MUSIC MOOD CLASSIFICATION OF TELEVISION THEME TUNES

M Mann
BBC R&D, London, UK
mark.mann@bbc.co.uk

T J Cox
University of Salford, UK
T.J.Cox@salford.ac.uk

F F Li
University of Salford, UK
F.F.Li@salford.ac.uk

ABSTRACT

This paper introduces methods used for Music Mood Classification to assist in the automated tagging of television programme theme tunes for the first time. The methods employed use a knowledge driven approach with tailored parameters extractable from the Matlab MIR Toolbox [1]. Four new features were developed, three based on tonality and one on tempo, to enable a degree of quantified tagging, using support vector machines, employing various kernels, optimised along six mood axes. Using a "nearest neighbour" method of optimisation, a success rate in the range of 80-94% was achieved in being able to classify musical audio on a five point mood scale.

1. INTRODUCTION

The BBC contains a vast archive of material estimated to be over a million hours, most of which has not been seen since it was first broadcast. The corporation is in the process of digitizing this archive, but very little is known about the programme's content. Consequently, various investigations are being carried out into the automatic classification of content and generation of metadata in order to enable searching and browsing of the archive when it is eventually published. However, because of the nature of the archive, the user may not necessarily know what is available. Therefore, researchers are investigating whether the user can browse the archive according to the mood of the programme they wish to see. One aspect of this is to attempt to determine the mood of the music contained within the programme and together with other audio and image recognition techniques [2], to tag a programme based on this.

Theme music is used to set the scene of a programme, so one would expect a happy, light tune to accompany an entertainment programme, and a dark, heavy tune for a serious, factual programme [3].

The field of Music Information Retrieval (MIR) is a well-established area of research with many methods and techniques for extracting audio features widely reported [4]. Consequently, the tools used in this paper are not in themselves novel, but the way in which they have been applied in this work is. In addition, this is arguably the first attempt to classify theme music, which is typically shorter than other pieces, using mood. The most common method for Mood-based Music Information Retrieval (M-MIR) classification in the literature thus far, has been to extract audio characteristics from music which are standalone values taken as an average over the entire piece or clip[6]. Certain audio characteristics can be very useful. For example, the mood heaviness scales roughly with the root mean square energy. To classify other, more complex moods, further such audio characteristics are added and processed with a support vector machine (SVM) classifier [5]. An SVM works as a binary classifier by taking a set of input data and predicts, for each given input, which of two possible classes the input is a member of. The justification of this approach is based on the supposition that the computer has the ability to cope with high-dimensionality and to determine trends to crudely mimic human perception. However, this method of approach does not take into account the inherent structure, order and progression of music. Characteristics extracted are often carried over from previous work into speech recognition and include Mel Frequency Cepstral Coefficients, entropy and flatness [6]. Such features are very useful but improved performance could be achieved by using common musical features such as tonality, dynamic range or tempo [7]. All but the most abstract music has a set of harmony and progression rules which are generally followed and are not always taken into account in determining the audio characteristics and features used in M-MIR literature to date.

This work details exploratory work with small datasets which will form the basis of more extensive investigations. It covers two new techniques for establishing features of variables which bear a greater resemblance to the tools used in musical composition, offering a better way for classifying the emotion of music. This includes a method for determining the overall tonality of the music, weighted tonal-

ity together with two new features of how these change during a piece and a more reliable tempo extractor. This work makes use of the Matlab MIR Toolbox but uses the output from the characteristics extractors to classify musical features in new ways. It differs from existing work in that it uses a knowledge-driven approach to quantify how extreme a mood is (e.g. is it quite happy or very happy?) and because it is examining theme music.

2. EXPERIMENTS

Upon starting this work it was clear that an adequate dataset for the aims of the project which described the mood of various theme tunes did not exist. Therefore, in order to gather sufficient ground truth data to train an SVM, a public engagement project entitled "Musical Moods" [8] was undertaken to obtain a dataset (in which the theoretical background, statistical data and reasoning for the dataset and the dataset itself can be found). This took the form of a survey in which 144 television theme tunes were rated by the general public on five point scales along the following emotional axes: happy-sad, light-heavy, dramatic-calm, masculine-feminine, playful-serious, relaxing-exciting. The axes were chosen to correlate with the semantic from Osgood's dimensional space; a three dimensional space incorporating Evaluation, Potency and Activity (EPA) [9].

