USING SELF-ORGANIZING MAPS TO CLUSTER MUSIC FILES BASED ON LYRICS AND AUDIO FEATURES

Arnulfo Azcarraga[1] and Calvin Enriquez[1]
[1]De La Salle University

Abstract: Self-Organizing Maps (SOM) are used not just as a clustering technique but mainly as a data visualization tool. A SOM is an unsupervised machine learning algorithm in that the dataset that is used for training does not have category information that accompanies the different training exemplars. Many domains of applications do not have labeled datasets, and this has made the SOM among the most widely used neural network algorithms. Although considered unsupervised, SOMs still need labeled datasets (possibly much smaller than the training set) in order to assign labels to the clusters that emerge in the map after training. In other words, labeling of SOMs is essentially supervised. In this research, we explore the possibility of using SOM on datasets with attributes that are of different nature such as the analog music signals on the one hand and the lyrics of the songs on the other. Experiments with the training of the SOM using analog attributes and lyric attributes individually, simultaneously, and consecutively show that the SOM trained using the analog signals combined with lyrics of the songs produced the best music clusters. Evaluation of the music clusters was done by computing the dispersion of nodes in the map associated with the same music genre.

Key Words: Self-Organizing Maps, Music, Lyrics, Random Projection

1. INTRODUCTION

Music is an integral part of our everyday lives, and with the advent of portable music players with large storage capacities to maintain large archives of songs, users need to be able to organize their archives in a manner that would allow them to retrieve the songs that they want to listen to. It would also be useful for the archive to be able to recommend songs related to the song that has just been played, as a new way of accessing music archives.

As an extension of an earlier work on a 3D music archive using self-organizing maps first studied by Manalili (2011), we study the effect of clustering music files using both audio features and song lyrics. In the earlier work, 68 audio features were used, out of close to 700 initial candidate features that were known in the literature, specifically those used in MusicMiner and jAudio. These same audio features were used in this study, and a subset of the same music collection was used. These audio features refer to the melody, pitch, timbre and other audible features of the song.

The music collection used in I3DMO (Manalili, 2011) had 1,000 songs, representing 10 different music genre: rock, metal, hip-hop, reggae, pop, classical, country, disco, blues, and jazz. Each genre had 10 albums, with 10 songs in each album. For the current study, only 360 songs were used from 4 genres which are *pop, rock, metal,* and *blues.* Only songs that had English lyrics are used – as this is the object of the current study. Since there were exactly 68 features based on audio signals, we sought to also use 68 features for the words in the song lyrics. As there were thousands of stemmed words used in the 360 songs that were part of the dataset, minus all the stop words, we used the Random Projection Method (RPM) (Dy, 2011) in converting the sparse term-frequency input vectors into highly compressed 68-feature vectors. The RPM method thus allowed us to manage the high-dimensionality of the dataset and at the same time convert the input file into an equivalent 68-feature lyric dataset that would be compatible with the 68-feature audio dataset.

The SOMs were trained using different training sets, each defined by the features used. A dataset was produced using just the audio signals, another dataset for the words in the lyrics as features, and a third dataset was built using both audio signals and song lyrics. The obtained clusters, demarcated using k-means clustering of SOM nodes, are compared and analyzed vis a vis the groupings of the songs based on their actual genre.

The rest of the paper is outlined as follows. Section 2 briefly describes the Self-Organizing Maps (SOM). Section 3 discusses the experiments performed on the music dataset. Section 4 discusses the results of the experiments, followed by conclusions and recommendations.

2. SELF-ORGANIZING MAPS

The SOM (Kohonen, 2001) algorithm uses a map that represents the data given by the user. The data consist of attributes and classes which will be used for clustering. Clustering is based on the distance of a node on the map that represents an instance of the data to another node on the map.

For the SOM to work, a map must first be initialized with random weights equal to the number of attributes a single instance in the data would have. These weights will be continuously corrected as the clustering progresses. However there is an alternative way to initialize the weights which is by using values from the instances of the data so that the weights would not be too random.

To start the algorithm, an instance of the data is taken and the attributes are compared to every node in the map using the Euclidean Distance Formula as shown in (Eq. 1). V stands for the vector of attributes in a single instance of a data. W is a vector of weights in a node. n is the total number of weights present in a single instance/node.

$$D = \sqrt{\sum_{i=0}^{i=n} (V_i - W_i)^2} \qquad \textbf{(Eq. 1)}$$

The node that has the smallest difference with the map is then awarded the Best Matching Unit (BMU) status. This means that the surrounding neighbor of the BMU will have their weights corrected. The weights change using a weight update rule as shown in the equation.

$$W_{t+1} = W_t + \alpha(V_t - W_t) \qquad \textbf{(Eq. 2)}$$

In the equation α changes as the number of iterations reach the desired amount of iteration prescribed by the user. α is the learning rate which decreases linearly as iterations are done. The neighborhood size also changes linearly as iterations are done, the neighborhood size is the number of nodes that gets changed every time a Best Matching Unit is chosen.

