An Unsupervised Approach to Anomaly Detection in Music Datasets

Yen-Cheng Lu[1] Chih-Wei Wu[2] Chang-Tien Lu[1] Alexander Lerch[2]
[1]Department of Computer Science, Virginia Tech
[2]Center for Music Technology, Georgia Institute of Technology
[1]{kevinlu, ctlu}@vt.edu, [2]{cwu307, alexander.lerch}@gatech.edu

ABSTRACT

This paper presents an unsupervised method for systematically identifying anomalies in music datasets. The model integrates categorical regression and robust estimation techniques to infer anomalous scores in music clips. When applied to a music genre recognition dataset, the new method is able to detect corrupted, distorted, or mislabeled audio samples based on commonly used features in music information retrieval. The evaluation results show that the algorithm outperforms other anomaly detection methods and is capable of finding problematic samples identified by human experts. The proposed method introduces a preliminary framework for anomaly detection in music data that can serve as a useful tool to improve data integrity in the future.

Keywords

Anomaly detection; music information retrieval; unsupervised

1. INTRODUCTION

Music information retrieval (MIR) is an active research area that integrates knowledge from a number of different fields, including Electrical Engineering, Computer Science, Psychology, and Musicology [7]. Many previous studies in this area have adopted machine learning techniques and applied them to audio data to build an intelligent system that understands music. The evaluation of such systems, as described by Schedl et al. [7], requires different datasets and annotations depending on the tasks. However, since the annotation process for music data is both complex and subjective, the quality of the annotations created by human experts can vary considerably from dataset to dataset, potentially introducing errors into the system and adversely affecting the performance. Finding a way to enhance the correctness of these datasets is thus crucial for the further improvement of MIR systems.

A typical example of the types of problems encountered concerns the Music Genre Recognition (MGR). According to Sturm [10], the most frequent used dataset in MGR is GTZAN [12], and many existing systems are evaluated based on their performance in classifying GTZAN audio data into its annotated genre class. However, Sturm points out that this dataset actually contains a significant fraction of corrupted files, repeated clips, and misclassified genre labels. These are clearly undesirable for the proper training of an MGR system.

In the data mining community, the corruptions listed above are referred to as anomalies or outliers. Identifying these outliers in a given dataset could be formulated as an anomaly detection problem [3]. Although anomaly detection methods are widely applied in various types of datasets, they are rarely discussed in the MIR community.

In this paper, we propose an unsupervised approach to address this problem and to detect the anomalies in music datasets. A statistics-based model is developed to capture the normal behavior of feature representations. Adding a Student-t prior to the latent error variable in the model maintains the robustness of the estimation of the normal behavior and absorbs the anomalous effects into this latent variable. The contributions of this paper are summarized as follows:

- **An unsupervised music anomaly detection approach:** An unsupervised approach for detecting anomalies in music datasets is proposed. No anomaly label is required.
- **A categorical anomaly detection model:** A regression based categorical anomaly detection model using a robust estimation strategy is also proposed. Furthermore, an approach to approximate the analytically intractable inference is also developed.
- **Benchmark experiments:** Results on the benchmark dataset demonstrated that the proposed approach outperforms existing state-of-the-art methods.

2. METHOD

2.1 Feature Extraction

In this paper, a set of baseline features based on Tzanetakis and Cook's features [12] is extracted for comparison with prior work in this area. The extracted features can be divided into three categories: spectral, temporal and rhythmic. All of these features are extracted using a block-wise analysis method. The process begins by down-mixing the audio signal to a mono signal, that is then segmented into overlapping blocks (block size: 23 ms, hop size: 11 ms). The temporal features are computed from the time domain signal of each block directly. The spectral features are computed from the

SIGIR '16, July 17-21, 2016, Pisa, Italy

DOI: http://dx.doi.org/10.1145/2911451.2914700

magnitude spectrum of each block using a Hann window. The rhythmic features are extracted from the beat histogram of the entire time domain signal. The selected features are as follows (for details of the implementations, please refer to [5]):

- **Spectral Features** ($d = 16$): Spectral Centroid (SC), Spectral Roll-off (SR), Spectral Flux (SF), 13 Mel Frequency Cepstral Coefficients (MFCCs)
- **Temporal Features** ($d = 1$): Zero Crossing Rate (ZCR)
- **Rhythmic Features** ($d = 8$): Period0 (P0), Amplitude0 (A0), RatioPeriod1 (RP1), Amplitude1 (A1), RatioPeriod2 (RP2), Amplitude2 (A2), RatioPeriod3 (RP3), Amplitude3 (A3).

