Multimodal Music Mood Classification using Audio and Lyrics

Cyril Laurier
Music Technology Group
Universitat Pompeu Fabra
c/ Ocata 1, 08003 Barcelona
cyril.laurier@upf.edu

Jens Grivolla*
Fundació Barcelona Media
Av. Diagonal 177, 08018 Barcelona
jens.grivolla@barcelonamedia.org

Perfecto Herrera
Music Technology Group
Universitat Pompeu Fabra
c/ Ocata 1, 08003 Barcelona
perfecto.herrera@upf.edu

Abstract

In this paper we present a study on music mood classification using audio and lyrics information. The mood of a song is expressed by means of musical features but a relevant part also seems to be conveyed by the lyrics. We evaluate each factor independently and explore the possibility to combine both, using Natural Language Processing and Music Information Retrieval techniques. We show that standard distance-based methods and Latent Semantic Analysis are able to classify the lyrics significantly better than random, but the performance is still quite inferior to that of audio-based techniques. We then introduce a method based on differences between language models that gives performances closer to audio-based classifiers. Moreover, integrating this in a multimodal system (audio+text) allows an improvement in the overall performance. We demonstrate that lyrics and audio information are complementary, and can be combined to improve a classification system.

1. Introduction

In the past few years, research in Music Information Retrieval has been very active. It has produced automatic classification methods in order to deal with the amount of digital music available. A relatively recent problem is the automatic mood classification of music consisting in a system taking the waveform of a musical piece and outputing text labels describing the mood in the music (as happy, sad, etc...). It has already been demonstrated that audio-based techniques can achieve satisfying results to a certain extent [11, 9, 7, 18, 21]. Using a few simple mood categories and carefully checking for reliable agreements between people, automatic classification based on audio features gives promising results. Psychological studies [1] have shown that part of the semantic information of songs resides exclusively in the lyrics. This means that lyrics can contain relevant emotional information that is not included in the audio. Indeed, Juslin [5] reported that 29% people mentioned the lyrics as a factor of how music expresses emotions, showing the relevance of studying the lyrics in that context.

Our focus is to study the complementarity of the lyrics and the audio information to automatically classify songs by mood. In this paper, we first present different approaches using audio and lyrics separately, and then propose a multimodal classification system integrating the two modalities.

2. Related Work

Although there is some existing work dealing with audio mood classification (e.g. [11, 9, 18, 21]), and also some recent literature about mood detection in text [2, 13], very little has been done so far to address the automatic classification of lyrics according to their mood. We have found no prior articles studying the combination of lyrics and acoustic information for this particular classification purpose. Mahedero et al. [10] reported promising results in using lyrics for thematic categorization suggesting that a mood classification is possible. Neumayer and Rauber [12] have shown the complementarity of audio and lyrics in the context of genre classification, which is also encouraging. Logan et al. [8] have investigated the properties of lyrics using Latent Semantic Analysis. They discovered natural genre clusters and their conclusion was also that lyrics are useful for artist similarity searches but the results were still inferior to those achieved using acoustic similarity techniques. However, they also suggested that both systems could profitably be combined as the errors of each one were different. Finally, studies in cognitive neuropsychology [14] also demonstrated the independence of both sources of information and so the potential complementarity of both melody and lyrics in the case of emotional expression.

[1] initially at the Music Technology Group, Universitat Pompeu Fabra

3. Database

For this study we use a categorical approach to represent the mood. We consider the following categories: happy, sad, angry, relaxed. Because these moods are related to basic emotions from psychological theories (reviewed in [5]) and also because they cover the four parts of the 2D representation from Russell [16] with valence and arousal dimensions. "Happy" and "relaxed" are with positive valence and respectively high and low arousal. "Angry" and "sad" have negative valence and respectively high and low arousal. As we do not want to restrict to exclusive categories, we consider the problem as a binary classification for each mood. One song can be "happy" or "not happy", but also independently "angry" or "not angry" and so on.

