The Recommendation Problem: More than Just Music

The use of recommendation systems is pervasive in our society, whether we are aware of it or not. Movies, books, music, and even news articles are now curated with one intention in mind: retain you, the user. This is done in a variety of ways, with the result always being some suggestion in the domain the user is seeking media in. With the explosion of individual- and user-based media, the research for these kinds of systems has seen a massive exploitation in the last decade. This advance of research has not all been without challenge. In fact, with newer systems being created, new problems have arisen with the construction of new and complicated models. The most pervasive of issues is the question of the user, and how to judge the quality of recommendations in the context of the user. The more individualized these recommendation systems become, the more focus that will be placed on models driven by user feedback. Moreover, the question of forming a perfect blend of collaborative, context-based, and content-based filtering to model the user holistically is widespread. There is hope as new modeling techniques and more powerful machines allow us the tools to research and understand these problems more fully. In this paper, I will survey and analyze the general framework that recommendation systems have found in the musical domain. The survey of works will be organized by looking at the changes in methodology and challenges though time.

The most prevalent method in recommendation systems is collaborative filtering. This algorithm is one which defines the problem within the context of users and items. In the case of music recommendation, the item is the song or music which may be recommended to the user. This algorithm works a bit like a nearest neighbor algorithm would. Items are recommended to users based on what user of similar characteristics also like. Imagine that two users share a musical similarity of 80%. In this context, the definition of musical similarity can most simply be thought of as “how alike are these two users.” The reasoning is that if these two users are alike, then the remaining 20% of musical “dissimilarity” may be a source of shared musical interest (Magno & Sable, 2008). Collaborative filtering has been found to be very effective, in part due to its ability to model both users and items in the same low-dimensional vector space (Schedl, 2019). There are three kinds of collaborative filtering algorithms that have been utilized for music recommendation: memory-based, model-based, and hybrid algorithms. Memory-based algorithms predict items based on a collection of previous ratings from the user group. Basically, if we looked at the aggregation of all the users in the group, how do they rate this song? Songs looked more favorably upon by the group are recommended to the user. The next is model-based algorithms. These enable more user interaction and consider more preferences of the user. They do this by allowing users to rate recommendations that they’re given. The final type of collaborative filtering is hybrid, which combines the two above algorithms and outperforms any one individually (Song et al., 2012).

While both intuitive and effective, this type of modeling raises the first problems and questions that still need to be accounted for in music recommendation. Firstly, and most obviously, music is subjective. Different users may relate to or listen to songs for different reasons. While one user might like a song for its use of lyricism, another might like it for the instrument it uses. Broadly, we have a problem with defining user similarly. Grouping users is difficult because they may value or weigh different aspects of a song more heavily than others (Schedl et al., 2013). The second issue that we run into is the ideal of the “cold-start.” This is a broad term in recommendation system research which refers to an algorithm not having the full range of data available to it. In effect, we are missing some of the momentum generated by the algorithm because we need users to provide their preferences for the model be provide recommendations. Also, new music can’t automatically be recommended to these groups of users. It would take one or many users from a group to find a new song on their own for this new song to be introduced to a user group (Oramas et al., 2017). This leads to the final issue in music recommendation created by collaborative filtering algorithms: diversity. There is often a lack of pleasant surprises in these kinds of algorithms. Songs that are popular with the general population will more likely be seen by any one user in a group (Song et al., 2012). Ultimately though the biggest problem with this kind of recommendation system is the cold-start problem and lack of autonomy by the music recommendation system.

While collaborative filtering has been the norm since the turn of the millennium, new approaches which model content are becoming even more prevalent in research. In this case context refers to two different, but related subgroups. The first of which is defined in the recommendation system field. These include things like artist metadata (biography, location, genre) and user-generated data (user ratings). The other related field is Music Information Retrieval (MIR), which focuses on more signal-based data like rhythm, tempo, and melody (Schedl, 2019). These musical data are then put into a model. There is an abundance of different applications of deep learning in context-based music recommendation models and the curious reader should be directed to *A Review on Deep Learning for Recommender Systems* (Batmaz et al., 2018). For the purposes of this review, I will focus on one such string of methods that is both intuitive and has been found to be effective. These methods have been derived from MIR techniques in signal processing which capture acoustic characteristics like pitch, rhythm, and timbre. Waveform data of a song is compacted into key features of the song called Mel-frequency cepstral coefficients (MFCC). Songs are compacted by shrinking waveform data into key features of the song. These low-level feature spaces can capture the “shape” of the sound, and in essence, the timbral quality and way the song sounds to a particular user. The timbral quality, or measure of timbre, has been shown to be effective in signal-based music similarity along with pitch and rhythm (Magno & Sable, 2008). Defining the feature space in this way provides us with a concise picture of what a song might look like. Moreover, it offers a standard way of viewing music from any different kind of genre. The only issue lies in the high dimensionality of these feature spaces. Schedl has shown that principal component analysis has been effective as a dimensionality reducing technique in this field (Schedl, 2019). van den Oord et al. suggests using these MFCCs as an input into a convolutional neural network (CNN). These types of models can extract hidden features from the spectrogram which are compared for similarity (Batmaz et al., 2018). This type of deep learning method helps capture important information that may otherwise go unknown to simple content-based methods (van den Oord et al., 2013). Another technique that has been used is K-means clustering and Gaussian mixture models (GMM). Clusters are formed using K-means from the MFCCs and these clusters are fit to a GMM. This multimodal approach will then lead us to form recommendations based on songs nearest a target user's initial song request (Magno & Sable, 2008). Another simpler method is one which calculates mean and variance of MFCCs over texture windows. These texture windows are defined as the minimum duration of time it takes for a user to identify the sound of the song (Magno & Sable, 2008). This gives each song a moving average of features, which can then be compared to others for similarity, and thus, recommendations.

