Characterizing and Classifying Music Subgenres

Adam Lefaivre
University of Lethbridge
Lethbridge, Canada
lefaivre@uleth.ca

John Z. Zhang
University of Lethbridge
Lethbridge, Canada
john.zhang@uleth.ca

***Abstract*—We consider the problem of subgenre classification in music datasets. We propose an adaptation of association analysis, a technique to explore the inherent relationships among data objects in a problem domain, to capture subgenres' characteristics through acoustical features. We further propose to use those characteristics to engage in a pairwise comparison among subgenres when classifying a new music piece. The initial investigation on our approach is examined through empirical experiments on a number of music datasets. The results are presented and discussed, with various related issues addressed.**

I. Introduction

Large quantities of digitized music data are now prevalent in massive online repositories and further used by streaming services. The main purpose of *Music Information Retrieval* (*MIR*) is to develop algorithmic solutions to ease music data curation tasks, and ultimately, to improve the experiences of listeners. Core tasks in MIR include: *automatic tag annotation*, *playlist recommendation*, *genre classification*, etc. [1].

It is obvious that many approaches in the MIR literature have discussed successful solutions to the genre classification problem. However the classification of music into subgenres is often neglected. It would be desirable to classify music into the finest category possible. Intuitively, this is the next step for successful genre classifications. In a practical sense, listeners may certainly spend more time listening to a particular subgenre than any other subgenres, or even other genres as a whole. We have noticed that many recent works do not address the problem of subgenre classification at large, though there are several works attempting to classify region-specific music, ballroom music [2] and Latin music [3].

Toward this end, we introduce a practical approach based on a pairwise comparison between subgenres, making use of the differences between them to conduct subgenre classification. In our approach, we employ association analysis to capture characteristic features for each subgenre and, when comparing two subgenres, we examine the differences between their corresponding characteristics. We show the effectiveness of our approach through empirical experiments on a variety of genres, with three (3) to six (6) subgenres each, for three (3) benchmark music datasets.

II. Related Works

An initial work in music genre classification is proposed by Tzanetakis and Cook [4], who created the GTZAN dataset, which is widely-used in the MIR community, despite its deficiencies [5]. Some earlier work presented by Silla *et al.* [6] proposes using ensemble techniques that combine a set of classical classification algorithms, such as *Support Vector Machines* (*SVM*), etc. to deal with the multi-label genre classification problem. More recent discussions include the one in [7], where *magnitude* and *tempo* features, and several off-the-shelf classifiers are used, showing a relationship between using a larger number of features and an improvement in classification. Medhat et al.[8] classify the ballroom music dataset with eight (8) genres, resulting in the best accuracy of 92.12%.

Quinto et al. [9] examine the effectiveness of deep learning classifiers using just *Mel-frequency cepstral coefficients* (*MFCC*) features on a dataset of three (3) jazz subgenres, which gives an average accuracy of 89.824%. Sousa et al.'s [10] classification of global and regional music datasets uses a new dataset called the *Brazilian Music Dataset* (*BMD*), derived based on the criticism [5] of the GTZAN dataset. With an SVM used, they achieve an accuracy of 79.7% for 10 GTZAN genres and 86.11% on the seven (7) BMD genres. Kizrak and Bolat [11] construct a dataset of 93 songs on the seven (7) most frequent Turkish Makams, achieving an accuracy of 96.57%. Soboh et al. [12] create a dataset based on Arabic music styles (four (4) genres, 100 songs each). *Dynamic*, *rhythmic*, and *timbral* features are derived and several classifiers are experimented with. An overall best accuracy of 80.25% is observed. Non-content-based classification is carried out by Neubarth et al. [13] to show the association of toponyms with *folk* music using textual features, in order to discover the relationships between the music and regions.

