MULTIMODAL SENTIMENT ANALYSIS ON SONGS USING ENSEMBLE CLASSIFIERS

Draft of April 24, 2015 at 18 : 28

BY

ESTEBAN GOMEZ

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science in Electrical and Computer Engineering
in the Undergraduate College of Engineering of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Adviser:

Minh N. Do

ABSTRACT

We consider the problem of performing sentiment analysis on songs by combining audio and lyrics in a large and varied dataset, using the Million Song Dataset for audio features and the MusicXMatch dataset for lyric information.

The algorithms presented on this paper utilize ensemble classifiers as a method of fusing data vectors from different feature spaces. We find that multimodal classification outperforms using only audio or only lyrics. This paper argues that utilizing signals from different spaces can account for interclass inconsistencies and leverages class-specific performance. The experimental results show that multimodal classification not only improves overall classification, but is also more consistent across different classes.

Keywords: Music Information Retrieval; Sentiment Analysis; Multimodal Classification; Classification Algorithms; Multimodal Fusion

ACKNOWLEDGMENTS

I want to thank Prof. Minh N Do, Ben Chidester and the rest of the Prof. Minh Do's research group for all the guidance and help they have provided through the development of this project.

TABLE OF CONTENTS

1 INTRODUCTION 1

2 PREVIOUS WORK 3
2.1 Audio Sentiment Analysis 3
2.2 Text Sentiment Analysis 3
2.3 Multimodal Classification 4

3 METHOD 6
3.1 Features Used 6
3.2 Classifiers 7
3.3 Fusion Methods 9

4 EXPERIMENT 13

5 RESULTS 15

6 FUTURE WORK 17

REFERENCES 18

1

INTRODUCTION

In recent years, music based services have been trying to make their applications more user-centered. One way to make sure that the user experience is foremost, is to provide services that match the user's current emotional state. This can be implemented by understanding the emotional content of the songs and playlists that are being played.

Sentiment analysis is the portion of Music Informational Retrieval (MIR) where an algorithm recognizes the main emotions that a song evokes. Emotions are subjective, so classifying media into distinct groups is a challenging problem. Most human subjects agree in broad strokes on emotional classifications. However it is not uncommon to find media inputs where there is no consensus, leading to inconsistencies in class groupings. As a result, sentiment analysis is still an open problem that requires the investigation of alternative methods that employ different modalities to counteract class inconsistencies. Success in this approach would allow a more robust classification algorithm that is better suited for real-world applications.

The motivation behind this research is to find a method that accounts for these inter-class inconsistencies across a large dataset. During initial testing it became apparent that certain classifiers perform well on specific sentiments and fail to learn features that represent others. This research seeks to account for this difference by combining the classifiers to output classifications in a more consistent manner, thus improving the reliability of the overall system.

This paper will be focusing on the employment of Support Vector Machines (SVM), Gaussian Naive Bayes and Multinomial Naive Bayes classifiers to recognize emotional information. The features that will be employed are MFCC-like vectors obtained from the response of the song's windowed spectrogram to different base functions to represent the audio portion of the song and a bag of words vector that contains the lyric information of the song. The modal fusion will be tested through both feature fusion and classifier fusion.

The dataset used for this paper is called the Million Song Dataset, and was compiled by Labrosa [1]. This dataset contains a million different songs represented by their pitch, loudness and timbre. The songs are also accompanied by plenty of metadata such as artist, release date and tags. Sentiment classification was obtained from these tags. If a sentiment was used to described a song, it is assumed that the song conveyed that sentiment. The lyric information was obtained from the musicXmatch dataset [2], which provides lyric information out of order in a bag-of-words format. Since there was no way to obtain semantic information from an unordered bag-of-words representation, this research did not focus on the impact of semantics on sentiment classification.

This document starts with a brief overview of previous work performed on this topic, followed by a description of the methods and the algorithms developed, then a section detailing the actual experiments, followed by the results and corresponding analysis and closes with suggestions for future work.

2

PREVIOUS WORK

There has been considerable amount of work done in the field of multimodal sentiment analysis. This section will briefly cover a portion of the relevant research that was considered during the development of the presented methodologies. The relevant topics that were researched for this paper were: Audio Sentiment Analysis, Text Sentiment Analysis and Multimodal Classification.

