Language Feature Mining for Music Emotion Classification via Supervised Learning from Lyrics

Hui He[1,2], Jianming Jin[2], Yuhong Xiong[2], Bo Chen[1], Wu Sun[3], and Ling Zhao[3]

[1] School of Information and Communication Engineering, Bejing University of Posts and Telecommunications, 100876 Beijing, P.R. China
{hh1012,chb615}@gmail.com
[2] HP Labs China, 100022 Beijing, P.R. China
{jian-ming.jin,yuhong.xiong}@hp.com
[3] YY Music Group
{xiaoge.sun,ling.zhao}@yy.com

Abstract. In recent years, efficient and intelligent music information retrieval became very important. One essential aspect of this field is music emotion classification by earning from lyrics. This problem is different from traditional text categorization in that more linguistic or semantic information is required for better emotion analysis. Therefore, we focus on how to extract useful and meaningful language features in this paper. We investigate three kinds of pre-processing methods and a series of language grams having different n-order under the well-known n-gram language model framework to extract more semantic features. Then, we employ three supervised learning methods (Naïve Bayes, maximum entropy classification, and support vector machines) to examine the classification performance. Experimental results show that feature extraction methods improve music emotion classification accuracies. Maximum entropy classification with unigram+bigram+trigram gets best accuracy and it is suitable for music emotion classification.

Keywords: emotion classification, n-grams, Naïve Bayes, Maximum Entropy, Support Vector Machine.

1 Introduction

The rapid growth of the Internet and the advancements of Internet technologies have made it possible for music listeners to have access to a large amount of online music data, including music sound signals, lyrics, biographies, and so on. This raises the question of whether computer programs can enrich the experience of music listeners by enabling the listeners to have access to such a large volume of online music data. Multimedia conferences, e.g. the International Conference on Music Information Retrieval (ISMIR) and Web Delivery of Music (WEBELMUSIC), have a focus on the development of computational techniques for analyzing, summarizing, indexing, and classifying music data.

Traditionally musical information has been retrieved and/or classified based on standard reference information, such as the name of the composer and the title of the work

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 426–435, 2008.

etc. But these are far from satisfactory. Huron points out that since the preeminent functions of music are social and psychological, the most useful characterization would be based on four types of information: the style, emotion, genre, and similarity [1].

The emotional component of music has attracted interest in the Music Information Retrieval (MIR) community, and experiments have been conducted to classify music by mood [2, 3, 4, 5]. Some investigate audio signals and some use lyrics to explore music emotion classification. In this paper, we choose to learn from lyrics for two main reasons. First, it is easy to get lyrics from Internet. Some websites provide free services to search, download and add lyrics. Second, it is much easier to process lyrics than audio signals since the preprocessing of audio signals is more complicated and time-consuming than lyrics. Thus it is more suitable for computing in mobile or MP3, MP4, etc.

However, previous work [6] suggests that lyrics are difficult for natural language processing. First, unlike other text data such as news, emotion words in lyrics have a very small number of occurrences. Instead, a large portion will be devoted to the background. This makes the term absolute frequency information not so useful, or even misleading. Second, because lyrics are free-style, approaches based on word positions also face difficulty. Third, lyrics downloaded from Internet may contain spelling mistakes, because most of lyrics are added by listeners. Therefore, in this paper, we adopt the popular n-gram language models and part of speech (POS) to mine kinds of language features. And then we employ three supervised learning methods to examine the classification performance.

The rest of this paper is organized as follow: In section 2, a briefly review of previous work on music emotion classification is presented. In section 3, how the n-gram based language features are extracted and weighted are described. Supervised learning methods are introduced in section 4. Section 5 presents our experiments and results analysis. Conclusions are derived in section 6.

2 Related Work

Music emotion classification has attracted interest in recent years. In related work, some researchers investigate audio signals and some use lyrics to explore music emotion classification. Thus, in this section, we will take a brief review of some related works.

2.1 Learning from Audio Signals

Relations between musical sounds and their impact on the emotion of the listeners have been studied for decades. The celebrated paper of Hevner [7] studied this relation through experiments in which the listeners are asked to write adjectives that came to their minds as the most descriptive of the music played. The experiments confirmed a hypothesis that music inherently carries emotional meaning.

Li [2, 3, 4] introduced Daubechies Wavelet Coefficient Histograms (DWCH) for music feature extraction to learn from audio signals and conducted a comparative study of sound features and classification algorithms on music emotion classification.

By combining DWCH with timbral features (MFCC and FFT), SVM achieved better performance.