The context-based models described above solve problems that were seen in the case of collaborative filtering, in particular, the cold-start problem. All music can be assigned these features from the MFCCs, from which similarity between the sounds of songs can be found. This also improves upon the popularity bias that collaborative filtering suffers from. In some literature, this is referred to as the “long-tail” problem (Oramas et al., 2017). How can we recommend novel, non-popular music to users who want to explore and get recommendations from the thin tails of their genre? The answer to this is still open-ended and openly researched as part of a more holistic user recommendation system. While these content-based models are great for providing some sort of metric to provide a measure of song similarity, they lack the user-centric design that collaborative filtering provided. Moreover, some content included in the content-based models like genre are still ill-defined. Certain circumstances define genre differently and in ways including geographically, technically, and lexically. It is also uncertain whether genre is something that applies to artists, albums, or songs themselves (Magno & Sable, 2008). Hybridization would help mitigate shortcomings of both approaches. Content-based models help solve the cold start problem by defining a feature space for music with no user interaction. Collaborative filtering allows user feedback in the recommendation system (Schedl et al., 2018).

Ideally, recommendation systems should provide users with a perfect recommendation. This presents a challenge beyond the scope of what the previous two models and their hybrid can handle. Each user is as individual and multifaceted as the next, which has been somewhat neglected in the field. Collaborative filtering provides some user-centric design elements, but these only personalize based on static user preference (Schedl et al., 2013). One final piece in the music recommendation engine is the user context. Music has already been defined within a context through genre, and both musical and user content are defined by timbre and user feedback, respectively. However, user context is ultimately a lacking aspect in music recommendation. User context in this case refers to characteristics like the users’ mood, the social context, and physiological aspects (Schedl et al., 2013). This is a problem that is much less easy to define than the user-item pairing question that has come to be associated with recommendation systems. On top of this interaction information, the intrinsic, extrinsic, and contextual information for users' matter. We are assuming that things like audio content can be predictive of what a user may like on their own, while it has been shown that acoustic properties are subjective in the perception of music (Schedl et al., 2018). This turns the problem of recommendation into the question "which of these songs are similar to this one for this group of users?" Collaborative filtering tries to remedy this, but not to the extent that is needed for a truly holistic recommendation. User feedback must be as multifaceted as the user seeking to answer questions not only about the users liking of the recommendation, but the intention of listening or the contextual compatibility (Schedl et al., 2013).

One final issue of music recommendation systems is the testing conditions for finding significant results. In the case of recommendations, it's tough to determine what the "better" model is. Is it the one with the statistically significant difference from the other? How were users feedback on the models implemented into the work? In practice, it's been shown that recommendation engines need to perform around 80% better for users to notice a practical difference (Urbano et al., 2012). What we really need to measure is the effect size of the difference between recommendation systems. Is there a practical effect, or just statistical significance? We can better achieve this result by providing results in terms of p-values as well as confidence intervals (Urbano et al., 2012).

Overall, the state of music recommendation is at a point of opportunity. User-centric and content-based models are pervasive as they are useful, but struggle in terms of finding a clean vector space to work in collaboratively. I think that future work in hybrid models of this kind can and should implement context-based information, which could intuitively fit into the larger feature space containing a holistic model of content and context for music and user alike.

Works Cited

Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, *52*, 1-37.

Magno, T., & Sable, C. (2008, September). A Comparison of Signal Based Music Recommendation to Genre Labels, Collaborative Filtering, Musicological Analysis, Human Recommendation and Random Baseline. In *ISMIR* (pp. 161-166).

Oramas, S., Nieto, O., Sordo, M., & Serra, X. (2017, August). A deep multimodal approach for cold-start music recommendation. In *Proceedings of the 2nd workshop on deep learning for recommender systems* (pp. 32-37).

Schedl, M., Flexer, A., & Urbano, J. (2013). The neglected user in music information retrieval research. *Journal of Intelligent Information Systems*, *41*(3), 523-539.

Schedl, M., Zamani, H., Chen, C. W., Deldjoo, Y., & Elahi, M. (2018). Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, *7,* 95-116.

Schedl, M. (2019). Deep Learning in Music Recommendation Systems. *Frontiers in Applied Mathematics and Statistics*, *5*. https://doi.org/10.3389/fams.2019.00044

Song, Y., Dixon, S., & Pearce, M. (2012, June). A survey of music recommendation systems and future perspectives. In *9th international symposium on computer music modeling and retrieval* (Vol. 4, pp. 395-410).

Urbano, J., Downie, J. S., Mcfee, B., & Schedl, M. (2012, December). How Significant is Statistically Significant? The case of Audio Music Similarity and Retrieval. In *ISMIR* (pp. 181-186).

Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. *Advances in neural information processing systems*, *26*.