MOODetector: A Prototype Software Tool for Mood-based Playlist Generation

Luís Cardoso[1,2] , Renato Panda[1,2] and Rui Pedro Paiva[1,3]

[1] Department of Informatics Engineering, University of Coimbra – Pólo II, Coimbra, Portugal
[2] {lfac, panda}@student.dei.uc.pt, [3] ruipedro@dei.uc.pt

Abstract. We propose a prototype software tool for the automatic generation of mood-based playlists. The tool works as typical music player, extended with mechanisms for automatic estimation of arousal and valence values in the Thayer plane (TP). Playlists are generated based on one seed song or a desired mood trajectory path drawn by the user, according to the distance to the seed(s) in the TP. Besides playlist generation, a mood tracking visualization tool is also implemented, where individual songs are segmented and classified according to the quadrants in the TP. Additionally, the methodology for music emotion recognition, tackled in this paper as a regression and classification problem, is described, along with the process for feature extraction and selection. Experimental results for mood regression are slightly higher than the state of the art, indicating the viability of the followed strategy (in terms of R^2 statistics, arousal and valence estimation accuracy reached 63% and 35.6%, respectively).

Keywords: music emotion recognition, music information retrieval, playlist generation, feature extraction, regression, classification.

1 Introduction

Presently, digital music repositories need more advanced, flexible and user-friendly search mechanisms, adapted to the requirements of individual users. In fact, "music's preeminent functions are social and psychological", and so "the most useful retrieval indexes are those that facilitate searching in conformity with such social and psychological functions. Typically, such indexes will focus on stylistic, mood, and similarity information." [1]. This is supported by studies on music information behavior that have identified music mood[1] as an important criterion for music retrieval and organization [2].

Music Emotion Recognition (MER) has received increased attention in recent years. In the first days of MER, most approaches were dealing with MIDI or symbolic representations [3]. More recently, the problem of emotion detection in audio music signals has received increasing attention [5]. Many limitations can still be found and

[1] Even though mood and emotion can be defined differently, the two terms are used interchangeably in the literature and in this paper. For further details, see [4].

several problems are open. In fact, the present accuracy of those systems shows there is plenty of room for improvement. In a recent comparison, the best algorithm achieved an accuracy of 65% in a task comprising 5 categories [6].

In this work we do not follow a categorical model of mood, where the objective is to classify songs according to a pre-defined number of categories. Instead, a dimensional model, namely Thayer's mood model [7], is employed (Fig. 1). Here, the emotion plane is regarded as a continuous space, with two axes - arousal and valence - forming 4 different quadrants: 1) exuberance (happy and energetic music); 2) anxiety (frantic and energetic music); 3) depression (calm and anxious music); 4) contentment (calm and happy music). Each point, then, denotes a different emotional state and songs are mapped to different points in the plane (Fig. 2). Compared to categorical approaches, dimensional models are less ambiguous, as no mood adjectives are employed.

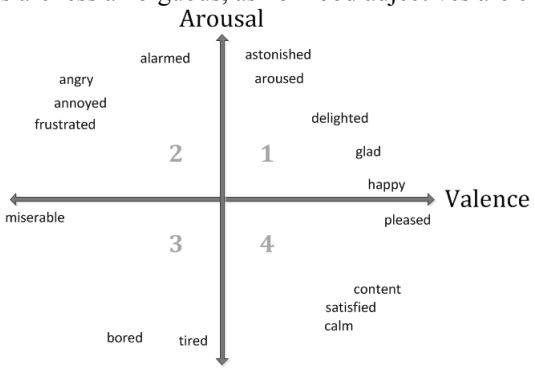

Fig. 1. Thayer's model of mood (adapted from [8]).

The aim of this article is to present MOODetector, a prototype application for the automatic generation of mood playlists, as well as the underlying methodology and current results. To this end, we built on Yang's work [8], where a regression solution to music emotion recognition was proposed. In addition, we aim to track mood in audio music, specifically its changes over time in terms of Thayer's quadrants.