2.1 Audio Sentiment Analysis

The study of the relationship between emotional content and audio signals is a very mature field. Researchers have expanded on the success found in the speech recognition community while using Mel-Frequency Cepstral Components (MFCC) to explore their uses in music modeling [3]. MFCCs are currently a staple in audio processing and are commonly used in MIR applications such as genre classification [4], since it is a quantifiable method for comparing the timbral texture of songs. Timbre has been used with some success to classify the emotional content of songs [5], however class inconsistencies proved to be a difficult challenge that create substantial misclassification between edge cases. Timbre has also been used to generate songs that evoke particular emotions [6]. These vectors have been commonly classified using Support Vector Machines (SVM) and Naive Bayes classifiers.

2.2 Text Sentiment Analysis

Similarly the study of the relationship between text and emotional content is quite developed. From predicting Yelp ratings based on the sentiment expressed on the review [7] to extracting the emotional progression of major literary pieces [6]. There are many methods to represent and extract emotional information from texts. The Yelp experiment uses statistical word vectors to capture word semantics and emotions as a probability. Other researchers have represented textual information in a bag-of-features framework and classified them using Naive Bayes, SVMs and Maximum Entropy classifiers to recognize positive or negative valances [8].

Capturing the semantic nuisances has also been an area of great interest, which is the study of how a given word might have different emotional value depending on its context. This level of analysis requires the creation of complex sentiment vectors that encode how meanings change based on semantics [9]. Similarly researchers have improved classification accuracy by preprocessing text [10], and use the cleaned data to capture emotional subtleties like the use of negation and modifiers to emotional words [11].

Although there is a great body of research on how to obtain rich sentiment vectors from text, the goal of this paper is demonstrate the added advantage of a multimodal approach. For that end, the lyric vectors were kept simple to clearly underline the benefit of combining them with audio information. In addition, it is necessary to point out that obtaining a large enough dataset of lyrics is difficult due to legal restrictions. As a result, the features used will be the unordered representation of the lyrics in a bag-of-words vector provided by the musicXatch dataset [2].

2.3 Multimodal Classification

Multimodal classification is the task of using feature vectors from different spaces, for example text and audio, to reach a single classification. There are two main methods of combining the information from both vector spaces: feature fusion and classifier fusion [12].

Feature Fusion is the technique that takes signals from different feature spaces and joins them to train a single multimodal classifier. The standard fusion method is called series fusion, which consists on concatenating the vectors together and training the classifier on the union of both spaces. Several alternatives have been suggested to maintain the same amount of expressibility in the vector while keeping the vector space as small as possible. Instead of concatenating the vectors together, it is possible to join vectors in parallel

[13] by making vectors from the linear combinations of a real-valued feature with another complex-valued feature. The benefit of the series fusion over the parallel method is that many diverse features can be fused together to obtain more robust data. As seen in the research by Liang et al., genre classification was improved by joining five different vectors all resulting from different preprocessing methods for text and audio vectors [14].

Classifier fusions consists on training an array of unimodal classifiers and using some function to consolidate the predictions [15]. This method seamlessly fuses features from very different spaces. Caridakis et al. combined facial expressions, body gestures and speech by having a classifier voting system where the class with most votes and higher probability was chosen amongst all the decisions [16]. The final decision-making process can be taken a step further by adding an additional classifier that learns from the decisions provided from classifier array [17]. The algorithms presented on this paper were largely based on this last approach.

Multimodal classification has been successful in improving the accuracy of classification [18] [12]. However, some of the previous work either ran the experiments on highly homogenous datasets where all the music was in the same language, belonged to the same genre or were carefully classified by a single subject thus eliminating class inconsistencies. The goal of this research is to obtain improved classification and reliability from a varied dataset through ensemble classifiers.

3

METHOD

Emotions are highly nuanced since there are many experiences that do not fall neatly within a category. The difference between bittersweet and nostalgic, for example, is very different to quantify. The six Ekman emotions are joy, sadness, anger, fear, disgust and surprise [6]. For this paper we will focus on classifying songs based on the first three emotions (joy/happiness, sadness and anger), mainly because there is not enough sample songs whose main emotions are the last three (fear, disgust and surprise).

