

# Machine Learning Approaches for Mood Classification of Songs toward Music Search Engine

Dang Trung Thanh, Kiyooki Shirai  
Japan Advanced Institute of Science and Technology

## 1 Introduction

Nowadays music is playing a more and more important role in human's life, whereas digital catalogs rapidly become larger and more inconvenient and inefficient to access. If we do not have a good method to explore music, a large amount of music will be fallen into oblivion.

The major way to search music is based on meta-data. Almost existing music-searching systems make use of manually assigned subjective meta-information to index the underlying music collections. Currently, there are three major types of metadata for a song that users can use to explore music stores: artist, genre and album. However moods or emotions of songs are not concerned much in current systems. We can see that human often wants to listen to music that fit best his current emotion. A grasp of emotions in songs might be a great help for us to effectively discover music. In this paper, we aimed at automatically classifying moods of songs based on lyrics and metadata, and proposed several methods for supervised learning of classifiers. In future, we plan to use automatically identified moods of songs as metadata in our music search engine.

## 2 Related Work

Recent developments in music information systems reflect the fact that new and possibly unconventional approaches are necessary to support users in finding desired music. Most of them carry out content-based analyses of the audio files or use collaborative recommendations to point users to music they might like. Knees et al. presents pioneering steps toward the challenging task of automatically building a search system which is capable of finding music that satisfies arbitrary natural language queries [1]. However, it can not consider emotions of songs. Currently, many works are devoted to use acoustical information to detect emotions [2, 3, 4]. However, lyrics, which are very important hints since they clearly show emotions, are not investigated much.

## 3 Mood Categories

In 2007, there is a famous contest about Audio Music Mood Classification (MIREX 07) [8]. The contest concentrated on using audio information to detect mood of music. They use 5 mood clusters in which each mood cluster is represented by a group of words which have close meaning. These mood clusters reduce the diverse mood space into a tangible set of categories, yet root in the social-cultural context of music. Therefore, we use these mood categories for classification. The table 1 shows more details about each mood cluster:

cluster 1 shows an exciting mood; cluster 2 a joyful, gentle mood; cluster 3 a sad and gentle mood; cluster 4 a funny mood and cluster 5 an aggressive mood.

Table 1: Mood Clusters

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Rowdy	Amiable/	Literate	Witty	Volatile
Rousing	Good natured	Wistful	Humorous	Fiery
Confident	Sweet	Bittersweet	Whimsical	Visceral
Boisterous	Fun	Autumnal	Wry	Aggressive
Passionate	Rollicking	Brooding	Campy	Tense/anxious
	Cheerful	Poignant	Quirky	Intense
			Silly	

## 4 Construction of Training Data

In this section, we will describe how to prepare our training data, the collection of songs tagged with their moods. We use a big blog site LiveJournal ([www.livejournal.com](http://www.livejournal.com)) which has more than 9,000 users and each blog entry is tagged with mood and music. Users can choose mood tag from 132 predefined moods of LiveJournal or input freely their current mood. Music tags are inputted arbitrarily by users, so we identify title and artist information by simple string matching with artist database obtained from Open Music website ([www.musicmoz.com](http://www.musicmoz.com)).

The next task is how to map mood keywords posted in LiveJournal to our mood categories. First, 50 most frequent mood keywords are chosen from LiveJournal. Then, we manually design rules to map a mood keyword to mood categories. The mapping rules can be a 1-to-1 or 1-to-many mapping. For example, "sad" mood is mapped to only cluster 3, but "tired" is mapped to both clusters 3 and 5.

In LiveJournal, a song can be tagged with many mood keywords. Thus, we choose the most frequent mood keyword as the unique mood of a song. Then, by using the mapping rules the mood keyword is mapped to the corresponding mood category. In ambiguous cases, the mood of song is determined according to sum of frequencies of mood categories associated with that song.