This paper is organized as follows. Section 2 presents the MOODetector application and its main functionalities. In section 3, the followed feature extraction, regression and classification processes are addressed. In section 4, experimental results are presented and discussed. Finally, conclusions from this study are drawn in section 5.

2 MOODetector Application

In this section an overview of the MOODetector application is presented, namely regarding the description of the user interface, main functionalities, the process of playlist generation and mood tracking visualization capabilities.

2.1 Application Overview

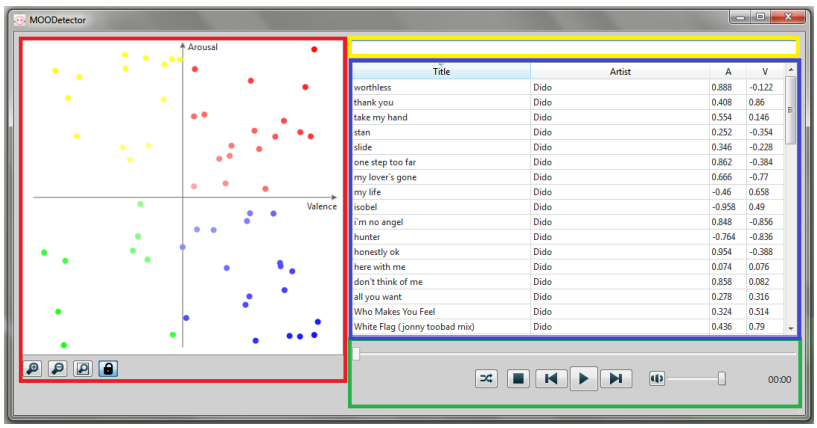

Fig. 2. MOODetector interface. Red is the Thayer's plot, green the multimedia controls, blue the playlist view and yellow the instantaneous search bar.

The MOODetector application is developed using the Qt Framework [9], a C++ application and UI development framework. It also uses the Marsyas[2] framework as part of the feature extraction logic, as well as the supervised learning algorithms offered by the libSVM[3] library for mood prediction. Both are open source projects, also developed in C++. Fig. 2 illustrates the user interface, which has five main components: i) the main window (where all the other components are); ii) the Thayer's plot (region highlighted with red); iii) the playlist view (blue region); iv) the multimedia controls (green region); iv) the instantaneous search bar (yellow region).

Regarding the Thayer's plane, songs are positioned according to the estimated arousal and valence (AV) values (the process for AV estimation is described in section 3.2). A color code is employed for song presentation in the plot: red, yellow, green and blue for the first, second, third and fourth quadrants, respectively. Additionally, the intensity of each dot (song) varies with the distance to the origin: lower if the song is close to the origin and getting higher as the distance increases.

As this application is meant to work as a typical media player, it has the basic usual music playback capabilities:

1. Playback control of mp3 songs (pause, stop, forward, backward, shuffle)
2. Volume control, mute, seek bar, and time label
3. Double click to play (in Thayer plot and playlist view)

It also has some more advanced music player capabilities like:

1. Instantaneous music search (by music name and artist)
2. Sort by song name, artist, arousal and valence values
3. Simple management of the library (adding and deleting songs)

[2] http://marsyas.info/
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

4. Music library statistics (library size, count by quadrant, analyzed songs, …)

In the realm of mood detection, among other functions, this system can:

1. Automatically estimate AV values for songs added to the library (sec. 3.2)
2. Allow the visualization of all or part of the music library in the Thayer's plot
3. Navigate in the mentioned visualization using zoom and panning
4. Locate a song in the Thayer's plot by clicking on the playlist view
5. See the numeric value of arousal and valence for a song
6. Automatically estimate the mood tracking of a song (the mood tracking estimation mechanism is described in section 3.2)
7. Display a visualization of the mood tracking of a song
8. Allow the user to change the AV values of a song by drag and drop in the Thayer's plot, besides manual edition of those values
9. Allow the user to reset one or all the changes of AV values