Lu [8] presented a hierarchical framework to automate the task of mood detection from acoustic music data. Three feature sets, including intensity, timbre, and rhythm were extracted to represent the characteristics of a music clip. Preliminary evaluations indicated that the proposed algorithm produced satisfactory results.

2.2 Learning from Lyrics

Most previous work learning from lyrics is on stylometric analysis. Li [4] studied the problem of identifying groups of artists by combining acoustic-based features and lyrics-based features and using a semi-supervised classification algorithm.

Work of music emotion classification via lyrics is rare. Lyrics is a special kind of text, so learning from lyrics can be taken as sentiment classification via texts. Similar work has been reported in [9] and improved work in [10]. In these two papers, the language feature extraction methods and machine learning techniques they used are of great help of our work.

3 Language Feature Extraction

Document representation is an important aspect in traditional topic-based categorization. In normal documents, there are always enough items to form term-vectors, while in lyrics, fewer items are there, especially for emotion words. Thus, we need to apply language model to capture more language features to express these lyrics.

3.1 Language Features

(1) Basic unigram. In topic-based categorization, the standard word-vector or so called bag-of-words is the basic and mostly used representation of document. For example, in a text snippet T consist of n words can be represented as a word set $\{w_1, w_2, \cdots, w_n\}$.

While being put into the n-gram framework, it can be viewed as a basic unigram form, which is a context free document representation model. In such model, each item is assumed to be independent of any other ones, thus the dependencies among items are absolutely ignored.

(2) Conventional n-gram. In a conventional n-gram ($n > 1$) model, the last $n-1$ items are viewed as the history information of the current item. It's obvious that a high order n-gram model can capture some short-distance dependencies by combining n sequential individual terms into a compounded feature, e.g., in a bigram model, sentence T will have a feature set $\{w_1w_2, w_2w_3, \cdots, w_{n-1}w_n\}$.

In this music emotion classification task, it can be noted that in addition to many words with obvious emotion tendencies, there are also a lot of tendentious phrases consist of two or three words. While a unigram model can make use of the individual words, n-gram models are good at taking advantage of those phrases or compounds, which are much more meaningful in emotion analysis.

(3) Part of speech. It is found in previous work that only a few words, whose part-of-speech are adjective, verb, noun, and adverb, can present text sentiment. So, one way of extracting features is to filter lyrics only retain words mentioned above.

3.2 Feature Selection and Weighting

As mentioned above, the feature set may consist of all possible language-grams. The original amount of these features will be too huge for an efficient document analysis system. On the other hand, a great percent of these language-grams are nothing but sequential words, which have neither linguistic structure nor semantic meaning and should be cleared out of our language model. Additionally, different language-grams are of different linguistic functions and semantic meanings, so they could not be viewed equally. Therefore, there are two procedures need to be perform on the original language-grams, one is feature selection, and the other is feature weighting.

Our emphasis was put on finding out what kinds of language features are useful. Thus, we mainly applied a simple feature elimination method that is to abandon those feature appearing fewer times than a threshold.

For feature weighting, we try three feature weighting methods: Boolean value, absolute term frequency and term frequency-inverse document frequency (TFIDF) weighting method. The following equation (1) is TFIDF weighting used in this paper:

$$w(f_k, l_i) = \frac{tfidf(f_k, l_i)}{\sqrt{\sum_{f_k} (tfidf(f_k, l_i))^2}} \tag{1}$$

$$tfidf(f_k, l_i) = \sqrt{tf(f_k, l_i)} \times \log\left(\frac{|D|}{df(f_k)}\right) \tag{2}$$

In equation (2), D is the corpus including all lyrics. $tf(f_k, l_i)$ is frequency of feature f_k occurs in lyrics l_i. $df(f_k)$ is lyrics frequency of feature f_k occurs in whole corpus. Equation (1) is normalization of equation (2).

4 Supervised Learning Methods

Our aim in this work is to examine whether it is helpful in music emotion classification by feature extraction from lyrics. We employ three standard algorithms: Naïve Bayes classification, maximum entropy classification, and support vector machines. The main ideas behind these three algorithms are quite different, but each has been shown to be effective in previous text categorization.

We use the following standard bag-of-features framework. Let $\{f_1, f_2, \cdots, f_m\}$ be a predefined set of m features that can appear in a lyrics. A feature can be a word or bigram etc. Let $n_i(l)$ be the number of times f_i occurs in lyrics l. Then, each lyrics l is represented by the lyric vector $\vec{l} = weighting(n_1(l), n_2(l), \cdots, n_m(l))$, $weighting()$ is weighting methods mentioned in section 3.2.