3.1 Features Used

The audio features used for the experiment are the timbre data from the EchoNest dataset. These features are zero-mean vectors of length 12, extracted by windowing the spectrogram in constant-width segments. Although the segment width varies from song to song, it is typically under a second long and it is determined so that the timbre and harmony are relatively uniform throughout.

The song is broken up into a series of non-overlapping windows $\mathcal{S}_i$ for the i^{th} segment. The images in Figure 1 are a visual representation of the basis functions g_j that were applied to each segment. An audio feature f is extracted from each segment and has the form of a vector of length 12, where each element of the vector is defined as follows:

$$f_j(x) = \sum_{p \in \mathcal{S}_i} g_j(p) \bullet x(p) \qquad j \in \{1, 2, ..., 12\}$$

The lyric information was obtained from the musicXmatch database [2] that provides the songs in a bag of words format. This format consists of a vector of length 5000 representing a stemmed dictionary where the value at each element is the number of times that a particular word is used in the song. Data is distributed in this form so that it respects artists' rights over the ordering of the words while still being able to represent the content.

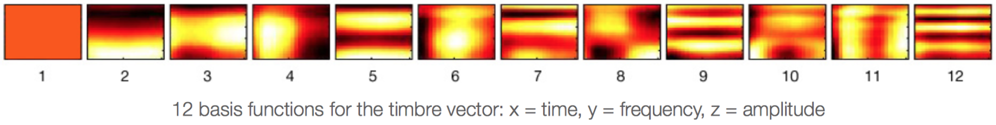
1
2
3
4
5
6
7
8
9
10
11
12
12 basis functions for the timbre vector: x = time, y = frequency, z = amplitude

Figure 3.1: EchoNest's basis functions used to generate MFCC like vectors.[1]

3.2 Classifiers

As mentioned in the Chapter 2, similar problems have been address with certain amount of success using Naive Bayes and SVMs. For the experiment the classifiers selected were the Gaussian Naive Bayes, Multinomial Naive Bayes and Kernel SVM with histogram intersection as the kernel.

3.2.1 Naive Bayes

The Naive Bayes classifier is a statistical classifier that selects the most-likely class given a particular data point. The classifier builds a statistical model for each class that follows a given distribution. Under the assumption that each feature is statistically independent, it computes the conditional probability for the input vector given each model and returns the model with the highest probability.

$$\hat{y} = \text{argmax}_y P(y)\Pi_{i=1}^n P(x_i|y)$$

The distinction between the Gaussian Naive Bayes and the Multinomial Naive Bayes is the distribution that is assumed for the conditional probability $P(x_i|y)$. The Gaussian Naive Bayes classifier, as the name suggests, assumes that data is normally distributed and that the covariance matrix are diagonal. It is a good starting point when little information is known about the data.

[1] `http://developer.echonest.com/docs/v4/_static/AnalyzeDocumentation.pdf`

The Multinomial Naive Bayes supposes that data follows a multinomial distribution, which is a generalization of the binomial distribution. The difference between binomial and multinomial is that in binomial the outcome of each of the k trials is either "yes" with a probability of p or "no" with a probability of $(1 - p)$. Under the multinomial distribution, the outcome of each of the k trials can be one of n different classes such that $\sum_{i=1}^{n} p_i = 1$. Since an event cannot occur a negative number of times, the multinomial distributions is well suited for problems where the vectors are non-negative such as for histogram analysis.

3.2.2 Support Vector Machines

Support Vector Machines are classifiers that construct hyperplanes that maximize the margin to the labeled data points. The margin is defined to be the shortest distance from hyperplane to any of the points. As a result a linear SVM finds a function that linearly separates the data points into clusters for classification. Soft SVMs exist that tolerate data that is not linearly separable by allowing some misclassification.

An SVM can be non-linear if the data points are preprocessed non-linearly before creating the hyperplane. Non-Linear SVMs rely on functions called kernels that map data points into a higher dimensional space where the points can be separated by a single hyperplane.