In regard to playlists we can:

1. Generate playlists based on one seed song (or point in the plane)
2. Generate playlists based on a desired mood trajectory path drawn by the user, according to the distance to the seed(s) in the TP
3. Filter playlists via instantaneous search (can be combined with the previous)
4. Export the playlist to a m3u file

2.2 Playlist generation

As mentioned in section 2.1, the system allows three types of playlist generation: i) based on a single seed song (or point in the plane); ii) based on a desired mood trajectory path drawn by the user, according to the distance to the seed(s) in the TP; and iii) via instantaneous search (and combined with the previous two).

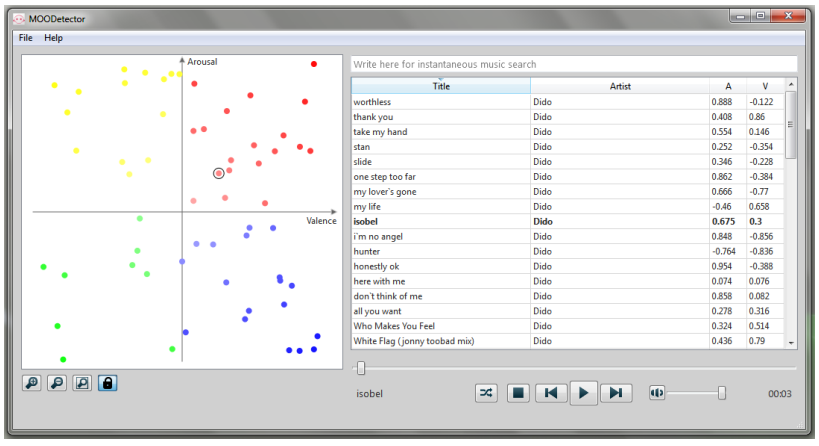

Fig. 3. Playlist generation using one seed song (the seed is the song in the plot with a black circumference surrounding it).

2.2.1 Single seed mode

In order to generate a playlist based on a single seed song, the desired seed is selected in the Thayer's plane. Then, the *N* closest songs, with a distance smaller than *threshold*, are added to the playlist (using the Euclidian distance). By default, the MOODetector system uses *N*=20 and *threshold*=0.35 (these parameters can be configured by the user). In Fig. 3 we can see the seed song that will be used in the playlist generation (black circumference around it). The resulting playlist is shown in Fig. 4. As it can be observed, it contains only 9 songs, according to the defined threshold.

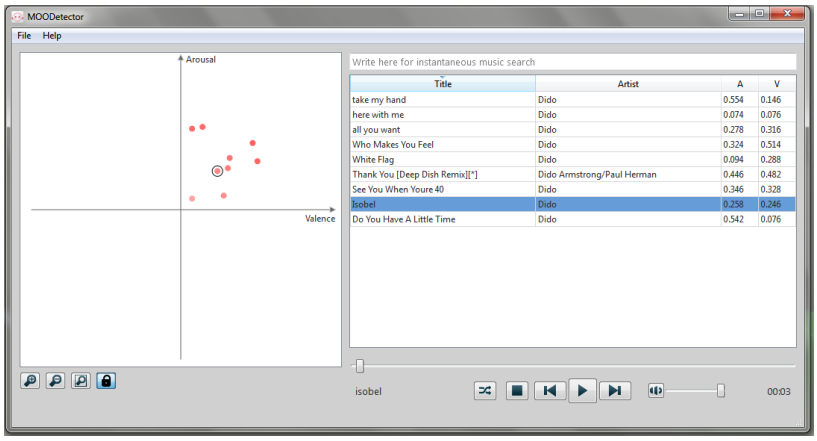

Fig. 4. The resulting playlist for the seed song in Fig. 3.