Spectral and temporal features are aggregated in texture windows of length 0.743 s following the standard procedure as outlined in [12]. The mean and standard deviation of the features within a window are then computed to create a new feature vector. Finally, all the resulting feature vectors are reaggregated with their mean and standard deviation, generating a single feature vector that represents each individual recording in the dataset.

2.2 Model Design

Given N feature vectors $X = \{X_1, ..., X_N\}$ and their corresponding categories $Y = \{Y_1, ..., Y_N\}$, where each $Y_n \in \{C_1, C_2, ..., C_M\}$, we formulate the relation of Y and X based on a linear assumption. The input-output relationship can be represented as a regression model:

$$g(Y) = X\beta + \varepsilon, \tag{1}$$

where g is the categorical link function, β is the regression coefficient matrix, and ε is a random variable that represents the white-noise vector of each instance. The link function g is a logit function that is paired with a category C_M, i.e., $\ln\left(P(Y_n = C_m)/P(Y_n = C_M)\right) = X_n\beta_m + \varepsilon_{nm}$. Since the probabilities of the categories will sum to one, we can derive the following modeling equations:

$$P(Y_n = C_m) = \frac{\exp\{X_n\beta_m + \varepsilon_{nm}\}}{1 + \sum_{l=1}^{M-1}\exp\{X_n\beta_l + \varepsilon_{nl}\}} \tag{2}$$

and

$$P(Y_n = C_M) = \frac{1}{1 + \sum_{l=1}^{M-1}\exp\{X_n\beta_l + \varepsilon_{nl}\}} \tag{3}$$

The coefficient vector β usually represents the decision boundary in a classification problem. Here, β is used to capture the normal behavior of the data. Following the convention, we assume that each β_m obeys a Gaussian distribution with a predefined mean vector and a predefined covariance matrix Σ_β, i.e.,

$$\beta_m \sim N(\beta_m|\mathbf{0}, \Sigma_\beta) \tag{4}$$

In traditional regression applications, the error factor ε is generally assumed to follow a Gaussian distribution. However, a Gaussian distribution lacks tolerance for anomalies since the probability distribution is near zero at points far away from the distribution mean. In the study of robust statistics, it is suggested to assume that this random variable follows a heavy-tailed distribution in order to improve the capability of capturing anomalies [11].

In this work, we assume that the error is a zero-mean Student-t random variable, which has been shown to be a useful way to improve the robustness of the logistic regression model [6]. The probability density function is as the following:

$$p(\varepsilon|\sigma_\varepsilon^2, df) = \frac{\Gamma(\frac{df+1}{2})}{\Gamma(\frac{df}{2})\sqrt{\pi df\sigma_\varepsilon^2}}\left(1 + \frac{\varepsilon^2}{df\sigma_\varepsilon^2}\right)^{1(\frac{df+1}{2})} \tag{5}$$

where σ^2 is the scaling parameter, and df is the number of the degrees of freedom. We utilize the Student-t variable as an "error-buffer" here to absorb the error introduced by the anomaly instances, thus allowing us to easily differentiate between the anomalies and errors.