For the experiments detailed on this paper, we used a non-linear SVM with a histogram intersection kernel. Histogram intersection is a method that finds the minimum intersection between two histograms [19], which means that it is generally more accurate similarity metric than euclidean distances for histograms. The histogram intersection kernel is defined by the function below:

$$k_{HI}(h_a, h_b) = \sum_{i=1}^{n} min(h_a(i), h_b(i))$$

The formula above details the metric used to compare the intersection of two histograms. The distance of a particular pair of histograms is the sum of minimum value of each column of the histogram. By computing the distance between every pair of histograms using this metric, the SVM can easily group similar data points together.

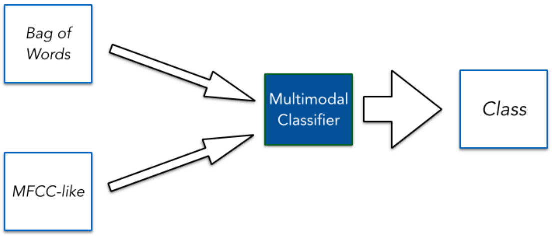

Figure 3.2: Series Fusion

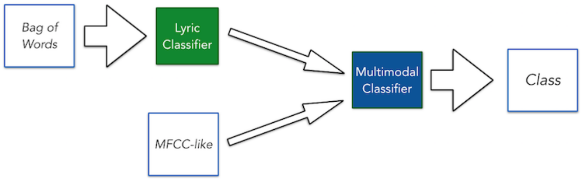

Figure 3.3: Lyrics Only Partial Ensemble

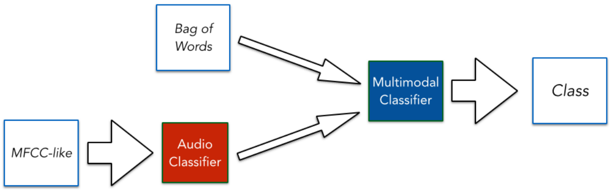

Figure 3.4: Audio Only Partial Ensemble

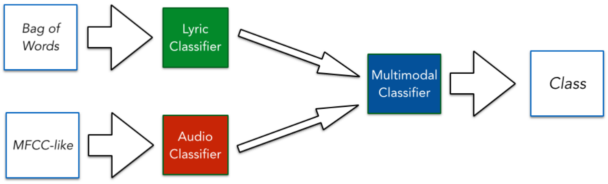

Figure 3.5: Full Ensemble

3.3 Fusion Methods

Using the features and the classifiers detailed above, four different fusion methods were used for classification: Series Fusion, two separate Partial Ensemble and Full Ensemble. These were defined by concatenating the feature vectors at different levels of processing. The figures above show a graphical representation of the different structures implemented.

It is important to note that out of the classifiers detailed above, only Gaussian Naive Bayes is suited to classify negative-valued features. As a result not all the classifiers can be used for all the vectors. The Multinomial Naive Bayes classifier requires that audio segments be normalized to be non-negative to be fitted to the multinomial distribution. We introduce the following classifier sets:

$\mathcal{A} = \text{set of audio classifiers} = \{\text{Gaussian NB}, \text{Multinomial NB}\}$

$\mathcal{L} = \text{set of lyric classifiers} = \{\text{Gaussian NB}, \text{Multinomial NB}, \text{SVM}\}$

$\mathcal{M} = \text{set of multimodal classifiers} = \{\text{Gaussian NB}, \text{Multinomial NB}, \text{SVM}\}$

The set $\mathcal{A}$ is used for audio features, the set $\mathcal{L}$ is used for lyric features and the set $\mathcal{M}$ is used for multimodal features. In the figures above, the squares labeled as Multimodal Classifier, Audio Classifier and Lyric Classifier represent a sampled classifier from the corresponding set.

3.3.1 Series Fusion

The Series fusion method is the basic approach discussed in the Chapter 2 of this document. This method concatenates each raw audio vector with the corresponding raw lyric vector. As a result each song has hundreds of segments, each of which is a vector of length 5012. This large dataset can be computationally prohibitive for certain classifiers. The classifier set for this structure is :

$$\text{Series} = \mathcal{M} \setminus \{SVM\}$$

The reason why SVMs were not included in this test was because the runtime complexity of training an linear SVM is at worst $O(d^2n)$ where d is the number of dimensions and n is the number of samples [20]. With a hundred vectors per song where each vector has 5012 dimensions, it becomes impractical to run this test.