Our collection is made of mainstream popular music. We have pre-selected the tracks using last.fm[1] tags. Last.fm is a music recommendation website with a large community that is very active in associating labels (tags) with music they listen to. These labels are then available to all the users. For each mood category we have generated a synonym set using Wordnet[2] and looked for the songs mostly tagged with these terms. We kept only songs having English lyrics and an entry in LyricWiki[3]. Then we asked listeners to validate this selection by mood. We considered a song to be valid if the tag was confirmed by at least one listener, as the pre-selection from last.fm granted that the song was likely to deserve that tag. We included this manual tag confirmation in order to exclude songs that could have gotten the tag by error, to express something else, or by a "following the majority" type of effect. The annotators were exposed to 30 seconds of the songs, first to avoid as much as possible changes in the mood, and then to speed up the annotation process. Therefore they could not listen to the whole lyrics, thus their judgment had to be biased toward an analysis of the audio. This might influence negatively the results if the mood of the lyrics is not coherent with the mood expressed by the music. In many cases both would match, in other cases it would introduce some error in the system. In total, 17 different evaluators participated and an average of 71.3% of the songs originally selected from last.fm were validated. The database is composed of 1000 songs divided between 4 categories of interest plus their complementary categories ("not happy", "not sad", "not angry" and "not relaxed"). We have used an equal distribution of these binary classes.

4. Audio Classification

To classify music by mood we used a state-of-the-art audio classification algorithm in a supervised learning approach. The features and the classifier were selected according to current litterature and the results from the Audio Mood Classification evaluation task held by the Music Information Retrieval Evaluation eXchange (MIREX) [4, 7].

In order to classify the music from acoustical information, we first extracted audio features of different kinds: timbral (for instance MFCC, spectral centroid), rhythmic (for example tempo, onset rate), tonal (like Harmonic Pitch Class Profiles) and temporal descriptors. All these descriptors are standard and derived from state-of-the-art research in Music Information Retrieval [6, 19].

We obtained the results shown in Table 1 using Weka [20] and 10 runs of 10-fold cross-validation. We report here the accuracies obtained using Support Vector Machines (SMO algorithm in Weka), with default parameters, normalizing the features and using a polynomial kernel. We also tried other classifiers (Random Forest or Logistic Regression are shown here for comparison), but found that Support Vection Machines performed better than others.

	SVM	Logistic	RandForest
Angry	**98.1%(3.8)**	95.9%(5.0)	95.4%(4.7)
Happy	**81.5%(11.5)**	74.8%(11.3)	77.7%(12.0)
Sad	**87.7%(11.0)**	85.9%(10.8)	86.2%(10.5)
Relaxed	**91.4%(7.3)**	80.9%(7.0)	91.2%(6.7)
Mean	**89.8%(8.4)**	84.4%(8.5)	87.6%(8.5)

Table 1. Classification accuracy using audio features, for each category against its complementary (with standard deviation)

The performances we obtained using audio-based classifiers are quite satisfying and even exceptional when looking at the "angry" category with 98% using SVM. All four categories reached classification accuracies above 80%, and two categories ("angry" and "relaxed") even above 90%. Even though these results can seem surprisingly high, this is coherent with other similar studies [18]. Moreover as we deal with binary comparisions on a balanced dataset, the random baseline is 50%. Also, the examples are selected and validated only when they clearly belong to the category or its complementary. This can bias the database towards very clear differences and so categories easier to classify.

5. Lyrics classification

In addition to the results from the audio analysis, lyrics can provide valuable information about the mood of a song. In this section we report three experiments. In the first one we used similarity between lyrics, feature vectors based on Latent Semantic Analysis dimensional reduction in the second, and in the third we propose a technique to select

[1] http://www.last.fm
[2] http://wordnet.princeton.edu
[3] http://lyricwiki.org

the most discriminative terms looking at the differences between language models.

The first two approaches treat the text in an unsupervised way, the representation in vector space is independent of the categories we are interested in. In the third approach, we use our categories (in a supervised process) to select an appropriate representation of the lyrics before addressing the classification task.

5.1. Experiment 1: Classification based on similarity using Lucene

Our first approach was based on the assumption that songs that are "similar" in a general sense are most likely similar for specific relevant aspects, such as genre, mood, etc.

