These Words are Music to My Ears: Recognizing Music Emotion from Lyrics Using AdaBoost

Dan Su and Pascale Fung
Department of Electronic & Computer Engineering, HKUST, Clear Water Bay, Hong Kong
E-mail: dsu@cse.ust.hk, pascale@ece.ust.hk

***Abstract*—In this paper, we propose using AdaBoost with decision trees to implement music emotion classification (MEC) from song lyrics as a more appropriate alternative to the conventional SVMs. Traditional text categorizations methods using bag-of-words features and machine learning methods such as SVM do not perform well on MEC from lyrics because lyrics tend to be much shorter than other documents. Boosting builds on a lot of weak classifiers to model the presence or absence of salient phrases to make the final classification. Our accuracy reached an average of 74.12% on a dataset consisting of 3766 songs with 14 emotion categories, compared to an average of 69.72% accuracy using the well-known SVM classification, with statistical significant improvement.**

I. Introduction

A computational music emotion classification (MEC) system has the potential to greatly enhance user experience with music, as well as contribute to more effective music data storage and management for music service providers.

Most existing methods for building a computational MEC system are based on the audio content of music [1], [2], [3], [4]. Recently, using lyrics as complementary features to audio signals for MEC have been proposed by many researchers [5], [6], [7], [8]. Current state-of-the-art lyrics-based MEC systems adopt one of two approaches — knowledge-based [9], [10] and statistical [5], [7], [8], [11], [12], [13].

Meyers' Lyricator system provides an emotional score for a song based on its lyrical content, where the overall emotional score is a summation of the emotion values of all the words in the lyrics [9]. The emotion values of words are described by Mehrabian and Thayer's PAD value [14]. This approach requires the prior knowledge of a dictionary of emotion words. If a song lyric does not contain sufficient emotion words, the emotion identification can be wrong.

In order not to rely on a prior knowledge base, researchers proposed statistical methods that use traditional text classification bag-of-words (BOW) features or N-grams with different weighting [5], [7], [11], [12], [13]. In particular, Hu and Downie found that the best performing individual feature types were the BOW features of content words with multiple orders of N-grams [7]. The machine learning classifiers they used were SVM with linear kernel and default parameter settings. This will be used as the baseline for our experiments.

Knoweldge-based methods tend to misclassify lyrics that do not contain the pre-selected semantic words. Statistical methods similar to those used for text categorization, with bag-of-words features like text frequency and inverse document frequency, are also limited in performance as pointed out by [15]. One hypothesis is that, unlike typical documents for text categorization, music lyrics are short.

We analyzed the lyrics of 3766 of mostly English songs from an online music guide found that the average number of words per lyrics is 150, and the average number of unique words per song is about 70, as shown in Fig 1 and 2. This shows that lyrics are shorter than, say, news stories or book chapters, the common target for text categorization.

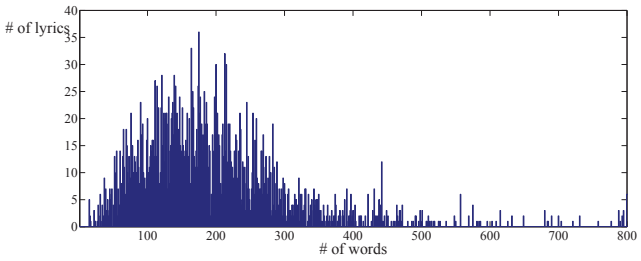

Fig. 1: Average Number of Words in Lyrics is 150

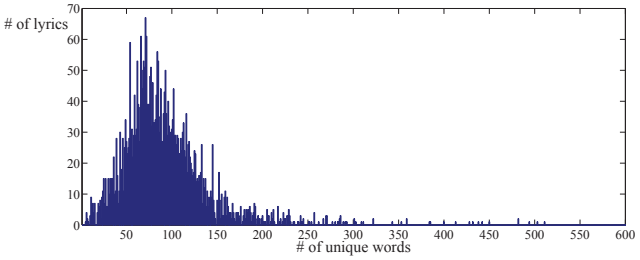

Fig. 2: Average Number of Unique Words in Lyrics is 70

We look to the approach previously used for user intention classification in call-routing applications, such as the AT&T's "How May I Help You?" system where an AdaBoost algorithm with decision stumps as weak classifiers is used to detect the caller intention and routes the call to the appropriate operator [16], [17]. Even though the calls are typically short, it has

been observed that boosting algorithms are able to make use of the presence or absence of a few salient phrases to make the correct classification.