Whilst the Musical Moods dataset was being gathered, it was necessary to use a development dataset upon which to experiment. Initial investigations attempted to find trends in features extracted using the MIR Toolbox [1] and the tracks tagged with mood-based adjectives in the Magnatagatune dataset [10]. Though only a small proportion of the dataset contained binary, rather than quantitative mood tags (i.e. happy-sad as opposed to very happy, quite sad), the Magnatagatune dataset was nevertheless considered useful for classification development and initially used to train single SVMs using a combination of the feature extraction tools available in MIR Toolbox.

Certain tools in the MIR Toolbox such as *mirrms*, which finds the root mean square of the energy of the track, *mirlowenergy*, which finds the percentage of the track time in which the audio is below a certain energy value and *mircentroid*, which finds the 'centre of gravity' in the frequency spectrum, were found to be very useful features to be incorporated into classification of some mood scales. Other tools were found to produce very useful results, but which needed to be enhanced and modified so that the extracted data could be converted into a useful, single number such as for the tools mentioned above in order to be used for classification.

2.1 Tonality

The first of these was *mirkeystrength*. There are 12 possible basic major chords and twelve possible minor chords in music. The function calculates and assigns a probability to each of the possible 24 chords at a sample rate that can be controlled with the function. For this investigation, half second intervals were used. The function calls another MIR toolbox function, *mirchromagram* [11], which calculates the energy distribution for each note in the diatonic scale. The pitches are then concatenated into one octave and normalized. Next, *mirkeystrength* cross-correlates the chromagram with the chromagram one would expect for each of the 24 chords and assigns a probability to each chord, where a probability of +1 for the tested chord would indicate a definite match whilst -1 would indicate a definite mismatch.

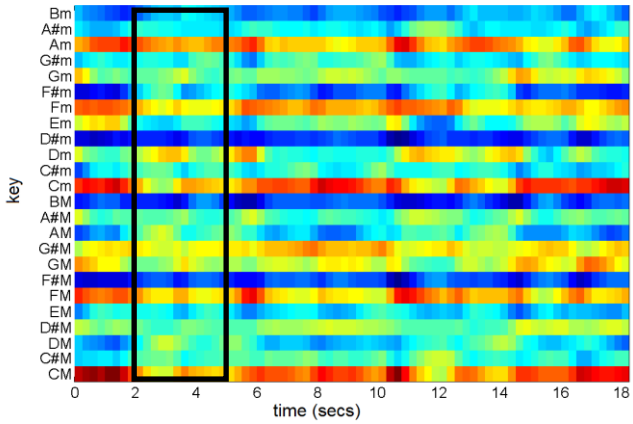

Figure 1. A graphical representation of the possible chords used in *Last of the Summer Wine* with time. Major chords are denoted with capital M, minor chords with a small m. The red colours denote a high degree of matching. Consequently, this piece is predominately in C major, though C minor gives quite a strong match also.

The reduction of a piece of music to major or minor chords is an oversimplification to a certain extent. Whilst major and minor chords are the basic construct of a piece of Western-style music, other chord types such as dominant sevenths, diminished sevenths, extended, other added tone and dissonant chords are used to great effect in music to elicit different emotions. However, by their nature, they are more complex and hence difficult to detect and can often be confused with major and minor chords. Consequently, when such chords are present, one would expect the key clarity to diminish. This can be seen in figure 1, which is the *mirkeystrength* chromagram for the theme tune of the BBC television programme *Last of the Summer Wine*. At around the 3 second mark (indicated by the black box) an added tone chord of C, D, F and A is played. The software understandably struggles to differentiate between D minor, A minor and F major chords as a consequence, with no as-

signed probability particularly high and no red in the figure at this point.

Nevertheless, *mirkeystrength* is an excellent tool because of the probabilities it associates with each chord. The ability to calculate tonality was thought to be of significant use because of it was perceived to have a correlation with mood axes such as happy-sad. Three features based on tonality were developed.