We defined the similarity between different songs in a way commonly used in document retrieval tasks. The representation of the songs is reduced to a bag of words, i.e. the set of words or terms used in a song as well as their frequency. This is then used, with the help of the Lucene document retrieval system[4], to rank documents by their similarity. The similarity measure used by Lucene essentially corresponds (with some performance tweaks) to the very common vector model of information retrieval [17], with tf.idf weighting in order to attribute more importance to those terms that are frequent in the given song, but less frequent overall in the collection.

The most classic approach for using similarity in a classification setting is the k-NN classifier. Based on a source item (in our case a song) for which the class is unknown, the k most similar items from the annotated collection are retrieved. Each of these provides a class label, and the majority label (the most represented one) is chosen as the predicted class of the source item.

5.1.1. Results

We conducted experiments with varying numbers of similar documents (k) to be taken into account. In general, a low k provides less stability, as the predicted label depends strongly on individual examples from the collection. Large ks on the other hand can mean that examples are taken into account that are not actually very similar (and thus representative) of the one that is to be classified. The optimum depends on the application and the distribution of the datapoints and can not be easily predicted a-priori.

While better than the random baseline for most of the moods (the baseline is 50%), the prediction power of the similarity-based approach for lyrics remains limited, with averaged accuracy around 60% as shown in Table 2. The most predictable category is "angry" and the least predictable is "sad".

	k=3	k=5	k=7	k=9	k=11
Angry	69.5%	67.5%	69.0%	68.5%	67.0%
Happy	55.9%	57.4%	60.9%	64.5%	64.1%
Sad	55.0%	52.8%	58.9%	54.5%	55.0%
Relaxed	61.8%	65.8%	61.0%	59.8%	59.1%
Mean	60.5%	60.9%	62.5%	61.8%	61.3%

Table 2. Classification accuracies using k-NN with a tf.idf-based distance on lyrics for different values of k

5.1.2. Limitations

It is difficult to directly integrate the results from both approaches as similarities for audio and lyrics are calculated in different ways. While on the audio side, the feature vectors can be used with different classification algorithms, this is not as easily the case for the lyrics. The typical sparse vector-of-terms representation of the lyrics generates a very high dimensionality, as the length of the vector is the full size of the vocabulary used in the entire collection. On our relatively small annotated collection the vocabulary size already reached over 7000 words, while more complete collections (e.g. the full LyricWiki) reach vocabulary sizes of several hundred thousand distinct words.

5.2. Experiment 2: Classification using Latent Semantic Analysis (LSA)

One approach to deal with the dimensionality problem is to project the lyrics into a lower-dimensional space that is manageable by generic classifiers. The most common method for this is Latent Semantic Analysis (LSA, [3]) which, similar to approaches like Principal Component Analysis (PCA), projects the data into a space of a given dimensionality, while maintaining a good approximation of the distances between data points.

In combination with tf.idf weighting, LSA allows us to obtain a low-dimensional representation of the data. The resulting dimensions tend to relate to clusters of similar documents, and the most significant terms contributing to those dimensions typically reflect the common vocabulary of groups of semantically related documents.

We conducted experiments to determine the impact of the number of dimensions used in the Latent Semantic Analysis on classification performances. As could be expected, performance (using lyrics alone) is very low for extremely low dimensionality and tends to improve with a greater number of dimensions. The peak performance

[4] http://lucene.apache.org/

(which remains quite moderate) is obtained at different numbers of dimensions for the different categories, in some cases at around 20-30 whereas in others it tends to further improve with a greater number of dimension.

5.2.1. Results

In Table 3 we show the results from this experiment. The accuracies presented here are averaged over the 10 runs of 10-fold cross-validation. The use of LSA does not dramatically improve performance compared to our first experiment, depending on the category it can even be worse. The reduction in dimensionality does, however, provide more flexibility, as different types of classifiers can be used on the resulting representation. The results shown here use a reduction to 30 dimensions.

