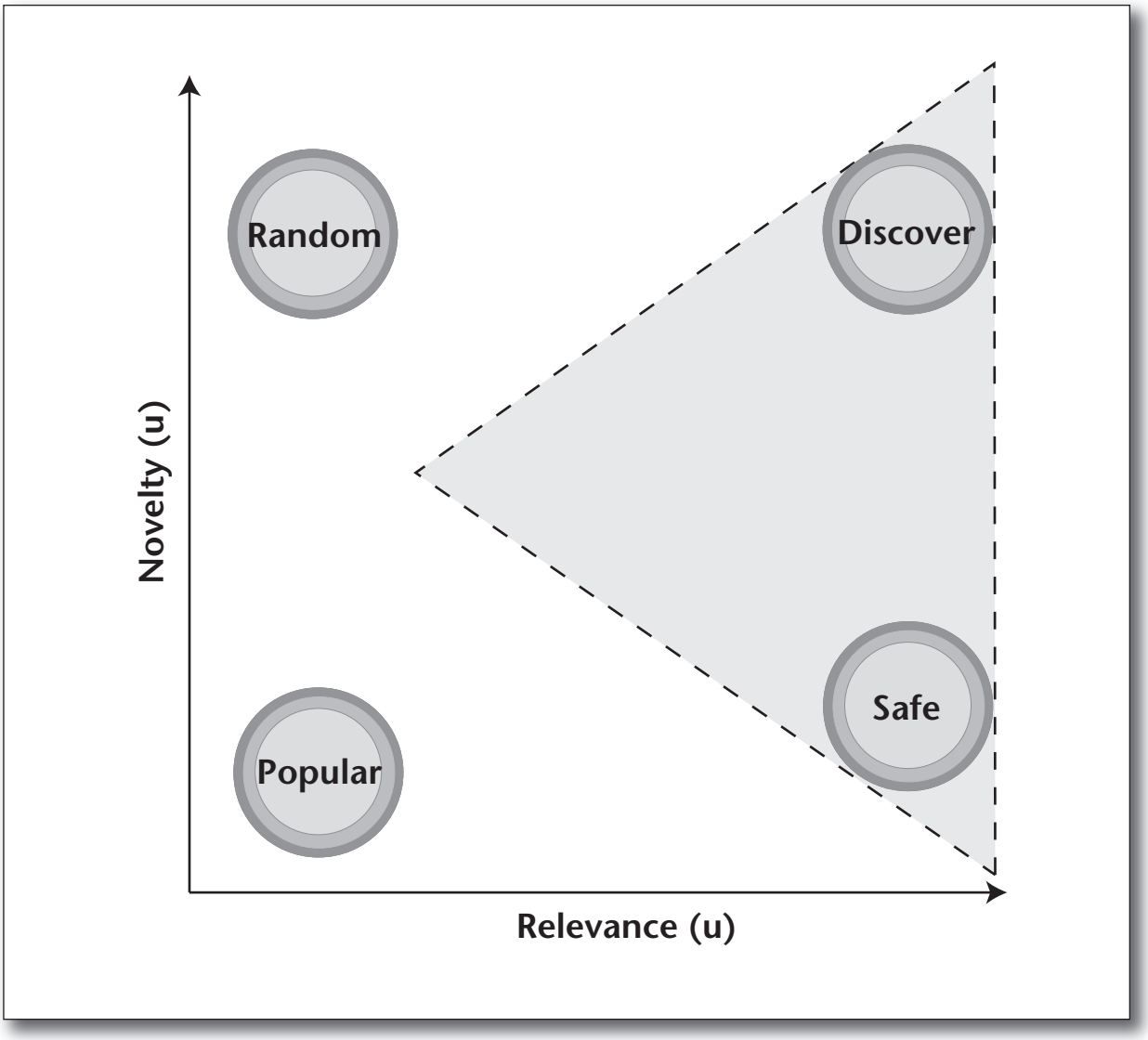

Figure 1. Trade-Off Between Novelty and Relevance.

recommendation technologies are used in the music domain. We describe the advantages and limitations of current state-of-the-art approaches: usage based (UB), which analyzes listener usage patterns; social based (SB), which mines web content; content based (CB), which derives item similarity directly from audio; and hybrid methods that combine these approaches. Finally, we present examples of UB, SB, and CB algorithms for generating personalized artist recommendations.

Recommending the Unknown

One of the challenges of building a music recommendation system is balancing recommendations between familiar and novel items. It has been largely acknowledged that item popularity can decrease user satisfaction by providing obvious recommendations (McNee, Riedl, and Konstan 2006). A user should be familiar with some of the recommended items, to improve confidence and trust in the system. Still, some items should be unknown to the user. The system should give an explanation of why those items were recommended. However, there is a trade-off between novelty and user's relevance. The more novel, unknown items a recommender presents to a user, the less relevant they can be perceived by him or her.