Finally, we collected about 6,000 songs tagged with mood categories. To check the quality of this dataset, we chose 50 songs randomly and checked mood of each song manually. The mood tags of 92% songs are correct. Furthermore, lyrics of songs in our collection are obtained from LyricWiki website ([www.lyricwiki.org](http://www.lyricwiki.org)).

## 5 Proposed Methods

There are two approaches to solve mood classification problems: using acoustical and verbal information. In this paper, we will concentrate on the latter, especially using lyrics of songs. We present three methods:

SVM classifier, Naive Bayes classifier and graph-based method.

## 5.1 SVM Classifier

SVM [6] classifiers are trained for mood categorization of songs. In this model, each song is represented as a vector with following features and weights.

### 5.1.1 Words Features

All words in a lyric of a song are used as features. Weight of each feature in a vector is defined as (1):

$$weight(w) = tfidf(w) \quad (1)$$

where  $tfidf$  is a product of  $TF$  (Term-Frequency) and  $IDF$  (Inverse-Document Frequency) of word  $w$  in our song collection.

### 5.1.2 Sentiment Words Feature

Although a lyric expresses personal feelings, it contains a lot of short and incomplete sentences. Because of this characteristic, it's hard to detect the mood of a song if we simply apply a traditional method of text categorization problem.

In lyric, sentiment words show feelings clearly, so they are important features in deciding the mood of a song. Beside, we can see that emotional polarity of a sentiment word ( $SW$ ) can be changed by a negation word and be made stronger or weaker by a modifier words ( $MOD$ ). For example, there's a positive meaning in "I love you", but if we use a negation as: "I don't love you", the emotional polarity is changed. We also see that the sentence "I love you very much" is stronger than "I love you". In brief, there are three types of sentiment events by which we can capture the mood of a song: occurrence of sentiment word ( $SW$ ), occurrence of sentiment word with negation ( $NEG-SW$ ) and occurrence of sentiment word with modifier ( $MOD-SW$ ). In the first attempts, we just consider  $SW$  and  $NEG-SW$ .

Sentiment words are collected from SentiWordNet (115,448 words) which is derived from WordNet and each word is assigned with three sentimental factors:  $fsp(SW)$ ,  $fsn(SW)$ ,  $fso(SW)$ . These factors show the level of positive, negative and objective of each word. We consider three features  $\{SW_p, SW_n, SW_o\}$  for each sentiment word  $SW$ . The weighting model is designed as below:

$$weight(SW_p) = (tf(SW) \times fsp(SW) + tf(NEG-SW) \times fsn(SW)) \times idf(SW) \quad (2)$$

$$weight(SW_n) = (tf(SW) \times fsp(SW) + tf(NEG-SW) \times fsn(SW)) \times idf(SW) \quad (3)$$

$$weight(SW_o) = tfidf(SW) \times fso(SW) \quad (4)$$

In (2) and (3),  $tf(NEG-SW)$  is multiplied by  $fsn(SW)$  for  $SW_p$  and  $fsp(SW)$  for  $SW_n$  because  $NEG-SW$  has an opposite polarity.

### 5.1.3 Artist Feature

We can observe that each artist/band often sings songs about a specific topic with a specific mood. For example, Eric Clapton often sings sad songs but Bob Marley likes singing happy songs. Table 2 shows a mood distribution of each artist/band extracted from our dataset.

Table 2: Mood Distribution on Artists

Artist	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
The Beatles	4	24	25	1	2
Metallica	1	4	32	4	10
Green Day	0	9	27	4	2
Evanescence	0	3	30	1	1
Bob Marley	0	13	5	0	1

Table 2 indicates that artists/bands are good features for identifying mood of a song. In the SVM classifier, we assign artist feature of each song with the following weight:

$$weight(a) = \log \frac{N}{N(a)} \quad (5)$$

where  $N$  is the number of songs in training data,  $N(a)$  is the number of songs of artist  $a$ .