4.1 Naïve Bayes

The Naïve Bayes classifier (NB for short) which is based on a simple application of Bayes rule is a simple but effective machine learning algorithm. It performs very well while being applied to text classification [11, 12]. Apply it to music emotion classification with lyrics, it can be presented as:

$$P(c \mid l) = \frac{P(c) \times P(l \mid c)}{P(l)} \tag{3}$$

where c denotes emotion category and l denotes lyrics. $P(c)$ is the prior distribution of a category. NB can be constructed by seeking the optimal category which maximizes the posterior $P(c \mid d)$.

Assume all of the attribute values are independent to the given category label. In addition $P(d)$ is a constant for every emotion category c, we can get:

$$c^* \propto \arg\max_{c \in C} \left\{ P(c) \times \prod_{j=1}^{m} P(f_j \mid c) \right\} \tag{4}$$

where a lyrics l is represented by a vector of m features which are treated as features appearing in the lyrics l, $\vec{l} = weighting(n_1(l), n_2(l), \cdots, n_m(l))$. $P(f_j \mid c)$ stands for the probability that the feature f_j occurs in a category c in training data, and Laplace smoothing method is usually chosen to estimate it to overcome the zero-frequency problem.

4.2 Maximum Entropy

Maximum entropy classification (MaxEnt, or ME, for short) is an alternative technique which has proven effective in a number of natural language processing applications [13]. Its estimate of $P(c \mid l)$ takes the following exponential form:

$$P_{ME}(c \mid l) = \frac{1}{Z(l)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(l, c) \right) \tag{5}$$

where $Z(l)$ is a normalization function. $F_{i,c}$ is a feature/class function for feature f_i and class c, defined as follow:

$$F_{i,c}(d, c') = \begin{cases} 1, & n_i(d) > 0 \quad and \quad c' = c \\ 0, & otherwise \end{cases} \tag{6}$$

Importantly, unlike Naïve Bayes, maximum entropy classification makes no assumptions about the relationships between features, and so might potentially perform better when conditional independence assumptions are not met.

The $\lambda_{i,c}$'s are feature-weight parameters; inspection of the definition of P_{ME} shows that a large $\lambda_{i,c}$ means that f_i is considered a strong indicator for class c. The

parameter values are not set so as to maximize the entropy of the induced distribution (hence the classifier's name) subject to constraint that the expected values of the feature/class functions with respect to the model are equal to their expected values with respect to the training data: the underlying philosophy is that we should choose the model making the fewest assumptions about the data while still remaining consistent with it, which makes intuitive sense.

4.3 Support Vector Machine

The support vector machine (SVM) is a powful supervised learning algorithm developed by Vapnik[14]. It has been successfully applied to text classification and performed very well. In its simplest form, the goal of a linear SVM is to find the hyperplane which can split different category examples by maximizing the distance between the nearest examples to the hyper-plane. Using a kernel function, the nonlinear SVM maps the input variables into a high dimensional space, and linear SVM can be applied in that space. In the binary classification, the corresponding decision function is :

$$F_{i,c}(d,c') = \begin{cases} 1, & n_i(d) > 0 \quad and \quad c' = c \\ 0, & otherwise \end{cases} \tag{7}$$

where K is a kernel function. Linear, polynomial kernel, Gaussian RBF kernel and sigmoid kernel are the typical kernel functions usually used in SVM.

5 Experiments

5.1 Dataset

In this paper, the music emotion classification was based on the lyrics of Chinese pop music, which is provided by YY Music Group. This data set consists of 1,903 songs. 803 songs are labeled with emotion of love, and 1100 songs are labeled as lovelorn songs. We measured the performance of different methods mentioned above using 5-fold cross validations.

5.2 Experiment Scheme

To prepare the lyrics, we automatically segmented these Chinese lyrics. We tried our methods on three kinds of segmented data sets. First data set was just segmented lyrics and no stop lists were used. Second was segmented lyrics having been deleted stop words[1], which were extensively used. Third data set was segmented lyrics with POS, and was filtered by POS and only retained words with POS of adjective, verb, noun, and adverb.

We attempted to model the potentially important contextual features. For this study, we focused on features based on unigrams, bigrams and trigrams. Since a great percent of these language-grams were nothing but sequential words, which had neither linguistic structure nor semantic meaning and would be cleared out of our language model, we ignored features that occurred 5 or fewer times in dataset.