Figure 1 presents the trade-off between novelty and relevance. The gray triangle represents the area on which a recommender should focus to provide relevant items to a user *u*. On the one hand, *safe* recommendations (bottom right) appear when the system recommends familiar and relevant items to *u*. On the other hand, the *discovery* process (top

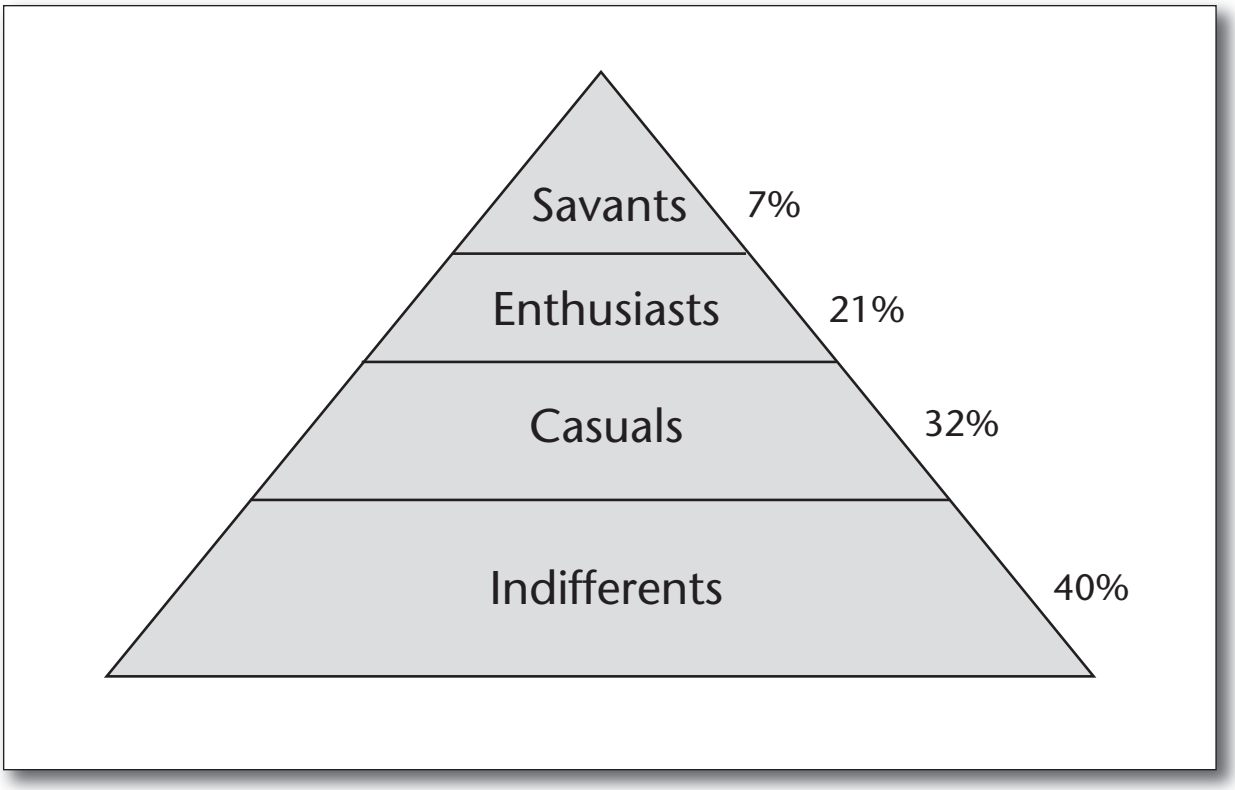

Figure 2. The Four Types of Music Listeners: Savants, Enthusiasts, Casuals, and Indifferents.

Each type of listener needs different types of recommendations.

right) starts when the system predicts unknown items that could fit in his or her profile. The provided recommendations should conform to the current user's intentions; sometimes a user is expecting familiar recommendations (safe state), while in other cases the user is seeking to actively discover new items. Outside the gray area, a recommender can either always play well-known, popular music (but this might not be relevant to a user), or just random music, where it might be novel for the user, but not related to his or her music taste at all.

Stereotyping the Listeners

The *Phoenix 2* UK Project summarizes the four degrees of interest in music, or type of listeners (Jennings 2007). This study is based on the analysis of thousands of subjects, with an age group ranging from 16 through 45. The classification, depicted in figure 2, identifies four groups of listeners: savants (7 percent), enthusiasts (21 percent), casuals (32 percent), and indifferents (40 percent). A music recommender should detect what type of listener a user is, as well as his or her current intentions, for example, whether the user is looking for safe recommendations or is in a more discovery-oriented mode. Depending on these factors the recommendations may vary a lot. Some aspects that can help to categorize a user are activity in the system: is the user listening to music all the time?, or is the user listening to trending artists before anyone else?, or is the user searching for music along the whole long tail of popularity (that is, listening not only to popular music, but also to unknown artists)?

Use Cases

Most current work in music recommendation focuses on three use cases: artist recommendation, neighbor recommendation, and playlist generation.