2.1.1 Weighted tonality

Data taken from *mirkeystrength* was used to find a meaningful feature for tonality which would have a correlation with the happy-sad axis and the Magnatagatune dataset. Bearing figure 1 in mind, it was clear that the feature needed to be weighted in some way. Indeed, without weighting, the correlation found between the happy-sad axis and the dataset was found to be poor. Consequently, the feature developed was named weighted tonality, *W*, and is defined as:

$$W = \frac{\left(\sum_{i=1}^{n} K_{max}\left(K_{maj} - K_{min}\right)\right)}{n} \quad (1)$$

Where:

K_{max} = peak tonality probability amplitude whether major or minor,

K_{maj} = peak major tonality probability amplitude,

K_{min} = peak minor tonality probability amplitude,

n = the number of time intervals used to classify the sample of music,

summed over all n and divided by n. Minor keys will therefore be of negative *W*.

This feature gives a much clearer representation of the overall tonality of the music under consideration because it emphasises certainty where it exists and minimizes uncertain contributions. This feature was combined with two other inputs, *mirrms* and *mircentroid*, to train an SVM on 99 tracks labelled with binary tags on the happy-sad axis. 82% of a further 194 tracks were then correctly classified, which is comparable with other success rates[6] and was considered to be a solid basis for the full investigation[13].

2.1.2 Weighted tonality differential

As well as the overall nature of the tonality in the music, it is also useful to know the frequency with which tonality changes. After taking time to study the sample set, it was found that moods such as exciting and dramatic tended to exhibit a more frequent change of tonality and dominant chord. Consequently, two features relating to the change in the in dominant chord were made. The first was a weighted tonality differential, which detects the rate at which the tonality changes during the course of the music.

It does this by finding the transitions and multiplying the transition with the sum of the certainties associated with the chords before and after the transition $\left|K_{maj} - K_{min}\right|_j + \left|K_{maj} - K_{min}\right|_{j+1}$ (where j corresponds to the certainty before and j+1 to the certainty after). It will only do this at transition locations. Where there is not a transition, the differential will be 0. This is then averaged over the number of time intervals, n. Again, because this weights the transitions with a certainty that the tonality change has happened, it gives greater emphasis to clearer transitions, thus filtering out transitions which may not have occurred.

2.1.3 Weighted chord differential

The second feature determined was a weighted chord differential, which detected the rate at which the dominant chord changed in the piece; the chord may change but this does not necessarily mean a change in tonality (for instance the chord can change from an A major to an E major chord).

It searches for the dominant chord, K_{max} and detects K_{max} transitions. The transition is weighted with a chord transition certainty, which is calculated by looking at the change in certainty of the two keys in question before and after the transition. Let us define K_i as the maximum certainty chord before the transition and K_{i+1} as the probability of this chord after the transition. Likewise L_i is defined as the certainty of the new chord before the transition and L_{i+1} as the certainty after it. The transition is weighted by the factor $(K_i - L_i) + (L_{i+1} - K_{i+1})$. Again, where a transition does not occur, the differential will be 0. This feature is averaged over all time intervals, n.

Because these features weight the transitions with a certainty that the chord change has happened, it gives greater emphasis to clearer transitions, thus filtering out uncertain transitions.

2.1.4 Testing of tonality features

The Magnatagatune dataset contained few tags on the relaxing-exciting axis. 150 production music clips tagged with exciting and relaxing in the BBC Archive by expert archivists were used as a ground truth dataset instead of Magnatagatune and although the dataset contained the occasional contentious tag, combining the differential features with weighted tonality, *mirrms* and *mirlowenergy* increased successful classification (by ~10%) and resulted in the correct classification of 37 out of 50 tracks on the re-

laxing-exciting axis (with the 100 remaining tracks used for training) which is comparable with other success rates [4] and was considered to be a solid basis for the full investigation.

2.2 Tempo

The final feature developed previous attempts to determine the tempo of music. The extraction of tempo is desirable because it correlates with mood scales such as exciting-relaxing. The field of beat extraction is a well-developed one, with a number of beat extractors competently able to extract the key beats at the last ISMIR conference. However, this is distinct from tempo, which is more subtle feature of the fundamental frequency and pace of the music. Beat extractors such as *beatroot* [13] and existing tempo extractors such as *mirtempo* often overestimate the tempo because they count the half or third beat (depending on the nature of the music), especially in pieces where instruments with high frequency transients such as percussion exist. Consequently, whilst being able to detect pieces of music with high tempo is relatively straightforward, pieces with low tempo are often labelled with twice or three times the actual value.