2.3 Approximate Inference

Since the response is a categorical variable, the inference becomes intractable. We make a Bayesian assumption to the model and use the variational-EM algorithm [1] to approximate the inference. We start from the joint distribution of the model:

$$p(Y, \beta, \varepsilon) \propto p(Y|\beta, \varepsilon)p(\beta)p(\varepsilon) \tag{6}$$

Suppose there is a proposal distribution $q(Y, \varepsilon, \beta)$ that approximates p, s.t. $q \simeq p$. Based on the structure of the model, we can factorize q into two parts, i.e., $q(\varepsilon)$ and $q(\beta)$. The estimation of the variational variables is updated by maximizing the optimal factors until the convergence criterion is satisfied. For each individual update we apply iterated re-weighted least squares (IRLS) to find the optimum. Applying the Taylor expansion to the log expectations above, we can obtain a quadratic form in $\ln q = -\frac{1}{2}\nu^T Q\nu + \nu^T b$, where ν represent the target variable to update, and

$$b(\nu) = \nabla\nabla_\nu q(\nu) - \nabla_\nu q(\nu) \tag{7}$$

$$Q(\nu) = \nabla\nabla_\nu q(\nu) \tag{8}$$

In each iteration, we update the value of ν by

$$\nu^{(new)} = Q^{-1}(\nu^{(old)})b(\nu^{(old)}) \tag{9}$$

Thus, by iteratively updating β and ε, with estimating the gradient and Hessian in each iteration, the process will converge to a local optimum of these variables.

2.4 Anomaly Detection Process

The full process, shown in Algorithm 1, consists of feature extraction and anomaly identification. It begins by taking the music clips and extracting their features. The anomaly identification process then takes the extracted feature vectors

Algorithm 1 Detection Process

Require: Dataset of music clips D
Ensure: The anomalous instances
set $[Y, X] = extractFeatures(D)$
set $\beta^* = \beta_0$, $\varepsilon^* = 0$
while Not converge **do**
  set $\beta^* = argmax_\beta(p(\beta|X, Y, \varepsilon^*))$
  set $\varepsilon^* = argmax_\varepsilon(p(\varepsilon|X, Y, \beta^*))$
  set $L = \ln(p(\theta|X, Y, \varepsilon^*))$
end while
set $AnomalySet = \phi$
for all ε_n^* in ε^* **do**
  if $\varepsilon_n^* > ErrorThreshold$ **then**
    put n in $AnomalySet$
  end if
end for
return $AnomalySet$

Method	P	R	F	AUC
Linear Detection	**0.59**	**0.55**	**0.57**	**0.91**
Clustering	0.23	0.23	0.23	0.74
KNN	0.27	0.26	0.27	0.78
LOF	0.26	0.25	0.25	0.74

Table 1: Average Detection Rate for Injected Data

and the corresponding categories as the input, and returns a set of indices of the anomalous instances as the output.

The process starts by taking the MGR dataset D and performing a feature extraction to obtain the feature vector X and its corresponding class labels Y. Next, the anomaly detection method starts with initial values of the variables β_0 and ε_0. The process iteratively updates the model variables β and ε using the approach introduced above (lines 3–7). After the variables have converged, the error variable ε is checked to identify anomalies (lines 8–13). As a reasonable assumption, similar to the analysis of Gaussian distributions, instances with ε greater than 3 times of the standard deviation are labeled anomalies.

3. EXPERIMENT

3.1 Experimental Design

The experiments were conducted on the popular GTZAN dataset [12]. This dataset consists of 10 music genres (blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock), each with 100 audio tracks. All tracks consist of a 30-second excerpt of complex mixtures of music.

For this study, two sets of experiments were conducted to test the performance of the proposed method and three other benchmark methods. In the first experiment, we used a purified GTZAN dataset that excludes the conspicuous misclassified and jitter music clips reported in [9]. An injection process was performed by randomly choosing 5% of the instances in each genre, and randomizing their genre labels to create outliers. Sturm's report identifies around 50 conspicuous files, which corresponds to about 5% of the total number of files in the dataset. We generated 10 random realizations of the dataset, and evaluated the measures with the average for this 10-run batch. This experiment simulates the best-case scenario, where the dataset is clean and all genres are well separated in feature space. The results thus serve as a sanity check for all the methods.

In the second experiment, we applied our method to the full GTZAN dataset directly. The outliers identified were then compared with the list reported in [9]. This experiment is effectively a real-world scenario in which case the automated anomaly detection methods are expected to find the outliers identified by human experts. Two sets of features represent the dataset, one is the feature set as described in Sect. 2 and the other a minimal feature set using only 13 MFCCs, as reported in the work of Hansen et al. [4].