	SVM	Logistic	RandForest
Angry	**62.1% (9.1)**	62.0% (10.2)	61.3 (11.5)
Happy	**55.2% (10.3)**	54.1% (12.5)	54.8 (10.7)
Sad	**66.4% (9.7)**	65.3% (11.0)	56.7 (12.1)
Relaxed	**57.5% (8.2)**	57.3% (9.1)	56.8 (9.79)
Mean	**61.3% (9.3)**	59.7% (10.7)	57.4% (11.0)

Table 3. Classification accuracies using LSA (30 dimensions) on lyrics (with standard deviation)

If our mood categories, as seems to be the case, do not relate to clusters of songs that would be considered similar according to the metrics used in document retrieval, this severely limits the potential of any approaches that are based on document distances with tf.idf weighting. LSA does not overcome this problem, as the distances between data points in the projected space directly reflect their tf.idf-based distance used as a basis for the transformation.

5.3. Experiment 3: Classification using Language Model Differences (LMD)

While distances between songs based on lyrics cannot separate our mood categories very well, lyrics convey other types of information to be exploited in pursuing their separation according to mood. In order to assess that potential, we analyzed the language models corresponding to the different categories ([15]). Figure 1 shows document frequencies (i.e. the proportion of documents containing a given term) for the 200 most frequent terms in the "angry" category, compared to the frequencies in the "not angry" class (results are similar for the other mood categories). As it can be expected, frequencies for many of the top-ranked terms coincide, as these terms are mainly function words (such as "and", "the", etc.) that are not related to a specific semantic content. However, there are very important differences for quite a number of other terms.

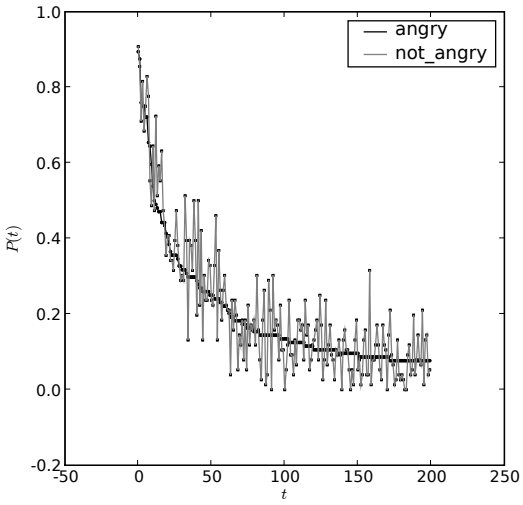

Figure 1. Document frequencies ($P(t)$) of terms in "angry" and "not angry" category where t is the term id.

Due to the very high dimensionality of the language models, some classifiers or feature selection techniques can have difficulties in exploiting this information. We therefore decided to extract a reduced list of relevant terms externally, while using the Weka framework to perform the classification. This is done by comparing the language models generated by the different categories an choosing the most discriminative terms from this comparison.

When comparing two language models, the simplest approach is to calculate the difference in document frequency for all terms. This can be computed either as an absolute difference, or as a relative change in frequency. Both of these, however, have important drawbacks. The absolute difference favors high-frequency terms, even when the relative difference in frequency is not very big. The relative difference on the other hand tends to favor low-frequency terms, especially those that do not occur at all in one of the language models (which results in a difference of 100%).

Example of terms ranked by absolute difference:

- *angry:* world, die, death, control, ...
- *not angry:* me, love, i'm, can, could, so, but, ...

Example of terms ranked by relative difference:

- *angry:* realms, dissolution, bear, four, thirst, perverted, evermore, ...
- *not angry:* chillin, nursery, hanging, scheming, attentive, lace, buddy, sweetest, endings, ...

We are interested in terms with a large relative difference (document frequency in one class being multiple times that in the other class), but that are quite frequent in order to cover a large amount of songs. Therefore, we need to find a measure that provides a good mixture of absolute and relative difference. This also has the effect of providing stable results for the selected top-ranked terms, as their frequency is sufficiently high to reduce to effect of chance variations in occurrence counts.