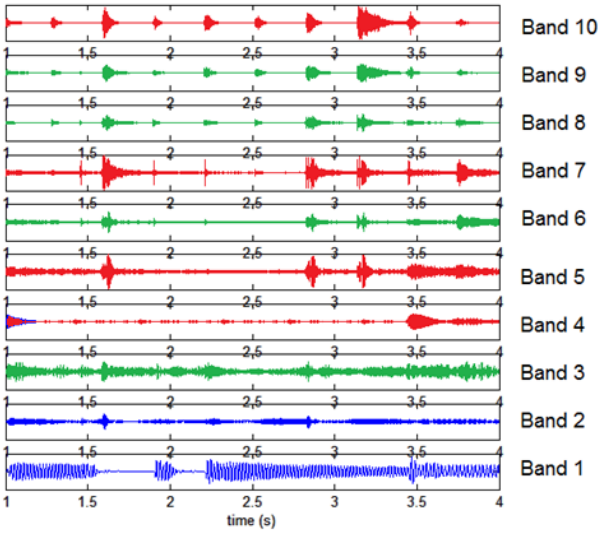

Figure 2. The filtered waveform of the BBC television theme tune *Eastenders*. Green waveforms (bands 3, 6, 8 & 9) indicate the bands in which an autocorrelation of onsets returns the musically correct tempo, red waveforms (bands 4,5,7 & 10) indicate where the function returns double the tempo. Blue (bands 1 &2) waveforms give neither.

Figure 2 illustrates a theme tune filtered into ten, roughly logarithmically equal frequency bands which roughly correspond to octaves using *mirfilterbank*. As can be seen, when this is done, the beat is clearly visible in certain bands, but the band in which they occur is not necessarily the same each time.

Tempo calculations were carried out on the filtered waveforms in figure 2 using *mirtempo*. The function *mirtempo* calculates the tempo by picking the highest peak in the autocorrelation function of onset detection. The green bands (3,6,8 & 9) indicate where the tempo was correctly identified, the red (4,5,7 & 10) where a tempo twice that of the correct tempo was calculated. Note that in no instance has a tempo half that of the correct tempo been found and that the correct beat can be clearly identified in the green waveforms.

The solution is to apply the mirtempo function to each of the ten filtered waveforms. The modal tempo is found and grouped into clusters. The standard deviation of the beats per minute inside each cluster is also noted to give a measure of how precise the extracted tempo is. It is therefore possible to return an unspecified tempo should this value go above a certain threshold.

When the data is clustered, the largest cluster (or mode) is found. The software then searches for a tempo within 15% of half the value of the mode. If this exists, the slower tempo is chosen as the correct tempo. The software then searches for a tempo within 15% of a third of the value of the mode. Again, if this exists, this slower tempo is chosen as the correct tempo. If neither a half nor a third tempo is found, the mean value of the modal cluster is chosen as the tempo.

In all cases tested so far, this has correctly identified the tempi of forty pieces of theme music. This is probably because the nature of the way in which the tempo is determined using the *mirtempo* function means that the tempo is always going to be over-estimated rather than underestimated and because tempo is a feature of the fundamental beat and pace of the music. Other extractors, such as *mirtempo* alone and *beatroot* only achieved success rates of 60-70% on the same theme music. The tempo extractor in this paper does not work quite as well for pieces with unusual time signatures such as 5/4 or 7/4, but these are not commonly used in theme tunes.

The features developed above complemented existing simpler features. Therefore, for each of the 144 theme tunes in the Musical Moods dataset, the following seven audio features were extracted: *mirrms*, *mirlowenergy*, *mircentroid*, weighted tonality, weighted tonality differential, weighted chord differential and tempo.

A mean score for each mood scale for each track in the Musical Moods data was calculated from the subjective testing and then normalized so that the lowest score was 0.5 and

the highest score 5.49. The means were then rounded to the nearest integer, giving a score 1-5. This aligns the data with the original scale with each number referenced to a tag (e.g. on the happy-sad scale, 1 would be associated with *very happy* whilst 4 would be associated with *quite sad*). Each integer was separated by an SVM classifier and trained as indicated in table 1.