Artist Recommendation

Artist recommendation is based on the classic user-item matching, in which artists are recommended to a user according to his or her profile. Artist recommendation can occur in a number of contexts such as assisting a user in planning a listening session, helping a listener sift through the latest batch of new artists, connecting a music fan with local concerts by artists that the fan might find interesting, or even helping a music fan rediscover music in his or her own personal collection.

Neighbor Recommendation

The goal of neighbor recommendation is to find users with similar music taste. Establishing a listener's musical neighborhood can help the user connect to other like-minded users allowing the user to discover new relevant music through his or her neighbors. Finding music listeners with similar music taste can help establish connections between users in social networks, promoting tight communities of people that share similar interests.

Playlist Generation

The goal of playlist generation is to help music listeners organize their listening by building ordered lists of songs. Playlists are often dependent on context. A user's playlist for exercise may contain only upbeat tracks of very familiar and highly rated songs, whereas a playlist for relaxing may contain only mellow songs by a mix of familiar and unknown artists. In some contexts, such as at a nightclub, the order of the songs in a playlist is important, while in other contexts a playlist focuses more on a desired emotional state, or acts as a background to an activity, for example, while working, while reading, while jogging, and so on (Cunningham, Bainbridge, and Falconer 2006).

Shuffling a list of songs is perhaps the simplest way to generate playlists from small music collections. A study by Leong, Vetere, and Howard (2005) argues that shuffle provides opportunities for unexpected rediscoveries and can, in some cases, reconnect users to songs with old memories. However, the serendipity of the shuffle play can be a disadvantage when songs are selected from a large pool. Extreme changes in music style can be particularly unpleasant to a listener.

Personalized playlists is another way to propose music to a user. Music is selected based upon the listener's preferences, within a particular context. The user can provide feedback (for example, *skip this song*, *ban this song*, *more like this*, and so on) and the system adapts to this feedback by steering the playlist toward songs similar to those preferred by the user and away from songs that are not preferred.

Group playlisting generates a playlist for shared listening that matches the taste of a group of users in a particular context such as at a party or club (Baccigalupo 2009).

Music Recommendation Approaches

There are three different ways to derive similarity among items (or users) and, based on that, to recommend music to the users.

Usage-Based (UB) Recommender

Music similarity can be derived from the users' listening habits. The main approach is collaborative filtering (CF), which analyzes the usage patterns. CF methods work by building a matrix M^{UB}, with m users and n items, that contains its interaction (for example, plays, ratings, page views, and others).

Early research on CF applied to the music domain was based on explicit feedback, using the ratings (for example, 1 to 5 stars) about songs or artists. *Ringo* (Shardanand 1994) is the first music recommender based on this approach. The author applies a user-based CF approach, using Pearson normalized correlation as similarity. The recommendations are computed as the mean of the ratings done by the similar users of the active user.

Nowadays though, tracking implicit feedback (for example, user listening habits) has become the most common way to gather data. The interaction between the users and the items is usually described by the songs a user listens to, or the total playcounts per artist. Note however, systems that track implicit feedback are less likely to be able to capture negative feedback.

Social-Based (SB) Recommender

Another approach to computing similarity among items is through web mining techniques, or exploiting social tagging information. Web mining techniques aim at discovering interesting information from the analysis and usage of web content. Similarity among artists is computed based on, for instance, cooccurrence analysis in web pages, songs played in the same session log, or the text analysis of album reviews. Social tagging aims at annotating web content using tags. A bottom-up classification emerges when all the annotations (tags) from a tagging community are merged.

In music, the majority of tags describe audio content. Genre, mood, and instrumentation account for 77 percent of the tags (Lamere 2008). Music items that share some tags (genre, instrumentation, artist location, active decades, moods, and so on) are more likely to be similar. Likewise, users that have applied the same tags are more likely to have similar listening tastes.

Both usage- and social-based recommendations have some limitations. New items or unknown items are difficult to recommend because there is not enough rating or tag data about the items to make recommendations. Popular items tend to be recommended more often, leading them to become even more popular, creating a feedback loop that results in a recommender that draws from a shrinking pool of only popular items. Due to this feedback loop, early raters of an item can have an overwhelming influence on the behavior of a recommender. A handful of early poor ratings

can doom even the most exceptional artist to obscurity (Celma 2010).

Content-Based (CB) Recommender

A content-based recommender extracts features directly from the music and uses these features to determine item similarity. Early work on audio similarity is based on low-level timbre descriptors, such as Mel frequency cepstral coefficients (MFCC). These approaches are aimed at deriving timbre similarity, but have also been used to take on other problems, such as genre classification. The *bag-of-frames* timbre approach models the audio signal using a statistical distribution of the audio features on short-time audio segments. Audio features are then aggregated using simple statistics (for example, mean and variance), or modeled as a Gaussian Mixture Model (GMM). However, as pointed out by Aucouturier and Pachet (2008), a timbre representation based on MFCCs and GMMs tends to create hubs. These are songs that are irrelevantly close to every other song.