This paper is organized as follows: Section 2 describes the methodology of applying AdaBoost with decision stumps to classify music emotion from lyrics, in Section 3 we show the experimental setups as well as results. Finally, we conclude in Section 4.

II. Methodology: Classifying Music Emotion from Lyrics via AdaBoost with Decision Stumps

AdaBoost is an ensemble machine learning method that combines many weak classifiers linearly, to form a single and accurate classifier. It has been found to work quite well empirically in call routing and facial detection system [16], [17], [18].

Fig 3 shows our proposed framework of training AdaBoost for MEC from lyrics. For each emotion category, an AdaBoost classifier maintains a weight distribution over all lyrics texts in the training set, and is trained in a sequential way by repeatedly calling weak classifiers. At each iteration, a weak classifier is trained based on the training set and the weight distribution. The final classification is made by a linear combination of weak classifiers from each iteration.

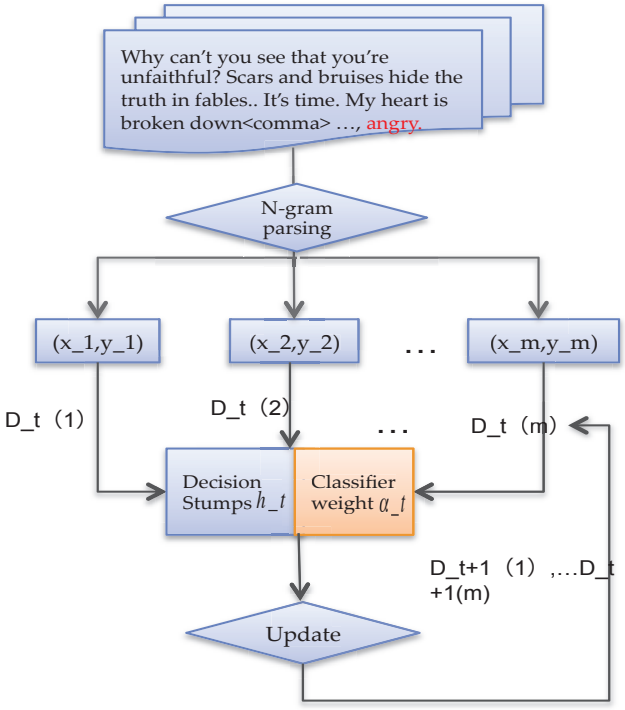

Fig. 3: Training Stage of AdaBoost for MEC from Lyrics

We segment each lyrics text in the training set into n-gram features, using x_i to represent the bag of n-grams for each lyrics text i, and $y_i \in \{+1, -1\}$ to represent the corresponding emotion label, where $+1$ indicate positive emotion in the lyrics text i and -1 negative, for a binary emotion classification task. Each input (x_i, y_i) will be assigned a weight $D_t(i)$ at AdaBoost learning iteration t, the weak classifier h_t and its corresponding weight α_t are trained based on the input feature sets (x_i, y_i), as well as the weight $D_t(i)$ of each lyrics. The nonnegative weights α_t represent how important h_t is for a overall classification.

The weight distribution D_t is initially uniform, at the end of each iteration, it is updated by the following equation (1). The weight of the incorrectly classified lyrics text are increased so that the weak classifier at next iteration $t+1$ will be forced to focus on classifying these particular lyrics.

$$D_{t+1}(i) = \frac{D_t(i)e^{(-\alpha_t y_i h_t(x_i))}}{Z_t} \tag{1}$$

where Z_t is a normalization factor so that $\sum_{i=1}^{m} D_{t+1}(i) = 1$ as befits a distribution.