### 5.1.4 Weighting by Entropy

In this method, entropy of the distribution probability of moods is integrated into the weighting model as (6):

$$weight(w) = \frac{tf(w)}{H(w)} \quad (6)$$

$$H(w) = - \sum_{m \in M} P(m|w) \log P(m|w) \quad (7)$$

In (7),  $P(m|w)$  is the probability that mood category of a song is  $m$  when its lyric contains the word  $w$ . If the probabilistic distribution  $P(m|w)$  is nearly uniform, i.e.  $H(w)$  is great,  $w$  would be ineffective to classify moods of songs. Thus we put lower weights if  $H(w)$  is great as indicated in (6).

## 5.2 Naive Bayes Classifier

### 5.2.1 Basic Model

In this model, the mood  $m$  for a song  $s$  is chosen such that  $P(m|L)$  is the greatest, where  $L$  stands for a lyric. According to Naive Bayes assumptions:

$$\begin{aligned} m_{select}(s) &= \arg \max_m P(m|L) \\ &= \arg \max_m P(m) \prod_{w \in L} P(w|m)^{TF(w,L)} \end{aligned} \quad (8)$$

where  $TF(w, L)$  is the frequency of term  $w$  in lyric  $L$ .  $P(w_i|m)$  is estimated with the following formula:

$$P(w|m) = \frac{1 + O(w, m)}{|F| + \sum_{w \in F} O(w, m)} \quad (9)$$

where  $O(w, m)$  is co-occurrence frequency of word  $w$  and mood  $m$ ,  $F$  is a set of all features (words) in lyrics in the training data.

### 5.2.2 Weighting for Chorus and Title Part

Each song has a specific structure in which each part plays a different role. A lyric is similarly divided into several parts: title, introduction, verse, chorus, bridge and outro. Among them, the most important parts related with a topic of a song are title and chorus.

**Title:** It describes theme of song shortly.

**Chorus:** The refrain of a song. A verse repeats at least twice with none or little differences between repetitions. Chorus parts tend to explicitly show the main theme of the song.

In a given song, the most important part is chorus which is repeated many times and contains almost meaning and emotion of a song. This is the part which listeners tend to remember. Thus words in the chorus part will be effective features to predict the mood of a song. Beside chorus, the title of a song which gives us the main topic of it also contains some useful mood information.

We consider modifying Naive Bayes classifier to put more weights for words in the CHORUS or TITLE part. In this model,  $P(w|m)$  is estimated as follows:

$$P(w|m) = \frac{1 + O(w, m) \times (1 + vote_w)}{|F| + \sum_{w \in F} O(w, m) \times (1 + vote_w)} \quad (10)$$

$$vote_w = \begin{cases} 0 & \text{if } w \notin CHORUS \wedge w \notin TITLE \\ V_c & \text{if } w \in CHORUS \wedge w \notin TITLE \\ V_t & \text{if } w \notin CHORUS \wedge w \in TITLE \end{cases} \quad (11)$$

where  $V_c$ ,  $V_t$  are parameters to put more weights for terms in CHORUS and TITLE parts. In this paper, these parameters are decided empirically.

### 5.2.3 Artist Feature

As mentioned in 5.1.3, artist is an important feature for mood detection problem. Each artist tends to compose or sing a kind of music. Assume that an artist  $art$  and a lyric  $L$  are independent, we have:

$$m_{select}(s) = \arg \max_{m \in M} P(m|L, art) \quad (12)$$

$$= \arg \max_{m \in M} P(L|m) \times P(m|art) \quad (13)$$

$P(L|m)$  is the same of the second term of (8), estimated by the following formula:

$$P(L|m) = \prod_{w \in L} P(w|m)^{TF(w, L)} \quad (14)$$

while  $P(m|art)$  is estimated as follows:

$$P(m|art) = \begin{cases} \left( \frac{N(art, m)}{N(art)} \right)^{wc} & \text{if } N(art) \geq X \\ \left( \frac{N(gnr, m)}{N(gnr)} \right)^{wc} & \text{if } N(art) < X \end{cases} \quad (15)$$