Similarity measures on top of the bag-of-frames approach include Kullback-Leibler (KL) divergence and the Earth mover's distance (EMD). KL divergence measures the relative (dis)similarity between two Gaussian distributions of data. EMD has been largely applied in the image community to retrieve similar images (Rubner, Tomasi, and Guibas 2000). In the music domain, EMD is defined as the minimum amount of work needed to change one audio signature to another (Logan and Salomon 2001). Furthermore, other semantic features can be obtained from the raw audio, including automatic rhythm description (Gouyon 2008), tonality and chord detection (Gómez 2006, Harte and Sandler 2005), music genre classification (Tzanetakis and Cook 2002), or mood detection based on a predefined set of labels (Laurier et al. 2010).

Content-based similarity methods have some limitations. They tend to find similar pieces that belong to different music genres. It is very unlikely that a user will love both a Franz Schubert piano sonata, and a Meat Loaf classic rock ballad just because the two contain a prominent piano melody. CB methods have high computational requirements. Applying CB methods to a multimillion song database can be a computational challenge.

Hybrid Recommender

The combination of any of the previous three approaches allows a music recommender system to minimize some of the issues that a single method has. Combining different methods can be done using a cascade approach, a step-by-step process. For example, to compute artist similarity one can first apply CF, and then reorder the results according to the semantic distance in the social space, or through the distance in the acoustic space. Another hybrid approach is to combine the output of separate approaches using, for instance, a linear combination of the scores of each recommendation technique. This is called the weighted approach.

One popular approach to combining CB and SB methods is to build a model that can be used to predict social tags directly from audio features. Using this model, poorly tagged or untagged music can be tagged based on audio features, allowing these items to be incorporated in a social-based recommender (Eck et al. 2008). Figure 3 presents an autotagging system. Audio features extracted from music are used to train a predictive model for a social tag such as *smooth jazz*. This model is then applied to poorly labeled tracks to determine which tracks should be tagged with *smooth jazz*. These automatically assigned tags also serve to smooth the semantic space from which similarities are made in the social-based recommender system.

Autotagging of music mitigates against the cold-start problem. However, there are some open issues with autotagging. Roughly 25 percent of social tags are not related to the audio (Lamere 2008). Tags such as *seen live*, *awesome*, *funny*, *great lyrics*, and *guilty pleasures* are difficult to predict directly from audio. Subtle distinctions such as the difference between *technical death metal* and *progressive death metal* are beyond the capabilities of current content-based classifiers.

The Radiohead Example

As an example, we present Radiohead's similar artists, for the three types of recommenders (usage, social, and content based). After that, we present music recommendation examples for a given user profile in *safe* recommendations and *discovery* modes, using Radiohead as seed artist.

Data Sets

We compiled the Last.fm 360K data set,[1] which contains the usage data (more than 17M rows) for almost 360,000 last.fm users and 160,000 unique artists, for the period of 2005–2008. The data set contains, for each user, the total playcounts of his or her top-n artists ($n \leq 100$). It is represented as a matrix M^{UB}, where rows are artists, columns are users, and M^{UB}_{ij} is the total playcounts of artist i for user j. The playcounts values are not normalized, thus we rescale the rows of the matrix, so that they all have unit Euclidean magnitude. This data set is the one used for usage-based recommendation.

From the Last.fm 360K data set, we also have gathered the most representative tags for all the artists.[2] This data conforms to the social tagging information of the artists. It is represented as a