To the best of our knowledge, currently there are only a few works in music subgenre classification. Kirss [14] creates a dataset of *electronic* music, with five (5) subgenres, 50 songs each. *Rhythmic* and spectral features are derived and the highest accuracy is 96.4% using an SVM. Chen [15] extracts rhythmic and spectral features for three (3) electronic music subgenres, 10 songs each. An accuracy of 80.67% is achieved. Tsatsishvili [16] creates a *metal* dataset that contains seven (7) subgenres, 30 tracks of each. Features (i.e. timbral) are extracted and the highest accuracy achieved is 45.7% using some classifiers from WEKA [17]. Mulder [18] collects songs from 17 metal subgenres, and musical *chroma interval* features are used. The average is not high, with an average accuracy of about 25%. However, none of these works propose new algorithmic approaches to the subgenre classification problem and only uses off-the-shelf ones.

III. Problem Statement and Approach

The musical contents from different genres contain rich information that makes them distinguishable from each other [1].

ISSN 2305-7254

Such information can be extracted and utilized. In this work, we present: (1) how to characterize each music subgenre by a set of acoustic features through association analysis, and (2) how to evaluate the subgenre of a new music piece through dichotomy-like pairwise comparisons.

Association analysis is first proposed by Agrawal *et al.* [19]. In a problem domain, a set of data items that "frequently" occur together shows some statistical relationship among them. Those frequent items are put into *frequent itemsets*, e.g., a frequent 3-itemset means the three items in the set occur together frequently. The *support* of an itemset is the percentage of the co-occurrence of the items in it. Only the itemsets whose support exceeds a *minimum support*, m_s, are frequent. We adapt the *Apriori* [19] association algorithm in our approach.

For subgenre classification, each piece in a music dataset is represented as a vector $P = \{p_1, p_2, \cdots, p_n\}$, where p_i is the value of the feature $f_i \in F$ and $F = \{f_1, f_2, \cdots, f_n\}$ is the acoustic feature set. These features are extracted using some software frameworks, e.g., *Marsyas* [20]. The approach to capture subgenre characteristics is shown in algorithm A1. The superscripts tr and te correspond to training and testing. Given a main genre, suppose that it has n subgenres and each subgenre G has a dataset of pieces labelled G. A set GS is randomly chosen from it, to balance the number of music pieces in each subgenre. Since Apriori handles discrete values, we discretize acoustic features' real values, using a binning method, and then normalize them. During this process, we encode each value for each feature systematically. After encoding, each piece is represented as a set of feature-value pairs, called an *fv-set*. For each frequent fv-set returned by Apriori, we set the number of feature-value pairs in it, to be at least 2. For the subgenre G, after this step, we obtain its M sets of frequent fv-sets, denoted as GS_i^F, where $i = 1, \cdots$, $i = M$ ($M = 10$ in our experiments), from which we produce a more representative characteristic set, called GS^C

A1: Characterizing music subgenres by feature-value pairs
1. For each binning method B
2. For each subgenre G's dataset GS
3. Apply B to GS
4. Split GS into training set GS^{tr} and testing set GS^{te}
5. For each G's GS^{tr}
6. Randomly generate M subsets of it (denoted as GS_i)
7. For each subgenre G and for each of its GS_i
8. GS_i^F = Call *Apriori* to GS_i with m_s
9. For each subgenre G
10. We append fv-sets from GS_i^F to GS^C and remove any duplicates

With the characteristic sets of individual subgenres ready, we classify an unseen music piece for its subgenre from the testing music dataset, which is represented as a vector of feature-value pairs and scored against a pair of subgenres by comparing the differences between their respective characteristic sets. The steps, which should be done per binning method and m_s, are summarized in Algorithm A2. For a new music piece P from a subset of GS^{te} (we create 10 subsets of GS^{te} for each genre to average its experimental accuracies), we maintain a score vector $(S_{G_1}, S_{G_2}, \cdots, S_{G_n})$, where S_{G_i} is the "score" of G_i for P.

