SONGSPOT: A LOCATION BASED MUSIC RECOMMENDATION SERVICE FOR ANDROID PHONES

Caleigh Mitchell
University of Victoria
caleighm@uvic.ca

ABSTRACT

SongSpot is a mobile application for context and location based music recommendation and discovery. Users check in to places to share and receive new music suggestions based on their location at the time.

Recommendations are determined on an individual basis to fit a user's personal musical preferences combined with the subset of music that the community feels fits the location.

SongSpot uses an analysis of the spectral features of the user's selected tracks with a machine-learning algorithm to determine an individual's musical preference, and then selects based on this from the pool of user submitted songs for an area.

Users also have the ability to view the most recently listened to tracks in a location, as well as for a particular user.

1. INTRODUCTION

There are already a large variety of music recommendation services available on the Internet. Some of the most well known of these are Pandora, Last FM, and Songza. Each of these services has a different approach to the problem of music recommendation and discovery. Pandora uses a panel of musical experts to categorize songs and recommend new music with similar characteristics. Last FM uses a similar approach, but rather than experts, they use crowd sourced tags submitted by Internet users, and base their recommendations on public opinion and trends. Both sites ask the user for input as to what their current musical preferences are, and select appropriate tracks that are similar to the music the user has identified.

Songza takes a different approach. Rather than asking the user for their musical preferences as a starting point, they take contextual information as input. They allow the user to select a day of the week, and time of day, as well as an intended activity and then recommend a variety of 'expertly' selected playlists optimized for the selected activity for the user to select from.

SongSpot combines these two approaches. It cross-references the user's predetermined musical preferences with contextual information to come up with an ideal recommendation. SongSpot uses crowd-sourced data to compile a database of songs specific to each location, and uses the user's individual preferences to select the most appropriate track from the pool for that location.

Some of the advantages of this particular setup involve utilizing the potential of smart phones (and other android devices) as a media platform. Taking advantage of their portability and the ability of the user to remain connected to services at all times, rather than just when a computer is available. Further than just convenience though, the experience of SongSpot is enhanced by the portability of the android platform.

SongSpot also takes advantage of the huge and constantly growing popularity of social media. One only needs to look at the success of sites like Facebook, Instagram, and Twitter, and the popularity of location-sharing services like Facebook's check-ins, Foursquare, and the beer recommendation site Untappd to realize the potential value that comes from people's desire to be connected to a community at all times.

2. PROJECT DESIGN

2.1 Overview

The bulk of the application will be implemented with the Android SDK, using Java and the Eclipse IDE. This will allow for the greatest amount of flexibility and support for both back and from end functionality. Java has a large selection of libraries and tutorials available for android development, as well as machine learning and database access. This will also allow for the simplicity of keeping most of the work in the same language. PostgreSQL will be used for managing the database of song sets for each location. Prototyping and initial visual design will be done using the online service Lucidchart.

The actual application will be written for android, as the mobile platform will allow for the greatest practical use as a check-in and location-sharing service, but

further development for the project could include a web-based interface as well, to reach a broader audience.
The initial version of the project will contain primarily locations on the Uvic campus, as well as major locations around Greater Victoria. It will also include functionality that allows users to add their own locations. User submitted locations would be reviewed to ensure they locations are not duplicated, and that they are valid places. Initially at least, private residences will not be included, and a new location must have at least five submitted tracks before it can be accessed to receive a recommendation.

2.1.1 Application Design

The application will consist of 6 main pages:
1. The main starting page, from which a user will be able to access their own profile for review and editing, and the option to check-in somewhere or add a new location to check in to.
2. Personal profile pages, which will contain the user's name, optional image, and a list of liked and highly rated tracks. This page will also include the option to edit this information.
3. The location pages will contain the location name, any details such as an address or description and reference image, and a list of recently listened to songs and the user who listened to them. Users will have the option to suggest a song, or to receive a recommendation. Either of these options, once completed, will add the user and song pair to the recent list.
4. Suggest a song – the user is taken to a page where they may suggest a song previously recommended for this location, or upload a new track.
5. Get a song – The user is taken to a page with an embedded track to listen to. Stop, play, fast-forward and rewind functionally are present. The user is then asked to rate the recommendation, and is then given the option to get another.
6. User profile pages – Reached upon clicking on a user-song pair, this page contains the user's name, an optional image, and a list of recently listened to and recommended tracks.