2.2.2 Path mode

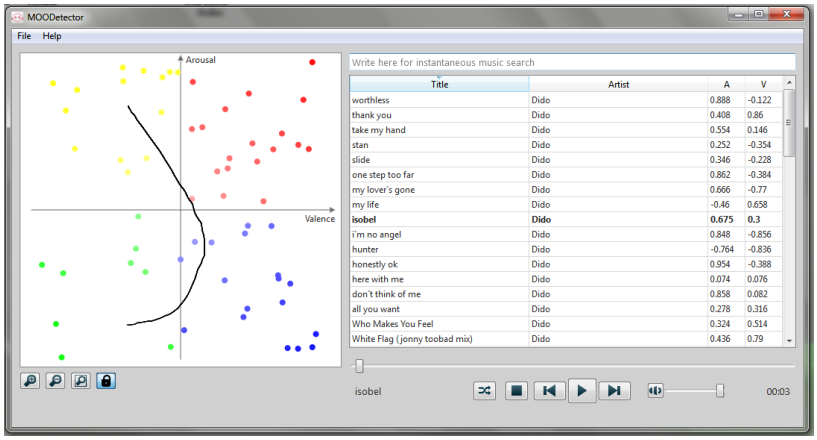

Fig. 5. An example of a line drawn by a user to generate a playlist using a path.

In this mode, the user draws a path in the Thayer's plot and the system computes the *N* closest songs to the path that have a distance less than *threshold*. This is done by

evenly dividing the path into *N* points (i.e., reference points) and then calculating the closest song to each point, with the restriction that the corresponding distance should not exceed *threshold*. As before, by default, the system uses *N*=20 and *threshold*=0.35. In Fig. 5 we can see the path that will be used for playlist generation. The resulting playlist is illustrated in Fig. 6.

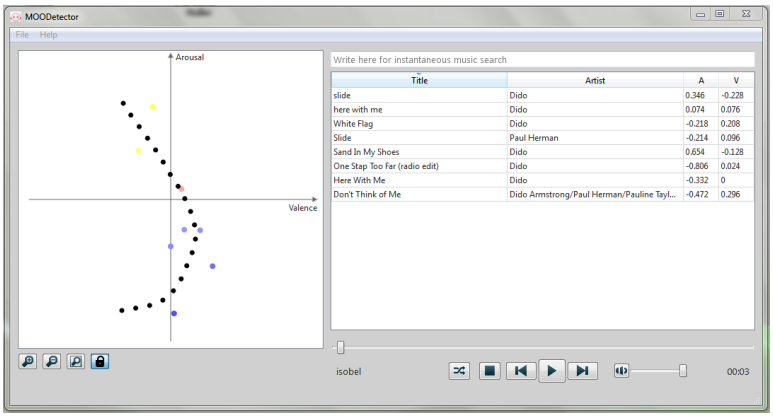

Fig. 6. The resulting playlist with reference points in black.

2.2.3 Instantaneous search mode

The user can also write in the search bar, which will automatically create playlists on the fly, by a song filtering process. If a song contains all the written words in any position of its title or artist name, the song is kept in the playlist; otherwise, it is removed. The comparisons are case insensitive. This mode can also be combined with both the single seed and path modes, for example, to find different versions of the same song. In Fig. 7 we can see the playlist that results from the search in Fig. 6, plus filtering with the search bar.

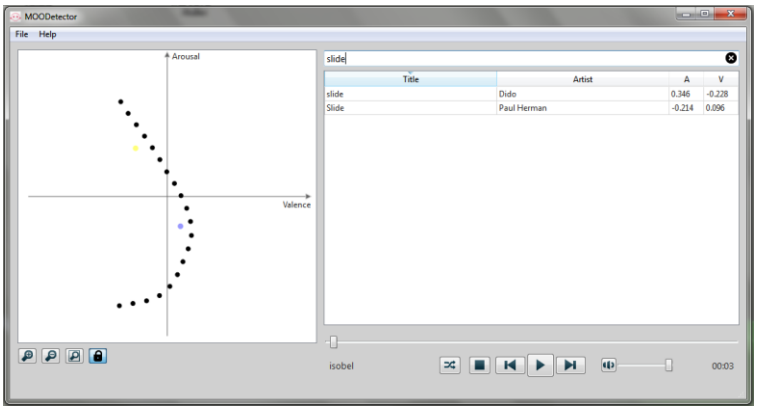

Fig. 7. The playlist resulting from the search in Fig. 5, plus filtering with the search bar.