Logic score	0	1
SVM1	1	2-5
SVM2	1,2	3-5
SVM 3	1-3	4,5
SVM 4	1-4	5

Table 1. A summary of how each SVM separated the mean mood scores.

To recover the mood score the classifications are summed together and one added as in equation 2:

$$\mu = C_1 + C_2 + C_3 + C_4 + 1 \quad (2)$$

Where:
C_1 = the classification of SVM1,
C_2 = the classification of SVM2,
C_3 = the classification of SVM3,
C_4 = the classification of SVM4,
μ = is the mood classification score.

Taking the example above, four classifications from each of the four SVMs of 1 1 1 0 would mean 3×1+1 = 4. Occasionally one would obtain a spurious result such as 1010. The same equation is applied in this instance and the track would therefore classify as a 3.

The dataset was randomly split into two, the first 94 tracks were used to train the SVMs, the final 50 to test the SVMs. The program optimized the classifier by choosing from five possible SVM kernels: linear, quadratic, cubic, Gaussian radial basis functions and multi-layer perception using the bioinformatics toolbox in Matlab for all of the possible 255 combinations of the 7 audio features.

Three methods for determining the best combination of features and SVM kernel were found. *A*, is the percentage of time that the classifier correctly identifies the correct mood score. *B*, is the average classification success rate for all four SVMs, *C*, is the percentage of time the classifier correctly identifies the correct score or classifies with the nearest integer (i.e. if the correct mood score of a track is 3 and the SVMs classify it as 2, 3 or 4, this would still count as a positive result towards *C* whereas a classification of 1 or 5 would not).

3. RESULTS

The aim of this section is to determine whether mood can be quantified for television theme tunes. The results for the above three methods are shown in table 2.

Mood Scale	*A*	*B*	*C*
Dramatic-calm	40%	85%	94%
Happy-sad	44%	84%	88%
Light-heavy	30%	79%	82%
Masculine-feminine	32%	80%	84%
Playful-serious	48%	81%	80%
Relaxing-exciting	36%	82%	88%

Table 2. The results obtained for each optimized feature.

The testing and training sets were then swapped and the same calculations carried out. On all mood axes *A*, *B* & *C* varied by an average of 2% with a close match in the audio features chosen. The use of weighted tonality, the differentials and the tempo extractor increase the successful classification percentages *B* & *C* by an average of ~20%. Table 3 uses the data in table 2 and gives the root mean square error with respect to a baseline in which each track is tagged with a score of 3 for each mood. All except the light-heavy scale show a marked improvement on the baseline. Much of the dataset results contained scores of 2, 3 or 4, and in general table 3 indicates the feasibility of quantifying the data by these methods.

Mood Scale	baseline	*A*	*B*	*C*
Dramatic-calm	1.16	0.96	0.99	0.93
Happy-sad	1.44	1.20	1.11	1.11
Light-heavy	1.15	1.19	1.64	1.64
Masculine-feminine	1.15	1.06	1.08	1.07
Playful-serious	1.47	1.22	1.33	1.11
Relaxing-exciting	1.29	1.05	1.09	1.00

Table 3. The root mean square error for *A*, *B* & *C* with a baseline of mood score 3.

The results in table 2 show that measure *A* gave the worst results, which is not entirely unexpected given the subtlety between the classifications. *B* has a higher success rate than *A* because it is a measure of how well the SVMs in table 1 are working which does not necessarily translate into an exact classification. The best success rates are achieved for the measure *C*, but this measure has the widest tolerance. However, what *C* does is to classify the audio so that most tracks are labelled with the correct tag or the one next to it. So, for instance, audio which is tagged as quite sad could actually be tagged as very sad or neither happy nor sad. Figure 3 illustrates how the distribution of

scores changes upon classification of 50 BBC television theme tunes on the dramatic-calm scale. Classifications of 1 or 5 decrease whilst classifications in the middle increase. Whilst the ground truth data is quite flat, classification optimised for *B* and *C* compresses the distribution into a large peak in the middle. This indicates that the algorithms for optimising the SVMs do not adequately account for extremes in mood (which could be explained by the dataset being too small, whereby the number of extreme mood samples is small). Optimising for *A* shows a better match in distribution in this example but this method shows great variation between the different mood axes and results in classifications that are more often wrong than correct.