Each page will also have a home button, and a profile button.

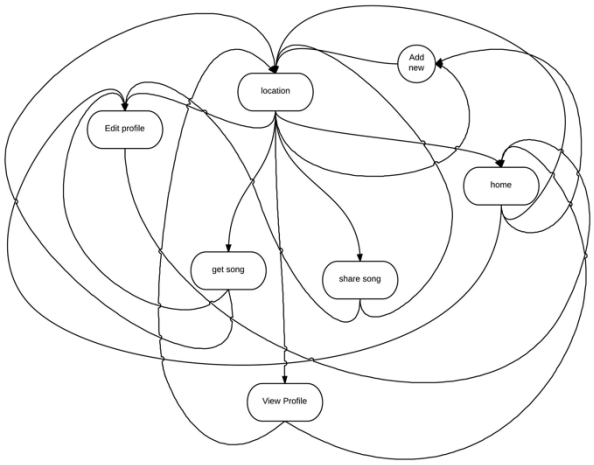

Figure 1. Page navigation within the app.

2.1.2 Recommendation System

Upon installing the app and registering a user name, the app will then be given access to the library of music stored on the user's device. It will use this selection of songs to construct a model of the user's taste. This model can be recalculated by the user at any time, and will also take into account the songs that the user chooses to share (if they are not already part of the calculation), and the ratings that the user gives to songs that they are recommended.

SongSpot is dependent on the submissions of users to populate the database from which it makes all recommendations. To allow for full functionality upon initial release, the database will be supplemented with a collection of tracks determined by surveys and solicitations from the Uvic community, and the general public. This initial data will be flagged, and removed once a quota of user submissions is reached.

3. PROGRESS REPORT

3.1 Prototype

Basic prototype completed using the online diagramming software Lucidchart. It is available to view at http://www.lucidchart.com/invitations/accept/5340ba30-ba5c-46fa-b8e3-06950a009433.

The functionality of the prototype is limited to page navigation, layout, and dialogue boxes. Its main purpose is to visualize the interface, and determine whether any initial changes are necessary before getting caught up in programming details. The next step will be to use these as a reference for implementing the basic android interface.

Figure 2. Main page and Location page from prototype

3.2 Tags vs Audio Feature Analysis

The Various methods exist for the comparison and classification of music and each of these have their own unique advantages and disadvantages depending on the application. As mentioned in the introduction, Pandora and Last FM use a system wherein tracks are annotated with tags, which contain descriptors and information about the piece of music, for example the mood and genre of the piece, whether or not it is instrumental, acoustic, performed by a male or female vocalist, contains banjo and slide guitar, or anything that the listener felt was valuable information for the classification of the track.

Another option for music classification is to use signal processing techniques to compare feature vectors. This would provide a balanced comparison method for each track, but would be more complex to implement within the time constraints of the project.

Problems with the tagging method arise when a song is added to the database that is new or relatively unknown and has no associated tags. This is known as the cold start problem. If there are no tags associated with the track there is no baseline for which to compare it with other tracks. Tagging systems also suffer from a popularity bias, wherein more popular well-known songs have a better collection of tags, being overly generalized and subjective, as well as vocabulary variations. One of the greatest advantages of the tag approach in this case is that there exist large databases of tagged music already. For the purposes of this project, due to time constraints and simplicity of implementation this becomes the most attractive option, despite some downsides.

3.3 Issues

3.3.1 Repetition of Recommendations

The problem arises, that by searching for the ideal track for a user in a particular situation, the user is limited to just one perfect choice, until such time as the requirements change and there is an option available that greater fits the constraints. This is obviously not ideal, as for a music sharing service of this nature; a user may wish to receive multiple recommendations at a time, or in similar situations. Therefore a constraint will be placed on the recommendations, requiring that any song or artist that has been recommended to the user in the last 10 recommendations be rejected, and the next best match be used instead.