2.3 Visualization of mood tracking

For each song, it is possible to visualize its mood variations throughout time. In Fig. 8, the smaller mood tracking plot illustrates quadrant changes in the Thayer plane across time. The same color code employed in the visualization of songs in the Thayer plane is kept. However, for simplicity, we decided not to include different color intensities, as the plots would contain excessive information. Instead, only the current quadrants are encoded in the color code, rather than exact AV information.

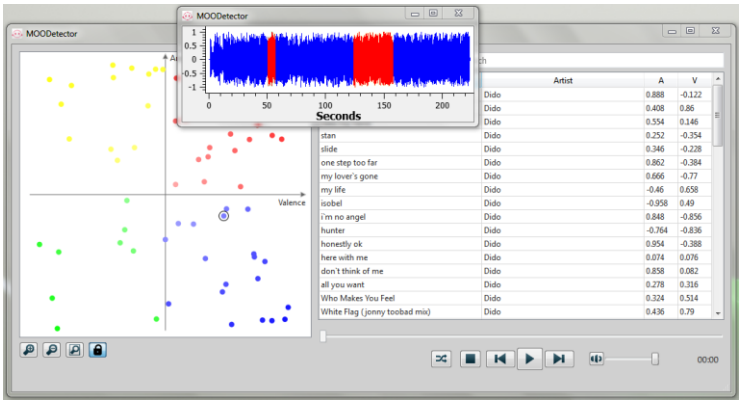

Fig. 8. The mood track visualization of a song.

3 Feature Extraction and Classification

The entire process of predicting a song's mood can be separated into three distinct stages. The first step consists in the extraction of audio features based on pitch, rhythm and timbral content of the audio signal of songs. After, the training process takes place, using both the extracted features and previously gathered AV annotations to create the mood models. Finally, these models are used to predict AV values for a target song or segment (in the case of mood tracking), based on its audio features.

3.1 Feature Extraction

Over the decades, several studies have been conducted to understand the relations between music attributes and evoked emotions. In one of the most recent overviews, Friberg [2] lists the following features as relevant for music mood analysis: timing, dynamics, articulation, timbre, pitch, interval, melody, harmony, tonality and rhythm. Other not included features but known to be relevant are, for example, mode, loudness or musical form [4].

One difficulty regarding the practical use of the listed features is related to the complexity to extract them from audio signals. As a result, most of the features

currently in use for music emotion recognition (MER) problems are the ones typically applied and known to perform well in other contexts such as speech recognition and genre classification. These low-level audio descriptors (LLDs) aim to represent attributes of audio like pitch, harmony, loudness, timbre, rhythm, tempo and so forth. LLDs are generally computed from the short-time spectra of the audio waveform, e.g., spectral shape features such as centroid, spread, skewness, kurtosis, slope, decrease, rolloff, flux, contrast or MFCCs [4]. Other methods have been studied to detect tempo and tonality.

To extract audio features, available audio frameworks can be used (e.g., Marsyas, MIR Toolbox, Psysound, etc.). Currently, our software tool extracts audio features from selected songs using the Marsyas framework, since its code is open source and it is available in C++, making it easier to integrate into our application.

For comparison purposes we also conducted our tests with two additional frameworks: MIR Toolbox[4] and PsySound[5]. These are implemented in Matlab but offer additional features currently not available in Marsyas, some of which have proved to be relevant [8]. Regarding Psysound, in order to compare results with previous studies, and given some limitations in the current version, a feature set obtained by Yang with an older version was used. In the future, it is our goal to integrate features from the MIR Toolbox and Psysound into our application.