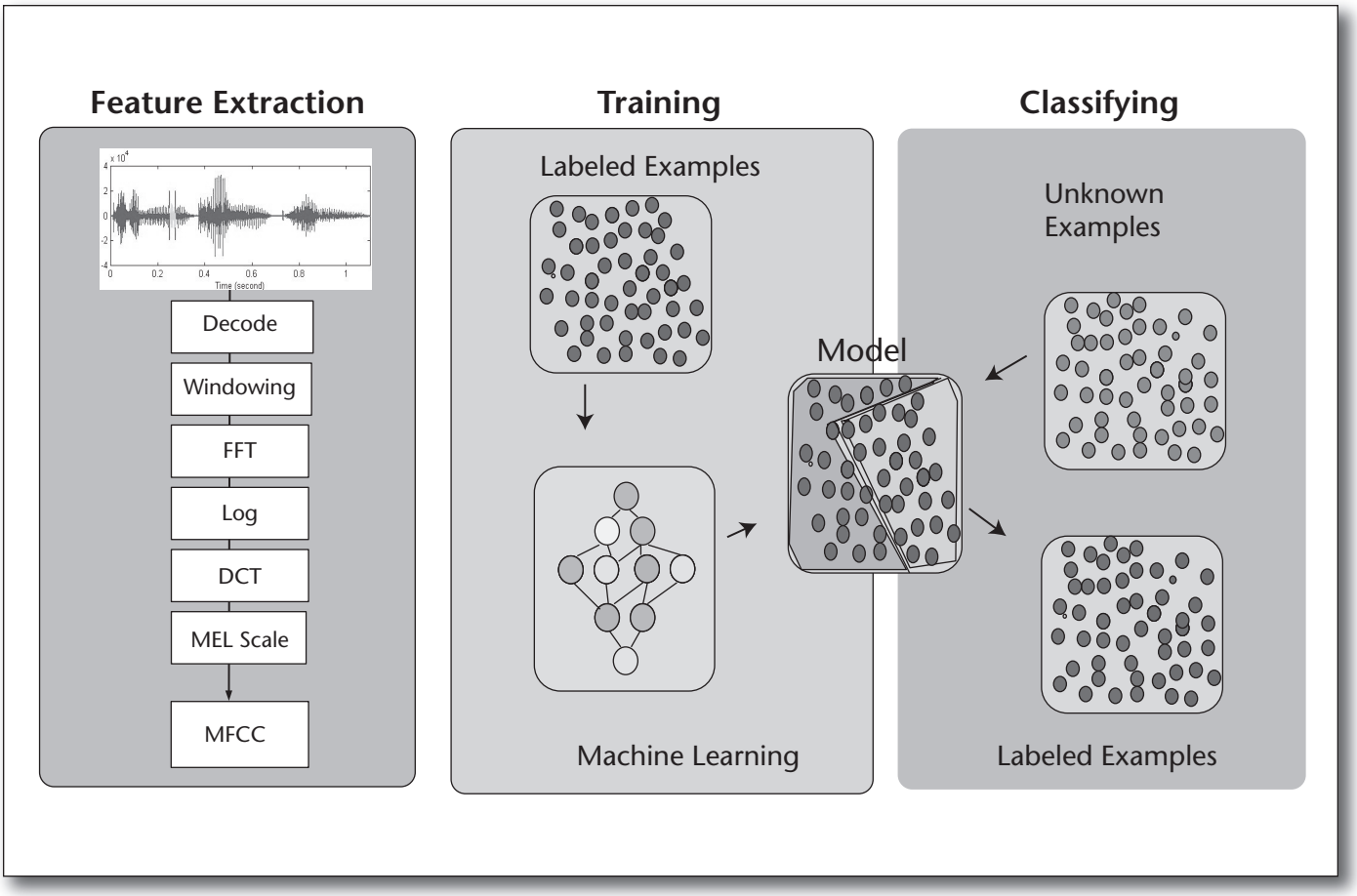

Figure 3. An Autotagging System.

matrix M^{SB}, where rows are artists, columns are tags, and M_{ij}^{SB} is the relevance value for artist *i* and tag *j*. In both matrices, we only use those artists that have three or more values (that is, at least three users played them, or have more than two different tags). Now, to compute similarity among the artists—either from the usage data, or from social tagging—we apply singular value decomposition (SVD), a matrix factorization technique, reducing the matrices to 100 factors or dimensions. Artist similarity is represented by the cosine distance between two vector rows in the reduced space.

To compute artist similarity using audio content-based similarity, we use an in-house music collection of more than 9 million tracks. The audio features include not only timbral features, such as MFCC, but musical descriptors related to rhythm (beats per minute, and perceptual speed), tonality (chroma features, key, and mode), genre estimation, and moods. Euclidean distance over a reduced space, using principal component analysis (PCA) and 25 dimensions, is used to compute song similarity. To compute artist similarity we retrieve the most popular tracks ($\mathcal{T}_a$) of an artist *a*, with a maximum of 100 tracks per artist. For each artist track, $t_i \in \mathcal{T}_a$, we obtain the most similar tracks (excluding those from artist *a*), and get the artists' names, $\mathcal{A}_{sim(ti)}$, of the similar tracks. Similar artists of *a* is composed by all $\mathcal{A}_{sim(ti)}$, aggregated by audio similarity:

$$similar(a) = \bigcup \mathcal{A}_{sim(t_i)}, \forall t_i \in \mathcal{T}_a$$

Similar Artists

Table 1 shows Radiohead most similar artists, using the three recommendation approaches. The value indicates how many users in the 360K data set have listened to the artist. Thus, it is an indicator of artist popularity. It is worth noting that the results from collaborative filtering (usage-based approach, UB) contains many more popular artists than the other two approaches. Similar artists from the social-based (SB) approach consists of a mix of popular and unknown *alternative rock*, *indie,* and *electronic* bands. The content-based (CB) approach is the one with less known artists. Interestingly enough, in CB there are a few female vocalists