The measure (3) we settled on is a compromise between absolute difference (1) and relative difference (2).

$$\Delta_{abs}(t) = abs(P(t|LM_1) - P(t|LM_2)) \quad (1)$$

$$\Delta_{rel}(t) = \frac{abs(P(t|LM_1) - P(t|LM_2))}{max(P(t|LM_1), P(t|LM_2))} \quad (2)$$

$$\Delta_{mixed}(t) = \frac{abs(P(t|LM_1) - P(t|LM_2))}{\sqrt{(max(P(t|LM_1), P(t|LM_2)))}} \quad (3)$$

where $P(t|LM_i)$ is the probability of term t occurring in a document represented by the language model LM_i, which is estimated as the document frequency of the term in the corresponding category (normalized by the number of documents).

Using this measure Δ_{mixed} gives us a nice list of terms that cover a good percentage of the songs, with very different distribution between the two categories, and that clearly make sense semantically:

- *angry:* death, control, die, dead, god, evil, hell, world, pain, fate, ...
- *not angry:* love, could, heart, can, i'm, were, blue, today, then, need, ...

5.3.1. Results

For each category, we selected the n terms with the highest Δ_{mixed}. We obtained a vector representation with n dimensions that can be used with different classifiers. We made 10 runs of 10-fold cross-validation (this includes the term selection, of course) and tried different values n. Depending on the categories the accuracy dropped under a certain value of n. For $n = 100$, we had relatively good results with no significant increase by changing its value for any of the categories. Classification performance is significantly better than with the distance based approaches, with accuracies in the 80% range using SVM as shown in Table 4. These results are also closer to those obtained using audio based descriptors. We ran the tests with several other classifiers (decision trees, kNN, logistic regression, random forest ...), some of which obtained good results also, but SVMs performed best overall. We therefore used the SVM classifier with this kind of data for our further experiments.

	SVM	Logistic	RandForest
Angry	**77.9%(10.3)**	60.6%(12.0)	71%(11.5)
Happy	**80.8%(12.1)**	67.5%(13.3)	70.8%(11.4)
Sad	**84.4%(11.2)**	83.9%(7.0)	75.1%(12.9)
Relaxed	**79.7%(9.5)**	71.3%(10.5)	78.0 (9.5)
Mean	**80.7%(10.8)**	70.8%(10.7)	73.7%(11.3)

Table 4. Classification performances using the 100 most discriminant terms, in parenthesis is the standard deviation

6. Combining Audio and Lyrics information

Both audio and lyrics can help in estimating the mood of a song. As these two modalities are quite different and potentially complementary, we chose to combine them in order to create a hybrid classification system.

We used two approaches to integrate these two information sources. The first one used separate predictions for audio and lyrics and combined them through voting. The second approach was to combine all features in the same space, having a vector composed of both audio and lyrics features. This allowed to use audio and lyrics information within one classifier. We only report here on the second approach, which gives slightly better results.

6.1. Mixed Feature Space

Having audio and lyrics information in the same vector allows to exploit interdependencies between aspects from both modalities.

As Table 5 shows, the combination of the language model differences with the audio descriptors yielded to relatively good results. For each category we show the accuracy of the SVM classifier for the audio analysis, for the lyrics analysis, and for the multimodal approach combining both. As in the previous experiments, the accuracies shown in Table 5 are averages over the 10 runs of 10-fold cross-validation.

This combination gives significant improvements over both individual approaches, leveraging the complementary information available from audio and lyrics, at least for two of the four categories: "happy" and "sad" with both a significant ($p < 0.05$ using a Paired T-Test) overall increase around 5% for both. For the angry and relaxed categories there is also a slight increase in classification performance. However, the extremely high baseline of over 98% accuracy on audio alone for the "angry" category, as well as the large difference in performance between lyrics and audio for "relax" limits the benefits of using a hybrid method. We should also notice that the multimodal approach reduces the

	Audio	Lyrics	**Mixed**
Angry	98.1%(3.8)	77.9%(10.3)	**98.3%(3.7)**
Happy	81.5%(11.5)	80.8%(11.2)	**86.8%(10.6)***
Sad	87.7%(11.0)	84.4%(11.2)	**92.8%(8.7)***
Relaxed	91.4%(7.3)	79.7%(9.5)	**91.7%(7.1)**

Table 5. Classification accuracies using audio features, lyrics with language model differences and finally a mixed feature space of both. We used SVM and in parenthesis is the standard deviation. '*' means that the increase compared to the best of the other methods is statistically significant ($p < 0.05$)

standard deviation of the accuracies between folds, which means that the systems are more robust.