Features from the three platforms were extracted in 512 samples' windows (23ms) and later transformed to song-level features by the MeanVar model [10], which compacts the feature set by computing means and variances.

In the conducted experiments, 189 audio clips of 25 seconds each were employed. These clips are mainly pop/rock and were gathered in a previous study by Yang, along with the respective AV annotations. The mood tracking approach was tested using full versions of 29 songs from the 189 clips, where two volunteers were asked to identify the mood changes across the Thayer's quadrants over time. Only the songs with above 80% agreement among volunteers were used in the mood tracking study. All songs were converted to WAV PCM format, with 22050 Hz sampling rate, 16 bits quantization and mono. The entire feature sets were normalized to the [0, 1] interval.

Further details regarding the feature extraction process can be found in a previous article by the team [11].

3.2 Regressor Training and Prediction

The following phase uses the extracted features and annotated AV values to train one arousal and one valence model. Support Vector Machines (SVMs) are employed based on the good results obtained in some MER problems [8]. Our implementation uses libSVM, a lightweight library written in C++ that provides both classification (SVC) and regression (SVR) solutions.

After the training process is completed, the resulting models, one for arousal and another for valence, can be used to predict AV values for a given set of audio features, placing each new song in the Thayer's plane.

[4] https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox
[5] http://psysound.wikidot.com/

As for mood tracking, the prediction process has some variations. Here, the used audio features represent each small segment of the song instead of the mean and variance values for the entire piece. The result is, rather than a single AV value, a collection of AV values, one for each segment (1.5-second windows, in this case), showing how the mood progresses over time. A second approach for mood tracking was also tested, transforming AV to quadrant annotations before the training process, using a classification model to predict quadrants changes instead of regression to predict AV mood tracking. Details on the mood tracking approach can be found in a previous paper by the team [12].

Given the high number of features, it is important to identify which ones are relevant to emotion detection and remove the less relevant ones. This was attained by using two feature selection algorithms – forward feature selection (FFS) [13] and RReliefF (RRF) [14]. One of the limitations of FFS is not taking into consideration the interdependence between groups of features. On the contrary, RReliefF does not assume feature independence. In addition, it also provides a weight to each feature in the problem under analysis.

In order to measure the performance of the regression models as well as to compare with previous approaches, R^2 statistics are used, "which is the standard way for measuring the goodness of fit for regression models" [8]. As for mood tracking, results are measured by calculating the matching ratio between annotations and classification predictions, i.e., the percentage of time with agreement between annotation and prediction.

4 Experimental Results

In this section we present the results obtained with the described approach.

4.1 Ground Truth Analysis

As mentioned, we employed the dataset and AV annotations gathered by Yang in a subjective study involving 253 volunteers, containing 194 music clips with 25-sec duration [8]. One limitation of this dataset is the unbalanced song distribution, as songs are not balanced across de four quadrants. This affects especially the second quadrant, which contains only 12% of all songs. As for mood tracking annotations, the main limitations of the collected annotations come from the fact that only two volunteers were used in this preliminary study. However, we are presently conducting an annotation study with a large population.

4.2 Regression Results

In the regression tests, 20 repetitions of 10-fold cross validation were run, ensuring that all songs are used in different groups for training and testing.

Various tests were run in to perceive the importance of individual features, as well as overall frameworks, on mood detection. Best results were obtained with FFS, using

a combination of all feature sets, reaching 63% for arousal and 35.6% for valence, using a total of 53 and 80 features respectively. Compared to the results reported by Yang [8], the prediction accuracy increased from 58.3% to 63% for arousal, and from 28.1% to 35.6% for valence, i.e., an improvement of 4.7% and 7.5%, respectively.