Usage Based (UB)	Social Based (SB)	Content Based (CB)
Beck$_{4867}$	dEUS$_{3865}$	Autumn Defense$_{50}$
dEUS$_{3865}$	Vita de Vie$_{203}$	Anathema$_{8498}$
Kashmir$_{1502}$	Airship$_{4}$	Augie March$_{460}$
Doves$_{3787}$	B. Corgan & M. Garson$_{7}$	Heather Duby$_{19}$
Interpol$_{20568}$	Ostava$_{198}$	The Standard$_{34}$
Twilight Singers$_{737}$	Galaxie 500$_{1112}$	Jenny Owen Youngs$_{435}$
Phoenix$_{6094}$	Enemy$_{268}$ Maxïmo Park$_{7562}$	Lavagance$_{73}$
South$_{414}$	State of Shock$_{149}$	Jane Weaver$_{10}$
Kings of Conv.$_{11020}$	This is serious mum!$_{177}$	Sad Riders$_{3}$
Magnet$_{469}$		The Chrysler$_{14}$

Table 1. Radiohead Top Ten Similar Artists Using the Three Recommendation Methods: Usage, Social, and Content Based.

The value indicates how many users in the data set have listened to the artist, an indicator of the artist's popularity.

whose voices resemble the voice of Radiohead's high-pitched singer.

Personalized Recommendations

Figure 4 depicts a last.fm user profile,[3] based on the user's top-20 most played artists, split on five clusters. We use *k*-means to clusterize the 20 artists from the profile. Artist similarity among them is derived from usage-based (CF) matrix M^{UB}, using cosine distance. The closest cluster to Radiohead is $C2$. That is the cluster that has its centroid "closer" to the artist vector.

Now imagine that this user wants to create a personalized playlist, based on seed artist Radiohead. To achieve this, first we get Radiohead top-100 similar artists and then rerank these results based on each artist's similarity with the user's $C2$ centroid. Table 2 presents the recommended artists for the user. We simply decided that *safe* recommendations contains the more popular artists, while the *discovery* mode has more unknown artists. Yet, all the results are relevant to the user profile. Note that in the CB approach all the artists are rare or unknown. Indeed, in pure CB music recommender systems there is no notion of what is popular or not.

In all the three approaches, an explanation of why those unknown artists—in the discovery mode—are recommended would be desirable. Otherwise, the user can perceive them as erroneous, unmeaningful predictions. On the other hand, another way to create a *safe*, comfortable, playlist could be to play music only from artists in the user profile that are similar to the seed artist, Radiohead.

Further Research

Research in music recommendation is multidisciplinary. It includes several areas such as search and filtering, data mining, machine learning, personalization, social networks, text processing, complex networks, user interaction, information visualization, and signal processing, among others. Furthermore, current research in recommender systems has a strong industry impact, resulting in many practical—and potentially successful—applications. Still, there are a number of open questions that could be addressed in further research. Some examples follow.

How well do music recommenders work? There is a lack of standardized data sets and objective evaluation methods to objectively compare different recommendation approaches. Without such data sets and methods it is difficult for researchers to share and compare recommendation approaches and results. Subjective evaluations by music listeners are also needed to understand how effective music recommendations really are. These evaluations tend to be very difficult to conduct and take a substantial amount of time even when they involve a small number of listeners. A standard framework that would allow researchers to conduct subjective music recommendation evaluations with a large set of listeners would accelerate progress in the field.

How to recognize and incorporate context into recommendations? Context is important in music recommendation. Understanding how to recognize the listener's context (exercising, exploring, working, driving, relaxing, and so on) and using this context to drive appropriate music choices is an