We choose decision stump as weak classifier h_t, as it has been shown to be successful in a lot of applications when combined with AdaBoost [16], [17], [19]. A decision stump has a basic form of one-level decision tree (stump) using confidence-rated predictions. At each iteration t, the decision stump classifier seeks for a distinguished n-gram stump w from the lyric texts of the training set, and two output values are also trained for each decision stump. At the testing stage, a simple check for the absence or presence of the n-gram stump w is conducted and the corresponding output value will be assigned accordingly. If we use $w \in x$ represent w occuring in input lyrics text x, we can formulate the decision stump classifier in the following form:

$$h_t(x) = \begin{cases} c_0 & \text{if } w \in x \\ c_1 & \text{if } w \notin x \end{cases} \tag{2}$$

where c_j is a real number, indicating the confidence level in assigning the emotion label to x.

Using T to denote the total iteration number, then the final overall classification can be represented as this:

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{3}$$

At the testing stage, for input lyrics x, the sign of $f(x)$ will be the prediction of whether x belongs to the relative emotion categories or not, and the magnitude of the prediction $|f(x)|$ is interpreted as a measure of "confidence" in the prediction.

We use icsiboost [1], an implementation of the AdaBoost.MH algorithm, a member of the boosting family of classifiers [20].

III. Experimental Setups and Results

A. Dataset & Evaluation Measure

Our music lyrics dataset consists of 3766 songs of Western music in 14 emotion categories (see Table I). The emotion labels are collected from an online music guide, which are claimed to be created by experts. The corresponding lyrics of all songs were automatically collected from two websites: LyricsDB and LyricWiki[2].

[1] http://code.google.com/p/icsiboost/

[2] http://lyrics.mirkforce.net/ http://www.lyricwiki.org/

TABLE I: Emotion Categories and Song Distributions of Our Dataset

Emotion	# of songs	Emotion	# of songs
sad	615	high	375
groovy	200	happy	401
lonely	332	sexy	315
energetic	339	romantic	187
angry	154	sleepy	156
nostalgic	131	funny	215
jazzy	54	calm	292

To train a binary classifier for each emotion category, we create each negative sample set by randomly selecting songs from other categories that do not appear in this set. We create positive and negative sample sets in the same size for each emotion category.

We use classification accuracy as the performance measure. For each emotion category, we show the average accuracy over a 10-fold cross validation.

B. Baseline Systems

In order to compare our proposed approach to other lyrics text mining methods for MEC task, we use two baseline systems. The first one is a re-implementation of one of the best performing system in [7], and the second one is re-implementation of one of the best performing systems in lyrics-based MEC among previous work [11], [12], [13], to our best knowledge.

- Baseline 1: Bag of content words (CW) (without stemming, since stemming did not give improvement in the performance) with multiple orders of uni + bi + trigrams, in Boolean representation as features, using SVM with linear kernel as the classifier. SMART stop word list was adopted.
- Baseline 2: Bag-of-words (BOW) with multiple orders of uni + bi + trigrams, in tf*idf representations, with SVM as classifiers.

C. Experimental Results

1) AdaBoost Performance: Table II shows the comparative results of the Baseline 1 and Baseline 2 with the proposed AdaBoost system. We also show results of Baseline 1 using bag-of-words. Our system achieved average accuracy improvements of 4.68% over Baseline 1 and 4.36% over Baseline 2. These accuracy improvements are statistically significant at 99.9% confidence level according to a two-proportion z-test.

2) N-gram Features: We then use different N-gram features in our AdaBoost experiment: word unigrams, unigrams and bigrams, unigrams and bigrams and trigrams. Table III shows the comparative results. Interestingly, unigram features seem to give the best MEC performance.

3) The Most Salient Words for Each Emotion Category Selected by Decision Stumps: Table IV shows the 10 most salient words for each emotion category, each word is the stump of one decision stump classifier. These salient words (stumps) are ranked according to the corresponding decision stump confidence value c_0.