A2: Evaluating pairwise music subgenres by feature-value pairs
1. For a new music piece P (represented by feature-value pairs)
2. For the characteristic sets of two subgenres G_i and G_j, GS_i^C and GS_j^C
3. Calculate the except difference of GS_i^C and GS_j^C,
4. i.e., $DC_{ij} = GS_i^C - GS_j^C$ and $DC_{ji} = GS_j^C - GS_i^C$.
5. Score on P using DC_{ij} and DC_{ji}
6. s_i = Counting(P, DC_{ij})
7. s_j = Counting(P, DC_{ji})
8. if $s_i > s_j$ then
9. $S_{GS_i} += electoral?1 : s_i$
10. else
11. $S_{GS_j} += electoral?1 : s_j$
12. Set the genre of the highest score to be the one for P.

There are musical and acoustic elements that are common to all genres, which cause confusions when classifying new pieces into their subgenres. We use a parameter ϕ to conduct a "fuzzy" check for whether an fv-set from GS_i^C appears in GS_j^C, and vice versa. For instance, if $\phi = 60\%$, then {b4, c3} matches with {a2, b4, c3} but not with {a2, b4, c2}. A higher ϕ results in a smaller number of removed fv-sets, and a lower one is stricter on removing fv-sets. We call ϕ the *strictness factor*. The except difference between the two characteristic sets, $GS_i^C - GS_j^C$, consists of those fv-sets that are present in GS_i^C but not in GS_j^C. We precalculate the except differences among all pairs of subgenres. The procedure *Counting* counts how many fv-sets in the except difference are a subset of P's feature-value vector. We implement two mechanisms of counting. The first is called *electoral voting* (i.e., winner takes all) and the second is called *popular voting*, which finds the actual number of times the fv-sets in the except difference appear in P (normalized based on the size of the difference). For the sake of space, the results from the popular voting mechanism are omitted, since this method obscures the except differences and yields a lower classification accuracy. We plan to investigate it more in our future work.

IV. Empirical Experiments

Through our experiments, we show that (1) our approach has the ability to report the distinguishable subgenres with high accuracies, and (2) that our approach performs with comparable accuracies on different datasets.

A. Experiment Setup

TABLE I. A SUMMARY OF THE FEATURES USED IN THE SUBGENRE EXPERIMENTS

Dataset	Feature Sets	Features	Parameters	Calculations
D_{LMD}	F_{LMD}^M	5 MFCCs, Spectral centroid, rolloff, and flux, zero crossings, low energy, relative amplitudes, beat per minute, max. periods of pitch peak (MARSYAS)	Unspecified in [21]	mean, variance
D_{CAL}	F_{CAL}^M	13 MFCCs (MARSYAS)	window size: 2048ms hop size: 1024ms sampling rate: 22050Hz	mean, std. dev.
D_{FMA}	F_{FMA}^{L1}	20 MFCCs (LibROSA)	window size: 2048ms hop size: 512 ms sampling rates: unchanged	mean, std. dev., skew, kurtosis, median, min, max
	F_{FMA}^{L2}	20 MFCCs, spectral contrast and centroid (LibROSA)	window size: 2048ms hop size: 512 ms sampling rates: unchanged	mean, std. dev., skew, kurtosis, median, min, max
	F_{FMA}^{J}	13 MFCCs, spectral centroid, zero crossings, strongest beat overall, beat sum overall, strength of strongest beat overall, strongest frequency via zero crossings (jAudio)	window size: 2048ms hopsize: 1024ms sampling rate: 22050Hz	mean, std. dev.

We include several benchmark music datasets from the MIR community in our experiments, as shown in Table I. Our experiments on DLMD use the same features (30 features in total) as given by Silla et al. [21] that can be found online (https://sites.google.com/site/carlossillajr/resources/the-latin-music-database-lmd). The acoustic features are extracted using the MARSYAS [20] framework. We will refer to this feature set as F^{M}_{LMD}.