7. Discussion and Conclusion

The results obtained with the different methods presented above are very encouraging, and the level of performance is good for many practical applications. This multimodal approach increases the performances for all the mood categories. We note very interesting results particularly for the "happy" and "sad" categories, in which the complementarity of lyrics and audio significantly increases the overall accuracy. Performance using audio purely is already very high for the "angry" category, limiting the potential impact of a multimodal approach. The same is true for the "relaxed" category, to a slightly lesser extent. These results prove that audio and lyrics information combined led to a better music mood classification system.

We should also comment that we have obtained the same trend in our results as Cho and Lee [2] who were working on affect recognition but using a technique based on a manually-built affect lexicon. They reported better results on "happy" and "sad" lyrics than on "violent" (which could be related to our "angry" category). The results we presented here confirm the relevance of the lyrics to convey emotions or at least that the mood expressed in music and acoustical data is correlated with information contained in the text.

In the future it will be interesting to compare our results to other approaches in affect recognition from text, like the methods based on common-sense or affective lexicons [2], and to investigate more advanced multimodal techniques.

8. Acknowledgement

This research has been partially funded by the EU Project Pharos IST-2006-045035.

References

[1] M. Besson, F. Faita, I. Peretz, A.-M. Bonnel, and J. Requin. Singing in the brain: Independence of lyrics and tunes. *Psychological Science*, 9, 1998.

[2] Y. H. Cho and K. J. Lee. Automatic affect recognition using natural language processing techniques and manually built affect lexicon. *IEICE - Transactions on Information and Systems*, E89-D(12), 2006.

[3] S. Deerwester, S. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. *Journal of the American Society for Information Science*, (6).

[4] J. S. Downie. The music information retrieval evaluation exchange (mirex). *D-Lib Magazine*, 12(12), 2006.

[5] P. N. Juslin and P. Laukka. Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening. *Journal of New Music Research*, 33(3), 2004.

[6] O. Lartillot and P. Toiviainen. Mir in matlab (ii): A toolbox for musical feature extraction from audio. In *Proceedings of the International Conference on Music Information Retrieval*, Vienna, Austria, 2007.

[7] C. Laurier and P. Herrera. Audio music mood classification using support vector machine. 2007.

[8] B. Logan, A. Kositsky, and P. Moreno. Semantic analysis of song lyrics. In *Proceedings of IEEE International Conference on Multimedia and Expo*, 2004.

[9] L. Lu, D. Liu, and H.-J. Zhang. *IEEE Transactions on Audio, Speech, and Language Processing*, (1).

[10] J. P. G. Mahedero et al. Natural language processing of lyrics. In *Proceedings of the 13th annual ACM international conference on Multimedia*, New York, NY, USA, 2005.

[11] M. I. Mandel, G. E. Poliner, and D. P. W. Ellis. Support vector machine active learning for music retrieval. *Multimedia Systems*, 12(1), 2006.

[12] R. Neumayer and A. Rauber. Integration of text and audio features for genre classification in music information retrieval. In *Proceedings of the 29th European Conference on Information Retrieval*, Rome, Italy, 2007.

[13] C. Ovesdotter Alm, D. Roth, and R. Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, 2005.

[14] I. Peretz, L. Gagnon, S. Hébert, and J. Macoir. *Music Perception*, (3).

[15] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Research and Development in Information Retrieval*, 1998.

[16] J. A. Russell. *Journal of Personality and Social Psychology*, (6).

[17] G. Salton, editor. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[18] J. Skowronek, M. F. McKinney, and S. van de Par. A demonstrator for automatic music mood estimation. In *Proceedings of the International Conference on Music Information Retrieval*, Vienna, Austria, 2007.

[19] G. Tzanetakis and P. Cook. Marsyas: a framework for audio analysis. *Organised Sound*, 4(3), 1999.

[20] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 1999.

[21] Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. H. Chen. *IEEE Transactions on Audio, Speech, and Language Processing*, (2).