SOM was chosen for this experiment among other clustering techniques because of the SOM algorithm's intuitiveness compared to other clustering techniques. The SOM can be used even by people who are not experts in the certain domain because of the visualization. SOM can provide information that is not easily seen using other clustering techniques.

The produced SOM was labeled using a method where all the dataset instance were compared to the nodes in the map, the node that has the lowest distance to the dataset instance gets a point for the genre the instance belongs to. In the end the node is labeled as the genre with the most points.

3. EXPERIMENTS

Data used in this research is a subset of music files from I3DMO. I3DMO used 1000 songs, the subset used in this research is only 360 songs from 4 genres: blues, metal, pop, and rock. 68 analog signals and 68 lyric features were used as the attributes of the music files. The analog signals that were used were the 68 analog signals used by I3DMO. The lyric~~s~~ attributes were from the frequency of all the words used in the music files; however, the frequency of words will result to 7107 lyric attributes, this may lead to drowning the analog signal attributes. To address this issue we use a method called the Random Projection Method (RPM) to reduce the dimensionality into the desired lesser dimensions.

To get the analog signal attribute, the songs underwent some pre-processing. The songs were segmented into 10 parts, the first and the last segments were discarded and then the standard deviation of the remaining segments were computed, if the value for the segment is more than or less than 2 standard deviations from the mean the value is discarded and is not used in computing the value for the song.

The analog signals were extracted using jAudio and MusicMiner. Initially there were 296 analog features, these were then reduced to 68 in I3DMO.

To get the lyric attributes the frequency of the words in a song is computed. A matrix is then formed with the words as the attributes. However, this matrix is sparse because some songs don't have intersecting lyrics. This results to a lot of data being zero. In order to reduce to reduce the dimensions, RPM was used.

RPM is a method to project a matrix with high dimensionality into a matrix with a lower dimensionality. The initial parameters needed by RPM are *n*, *m*, and *r*. *n* is the original dimensionality of the matrix. *m* is the target dimension of the reduced matrix. *r* is the number of times a word is repeated from the original matrix to the reduced matrix.

1. For each item in *n*, the value would be put randomly in an *m* x *n* matrix *r* times.
2. Then the randomly assigned values would be added to produce the new value for the attribute.

The final result would be the dataset with *m* attributes. In this experiment the 68 was chosen to be the number attributes so that it would be the same with the analog signals.

The final values for the attributes were further normalized, this was done because the analog signals were normalized to have a range of 0 to 1. This means that the lyric attributes must also be transformed to data with range of 0 to 1. This was done by applying equation 3.

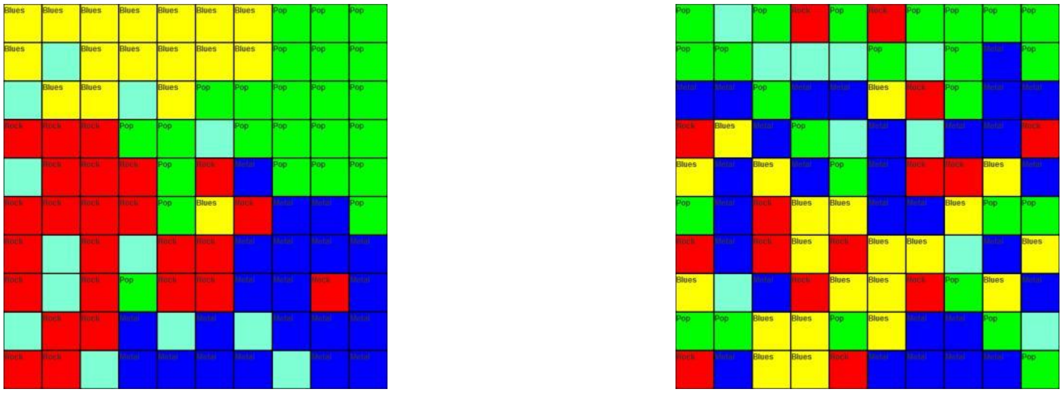
$$val_{new} = \frac{val_{old}}{\max - \min} \qquad \textbf{(Eq. 3)}$$

Maintaining the balance and consistency between the two kinds of attributes is important because if one is greater than the other the SOM might turn out to be more bias towards one kind of feature over the other.

4. RESULTS AND DISCUSSIONS

The following are the SOMs produced by using Analog attributes and Lyric attribute individually as shown in figure 1. The next figure shows the training using both the Analog attribute and the Lyric attribute.