In addition, we also include two other large benchmark datasets: Cal10k [22] and the Free Music Archive [23], denoted as D_{CAL} and D_{FMA}, respectively. D_{CAL} has a subset of features taken from the middle segment, using MARSYAS. We will refer to this feature set as F^{M}_{CAL}. For D_{FMA}, we are using the "large" version of the dataset (106,574 songs total) with two acoustic feature sets extracted using LibROSA [24] from the middle 30 seconds per song, as presented by Defferrard et al. [23]. We denote these feature sets as F^{L1}_{FMA} and F^{L2}_{FMA}, respectively. To be even more complete, we also create a less sizeable but more diverse set of features extracted using jAudio [25], with similar parameters to D_{CAL}, we will refer to this feature set as F^{J}_{FMA}.

For all experiments, we equalize the number of songs per subgenre. Three binning methods are used to preprocess music data, including: *equal width* (B_{ew}), *equal frequency* (B_{ef}), and *equal width with Rice Rule* (B_{rr}) [26]. We then used an 80% 20% split for training and testing, with $M = 10$. 80% of the testing songs are then used to generate 10 random subsets for each accuracy calculation. Songs that are labelled as more than one subgenre are removed from further processing.

B. Experiment Results and Discussions

1) On dataset D_{LMD}: We first discuss the results on D_{LMD}, in which the genres can be considered as subgenres, due to the nature of the dataset. In Figure 1, we observe that with a lower support, a greater volume of complex characteristic relationships are present for each genre, and thus a higher classification accuracy is achieved after pair-wise comparisons. In Figure 2 we notice the trend that for stricter removal thresholds, accuracies become greater for distinguishable genres (i.e. *tango*, *bachata*) and can sometimes be lower for less distinguishable genres. Some fine tuning of this threshold may be needed, although the strictest ϕ value generally produces a better average overall accuracy. In Figure 3 we see that B_{ef} performs in a stable manner, and is less prone to accuracy drops for certain subgenres, which might occur with B_{ew}, or B_{rr}. Note that the binning method can exacerbate the difference between distinguishable and non-distinguishable genres, and make the classification less stable.

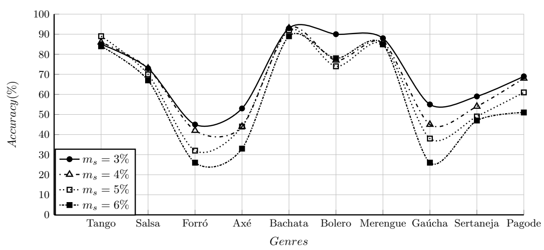

Fig. 1. *Analysis of m_S with parameters: D_{LMD}, F^{M}_{LMD}, $\phi = 0.4$, B_{ef}, 3000 songs per subgenre.*

We have noticed that a low support, with B_{ef} and a strict ϕ threshold, yields better classification accuracies than the other parameter combinations. In the case of classifying "hard" subgenres these successful parameters may not function in the same manner due to the similarity between them, and may need further tuning.

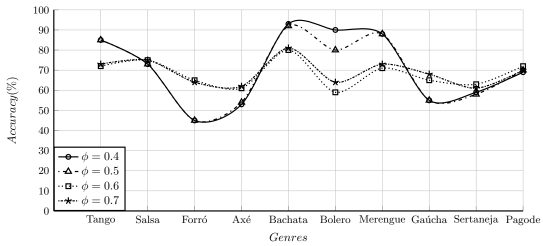

Fig. 2. *Analysis of ϕ with parameters: D_{LMD}, F^{M}_{LMD}, $m_S = 3\%$, B_{ef}, 3000 songs per subgenre.*

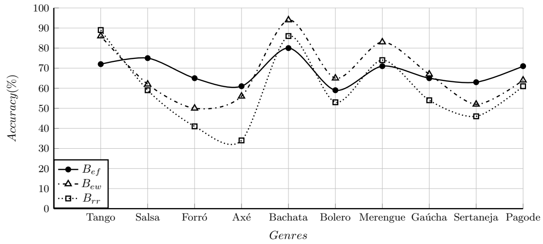

Fig. 3. *Analysis of binning with parameters: D_{LMD}, F^{M}_{LMD}, $m_S = 3\%$, $\phi = 0.6$, 3000 songs per subgenre.*