3.3.2 User-Submitted Data

SongSpot initially includes a set of predetermined tracks, as well as base set of major locations. Users will then have the ability to add to these as they like. With this ability, the issues of duplicates and variation occur. For the system to be effective there can't be five versions of the same place, as then the song set will be divided between these, leading to less successful recommendations. In order to prevent this occurrence, each newly submitted location will only be made public once it has been checked for uniqueness. At this stage it will also be checked to ensure it fits the requirements of a location, which have been determined to be that it must be a real place, within the bounds of the app (at this point, Greater Victoria), and not a private residence. These constraints ensure that each location is a public place, visited by a reasonable number of people that will participate and encourage the growth of the system. Initially, this process will be complete manually, at regular intervals, but will eventually need to be automated.

3.3.3 Music Sharing

An earlier version of this project was essentially an inversion of the current concept. Rather than having users receive a recommendation from a database of shared songs that matched their musical preferences, it created a profile for each location based on the recommended songs, and would search the user's music library and select for them a song appropriate for the location. This idea was aban-

doned due to a greater desire to focus on the potential for new musical discovery and exploration.

A further version was more like the current concept, but involved the users actually downloading a copy of the recommended track to listen to on their mobile device. This was problematic due to regulations surrounding reproduction and sharing of copyrighted material. The current incarnation, which features a streaming track embedded in the page, appears to have dealt with this issue, though further research is required for confirmation.

4. FINAL REPORT

4.1 Application Details

The demo version of the app has been simplified to allow for completion within the time frame of the class. Profile functionality has not been implemented at this point, but each instance of the application on a device functions as an individual user profile. Personal user data is stored locally on the device (i.e. musical preferences), while song and location information is stored online. The original plan was to use postgreSQL for data storage due to background knowledge of the system, but due to a change in software used to implement the current version of the application, all public data is presently stored on the google cloud system.

The user's musical preferences are stored in the form of a bank of associated with the songs that they have indicated a preference for. A tally is kept of each occurrence of each tag and then the tags are ranked by number of occurrences.

When a user check in and requests a song from a location, the system looks at the tags listed for each song and selects the song that best matches the users indicated preferences.

The system selects the top tag from the user's profile, and loops through the pool of songs checking each one for the presence of that tag. Each song containing the tag is returned. If no songs are returned, the system moves on to the next tag. If there is only one, that song is selected and played for the user. If there are multiple songs returned, the system moves on to the next tag in the list. The subset of returned songs is checked, and any that contain that tag are returned. If there are none, it moves on to the next tag. This process is repeated until one song remains, in which case that song is returned and played, or no more tags remain. If the system runs out of tags, then one song is selected from the remaining subset at random and returned.

Adding a location is straightforward. When users submit a new location, a new table is created within the database. Once created, new songs can then be added to that table. Currently there is nothing in place to restrict what can and can't be added as a location, but items that do not fit previously listed constraints can be monitored for and removed.

The tags used in the recommendation system are from Last.fm, sourced using a separate script that is run regularly, when a new group of songs are added to the database. A selection of songs was added to the system as an initial pool. Once a sufficient number of user generated songs have been added, the initial songs will be removed. Song addition feature is currently in the form of a suggestion box, and suggested songs are uploaded manually at set points. Songs are currently hosted on the visual arts file system and stream from there.

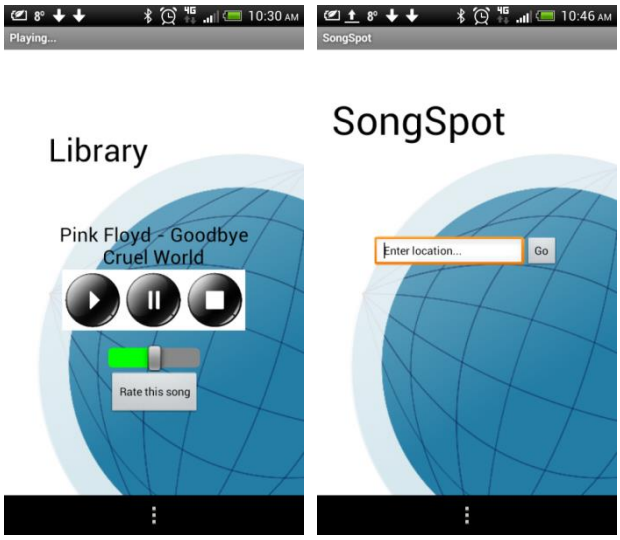

Figure 3. Screen capture of recommendation page and main page from functional demo app.