[1] http://bbs.langtech.org.cn/

The implementation we used for maximum entropy classification is modified on the basis of SS Maxent toolkit[2]. We used Chih-Chung Chang's LIBSVM[3] package for training and testing, with all parameters set to their default values. In this paper, we employ linear kernel as kernel function.

5.3 Experiment Results and Analysis

As described in section 3.2, first, we tested the three weighting methods as Boolean, absolute term frequency, and TFIDF. To make a comparision, these methods have been applied on three kinds of datasets and with their unigram, unigram+bigram and unigram+bigram+trigram language feature sets respectively.

Experiment results are shown in Table1~Table3 with three different preprocessing datasets mentioned in experiment scheme. In these tables, the mark "ABS" stands for the absolute term frequency weighting, "BOOL" for Boolean form, and "TFIDF" for term frequency-inverse document frequency weighting method. Results are measured by accuracy.

(1) Different supervised learning algorithms. As a whole, the supervised learning algorithms clearly surpass the random-choice baseline of 50%. Maximum entropy classification and SVM perform much better than Naïve Bayes. In three different preprocessing datasets, accuracies of maximum entropy classification and SVM with same features and weighting methods are nearly the same.

(2) Different preprocessing. As can be seen from Table 1 and Table 2, a little improvement has been made by deleting stop words with same conditions (including same features, weighting methods and supervised learning algorithms). It seems that some stop words are noisy data in music emotion classification. Thus, if a better result is required, an elaborate stop word list should be selected.

Compared with Table 1 and Table 3, accuracies of lyrics with POS filtered processing decline a little with cases of Naïve Bayes and maximum entropy classification. ME with features of unigrams+bigrams+trigrams obtains best accuracy of all, which

Table 1. Average 5-fold cross-validation accuracies (Lyrics segmented only)

Features	Weighting	NB	ME	SVM
Unigrams	ABS	64.37%	86.71%	**86.76%**
Uni+bigrams	ABS	66.74%	87.86%	**88.07%**
Uni+bi+trigrams	ABS	68.94%	87.39%	**87.91%**
Unigrams	BOOL	64.37%	86.71%	**87.49%**
Uni+bigrams	BOOL	66.74%	87.86%	**88.07%**
Uni+bi+trigrams	BOOL	68.94%	87.39%	**88.49%**
Unigrams	TFIDF	58.85%	**89.28%**	88.56%
Uni+bigrams	TFIDF	54.70%	**89.28%**	88.81%
Uni+bi+trigrams	TFIDF	53.86%	**89.65%***	89.28%

[2] Offered by Tsujii laboratory, at University of Tokyo, on http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/maxent/

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

Table 2. Average 5-fold cross-validation accuracies (Lyrics segmented and deleting stopwords)

Features	Weighting	NB	ME	SVM
Unigrams	ABS	68.94%	**87.97%**	86.60%
Uni+bigrams	ABS	70.84%	**88.65%**	87.70%
Uni+bi+trigrams	ABS	70.94%	**88.49%**	88.28%
Unigrams	BOOL	68.94%	**87.97%**	87.81%
Uni+bigrams	BOOL	70.84%	88.65%	**88.75%**
Uni+bi+trigrams	BOOL	70.94%	88.86%	**89.23%**
Unigrams	TFIDF	61.32%	88.49%	**89.02%**
Uni+bigrams	TFIDF	55.33%	89.18%	**89.34%**
Uni+bi+trigrams	TFIDF	54.91%	88.70%	**89.21%**

Table 3. Average 5-fold cross-validation accuracies (Lyrics segmented with POS filtered)

Features	Weighting	NB	ME	SVM
Unigrams	ABS	64.30%	**86.49%**	85.76%
Uni+bigrams	ABS	65.72%	**87.38%**	86.71%
Uni+bi+trigrams	ABS	66.14%	**87.65%**	86.23%
Unigrams	BOOL	64.30%	**86.49%**	86.39%
Uni+bigrams	BOOL	65.72%	87.38%	**88.39%**
Uni+bi+trigrams	BOOL	66.14%	87.65%	**87.81%**
Unigrams	TFIDF	59.99%	88.70%	**89.20%**
Uni+bigrams	TFIDF	55.42%	89.12%	**89.43%**
Uni+bi+trigrams	TFIDF	54.84%	89.17%	**89.33%**

shows that high order of grams captures more semantic features, so best results are achieved. However, SVM with POS filtered increases, especially SVM with features of unigrams+bigrams and TFIDF weighting get the highest accuracy in Table 3. We suppose that the retained words with POS of adjective, verb, noun, and adverb represent main emotion expressed by songs.