2) On dataset D_{CAL}: The subgenre experiments for the dataset D_{CAL} in Table II are examined. We can easily see the effectiveness of our approach in Table II(a), and the same trend that is also present in the experiments for D_{LMD} (i.e. a lower ϕ threshold produces better accuracies). In Table II(b) we can see that our approach is effective in differentiating between two types of *gospel* music (i.e. far above a chance percentage of 25%). Next, in Table II(c), for $\phi = 0.3$ we note that *solo* piano pieces are classified correctly 100% of the time, and those pieces with heavy violin instrumentation are often classified correctly. This tells us that our approach should be well-suited for instrument recognition tasks. We see that *jazz* and *hiphop* have effective accuracies as well, as the subgenres *electro*, and *smooth jazz* achieve classification accuracies in the 80% range. *Jazz fusion* is expected to have higher classification accuracies, as elements of *rock* should make this subgenre distinguishable. This is not observed in our experiments. We also predicted the confusion of *classic hip hop* with *electro*, and this is found to be true. The most notable results, after the impressive classification of *reggae*, is the classification of *pop*, as shown in Table II(f). *Pop* is one of the least suspected genres to have effective classification accuracies, given the similarity of its subgenres. However, the majority of the subgenres (especially for $\phi = 0.4$) are classified successfully.

Note that the trend, i.e., the strictest ϕ value yields the best average classification accuracy, is still present, except for *jazz* and *pop*, where the highest accuracies are achieved with a moderate strictness. Since we experiment on m_s values from 3% to 7%, we find that an m_s value of 3% and 4% percent achieves lower classification accuracies. Since subgenres are very similar, lower support itemsets will be removed by a strict ϕ value and with a higher m_s, frequently found characteristics

TABLE II. RESULTS FOR DCAL

ϕ	dub	reggaeton	dancehall	ska	avg
0.3	77	88	86	78	82.25
0.4	66	80	72	66	71
0.5	66	78	75	65	71
0.6	43	84	63	76	66.5
0.7	41	79	65	79	66

(a) *Reggae,* D_{CAL}*,* F^M_{CAL}*,* B_{ef}*,* $m_s = 5\%$*, 71 songs per subgenre.*

ϕ	soul	funk	traditional gospel	contemporary gospel	avg
0.3	49	49	74	60	58
0.4	30	53	69	35	46.75
0.5	43	43	70	41	49.25
0.6	40	39	51	54	46
0.7	41	43	48	59	47.75

(b) *Rhythm and Blues (RnB),* D_{CAL}*,* F^M_{CAL}*,* B_{ef}*,* $m_s = 7\%$*, 54 songs per subgenre.*

ϕ	piano concerti	symphonic	violin features	piano solo	choral	avg
0.3	49	53	63	100	31	59.2
0.4	61	55	20	95	20	50.2
0.5	61	58	31	90	20	52
0.6	59	30	48	50	49	47.2
0.7	59	30	41	89	51	54

(c) *Classical,* D_{CAL}*,* F^M_{CAL}*,* B_{rr}*,* $m_s = 6\%$*, 53 songs per subgenre.*

ϕ	smooth jazz	jazz fusion	avante garde jazz	bebop	swing	avg
0.3	66	36	41	57	58	51.6
0.4	85	39	41	67	49	56.2
0.5	86	33	45	68	49	56.2
0.6	56	24	34	52	58	44.8
0.7	48	24	37	43	60	42.4

(d) *Jazz,* D_{CAL}*,* F^M_{CAL}*,* B_{ef}*,* $m_s = 6\%$*, 60 songs per subgenre.*