In (15),  $N(art)$  is the number of songs of the artist  $art$ , while  $N(art, m)$  is the number of songs of  $art$  tagged with mood  $m$ . In case that  $N(art)$  is not great, estimated  $P(m|art)$  might be unreliable. For smoothing, we estimated  $P(m|gnr)$  instead of  $P(m|art)$  when

$N(art)$  is less than a certain threshold  $X$ , where  $gnr$  stands for the genre of a song. We set  $X = 5$ . Finally,  $wc$  is a parameter defining the weight for  $P(m|art)$  compared with  $P(L|m)$ .

### 5.3 Graph-Based Method

In many settings, the usual approach for classification problem does not exploit the available information about relationships between data items. Using the relationship information, we can construct a graph  $G$  in which each data item is a node and each relationship forms an edge between the corresponding nodes. Then the classification problem can be formulated as a graph labeling or coloring problem on such a graph.

In our problem, each node is a song and links are created by artist relationship (two songs are connected if they belong to the same artist). In order to try the first attempt on evaluating affection of graph-based approaches to our mood classification problem, we apply the simplest method of Oh et al. [5]. Firstly, a graph of test data is built based on links of data items. Then, they use Naive Bayes model to classify test data. Finally, in order to consider neighbors' affection to a data item, they classify test data again using the test graph. The classification model for the final phase as follow:

$$C_{select} = \arg \max_C P(C|G, T) \quad (16)$$

$$= \arg \max_C P(T|C)P(C|G) \quad (17)$$

$$= \arg \max_C \prod_{i=1}^{|T|} P(t_i|C)^{N(t_i, d)} \times Neighbor_d(C) \quad (18)$$

where  $C$  stands for a category (mood),  $T$  a document (lyric),  $G$  a graph and  $N(t_i, d)$  frequency of the term  $t_i$  in the node  $d$ .

Neighbor function is calculated from neighbor data items, not from training data. It basically computes the degree to which the current class  $c$  is supported by the neighbors of a node  $d$ .

$$Neighbor_d(C) = \frac{l_d(C)}{l_d} \times w_L \quad (19)$$

Here  $l_d$  and  $l_d(C)$  represent the number of all links from or to  $d$  and the number of neighbors having class label  $c$ , respectively. In addition,  $w_L$  represents the average weight for all the links and indicates to what extent the categories of the neighbor nodes are confident.

Then we propose the new model which is an extension of Oh's method. First, we construct two graphs,  $G_a$  and  $G_g$ . Songs of the same artist are connected in  $G_a$ , while songs in the same genre are connected in  $G_g$ . We primary use  $G_a$ . However, if the number of neighbors is small, the neighbor function would be unreliable. In such cases, we use  $G_g$  where the number of neighbors will be much greater than in  $G_a$ . Neighbor function in our new model is summarized as follows:

$$Neighbor_d(C) = \begin{cases} \left( \frac{l_d(C)}{l_d} \right)^{wc} & \text{in } G_a \text{ if } l_d \geq X \\ \left( \frac{l_d(C)}{l_d} \right)^{wc} & \text{in } G_g \text{ if } l_d < X \end{cases} \quad (20)$$

Here we set  $X$  is 5. Note that the parameter  $w_L$  in (19) is replaced with  $w_c$  in (20), that is, we modified the way to decide the weight for categories in neighbors.

## 6 Evaluation

In this section, we will report the results of experiments to evaluate our methods. First, we define the baseline as the naive system which always selects the most frequent mood (cluster 3 in our training data).

### 6.1 SVM Classifier

We trained the following four SVM classifiers:

- SVM-AW: use all words as features
- SVM-SW: use all words and sentiment words as features
- SVM-AA: use all words and artist as features
- SVM-AE: use all words and entropy weighting model

Table 3 reveals the accuracies of SVM classifiers evaluated by five-fold cross validation as well as the baseline (BL).