4.2 Things Learned

The greatest challenge in this project was learning how to develop an android application from scratch, and how to integrate the concepts and ideas that I'm familiar with a new development environment. Adjusting goals and changing elements from the initial plan was also difficult. Determining what elements of the application were essential, what could be compromised on, or eliminated now and added back in later in the interests of having a demonstrable prototype finished within the time frame.

It was determined that the essential part of the application is the basic framework, consisting of the song selection algorithm, the database of locations and song-tag groupings, and the ability to directly stream the songs within the app. Having the combination of online and local storage enabled the basic functionality of a user profile without the actual implementation.

The demo version of the app was constructed using the MIT App Inventor 2. App Inventor 2 is a graphical interface that has both a visual editor and a graphical drag and drop coding interface. This software simplified the learning process a great deal, but at the same time placed a lot of restrictions on what could be done.

A final version is still in progress using the Android development toolkit, written in a combination of java and XML. This allows for more freedom in the design of the interface, as well as the flexibility to more easily implement other non-essential elements (detailed in the following section).

4.3 Future Work

The demo version of the app was streamlined to the most basic functions. As this is something I am interesting in continuing to develop, there are a number of things from the original idea that need to be added back in, as well as extra features that could be added to improve the overall experience for the user.

4.3.1 User Profiles

In its current state, the application stores all personal user data on the device that is running it. Any and all users of that device are treated as the same for the purposes of the demo version, and users cannot share data between devices. It is possible for an individual to have the app on two different devices, and receive entirely different recommendations due to a different usage history on each device. This is not ideal, and the next step to developing the application will be to implement individual users profiles that are stored online and can be signed in to and accessed from any android device, and such that multiple users can use the app on the same device.

Initial user profile will store a username and password, and tag collection. The next stage will store a list of the user's 10 most recent songs, both submitted and received. Once profiles are implemented, a search function such that users can search for other user's profiles will follow, and then after that, the inclusion of visitor data at locations.

4.3.2 Redesign of decision making algorithm

The decision making algorithm currently in use is simple comparison of song tags, wherein the most frequently occurring characteristic are prioritized in the selection of new songs. The songs types that the user has indicated the highest preference for will be used to select the next recommendation, and if the user likes that song, that will strengthen the preference for that particular characteristic. This is likely to lead to issues of the user repeatedly receiving the same song as a recommendation, and a pool of songs that all sound similar. An improvement on this algorithm would allow for more variance in the selection, so that the user gets songs that they are likely to enjoy, but not songs that all sound the same. Using classifiers and machine learning algorithms to create a model for the user's profile that better reflects the users overall and varied music tastes. This would enable the discovery of more music the user would potentially enjoy, and reduce the potential for repetition.

Once this is implemented, it wouldn't be much further to apply the same profile construction algorithm to the locations, and then have the recommendations come from a general pool of music that is then selected from by an intersection of the two profiles.

This could also be used to ease the addition of new locations by adding types (e.g. school, café, transportation hub, office, library etc.) which can be applied to each new location as an initial profile that will then develop as the app is used.

4.3.3 Additional Functionality

The initial implementation of the project as a mobile app was the most logical, as one of the main ideas behind the application is related to the mobility of the user. Expanding to other platforms once the android app is completed would help to reach a greater audience. Moving beyond android and implementing the project as an iPhone app, as well as making it available as a web app would enable greater access to more people, and a greater user base would increase the number of songs added, and feedback received, which would improve the overall function of the application.