From Table IV we can see that the AdaBoost algorithm with decision stumps not only selects words that match well to intuitive understanding for each emotion category, such as, "hat", "clouds", "guitar" for romantic emotion, it also selects

TABLE II: AdaBoost Performs Significantly Better than Baseline Systems for MEC from Lyrics (Acc %)

Emotion (# of songs)	Baseline 1 uni+bi+tri (Boolean)		Baseline 2 uni+bi+tri (tf*idf)	AdaBoost uni+bi+tri (Boolean)
	CW	BOW	BOW	BOW
sad(615)	68.13	67.47	68.28	**70.57**
high(375)	68.13	68.66	67.20	**74.74**
groovy(200)	70.25	70.00	70.25	**75.50**
happy(401)	64.96	66.06	65.56	**68.75**
lonely(332)	67.17	67.16	67.01	**70.35**
sexy(315)	68.80	**71.63**	70.84	69.00
energetic(339)	63.41	64.00	63.69	**72.44**
romantic(187)	68.41	69.28	69.80	**73.20**
angry(154)	76.81	78.73	78.71	**81.66**
sleepy(156)	74.17	71.63	70.65	**81.58**
nostalgic(131)	73.27	75.58	76.73	**81.02**
funny(215)	65.84	66.96	65.57	**70.00**
jazzy(54)	72.83	72.33	73.50	**76.00**
calm(292)	67.40	66.55	66.38	**70.35**
average	69.26	69.72	69.58	**73.94**

TABLE III: AdaBoost Systems using Different N-gram Features for MEC from Lyrics (Acc %)

Emotion (# of songs)	AdaBoost System		
	uni	uni+bi	uni+bi+tri
sad(615)	69.43	69.67	**70.57**
high(375)	72.80	73.55	**74.74**
groovy(200)	**77.25**	77.00	75.50
happy(401)	68.68	68.17	**68.75**
lonely(332)	70.32	70.18	**70.35**
sexy(315)	**72.14**	70.17	69.00
energetic(339)	70.34	71.85	**72.44**
romantic(187)	**74.81**	72.13	73.20
angry(154)	**82.14**	79.51	81.66
sleepy(156)	77.94	78.35	**81.58**
nostalgic(131)	80.63	80.63	**81.02**
funny(215)	**70.95**	71.84	70.00
jazzy(54)	**76.99**	74.50	76.00
calm(292)	**73.24**	70.51	70.35
average	**74.12**	73.43	73.94

TABLE IV: The 10 Most Positive Terms Related with Each Emotion Category Selected by Decision Stumps

sad	**high**	**groovy**	**happy**	**lonely**	**sexy**	**energetic**
city	alright	guess	walls	across	fine	ran
dead	now	mean	fine	sorrow	please	true
apart	things	calling	chance	strange	watching	looking
all	good	number	clothes	mean	brain	told
things	towards	car	dying	telling	heads	words
steal	hair	else	made	fear	apart	cheap
spin	ever	street	sugar	became	je	blood
arm	few	money	seemed	ones	weren't	c'mom
bones	dear	got	places	flame	car	fed
eyes	streets	feeling	children	slow	breathe	win
calm	**angry**	**sleepy**	**nostaltic**	**funny**	**jazzy**	**romantic**
without	am	sitting	guess	guess	you'll	strange
city	future	trees	across	seem	song	real
further	mean	cold	happy	no	or	wake
rise	I	watching	door	word	you've	across
away	color	am	changed	fine	do	clouds
apart	throat	found	you	isn't	on	you
one's	things	gold	across	down	day	loving
lie	fuck	fingertips	past	get	tide	guitar
beside	me	wake	apart	hair	loved	party
time	you	late	door	those	make	hat

some non-content words, such as "I", "you", "me", to classify angry emotion. It is also interesting to observe the salient presence of the French word "je" in lyrics of the sexy category.