This fusion method involves taking a classifier from the set *Series* and training it on the dataset resulting from the concatenation of the audio vectors and bag of words.

3.3.2 Audio Partial Ensemble

The Audio Partial Ensemble methods come from noting that the complexity of the Series Fusion method stemmed from the fact that the amount of vectors for each of the modes was lopsided. Each song has exactly one bag of words vector of length 5000, while having hundreds of audio segments of length 12. Audio Partial Ensemble explores preprocessing those hundreds of segments and representing them as a single vector that can be combined with the bag of words.

This method initially classifies all the song's audio segments, and builds a histogram for each song containing how many of these segments were classified for each emotion. As a result the audio data goes from being represented by hundreds of vectors of length 12, to a single histogram of length 3 that provides some concrete information of the emotional content of the song.

The classifier set for Audio Partial Ensemble is defined as all the combinations between the Multimodal classifier set and the Audio classifier set. The resulting set is as follows:

$$\text{Audio Partial Ensemble} = \mathcal{A} \times \mathcal{M}$$

For this fusion method a classifier is taken from $\mathcal{A}$, and is used to build the emotional histogram from the audio segments. This histogram is then concatenated to the bag of words to create the multimodal vector, which is used to train the classifier from $\mathcal{M}$.

3.3.3 Lyric Partial Ensemble

The Lyric Partial Ensemble is very similar to the Audio Partial Ensemble, with the modification that the preprocessing is performed to the bag of words vector instead of the audio.

The classifier set for Lyric Partial Ensemble is defined as all the combinations between the Multimodal classifier set and the Lyric classifier set.

$$\text{Lyric Partial Ensemble} = \mathcal{L} \times \mathcal{M}$$

This fusion method takes a classifier from $\mathcal{L}$ and determines the likelihood that each bag of words belong to each class, resulting in a vector of length 3. This likelihood vector is concatenated to the MFCC-like vectors from the audio data to create the multimodal features. A classifier from $\mathcal{M}$ is used to make the final decision based on this vector.

3.3.4 Full Ensemble

Taking both Partial Ensembles to the next logical step, Full Ensemble preprocesses both the audio and the lyric vectors and concatenates the the intermediate results into a single multimodal vector. The audio vectors are converted into the same emotional histogram used in the Audio Partial Ensemble method. The bag of words, like in Lyric Partial Ensemble, is used to create a likelihood vector. The emotional histogram and the bag of words classification are concatenated and used as a single vector for multimodal classification.

The classifier set for Full Ensemble is defined by all combinations between the Audio, Lyric and Multimodal classifier sets. The resulting set is defined as follows:

$$\mathrm{Full} = \mathcal{A} \times \mathcal{L} \times \mathcal{M}$$

4

EXPERIMENT

Given the size of the dataset, it was necessary to perform significant preprocessing for each test to run in a sensible amount of time. The experiments require that every song has audio vectors, a lyric vector and a true emotional label. To get the subset of the Million Song Dataset that had these characteristics, it was necessary to perform two levels of preprocessing. First we needed to identify which songs had sentiment labels. From this subset, we needed to identify which songs were also included in the musciXmatch dataset. As a final step, the dataset was truncated so that each class had the same number of samples to avoid having any misrepresentation.

Song labeling was performed by looking at the mbtags provided in the metadata provided by the MSDS. If the tags contained the word 'happy', 'angry' or 'sad', the corresponding song was considered to be part of that class. This method resulted in three distinct sets. These sets were cross referenced with the musicXmatch dataset to determine which songs contained both an emotion label and a bag of words. Out of the intersection between the two sets, 1000 songs for each sentiment were randomly selected.

The algorithms detailed in the Chapter 3 were implemented on python using the sklearn toolkit[1]. A python module was developed to handle sampling and extracting features and class vectors, training classifiers and creating the emotional histogram. These functions were built into a single module to ensure that all the tests ran on the same code, thus avoiding any issues with accuracy difference due to code conflicts. To streamline the training and testing processes, 100 randomly selected audio segments were sampled for each song. This allowed the test program to quickly identify where the prediction for one song ended and the next one started without much overhead.

The module was then used to implement five different programs: Series Fusion, Partial Ensemble, Full Ensemble, Audio Only and Lyrics Only. It was important that each program could take user input to select which classifier is to be used for what portion of the algorithm. A script was written around these programs to try all classifier combinations as described in Chapter 3.