An additional direction that the project could take would be to take the recommendations one step further, to allow users the option to receive a playlist of songs, of their chosen length, rather than just one song.

5. REFERENCES

[1] C. Mascolo, A. Noulas, M. Pontil, and S. Scellato: "An Empirical Study of Geographic User Activity Patterns in Foursquare," *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pp. 570-573, 2011.

[2] H. Cramer, L. Holmquist, and M. Rost: "Performing a Check-in: Emerging Practices, Norms and

'Conflicts' in Location-Sharing Using Foursquare" *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 056-066, 2011.

[3] J. Cranshaw, J. Hong, J.Lindqvist, J. Wiese, J. Zimmerman: "I'm the Mayor of My House: Examining Why People Use foursquare – a Social-Driven Location Sharing Application," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2409-2418, 2011.

[4] H. Chen, A.L.P. Chen: "A music recommendation system based on music data grouping and user interests," *Proceedings of the tenth international conference on Information and knowledge management*, pp. 231-238, 2001.

[5] U Shardanand, and P. Maes: "Social Information Filtering: Algorithms for Automating 'word of mouth'," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 210–217, 1995.

[6] B. Logan: "Music Recommendation from Song Sets," *Proceedings of the International Symposium on Music Information Retrieval*, 2004.

[7] B. Logan: "Content-Based Playlist Generation: Exploratory Experiments," *Proceedings of the International Symposium on Music Information Retrieval*, 2002.

[8] B. Logan, and A. Salomon: "A Music Similarity Function Based on Signal Analysis," *Proceedings of the IEEE International Confrence on Multimedia*, pp 190, 2001

[9] A. Crossen, J, Budzik, and K. J. Hammond: "Flytrap: Intelligent Group Music Recommendation" *Proceedings of the 7th International Conference on Intelligent User Interfaces*, pp. 184-185, 2002.

[10] D. Eck, P. Lamere, T. Bertin-Mahieux and S. Green: "Automatic Generation of Social Tags for Music Recommendation." *Paper presented at the meeting of the NIPS*, 2007.

[11] S. Cho, H. Park, and J. Yoo: "A Context-Aware Music Recommendation System Using Fuzzy Bayesian Networks with Utility Theory," *Proceedings of the Third International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 970-979, 2006.

[12] E. Not, D Petrelli, C. Strapparava, O. Stock, and M. Zancanaro: "Modelling and Adapting to Context," *Personal and Ubiquitous Computing*, Vol. 5, No. 1, pp. 20-24, 2001.

[13] G. Adomavicius, and A Tuzhilin: "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extentions," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp. 734-749, 2005.

[14] P. Cano, M. Koppenberger, and N Wack: "Context-Based Music Audio Recommendation," *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pp. 211-212, 2005.

[15] M. Kaminskas, and F. Ricci: "Location-Adapted Music Recommendation Using Tags," *Proceedings of the 19th International Conference on User Modeling, Adaptation, and Personalization*, pp. 183-194, 2011.

[16] G. Jawajeer, P. Kostkova, and M. Szomszor: "Comparison of Implicit and Explicit Feedback from an Online Music Recommendation Service," *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pp. 47-51, 2010.

[17] J. Donaldson: "A Hybrid Social-Acoustic recommendation System for Popular Music," *Proceedings of the 2007 ACM conference on Recommender Systems*, pp. 187-190, 2007.

[18] Jin Chun Lee and Jae Sik Lee: "Context awareness by case-based reasoning in a Music Recommendation System," *Proceedings of 4th International conference on Ubiquitous Computing Systems*, pp45-58, 2007.

[19] Y. Zhang, D. O Seaghdha, D Quercia, and T. Jambor: "Auralist: Introducing Serendipity into Music Recommendation," *Proceedings of 5th ACM International Conference on Web Searching and Data Mining*, pp 13-22, 2012.

[20] P. Kazienko: "Web-Based Recommender Systems and User Needs – the Comprehensive View," *Proceedings of 2008 Conference on New Trends in Multimedia and Network Information Systems*, pp 243-258, 2008.