(3) Different weighting methods. In general, the performances of TFIDF are obviously better than Boolean weighting and absolute term frequency in the three tables. This may due to that the TFIDF weighting is a discount weighting method between the Boolean weighting and absolute term frequency weighting. It makes the feature space more smoothing between high frequency terms and low frequency terms while still remaining differences among features.

Interestingly, accuracies of Naïve Bayes with TFIDF weighting drop 10%. It is conflicted with traditional topic categorization. We speculate that this perhaps due to topic being conveyed mostly by particular content words that tend to be repeated, but this remains to be verified in music emotion classification with Naïve Bayes.

(4) Different n-gram features. We also studied the use of bigrams and trigrams to capture more context in general. It can be seen from these three tables that all accuracies with same algorithm and weighting methods are improved by adding high order

grams except Naïve Bayes with TFIDF weighting. We suspect that this is because TFIDF weighting is not suitable for Naïve Bayes.

6 Conclusions and Future Work

In this paper, we focus on how to extract useful and meaningful language features. We investigate three kinds of preprocessing methods and a series of language grams having different n-order under the well-known n-gram language model framework. Then we employ three prevail supervised learning methods to examine the classification performance. Experiment results show that feature extraction methods improve music emotion classification accuracies. ME with unigram+bigram+trigram get best accuracy and it is suitable for music emotion classification.

The work reported here is still worth in-depth studying. The n-gram based feature representation is useful, but feature sets contain noise and useless grams. In this paper, we just ignore them by feature frequency. Some supervised feature selection and extraction methods could be adopted to improve accuracy and efficiency.

Acknowledgements

This research is partially supported by the National High-tech Development Plan under Grant No.2007AA01Z417, and the 111 Project under Grant No. B08004. The authors would like to thank YY music group for provision of music emotion corpus, who has spent great efforts in intelligence music service.

References

1. Huron, D.: Perceptual and Cognitive Applications in Music Information Retrieval. In: Proc. Int. Symp. Music Information Retrieval (2000)
2. Li, T., Ogihara, M.: Detecting Emotion in Music. In: Proc. Fifth Int. Symp. Music Information Retrieval (ISMIR 2003), pp. 239–240 (2003)
3. Li, T., Ogihara, M.: Content-based Music Similarity Search and Emotion Detection. In: Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, pp. 705–708 (2004)
4. Li, T., Ogihara, M.: Toward Intelligent Music Information Retrieval. IEEE Transactions on Multimedia 8(3), 564–574 (2006)
5. Skowronek, J., McKinney, M., van de Par, S.: A Demonstrator for Automatic Music Mood Estimation. In: Proc. 8th Int. Symp. Music Information Retrieval (ISMIR 2007) (2007)
6. Scott, S., Matwin, S.: Text Classification Using WordNet Hypernyms. In: COLING-ACL 1998 Workshop, pp. 38–44 (1998)
7. Hevner, K.: Experimental Studies of the Elements of Expression in Music. Amer. J. Psychol. 48, 246–268 (1936)
8. Lu, L., Liu, D., Zhang, H.: Automatic Mood Detection and Tracking of Music Audio Signals. IEEE transactions on audio, speech, and language processing 14(1), 5–18 (2006)
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification Using Machine Learning Techniques. In: Proceedings of the Conference on Empircal Methods in Natural Language Processing, Philadelphia, US, pp. 79–86 (2002)

10. Pang, B., Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cus. In: Proceedings of 42nd Meeting of the Association for Computational Linguistics, Barcelona, ES, pp. 271–278 (2004)
11. Mccallum, A., Nigam, K.: A Comparison of Event Models for Naïve Bayes Text Classification. In: Proceedings of AAAI 1998 Workshop on Learning for Text Categorization, pp. 41–48 (1998)
12. Rennie, J.D.M., Shih, L., Teevan, J., Karger, D.R.: Tackling the Poor Assumption of Naïve Bayes Text Classifiers. In: Proceedings of the 20th International Conference on Machine Learning (ICML 2003) (2003)
13. Berger, A.L., Della Pietra, S.A., Della Pietra, V.J.: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics 22(1), 39–71 (1996)
14. Vapnik, V.: Principles of Risk Minimization for Learning Theory. In: Lippman, D.S., Moody, J.E., Touretzky, D.S. (eds.) Advances in Neural Information Processing Systems, pp. 831–838. Morgan Kaufmann, San Francisco (1992)