The number of used features was high, in part due to the FFS working mode. Although RRF results were lower, they were in many cases obtained resorting to less features, helping us to identify the most important features for both problems (AV).

The remaining tests highlighted MIR Toolbox features as achieving better results, especially on valence, with R^2 attaining 25.7%. PsySound followed, with a valence accuracy of 21% and Marsyas scored the lowest, only 4.6%, proving to be quite ineffective for valence prediction in this dataset. In terms of arousal, all the frameworks had a close score. A summary of the results is presented in Table 1.

Table 1. Results of the regression tests.

	All features (AV)		FFS (AV)		RReliefF (AV)	
PsySound	58.7%	12.7%	60.3%	21.0%	60.1%	21.1%
MIR Toolbox	58.2%	8.5%	58.7%	25.7%	62.1%	23.3%
Marsyas	52.9%	3.7%	56.0%	4.6%	60.0%	10.4%
All	57.4%	19.4%	62.9%	35.6%	62.6%	24.5%

4.3 Mood Tracking Results

The tracking results were measured by calculating the matching ratio between predictions and the respective annotations for full songs. Without feature selection and using the quadrants approach, the results of each framework, used separately, were generally similar (around 53% accuracy). Combining both frameworks, the accuracy increased marginally. Using feature selection, results rose nearly 2% for the MIR Toolbox (MIR TB), while dropping 1% for Marsyas.

The regression approach was used to predict AV values for the full songs. To measure accuracy, the predicted AV values were converted to quadrants and matched against the manual annotations. For Marsyas, using the entire feature set, results dropped to 48.9%. However, using the MIR Toolbox, results went up to 56%. This is consistent with results in section 4.2, where the MIR Toolbox performed better for valence prediction, while Marsyas features was ineffective. Applying feature selection improved all results, with the MIR Toolbox reaching 56.3%.

A summary of the most relevant results is shown in Table 2. PsySound was not tested due to the lack of support for feature extraction in several segments.

Table 2. Results of the mood tracking tests.

	All features		Feature Selection	
	Quadrants	AV	Quadrants	AV
Marsyas	53.5%	48.9%	52.6%	50.9%
MIR TB	52.7%	56.0%	54.5%	56.3%
All	53.7%	56.0%	54.7%	55.0%

4.4 Playlist Generation Results

To evaluate the quality of the single seed playlist generation strategy, we tested a regressor-based distance strategy. In this method, distances are calculated using the predicted AV values returned by the regression models. The predicted distances were compared to the reference distances resulting from the real AV annotations.

To this end, the dataset was randomly divided into two groups, balanced in terms of quadrants. The first, representing 75% of the dataset was used to train the regressor. Next, the resulting model was used to predict AV values for the remaining 25% songs[6]. From this test dataset, a song is selected and serves as the seed for automatic playlist generation. Using the seed's attributes, similarity against other songs is calculated. This originates two playlists ordered by distance to the seed, one based on the predicted and another on the annotated AV values. The annotation playlist is then used to calculate the accuracy of the predicted list, by matching the top 1, 5 and 20 songs, counting how many songs in each top are common between both playlist. The entire process is repeated 500 times, averaging the results.

Results were very similar among the three audio frameworks. The best results were attained using FFS for the combined feature set of all frameworks, with a matching percentage of 6.2% for top1, 24.8% for top5 and 62.3% for top20. Detailed results are presented in Table 3. The lower results in smaller playlists are mostly caused by the lack of accuracy when predicting valence. Still, best results are obtained with longer playlists, as normally used in a real scenario.

Table 3. Results of the playlist generation based on one seed song.