ϕ	underground hiphop	southern rap	classic hiphop	electro	avg
0.3	57	38	47	83	56.25
0.4	38	43	47	82	52.5
0.5	38	46	43	83	52.5
0.6	40	66	24	81	52.75
0.7	33	60	27	89	52.25

(e) *Hip Hop,* D_{CAL}*,* F^M_{CAL}*,* B_{ef}*,* $m_s = 6\%$*, 57 songs per subgenre.*

ϕ	new age pop	pop rock	dance pop	classic pop	teen pop	avg
0.3	80	29	70	55	53	57.4
0.4	78	31	71	53	67	60
0.5	76	32	66	55	63	58.4
0.6	78	37	46	25	73	51.8
0.7	78	41	47	23	75	52.8

(f) *Pop,* D_{CAL}*,* F^M_{CAL}*,* B_{ef}*,* $m_s = 5\%$*, 67 songs per subgenre.*

TABLE III. *RESULTS FOR* D$_{FMA}$

ϕ	bluegrass	rockabilly	country and western	avg
0.2	91	75	91	85.67
0.3	86	60	100	82
0.4	83	63	92	79.33
0.5	93	69	82	81.33
0.6	93	60	59	70.67
0.7	91	63	58	70.67

(a) *Country,* D_{FMA}*,* F^{L2}_{FMA}*,* B_{ew}*,* $m_s = 15\%$*, 64 songs per subgenre.*

ϕ	krautrock	new wave	post rock	shoe-gaze	industrial	prog-ressive	avg
0.2	49	64	31	35	28	57	44
0.3	45	64	31	34	31	53	43
0.4	48	64	35	30	34	47	43
0.5	49	65	35	31	35	48	43.83
0.6	39	74	28	28	33	39	40.17
0.7	39	68	30	29	35	30	38.5

(b) *Rock,* D_{FMA}*,* F^{L2}_{FMA}*,* B_{ef}*,* $m_s = 7\%$*, 232 songs per subgenre.*

ϕ	thrash	black metal	death metal	sludge	grindcore	avg
0.2	41	80	35	93	46	59
0.3	49	82	41	95	56	64.6
0.4	49	82	41	95	56	64.6
0.5	35	93	39	95	58	64
0.6	34	82	32	82	46	55.2
0.7	35	82	30	82	45	54.8

(c) *Metal,* D_{FMA}*,* F^{L1}_{FMA}*,* B_{ef}*,* $m_s = 12\%$*, 82 songs per subgenre.*

ϕ	hardcore	post punk	electro punk	no wave	avg
0.2	81	50	85	63	69.75
0.3	79	43	88	54	66
0.4	75	43	88	53	64.75
0.5	80	43	89	55	66.75
0.6	61	41	91	51	61
0.7	58	39	89	49	58.75

(d) *Punk,* D_{FMA}*,* F^{L2}_{FMA}*,* B_{ef}*,* $m_s = 8\%$*, 211 songs per subgenre.*

ϕ	house	glitch	drum and bass	downtempo	dubstep	avg
0.3	47	41	47	60	61	51.2
0.4	43	37	45	57	54	47.2
0.5	44	41	46	58	53	48.4
0.6	32	34	44	34	44	37.6
0.7	33	30	33	38	42	35.2

(e) *Electronic,* D_{FMA}*,* F^{J}_{FMA}*,* B_{ef}*,* $m_s = 7\%$*, 189 songs per subgenre.*

ϕ	British folk	free folk	freak folk	avg
0.3	71	29	50	50
0.4	67	53	55	58.33
0.5	69	59	54	60.67
0.6	63	36	59	52.67
0.7	60	51	47	52.67

(f) *Folk,* D_{FMA}*,* F^{J}_{FMA}*,* B_{ef}*,* $m_s = 8\%$*, 89 songs per subgenre.*

for a subgenre are found. We can see here that for close subgenres, having the lowest reasonable m_s value does not always give the best result.