3.2 Benchmark Methods

We compared the results obtained using our approach with those of three unsupervised benchmark methods. Due to the small number of true anomaly labels and their uneven distributions across different genres, the application of a supervised anomaly classifier was deemed not feasible and thus excluded in this experiment.

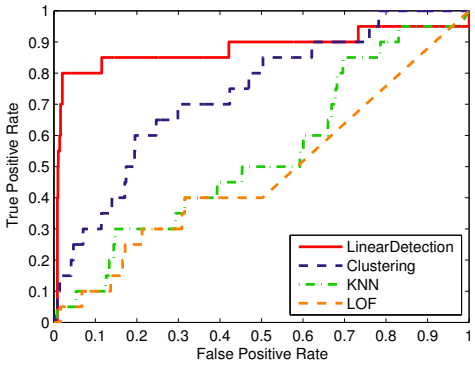

Figure 1: ROC curves on injected data

1. *Clustering*: an intuitive method for detecting misclassified music clips is to group them into M clusters. Ideally, all of the music clips in the same cluster should belong to the same genre so the instances with different category indices than the majority are labeled as anomalies. In this experiment, we used k-means clustering with the centroids initialized using the mean of each genre.
2. *k-Nearest Neighbor* (KNN): the KNN-based method calculates the distance from each instance to its k-nearest neighbor and marks those points with high average distances to their neighbors as anomalies. In our experiment, values for k between 1 and 20 were tested and the value with the best result were chosen. We set $k = 10$ in the injection set and $k = 1$ in the expert-label set.
3. *Local Outlier Factor* (LOF): LOF [2] is one of the most popular anomaly detection methods. Similar to the KNN method, it calculates the local density of each instance that can be estimated from the distance of its k-nearest neighbors. This approach compares an object's local density to its neighbors' and identifies the objects with significant lower local densities as anomalies. The value of k was chosen using the same approach as for the *KNN* method. We set $k = 15$ in both the injection experiment and the expert-label experiment.

3.3 Experimental Results

Our evaluation of the results obtained for the four methods was based on four standard metrics: precision P, recall R, F-measure F, and Area Under ROC Curve AUC. The first set of experiments was conducted on the injected data set. As shown in Table 1, the proposed method outperformed the three benchmark methods on all of the metrics.

Figure 1 shows the ROC curves for injected data. In the figure, *LinearDetection* refers to the proposed method, which is based on a linear assumption. Since our method has the best performance with respect to the AUC, the anomaly scores given by our approach provides the most useful information for ranking the significance of the detected anomalies. The advantage of the proposed method is that the regression model captures the input-output relationship between the features and the genres, while the benchmark methods rely solely on the distribution of features and may thus fail to capture the actual pattern.

We also compared our feature set with a minimal feature set containing only MFCCs, as used in [4]. As shown in Table 2, the performance metric for all the methods dropped

Method	P	R	F	AUC
Linear Detection	**0.52**	**0.40**	**0.45**	**0.87**
Clustering	0.08	0.07	0.07	0.61
KNN	0.12	0.11	0.12	0.68
LOF	0.13	0.13	0.13	0.70

Table 2: Average detection rate for injected data with only MFCCs features

Method	P	R	F	AUC
Linear Detection	**0.18**	**0.23**	**0.20**	**0.63**
Clustering	0.08	0.07	0.07	0.41
KNN	0.10	0.09	0.10	0.50
LOF	0.10	0.09	0.10	0.51

Table 3: Detection rates for GTZAN data (Sturm's anomalies)

in this case. In particular, the performance of the *Clustering* method wa significantly dropped. This observation suggests that in MFCC feature space, the genres overlap substantially and cannot be separated by clustering.