2(a) shows the SOM produced when trained using the attributes simultaneously. 2(b) shows the SOM produced when trained using the Analog attribute and then the Lyric attribute. 2(c) shows the SOM produced when trained using the Lyric attribute and then the Analog attribute. In all the figures the green boxes are the nodes that are *pop*. The yellow nodes are *blues*. The red nodes are *rock*. The blue nodes are *metal*.



Blues	Blues	Blues	Blues	Blues	Blues	Blues	Pop	Pop	Pop
Blues		Blues	Blues	Blues	Blues	Blues	Pop	Pop	Pop
	Blues	Blues		Blues	Pop	Pop	Pop	Pop	Pop
Rock	Rock	Rock	Pop	Pop		Pop	Pop	Pop	Pop
	Rock	Rock	Rock	Pop	Rock	Metal	Pop	Pop	Pop
Rock	Rock	Rock	Rock	Pop	Blues	Rock	Metal	Metal	Pop
Rock		Rock		Rock	Rock	Metal	Metal	Metal	Metal
Rock		Rock	Pop	Rock	Rock	Metal	Metal	Rock	Metal
	Rock	Rock	Metal		Metal		Metal	Metal	Metal
Rock	Rock		Metal	Metal	Metal	Metal		Metal	Metal

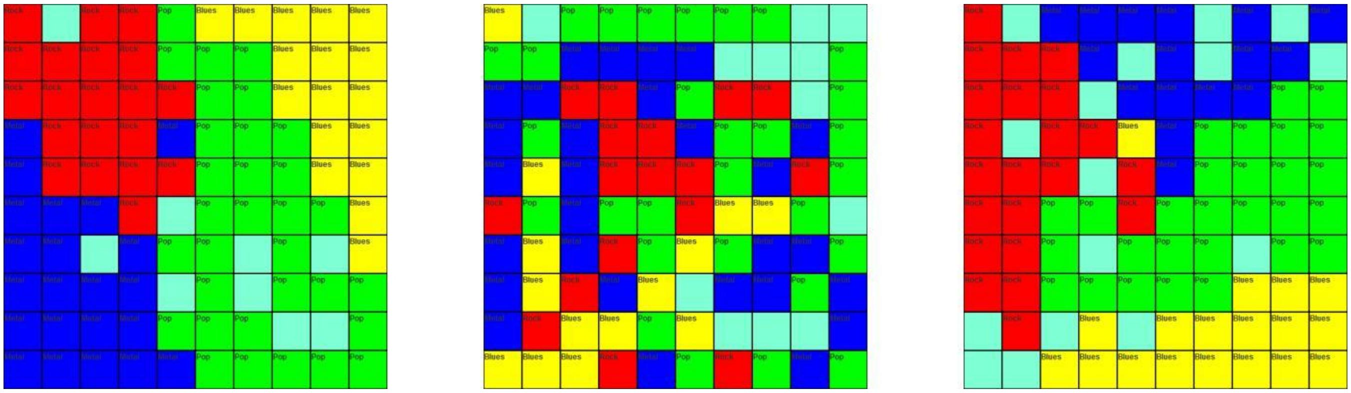
(a)

Pop		Pop	Rock	Pop	Rock	Pop	Pop	Pop	Pop
Pop	Pop				Pop		Pop	Metal	Pop
Metal	Metal	Pop	Metal	Metal	Blues	Rock	Pop	Metal	Metal
Rock	Blues	Metal	Pop		Metal		Metal	Metal	Rock
Blues	Metal	Blues	Metal	Pop	Metal	Rock	Rock	Blues	Metal
Pop	Metal	Rock	Blues	Blues	Metal	Metal	Blues	Pop	Pop
Rock	Metal	Rock	Blues	Rock	Blues	Blues		Metal	Blues
Blues		Metal	Rock	Blues	Blues	Rock	Pop	Blues	Metal
Pop	Pop	Blues	Blues	Pop	Blues	Metal	Metal	Pop	
Rock	Metal	Blues	Blues	Rock	Metal	Metal	Metal	Metal	Pop

(b)

Figure 1. (a) SOM using Analog attributes as input features, (b) SOM using Lyric attributes as input features

As can be seen in figure 1(a), the nodes with the same music genre are geographically clustered with each other. The nodes in the middle of the figure shows some of the music genre can be mistaken for another music genre. The SOM that was trained using Lyric attribute is shown in figure 2(b). As can be seen from the figure, the nodes associated to the different music genre are not geographic located near each other in the map. This is attributed primarily to the fact the some of the words from the lyrics actually do intersect. For example, the word man can actually be used in different genre and with different meanings. This makes the lyric attributes not effective for training the SOM. Upon observation of the SOM produced by the Analog attribute and Lyric attribute, we can see that when Analog attributes are used to train the SOM, it produces a map that shows clusters compared to the SOM trained using the Lyric attribute as input features.