3) On dataset D_{FMA}*:* Next, the experiment results shown in Table III are examined. Successful accuracies of subgenres are found with MFCCs for just one genre (i.e. *metal*). All other genres' subgenres have improved accuracies after including further features. For *country*, we see an average accuracy of 85.67% for three subgenres. This is what is expected, since these subgenres are quite dissimilar. One peculiarity is that B_{ew} provides successful predictions for *country*, unlike the other genres. Given that there are six (6) overlapping subgenres for *rock*, in Table III(b), we do not deem this as an unsuccessful classification, since three (3) subgenres are classified at around 50%. Furthermore, *progressive rock* and *new wave* are classified more distinctly, and *post rock* and *shoegaze* are confused, which matches our intuition. Next, *metal*'s subgenres are classified quite successfully. We expect that *black metal* and *sludge* would provide higher accuracies compared to other metal subgenres. *Death metal* is similar to both *thrash* and *grindcore*. So this accounts for some of the misclassifications between the three subgenres. Most of the classifications for *punk* are acceptable, with *electropunk* and *hardcore* being more successful. This once again satisfies our intuition. *Electronic* is classified as good as the other genres. However, we see in Table IV that a higher classification accuracy is found for more subgenres of *electronic* music. This may be due to the similarity of subgenres chosen in the D_{FMA} dataset, as well as the subset of songs chosen for experiments. Finally, we see practical classification accuracies for *folk*, as shown in Table III(f), despite the overlap between its subgenres.

TABLE IV. ELECTRONIC, D_{CAL}, F^M_{CAL}, B_{ew}, ms = 6%, 56 SONGS PER SUBGENRE

ϕ	drum and bass	trance	triphop	techno	industrial	ambient	avg
0.3	17	39	43	66	72	80	52.83

4) General remarks: The binning method B_{ef} is able to withstand very strict ϕ percentages, whereas B_{ew} and B_{rr} sometimes return empty sets after pairwise comparison for the same strictness threshold. This is due to the same number of values present in every bin with B_{ef}. Having the same number of values in each bin avoids the possibility that a certain set of bins will contain most of the values. However, this possibility still remains for B_{ew} and B_{rr}. So after training, plenty of pairwise removal can be done for them. The number of bins is not further explored, for the sake of space, for the subgenre classification tasks. The number of bins and the binning types become sensitive for subgenre classification. Further future investigation is needed.

Another factor to consider while dealing with each subgenre's GS^C is the overall complexity. With a greater number of unique fv-sets for the M generations of GS_i^F we find that, with a small enough m_s, the complexity of the pairwise removal tasks becomes burdened. With a suitable m_s value, and M parameter, a trade-off between complexity and classification accuracy can be made.

V. Conclusion

In this paper, we propose a novel approach to the subgenre classification problem in music. We have shown that our approach is able to classify subgenres effectively. Our experiment results are easily reproducible, and can be used to compare our approach with others'. Our work is the first attempt (to our current knowledge) of subgenre recognition at such a detailed level. It should be noted that since the highest classification accuracies are achieved with a more diverse set of features than just MFCC features, we believe that with additional features included (i.e. content-based, visual, lyrical, etc.), the classification accuracies could be even more useful for musical data curation tasks.

One of the merits of our approach, among others, is that it is able to store the characteristics of the subgenres that make the particular subgenres distinsguishable, which would be beneficial to other tasks in the MIR community, i.e. feature reduction, verifying the consistency of tag annotations, etc. We have also observed that our approach can recognize various instruments, and may even be able to determine various styles of music production, but we need more experiments to support this. This will be our very next task in the future.

References

[1] T. Li, M. Ogihara, and G. Tzanetakis, *Music Data Mining*. Belmont, CA: CRC Press Wadsworth, 2011.

[2] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, "Evaluating rhythmic descriptors for musical genre classification," in *Proceedings International Conference, Audio Engineering Society*, 2004, pp. 196–204.