It is important to mention that the Audio Only benchmark showed that Gaussian Naive Bayes consistently provided better results than the Multinomial Naive Bayes, which allowed us to trim a subset of the combinations used for the Full Ensemble tests by requiring audio preprocessing to be done with a Gaussian Naive Bayes classifier. As a result, we ran six different unimodal benchmarks, nine different configurations for Full Ensemble tests, eighteen different Partial Ensemble tests and two Series fusions, for a total of thirty-five different runs.

[1] The histogram intersection kernel code was obtained from kuantkid's branch of the open source scikit-learn project.
`https://github.com/kuantkid/scikit-learn/commit/16c82d8f2fe763df7bfee9bbcc40016fb84affcf`

5

RESULTS

Mauris diam sapien, consequat non velit at, viverra pretium est. Mauris quis leo est. Praesent dictum posuere accumsan. Suspendisse feugiat metus quis risus ultrices, lobortis aliquam purus faucibus. Curabitur viverra tempus massa vitae blandit. Etiam sodales facilisis ullamcorper. Morbi porttitor consectetur est, sit amet vestibulum turpis euismod sagittis. Morbi nec odio nulla. Nunc mollis eros ut velit aliquam mollis. Pellentesque feugiat suscipit nibh, sed consequat eros venenatis at. Quisque nec risus justo. Sed mauris sapien, tincidunt ut dictum a, accumsan faucibus augue. Suspendisse sollicitudin congue arcu, efficitur vulputate nisi semper vitae. Morbi porttitor sodales lorem ac venenatis. Cras vitae tellus ultrices erat ultrices congue posuere ut velit. Quisque id tortor vel arcu efficitur posuere.

Sed molestie nibh lacinia rutrum mattis. Praesent dignissim eget turpis in porta. Fusce ac enim at augue dignissim rhoncus ac quis elit. Etiam accumsan, magna vitae tristique tincidunt, lectus turpis feugiat magna, sit amet consectetur ante risus non erat. In tincidunt ultrices nisl vel dignissim. Phasellus ipsum leo, feugiat et imperdiet nec, cursus et ex. Duis vehicula, sapien ut congue scelerisque, augue leo dictum ex, eget ultricies urna nisi et diam. Aliquam ullamcorper, tellus aliquet bibendum ullamcorper, neque nisl euismod diam, at facilisis ex nisl in leo. Mauris auctor sodales lorem id ultrices. Nullam mauris ligula, sollicitudin at leo et, eleifend tempus ex. Suspendisse nec blandit risus. Suspendisse sollicitudin ipsum vitae orci pulvinar, convallis ullamcorper orci molestie.

Vestibulum egestas dictum eros eu posuere. Vestibulum quis tincidunt sem. Etiam in facilisis dui. Cras viverra magna eget fringilla semper. Etiam ut orci fringilla, faucibus sem ac, facilisis dolor. Nunc molestie, neque id varius fermentum, ipsum sem tempus neque, quis commodo dui tortor placerat nulla. Morbi posuere nisl vitae posuere varius. Phasellus ac accumsan lectus. Nunc dictum fermentum vehicula. Morbi fermentum, odio id sollicitudin sodales, magna nibh vulputate tortor, eget ultricies lacus sem a sem. Quisque pellentesque id libero ac efficitur. Nulla aliquet facilisis venenatis.

Aliquam blandit cursus faucibus. Integer semper est a laoreet tincidunt. Suspendisse potenti. Quisque eget ullamcorper ligula, porta ullamcorper neque. Suspendisse vel justo arcu. Nam pretium turpis id lacinia ultrices. Phasellus ullamcorper lorem est, quis condimentum sem molestie sit amet. Etiam pellentesque sodales ante, vitae rhoncus nisl ornare non. Etiam nec ante laoreet, maximus lectus ac, tempus orci. Proin efficitur ultricies purus, vel volutpat purus ultrices sed. Fusce quis est ex. Nunc suscipit urna et arcu mattis, id vestibulum purus placerat.