Table 3: Accuracies of SVM Classifiers

BL	SVM-AW	SVM-SW	SVM-AA	SVM-AE
54.00%	50.58%	52.73%	52.41%	52.11%

Table 3 shows that artist and sentiment word features are good for mood classification. Accuracy is improved 2% in average on three methods compared with SVM-AW, although they do not outperform the baseline. However, sentiment word features are not as good as we expect. The reason is that number of sentiment words is very few, so this method can not capture moods of songs.

### 6.2 Naive Bayes Classifier

We evaluated following four Naive Bayes classifiers:

- NB: basic model
- NB-C: Naive Bayes with weighting chorus part
- NB-T: Naive Bayes with weighting title part
- NB-A: Naive Bayes with artist features

Parameters  $V_c$ ,  $V_t$  and  $w_c$  are optimized in the validation data, which is mutual exclusive with both test and training data. Table 4 shows the accuracies of Naive Bayes classifiers as well as optimized parameters.

Table 4: Accuracies of Naive Bayes Classifiers

BL	NB	NB-C	NB-T	NB-A
54.00%	53.40%	56.44% ( $V_c=7.2$ )	54.92% ( $V_t=1.6$ )	57.44% ( $w_c=10.5$ )

This result shows that Naive Bayes classifier works better than SVM classifier on mood classification problem. More weighting for words in title and chorus can improve the system. The highest improvement is the method using artist feature NB-A.

## 6.3 Graph-Based Method

We evaluated following three graph-based methods:

- NB: content-based Naive Bayes classifier
- GC-Oh: graph-based method by Oh et al.
- GC-New: graph-based method with our extension of neighbor function ( $w_c = 11$ )

Table 5: Accuracies of Graph-based Method

BL	NB	GC-Oh	GC-New
54.00%	53.40%	53.75%	56.66%

This result shows that relationship information among songs is useful for mood classification. Our result is improved 3.2% to Naive Bayes and 2.9% to GC-Oh in which only artist relationship is used to build graph. If other types of relationship are used, we believe that the result will be much better. Applying more sophisticated graph-based classification methods such as [9, 10] will also improve accuracy.

## 7 Conclusion

The accuracy of mood classification methods is not good enough to apply for a real music search engine system. There are two main reasons: mood is a subjective metadata; lyric is short and contains many metaphors which only human can understand. However the experiments showed that artist, sentiment words, putting more weight for words in chorus and title parts are effective for mood classification. Graph-based method promises a good improvement if we have rich relationship information among songs. In future, we plan to integrate audio information with lyric for further improvement.

## References

- [1] Peter Knees, Tim Pohle, Markus Schedl, Gerhard Widmer. "A Music Search Engine Built upon Audio-based and Web-based Similarity Measures". In SIGIR, 2007.
- [2] T. Li and M. Ogihara. "Content-based music similarity search and emotion detection". 2004
- [3] B. Y. Chua and G. Lu. "Determination of Perceptual Tempo of Music". In CMMR 2004
- [4] D. Liu, L. Lu and H-J. Zang. "Automatic Mood Detection from Acoustic Music Data". 2003
- [5] Oh, S. H. Myaeng, and M.-H. Lee. "A Practical Hypertext Categorization Method using Links and Incrementally Available Class Information". In SIGIR, 264-271, 2000.
- [6] V. Vapnik. "The Nature of Statistical Learning Theory". Springer, New York, 1995.
- [7] A. Klautau. "Speech recognition based on discriminative classifiers". In SBT, 2003.
- [8] <http://www.music-ir.org/mirex/2007>
- [9] R Angelova, G Weikum. "Graph-based Text Classification: Learn from Your Neighbors". In SIGIR, 2006.
- [10] Q. Lu, L. Getoor. "Link-based classification". In ICML, 2003.