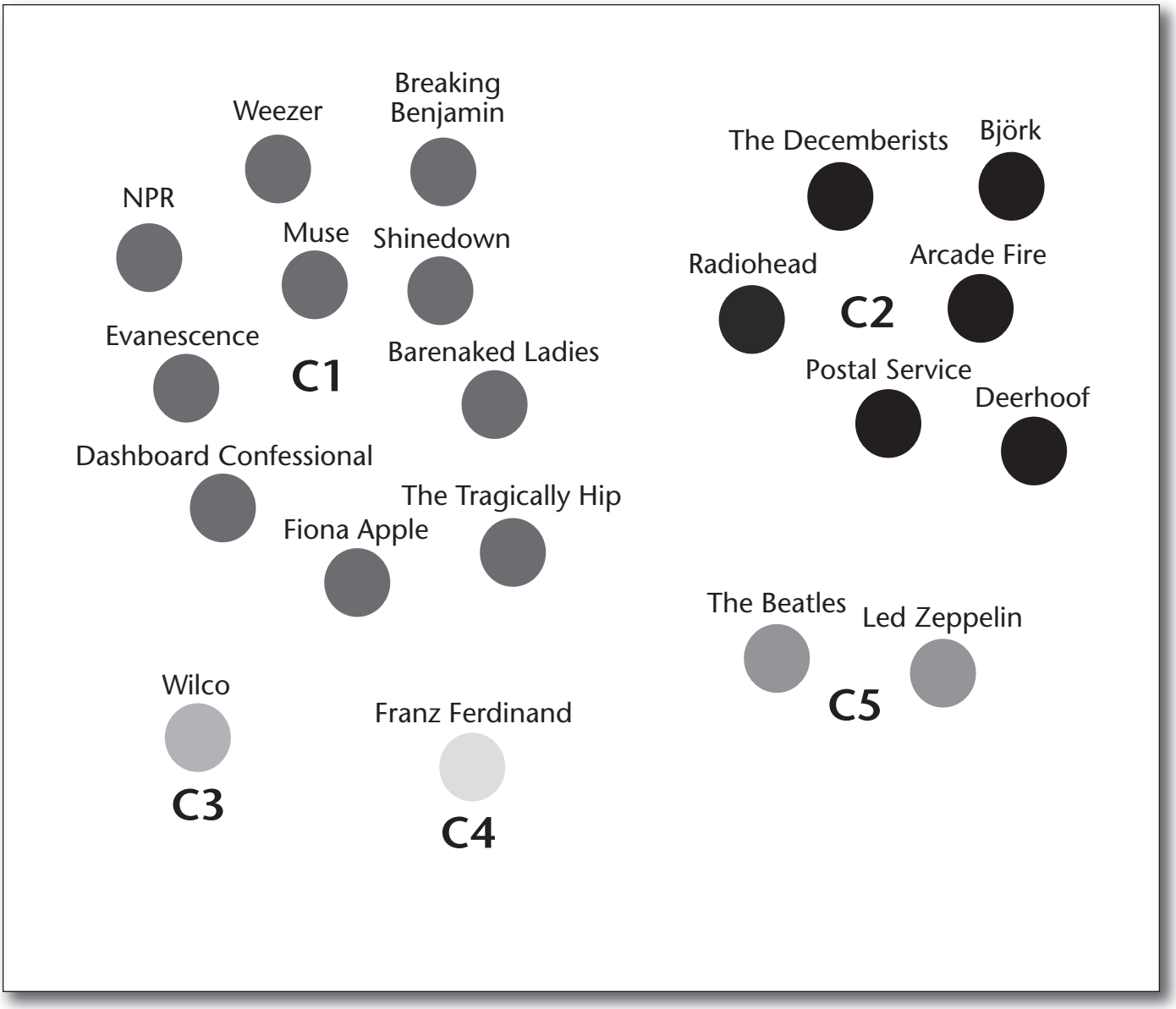

Figure 4. A User Profile Based on His or Her Top-20 Most Played Artists, Split on Five Clusters Using k–means Clustering.

Similarity among the artists is derived from usage-based (CF) matrix M^{UB}. The closest cluster to Radiohead is $C2$.

important consideration in the design of future music recommendation systems. Understanding a listener's intentions (exploring for new music, or seeking comfortable old favorites) and providing the most suitable recommendations to satisfy the listener is an open research problem.

How to make recommendations for all music? Current commercial recommenders tend to recommend only popular content. It is important for recommenders to be able to consider all music including new, unknown, and unpopular content. The dynamic nature of music presents a particular challenge. Artists can change their music style from one release to the next, and listener's tastes can change from week to week. Since there are millions of artists, hundreds of millions of tracks, and many millions of music listeners to consider, building a system that can effectively personalize recommendations for each user that reaches all content is a challenge.

What effect will automatic music recommenders have on the collective music taste? A music recommender can create a feedback loop that can drive listeners to a diminishing pool of popular artists. Understanding the effects that recommendation tools can have on music consumption can help inform the design of systems that yield more diverse recommendations.

	Usage Based (UB)	Social Based (SB)	Content Based (CB)
Safe	Interpol$_{20568}$ Beck$_{4867}$ Doves$_{3787}$ Raconteurs$_{6038}$ P. Bjorn & John$_{4736}$	Daughtry$_{2536}$ dEUS$_{3865}$ Maxïmo Park$_{7562}$ British Sea Power$_{1783}$ Nada Surf$_{5097}$	Robag Wruhme$_{129}$ Sono$_{169}$ YACHT$_{155}$ Märtini Brös$_{126}$ Move D$_{152}$
Discovery	Tiger Saw$_{3}$ Morning Runner$_{101}$ The Tacticians$_{4}$ Naked Lunch$_{115}$ Deportees$_{158}$	Vespertine$_{8}$ Airship$_{4}$ State of Shock$_{149}$ Dream City Film$_{36}$ Raining Pleasure$_{120}$	Fred Giannelli$_{4}$ Stereomatic$_{29}$ Freeform$_{76}$ Sex In Dallas$_{20}$ Heather Duby$_{1}$

Table 2. Personalized Recommendations for the User Represented in Figure 4, Based on Radiohead Seed Artist.

Safe mode contains more popular artists (in terms of the number of users that have played them). *Discovery* mode contains a list of more unknown, long-tail, artists.