Nulla non interdum nulla, vel convallis neque. Praesent eu sapien id tellus convallis bibendum. Integer quis justo et est lacinia consectetur. Praesent nec tellus vitae lacus convallis egestas. Pellentesque aliquam semper quam ut tempor. Sed posuere condimentum arcu, et vulputate nibh sodales ac. Aliquam imperdiet in nulla ac vulputate. Aenean nibh risus, vulputate non risus quis, dignissim pellentesque velit. Donec mi neque, aliquet ut sapien id, lobortis elementum ipsum. Nulla mattis accumsan eros, convallis faucibus ante congue vitae. Nunc pretium ultrices libero. Maecenas eu felis massa.

Donec in enim eget lacus facilisis suscipit. Vestibulum elementum lorem vitae felis euismod tincidunt. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam nec eleifend tellus, ut venenatis metus. Aenean sit amet nisl pellentesque, iaculis purus vel, ornare lorem. Maecenas efficitur porttitor nibh, non malesuada nunc hendrerit ac. Nunc condimentum, orci at porta aliquam, urna purus rutrum mauris, at mattis lorem erat id enim. Nam auctor lacinia neque. Nullam gravida erat sollicitudin, placerat ante a, tincidunt lorem. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In iaculis ipsum nec velit imperdiet, faucibus commodo ipsum tempus.

6

FUTURE WORK

Keywords: Signal Processing; Sentiment Analysis; Machine Learning; Feature Fusion; Multimodal Classification ; I NEED TO DISCUSS THIS PART

REFERENCES

[1] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[2] musicXMatch, "The musixmatch dataset," 2011. [Online]. Available: http://labrosa.ee.columbia.edu/millionsong/musixmatch

[3] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *In International Symposium on Music Information Retrieval*, 2000.

[4] G. Tzanetakis and G. Essl, "Automatic musical genre classification of audio signals," in *IEEE Transactions on Speech and Audio Processing*, 2001, pp. 293–302.

[5] T. L. University and T. Li, "Detecting emotion in music," 2003.

[6] H. Davis and S. M. Mohammad, "Generating music from literature," presented at the EACL in Gothenburg, 2014.

[7] J. Jong, "Predicting rating with sentiment analysis," 2003.

[8] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, ser. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. [Online]. Available: http://dx.doi.org/10.3115/1118693.1118704 pp. 79–86.

[9] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002491 pp. 142–150.

[10] E. Haddi, X. Liu, and Y. Shi, “The role of text pre-processing in sentiment analysis,” *Procedia Computer Science*, vol. 17, no. 0, pp. 26 – 32, 2013, first International Conference on Information Technology and Quantitative Management. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913001385

[11] Y. Xia, K. fai Wong, L. Wang, and M. Xu, “Sentiment vector space model for lyric-based song sentiment classification.”

[12] J. Zhonga, Y. Chenga, S. Yanga, and L. Wena, “Music sentiment classification integrating audio with lyrics,” 2012.

[13] J. Yang, J.-y. Yang, D. Zhang, and J.-f. Lu, “Feature fusion: parallel strategy vs. serial strategy,” *Pattern Recognition*, vol. 36, no. 6, pp. 1369–1381, 2003.

[14] D. Liang, H. Gu, and B. O?Connor, “Music genre classification with the million song dataset,” *Machine Learning Department, CMU*, 2011.

[15] K. Ahmadian and M. Gavrilova, “A multi-modal approach for high-dimensional feature recognition,” *The Visual Computer*, vol. 29, no. 2, pp. 123–130, 2013.

[16] G. Caridakis, G. Castellano, L. Kessous, A. Raouzaiou, L. Malatesta, S. Asteriadis, and K. Karpouzis, “Multimodal emotion recognition from expressive faces, body gestures and speech,” in *Artificial intelligence and innovations 2007: From theory to applications.* Springer, 2007, pp. 375–388.

[17] S. Li and C. Zong, “Multi-domain adaptation for sentiment classification: Using multiple classifier combining methods,” in *Natural Language Processing and Knowledge Engineering, 2008. NLP-KE’08. International Conference on.* IEEE, 2008, pp. 1–8.

[18] 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.

[19] S. Maji, A. C. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[20] O. Chapelle, “Training a support vector machine in the primal,” *Neural Computation*, vol. 19, no. 5, pp. 1155–1178, 2007.