(a) (b) (c)

Figure 2. (a) SOM using combined Analog and Lyric attributes as input features, (b) SOM trained first with Analog attributes then retrained with Lyric attribute as input features, (c) SOM trained first with Lyric attributes then retrained with Analog attribute as input features.

The SOM where both Analog and Lyric attributes are used are shown in figure 3. 3(a) shows the SOM produced when trained using Analog and Lyric attribute simultaneously. 3(b) shows the SOM produced when trained first using Analog attribute then retrained using Lyric attribute. 3(c) shows the SOM produced when trained using Lyric attribute then retrained using Analog attribute. As seen in figure 2(a), the map is clustered where the genres can be differentiated from each other. The difference in position compared to the analog attribute is due to the fact that random weights are assigned initially. However, we can see that each genre actually takes a quadrant of the map. Also shown in the map is that there are less mix genres in the middle, although there is evidence of some mix nodes it is more organized compared to the SOM using only analog attributes. Figure 2(b) shows that retraining the SOM with lyric attributes caused the previously organized SOM from figure 1(a) into a map that has nodes that are not geographically near each other. It seems that the lyric attributes may cause the SOM to disorganize the map. Figure 2(c) shows that retraining using analog attributes always organizes the map in a way that there are distinct clusters for each genre.

Table 1. Dispersion Table

Class	Audio	Lyrics	Combined	Audio then Lyrics	Lyrics then Audio
Rock	2.495519666	3.553870958	1.772465097	2.80249402	2.266408677
Pop	2.316024057	4.280855445	2.812945496	3.752367028	2.483726064
Metal	2.186753858	3.64432557	2.093432943	3.796366873	2.009421813
Blues	2.013799654	2.830459706	1.926438751	2.911594531	2.268212811
Average	2.253024309	3.57737792	2.151320572	3.315705613	2.256942341

Table 1 shows the dispersion of nodes per genre based on figures 1 and 2. Dispersion was computed by taking the location of each node and getting the average of the distances in order to get the centroid for a music genre. The distance of each node associated to the music genre is then computed and averaged in order to get the dispersion value per music genre. This was done in all SOMs that were trained.

Red highlights are the highest per row and green highlights are the lowest per row. As seen in the table training SOM last with audio or combination with audio yields lesser dispersion compared to the ones trained with lyrics last. The lyric attribute might be causing the SOM to be more dispersed because some words can appear in the songs of different genre. These words might not be used with the same meaning but it falls under the same attribute because the only the term frequency is used as attributes.

The result also shows that the combined attribute actually performed best compared to the others. This might be because the audio is a good way to get the dispersion however when lyrics are added some fine tuning in the node may have occurred causing the nodes to be more compact compared to just using audio.

5. CONCLUSION AND RECOMMENDATIONS

Using both Analog and Lyric attribute simultaneously gives a better SOM compared to just using the Analog attributes. Using just the Analog attribute already gives a good SOM however adding the lyrics actually helps in creating better clusters. The Lyric attribute, however, cannot be used alone, the SOM tends to produce a disorganized map when trained using the lyric attribute.

In order to use attributes of different nature, extensive study and attribute extraction must be done. Also, choosing the correct attributes must be done carefully. It seems that there can still be some processing that should be done on the lyric attribute. Using semantics on the words might prove to be useful so that words with different meaning may be considered entries as lyric attributes. It might be to the advantage of the SOM to get attributes that are not common to other clusters as this may cause confusion to the SOM. However, extensive NLP work might be needed for this.

Recommendations when fusing attributes for SOM is to carefully choose the correct attributes and proper pre-processing so that the SOM would be able to use the attributes to their fullest potential. Also another way is to assign weights to the attribute that produces better clusters and give less weights to the attributes that may disorganize the map.

6. REFERENCES

Azcarraga, A., & Manalili, S. (2011). Design of a structured 3D SOM as a music archive, Springer-Verlag Lecture Notes Series: Proceedings of the 8th international conference on advances in self-organizing maps (pp. 188–197).

Azcarraga, A. et al.(2008). Improved SOM Labeling Methodology for Data Mining Applications. Soft Computing for Knowledge Discovery and Data Mining

Dy, J. B. S. (2011). Keyword Extraction for Very High Dimensional Datasets using Random Projection as Key Input Representation Scheme. Master's thesis, De La Salle University - Manila.

Kohonen, T. (2001). Self-Organizing Maps. Berlin: Springer.

Kohonen, T. (2000), Self-organization of a massive document collection, IEEE Transactions on Neural Networks 11(3): 574-585.

Manalili, S. (2010). I3DMO: An Interactive 3D Music Organizer. MS Thesis, College of Computer Studies, De La Salle University, Manila, Philippines.