Conclusions

Music is inherently different than other types of media. The space of recommended items is extremely large—compared to other domains—with a typical online music store offering 10 million titles to chose from. People interact with music differently than they do with other types of media. A new song can be auditioned in a matter of minutes whereas a movie may take a couple of hours to watch, and a book may take a dozen hours to read. People enjoy listening to music over and over, but it is the rare book that is read more than once. Listeners vary their music preference based upon context and activities. A playlist for jogging is likely to be very different than a playlist created by the same user for relaxing. Listeners enjoy listening to sequences of songs, often getting as much enjoyment from the song transitions as from the songs themselves. The uniqueness of music as a recommendation domain presents challenges not seen in other recommender domains. It is important to consider the special nature of music when building recommenders for music.

We are seeing huge changes in the world of music. Soon a listener will be able to listen to nearly any song that has been ever recorded at any time. However, users can be overwhelmed by these millions of listening options. Recommendation technologies that help users find new and interesting music are becoming increasingly important as a way to help the music listener sift through all of these options. These technologies will be integral in helping the next generation of music listeners find that next favorite song.

Acknowledgments

This work has been partially funded by the Buscamedia project (CDTI CEN-20091026). The authors would like to specially thank Keith Emerson and Timothy John Taylor for providing inspiration and encouragement to continue pursuing this subject area.

Notes

1. See www.dtic.upf.edu/~ocelma.
2. Using the last.fm API, we retrieved a list of tags with a normalized relevance value of 1..100 for each artist.
3. Last.fm user profile lamere.

References

Aucouturier, J.-J., and Pachet, F. 2008. A Scale-Free Distribution of False Positives for a Large Class of Audio Similarity Measures. *Pattern Recognition* 41(1): 272–284.

Baccigalupo, C. 2009. Poolcasting: An Intelligent Technique to Customise Music Programmes for Their Audience. Ph.D. Dissertation, Computer Science, Universitat Autònoma de Barcelona, Catalonia, Spain.

Celma, O. 2010. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Berlin: Springer.

Cunningham, S. J.; Bainbridge, D.; and Falconer, A. 2006. More of an Art Than a Science: Supporting the Creation of Playlists and Mixes. Paper presented at the 7th International Conference on Music Information Retrieval, Victoria, BC, Canada, 8–12 October.

Eck, D.; Lamere, P.; Bertin-Mahieux, T.; and Green, S. 2008. Automatic Generation of Social Tags for Music Recommendation. In *Advances in Neural Information Processing Systems 20*. Cambridge, MA: The MIT Press.

Gómez, E. 2006. Tonal Description of Polyphonic Audio for Music Content Processing. *INFORMS Journal on Computing* 18(3): 294–304.

Gouyon, F. 2008. *Computational Rhythm Description*. Saarbrücken, Germany: VDM Verlag Dr. Müller GmbH & Company KG.

Harte, C. A., and Sandler, M. 2005. Automatic Chord

Identification Using a Quantised Chromagram. In *Proceedings of the 118th Convention. of the Audio Engineering Society*. New York: Audio Engineering Society, Inc.

Jennings, D. 2007. *Net, Blogs and Rock 'n' Roll: How Digital Discovery Works and What It Means for Consumers*. Boston: Nicholas Brealey Publishing.

Lamere, P. 2008. Social Tagging and Music Information Retrieval. *Journal of New Music Research* 37(2): 101–114.

Laurier, C.; Meyers, O.; Serrà, J.; Blech, M.; Herrera, P.; and Serra, X. 2010. Indexing Music by Mood: Design and Integration of an Automatic Content-Based Annotator. *Multimedia Tools and Applications* 48(1): 161–184.

Leong, T. W.; Vetere, F.; and Howard, S. 2005. The Serendipity Shuffle. In *Proceedings of the 23rd International Conference on Human Factors in Computing Systems*, 1–4. New York: Association for Computing Machinery.

Logan, B., and Salomon, A. 2001. A Music Similarity Function Based on Signal Analysis. *IEEE International Conference on Multimedia and Expo, 2001*, 745–748. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

McNee, S. M.; Riedl, J.; and Konstan, J. A. 2006. Being Accurate Is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, 1097–1101. New York: Association for Computing Machinery.

Rubner, Y.; Tomasi, C.; and Guibas, L. 2000. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* 40(2): 99–121.

Shardanand, U. 1994. Social Information Filtering for Music Recommendation. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Tzanetakis, G., and Cook, P. 2002. Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing* 10(5): 293–302.

Òscar Celma is the chief innovation officer at Barcelona Music and Audio Technologies (BMAT). He received his Ph.D. in computer science and digital communication from the Pompeu Fabra University (Barcelona, Spain) in 2008. His research interests include music recommendation, music discovery in large collections, complex networks in music informatics, and multimedia ontologies and the semantic web.

Paul Lamere is the director of developer platform for the Echo Nest, a music intelligence company located in Boston, MA. Lamere is interested in using technology to help people explore for new and interesting music. He is active in both the music information retrieval and the recommender systems research communities. Lamere authors a popular blog on music technology at MusicMachinery.com.