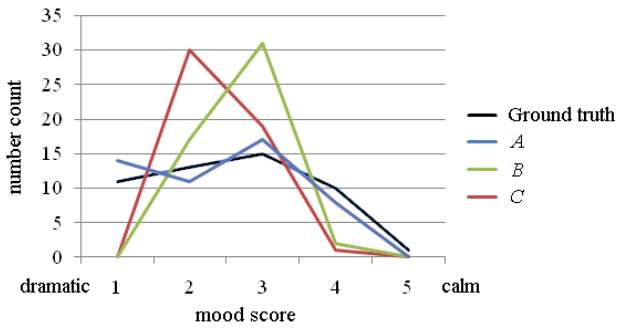

Figure 3. The distribution of mood scores having optimised for *A* (correct score), *B* (average SVM success percentage) and *C* (correct score or nearest integer).

4. CONCLUSION & FUTURE WORK

This is a promising first step towards a scaled classification of television theme music mood. The use of weighted tonality enabled a correct classification of 171 out of 194 tracks (82%) using a further 99 tracks which were labelled with binary tags on the happy-sad axis. The use of weighted tonality differentials resulted in the correct classification of 37 out of 50 tracks on the relaxing-exciting axis (with the 100 remaining tracks used for training). The enhanced tempo extractor correctly identified 40/40 tempi of theme music. The use of the above combined increased the successful classification percentages *B* & *C* by an average of ~20% to accuracies of up to 94% for some mood scales. Improvements need to be able to classify extremes of emotion. The work covered in this paper does not cover how the extracted audio features coincide with each other temporally. For instance, a sudden, very loud minor chord may invoke a more complex mood than the time-averaged moods determined here. Measures of dynamic progression cross-correlated with tonality, spectral centroid and tempo are a possible means to enable a better classification of more complex and stronger emotions over shorter timescales and should be the focus of future work.

5. REFERENCES

[1] O. Lartillot, P. Toiviainen, "A Matlab Toolbox for Musical Feature Extraction from Audio*", Proceedings of the International Conference on Digital Audio Effects, Bordeaux*, 2007.

[2] S. Davies, D. Bland, and R. Grafton, "A Framework for Automatic Mood Classification of TV Programmes," *Proceedings of the 5th International Conference on Semantic and Digital Media Technologies*, Saarbrucken, Germany, 2010.

[3] K. Negus and J. Street. "Introduction to Music and Television," *Special Issue, Popular Music*, No. 21, pp 245-248.

[4] Y.E. Kim et al. "Music Emotion Recognition: A State of the Art Review," *Proc. 11th Intl. Soc. for Music. Inf. Retrieval Conf.*, pp. 255-66, 2010.

[5] I. Steinwart, A. Christmann, *"Support Vector Machines,"* Springer, 2008.

[6] M. Xu et al. M Xu, LY Duan, J Cai, LT Chia, C Xu. "*Advances in Multimedia Information Processing,"* PCM, Springer, 2004,

[7] Anon., *"Rudiments and Theory of Music,"* Associated Board of the Royal Schools of Music, 1958.

[8] S. Davies, T.J. Cox, P. Allen, "Musical Moods: A Mass Participation Experiment for Affective Classification of Music," *Proc. 12th Intl. Soc. for Music. Inf. Retrieval Conf.*, (accepted), 2011.

[9] C. E. Osgood, G. Suci, P. Tannenbaum, "*The measurement of meaning*," University of Illinois Press, Urbana, USA, 1957.

[10] E. Law, K. West, M. Mandel, M. Bay, S. Downie, "Evaluation of algorithms using games: the case of music tagging," *Proc. 11th Intl. Soc. for Music. Inf. Retrieval Conf.*, pp. 387-392, 2009.

[11] O. Lartillot, P. Toiviainen, T. Eerola, *" Studies in Classification, Data Analysis, and Knowledge Organization,"* Springer-Verlag, 2008.

[12] M. Mann, "Processing audio data for producing metadata," *UK Pat. App. P/66699.GB01/IML/kz*, 2011

[13] S. Dixon "Evaluation of the Audio Beat Tracking System BeatRoot," *Journal of New Music Research*, Vol. 36, No. 1, pp. 39-50, 2007.