IV. Conclusions

We proposed using AdaBoost with decision stumps to recognize music emotion from lyrics, as a more efficient alternative to the conventional SVM classifiers. The accuracy of our system reached an average of 74.12% on a dataset consisting of 3766 songs with 14 emotion categories, compared to an average accuracy of 69.72% achieved by an implementation of the well-known SVM classification approach, with a statistically significant improvement at 99.9% confidence. Our method is not constrained by the number of emotion categories, and is also language independent.

Acknowledgment

This research is partially supported by the grant RDC R5437 from Velda Limited.

References

[1] T. Li and M. Ogihara, "Detecting emotion in music," in *Proceedings of the International Symposium on Music Information Retrieval, Washington DC, USA*, 2003, pp. 239–240.

[2] Y.H. Yang, C.C. Liu, and H.H. Chen, "Music emotion classification: a fuzzy approach," in *Proceedings of the 14th annual ACM international conference on Multimedia*. ACM, 2006, pp. 81–84.

[3] L. Lu, D. Liu, and H.J. Zhang, "Automatic mood detection and tracking of music audio signals," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 5–18, 2006.

[4] H. Chen and Y. Yang, "Prediction of the distribution of perceived music emotions using discrete samples," *Audio, Speech, and Language Processing, IEEE Transactions on*, , no. 99, pp. 1–1, 2011.

[5] C. Laurier, J. Grivolla, and P. Herrera, "Multimodal music mood classification using audio and lyrics," in *Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on*. IEEE, 2008, pp. 688–693.

[6] B. Schuller, F. Weninger, and J. Dorfner, "Multi-modal non-prototypical music mood analysis in continuous space: Reliability and performances," in *Proceedings 12th International Society for Music Information Retrieval Conference, ISMIR*, 2011, pp. 759–764.

[7] X. Hu and J.S. Downie, "Improving mood classification in music digital libraries by combining lyrics and audio," in *Proceedings of the 10th annual joint conference on Digital libraries*. ACM, 2010, pp. 159–168.

[8] Rada Mihalcea and Carlo Strapparava, "Lyrics, music, and emotions," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea, July 2012, pp. 590–599, Association for Computational Linguistics.

[9] O.C. Meyers, *A MOOD-BASED MUSIC CLASSIFICATION AND EXPLORATION SYSTEM*, Ph.D. thesis, Massachusetts Institute of Technology, 2007.

[10] Y. Hu, X. Chen, and D. Yang, "Lyric-based song emotion detection with affective lexicon and fuzzy clustering method," in *Proceedings of ISMIR*, 2009.

[11] H. He, J. Jin, Y. Xiong, B. Chen, W. Sun, and L. Zhao, "Language feature mining for music emotion classification via supervised learning from lyrics," *Advances in Computation and Intelligence*, pp. 426–435, 2008.

[12] Y.H. Yang, Y.C. Lin, H.T. Cheng, I.B. Liao, Y.C. Ho, and H. Chen, "Toward multi-modal music emotion classification," *Advances in Multimedia Information Processing-PCM 2008*, pp. 70–79, 2008.

[13] M. Van Zaanen and P. Kanters, "Automatic mood classification using tf*idf based on lyrics," *ISMIR2010*, pp. 75–80, 2010.

[14] A. Mehrabian and J.A. Russell, *An approach to environmental psychology.*, the MIT Press, 1974.

[15] D. Yang and W.S. Lee, "Music emotion identification from lyrics," in *Multimedia, 2009. ISM'09. 11th IEEE International Symposium on*. IEEE, 2009, pp. 624–629.

[16] G. Tur, D. Hakkani-Tur, L. Heck, and S. Parthasarathy, "Sentence simplification for spoken language understanding," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5628–5631.

[17] R.E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2, pp. 135–168, 2000.

[18] G.D. Fabbrizio, D. Dutton, N.K. Gupta, B. Hollister, M. Rahim, G. Riccardi, R. Schapire, and J. Schroeter, "AT&T help desk," in *Seventh International Conference on Spoken Language Processing*, 2002.

[19] P. Viola and M.J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[20] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet, "Icsiboost," http://code.google.come/p/icsiboost, 2007.