		Psy15	Psy44	MIR	Marsyas	ALL
Top 1	All	4.2%	4.1%	3.6%	4.0%	4.2%
	FFS	5.6%	3.8%	5.2%	4.4%	6.2%
	RRF	5.1%	4.6%	5.6%	4.6%	5.2%
Top 5	All	21.1%	20.9%	22.8%	18.1%	21.0%
	FFS	21.5%	21.1%	22.0%	19.8%	24.8%
	RRF	21.9%	22.1%	23.3%	18.7%	10.4%
Top 20	All	61.9%	60.5%	62.7%	58.5%	60.7%
	FFS	62.0%	61.9%	62.5%	60.0%	62.3%
	RRF	61.0%	60.8%	61.7%	57.4%	61.6%

5 Conclusions

In this paper, we present a prototype application and an approach for the automatic creation of mood playlists in the Thayer plane.

Regarding AV prediction accuracy, we were able to outperform Yang's previous results using forward feature selection on a set of features extracted from three frameworks (PsySound, MIR Toolbox and Marsyas), reaching 63% average accuracy

[6] This 75-25 division was necessary so that the validation set was not too short, as we want to evaluate playlists containing up to 20 songs.

for arousal and 35.6% for valence, in terms of R2 statistics. RReliefF was also important to highlight the most interesting features to the problem. As for playlist generation, matching for top1 was low, averaging 5% between all frameworks, with top20 presenting some reasonable results, of around 60%. In terms of mood tracking, 56.3% classification accuracy was attained.

The achieved results show that there is room for improvement, but are encouraging and have already improved the current state of the art.

In the future, we intend to perform a larger-scale annotation study for mood tracking, implement novel features, as identified in the literature review, and exploit the information conveyed in the lyrical part.

References

1. Huron, D.: Perceptual and Cognitive Applications in Music Information Retrieval. In: International Symposium on Music Information Retrieval (2000)
2. Friberg, A.: Digital Audio Emotions - An Overview of Computer Analysis and Synthesis of Emotional Expression in Music. In: International Conference on Digital Audio Effects. pp. 1--6. , Espoo, Finland (2008)
3. Lu, L., Liu, D., Zhang, H.-J.: Automatic Mood Detection and Tracking of Music Audio Signals. IEEE Transactions on Audio, Speech and Language Processing, 14, 5--18 (2006)
4. Meyers O. C.: A mood-based music classification and exploration system. M.Sc. thesis, Massachusetts Institute of Technology, 2007
5. Kim Y. E., Schmidt E. M., Migneco R., Morton B. G., Richardson P., Scott J., Speck J. A., Turnbull D.: Music Emotion Recognition: A State of the Art Review. In 11th Int. Society for Music Information Retrieval Conf. (2010).
6. Music Information Retrieval Evaluation eXchange - MIREX'2010 Results, http://www.music-ir.org/mirex/wiki/2010:MIREX2010_Results
7. Thayer, R. E.: The biopsychology of mood and arousal. Oxford University Press (1989)
8. Yang, Y.-H., Lin, Y.-C., Su, Y.-F., Chen, H.H.: A Regression Approach to Music Emotion Recognition. IEEE Transactions on Audio, Speech, and Language Processing. 16, 448--457 (2008)
9. Qt – Cross-platform application and UI framework, http://qt.nokia.com/
10. Meng, A., Ahrendt, P., Larsen, J., Hansen, L.K.: Temporal Feature Integration for Music Genre Classification. IEEE Transactions on Audio, Speech and Language Processing. 15, 275--9 (2007)
11. Panda, R., Paiva, R.P.: Automatic Creation of Mood Playlists in the Thayer Plane: A Methodology and a Comparative Study. 8th Sound and Music Computing Conference (accepted for publication) (2011)
12. Panda, R., Paiva, R.P.: Using Support Vector Machines for Automatic Mood Tracking in Audio Music. 130th Audio Engineering Society Convention (2011)
13. Chiu, S.L.: Selecting input variables for fuzzy models. Journal of Intelligent and Fuzzy Systems. 4, 243--256 (1996)
14. Robnik-Šikonja, M., Kononenko, I.: Theoretical and Empirical Analysis of ReliefF and RReliefF. Machine Learning. 53, 23--69 (2003)