[3] C. N. Silla Jr., A. L. Koerich, and C. A. A. Kaestner, "The latin music database," in *Proceedings of the 9th International Conference on Music Information Retrieval*, Philadelphia, USA, 2008, pp. 451–456.

[4] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[5] B. L. Sturm, "The gtzan dataset: Its contents, its faults, its effects on evaluation, and its future use," *Computing Research Repository*, vol. abs/1306.1461, 2013.

[6] C. N. Silla, A. L. Koerich, and C. A. A. Kaestner, "A machine learning approach to automatic music genre classification," *Journal of the Brazilian Computer Society*, vol. 14, no. 3, pp. 7–18, 2008.

[7] R. Ajoodha, R. Klein, and B. Rosman, "Single-labelled music genre classification using content-based features," in *Proceedings Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference*, Nov 2015, pp. 66–71.

[8] F. Medhat, D. Chesmore, and J. Robinson, "Automatic classification of music genre using masked conditional neural networks," in *Proceedings IEEE International Conference on Data Mining*, Nov 2017, pp. 979–984.

[9] R. J. M. Quinto, R. O. Atienza, and N. M. C. Tiglao, "Jazz music subgenre classification using deep learning," in *Proceedings IEEE Region 10 Conference*, 2017, pp. 3111–3116.

[10] J. M. de Sousa, E. T. Pereira, and L. R. Veloso, "A robust music genre classification approach for global and regional music datasets evaluation," in *IEEE International Conference on Digital Signal Processing*, Oct 2016, pp. 109–113.

[11] M. A. Kand B. Bolat, "A musical information retrieval system for classical turkish music makams," *Simulation*, vol. 93, no. 9, pp. 749–757, Sep. 2017.

[12] L. Soboh, I. Elkabani, and Z. Osman, "Arabic cultural style based music classification," in *International Conference on New Trends in Computing Sciences*, Oct 2017, pp. 6–11.

[13] K. Neubarth, I. Goienetxea, C. Johnson, and D. Conklin, "Association mining of folk music genres and toponyms," in *Proceedings International Society for Music Information Retrieval Conference*, 2012, pp. 7–12.

[14] P. Kirss, "Audio based genre classification of electronic music," Master's thesis, University of Jyväskylä, 2007.

[15] A. C. Chen, "Automatic classification of electronic music and speech/music audio content," Master's thesis, University of Illinois at Urbana-Champaign, 2014.

[16] V. Tsatsishvili, "Automatic subgenre classification of heavy metal music," Master's thesis, University of Jyväskylä, 2011.

[17] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Morgan Kaufmann Publishers Inc., 2016.

[18] D. G. J. Mulder, "Automatic classification of heavy metal music," Master's thesis, University van Amsterdam, 2014.

[19] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Record*, vol. 22, no. 2, pp. 207–216, 1993.

[20] G. Tzanetakis. (2017, October) Music analysis, retrieval and synthesis for audio signals. [Online]. Available: http://marsyas.info/index.html

[21] C. N. Silla, A. L. Koerich, and C. A. A. Kaestner, "Automatic music genre classification using ensemble of classifiers," in *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 1687–1692.

[22] D. Tingle, Y. Kim, and D. Turnbull, "Exploring automatic music annotation with acoustically-objective tags," in *Proceedings of the international conference on Multimedia information retrieval*. ACM, 2010, pp. 55–62.

[23] K. Benzi, M. Defferrard, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proceedings International Society for Music Information Retrieval*, 2017, pp. 316–323.

[24] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa : Audio and music signal analysis in python," in *Proceedings Python in Science Conference*, 2015, pp. 18–25.

[25] D. McEnnis, I. Fujinaga, C. McKay, and P. Depalle, "Jaudio: A feature extraction library," in *Proceedings the International Society of Music Information Retrieval*, 2005, pp. 600–603.

[26] D. M. Lane. (2017, December) Online statistics education: A multimedia course of study. rice university.