In the second set of experiments, the anomaly detection process was applied to the full GTZAN dataset and aim to detect the misclassified music clips reported by Sturm [9]. The results are shown in Table 3. Although our method still outperformed the benchmark methods, the performance decreased noticeably compared to the injection setup. Based on the metrics utilized here, none of these methods are able to detect anomalies with high accuracy. Comparing our results with the anomalies reported in [9], we found that our method is able to detect the jitter clips (*reggae*: No. 87), and some of the obviously misclassified clips, including *reggae*: No. 88, *disco*: No. 41, *pop*: No. 81, *hip-hop*: No. 31, all of which achieved high anomaly score with our method. Interestingly, we also found that neither our method nor any of the benchmark methods could successfully detect anomalies in the metal genre (*metal*: Nos. 46–57). These music clips are in fact punk rock but are annotated as metal in the GTZAN dataset. It can be observed that the extracted features for these music clips are very similar to other metal clips. This implies that the features used for this approach are not able to sufficiently differentiate *punk rock* from *metal*. One important task for future work is therefore to improve the identification of representative features to allow for better differentiation between similar genres.

It should also be pointed out that while people mostly agree on the genre labels *hip-hop* and *blues*, they tend to disagree on the category *rock* [8], demonstrating the inherent ambiguity of many music genres. This trend can be observed in Table 4, where the detection rates for *metal* and *rock* are the lowest among all the genres. This is most likely due to the inherent ambiguity in these genre categories, which could result in inconsistent patterns in the feature space and thus increase the difficulty of anomaly detection. All of the methods failed to detect the alternative rock clips reported by Sturm (*metal*: Nos. 96–99); the feature vectors of these clips are generally similar to those of other metal music clips. To the best of our understanding, these clips can also be categorized as nu-metal, which is a sub-category of metal music in the taxonomy of genre. This again highlights the natural ambiguity involved in classifying music. One potential solution is to develop a multi-class modeling method to tolerate this ambiguity and suggest alternative classes that might also be applicable to individual music clips.

Genre	# Anomalies	# Positives	P	R
Blues	0	1	N/A	N/A
Classical	0	0	N/A	N/A
Country	4	8	0.13	0.25
Disco	7	9	0.33	0.42
Hip-hop	2	6	0.17	0.50
Jazz	2	0	N/A	0.00
Metal	16	1	0.00	0.00
Pop	3	6	0.33	0.67
Reggae	7	10	0.30	0.43
Rock	2	15	0.00	0.00

Table 4: Detection rate on GTZAN data by genre

4. CONCLUSION

In this paper, we have presented an unsupervised approach for automatic anomaly detection in music datasets. The proposed approach incorporates a novel statistical model to identify suspect files in existing music datasets with no training data required. The experimental results demonstrated that our method outperformed other benchmark anomaly detection approaches when applied to the GTZAN dataset. In the future, more features will be investigated to find better and meaningful representations for anomaly detection in music datasets.

5. REFERENCES

[1] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. *SIGMOD Record*, 29(2):93–104, May 2000.

[3] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computer Survey*, 41(3):15:1–15:58, July 2009.

[4] L. Hansen, T. Lehn-SchiÃÿler, K. Petersen, J. Arenas-Garcia, J. Larsen, and S. Jensen. Learning and clean-up in a large scale music database. In *European Signal Processing Conference (EUSIPCO)*, pages 946–950, 2007.

[5] A. Lerch. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. John Wiley and Sons, 2012.

[6] C. Liu. *Robit Regression: A Simple Robust Alternative to Logistic and Probit Regression*, pages 227–238. John Wiley & Sons, Ltd, 2005.

[7] M. Schedl, E. Gómez, and J. Urbano. *Music Information Retrieval: Recent Developments and Applications*, volume 8. Now Publishers Inc., Hanover, MA, USA, 2014.

[8] M. Sordo, O. Celma, M. Blech, and E. Guaus. The Quest for Musical Genres: Do the Experts and the Wisdom of Crowds Agree? In *Int. Conference on Music Information Retrieval (ISMIR)*, pages 255–260, 2008.

[9] B. L. Sturm. An analysis of the GTZAN music genre dataset. In *Proceedings of the second international ACM workshop on Music Information Retrieval with user-centered and multimodal strategies*, 2012.

[10] B. L. Sturm. The State of the Art Ten Years After a State of the Art: Future Research in Music Information Retrieval. *Journal of New Music Research*, 2013.

[11] D. E. Tyler. Robust statistics: Theory and methods. *Journal of the American Statistical Association*, 103:888–889, 2008.

[12] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.