

COMPUTING A SONG’S DECADE

Chris Latina

Georgia Tech Center for Music Technology
840 McMillan St. Atlanta, GA
chris.latina@gatech.edu

Liang Tang

Georgia Tech Center for Music Technology
840 McMillan St. Atlanta, GA
liangmt@gatech.edu

ABSTRACT

Is it possible to compute a song’s decade from a short audio clip? This paper describes an algorithm that makes use of instantaneous audio features, metadata, and traditional machine learning algorithms to estimate a song’s decade. Inspired by the accuracy and consistency of the Discogs database, we wanted to apply the approaches commonly used to classify genre to additional song and album metadata. Our task is to determine which features are most relevant for the task and more generically, whether a hierarchical approach to classification using genre can help predict other metadata about a set of audio clips, such as the year of release.

Index Terms— Year Computation, Machine Learning, Audio Content Analysis, MIR, GTZAN

1. INTRODUCTION

MIR is an active field of machine learning research. Users of online music services are very likely to search for music by genre or style, so researchers have attempted to understand how to automatically classify music by these labels. For instance, overarching genres like rock or disco likely exhibit enough distinction for computers to effectively distinguish between them [1].

The dataset used for training and testing the machine learning model plays a significant role in its results. Many datasets that exist for genre classification only contain the minimum amount of information, namely the audio snippet and a label. Recently, researchers have taken effort to compile more complete and accurate datasets in different ways [2]. One approach is to supplement the existing data by providing music-related metadata information such as the song title, album name, year and even price of the release.

Discogs.com is a user-sourced database and marketplace for album releases on vinyl, cassette, CD, and digital formats. Unlike last.fm, the data is extremely complete and accurate because the website is heavily moderated. Notably, the Discogs API mentions that each release has a “master format”, typically the first release chronologically along with metadata such as genre, styles (subgenres), and year.

In this paper, the problem of automatically computing song’s decade is addressed. More specifically, the technique of using the combination of traditional music genre classification approaches with music metadata to estimate the year of a song is proposed. Although there has been significant work in the development of music information retrieval especially for genre classification, there has been relatively little work in the development of song’s year or decade computation.

2. RELATED WORK

In music, like in other forms of art, style represents a classification of the medium. Likewise, a style is often associated with an era. Art historians often map the trajectory of the two in order to decipher the influence of one school of artists on another. In regards to classification with metadata, “Computation Analysis Of Musical Influence” by Nick Collins mentions “...sourcing data from allmusic.com, and utilising python APIs for last.fm, EchoNest, and MusicBrainz.” In his approach, Collins also cross referenced the year of the primary release with Discogs.com for accuracy. This paper touches upon the idea of a more semantic understanding of the output of classifiers.

In that effort, several papers have explored the efficacy of learning algorithms to predict genres. In his paper, George Tzanetakis effectively classified genres on live radio broadcasts using a Gaussian classifier [5]. Mandel used Support Vector Machine (hereafter referred to as *SVM*) on artist and album-level features to make similar classification as well [6]. Also, another study explored mixtures of Gaussians and K-Nearest-Neighbors (hereafter referred to as *KNN*) for music classification [7]. Each of these studies used similar features - Mel-Frequency Cepstral Coefficients (hereafter referred to as *MFCC*) and chroma features of audio to make the classification.

3. ALGORITHM OVERVIEW & DESCRIPTION

3.1. Audio Feature Extraction

Feature extraction is the process of computing a compact numerical representation that can be used to characterize a seg-

ment of audio [5]. In terms of MIR, the design of descriptive features for specific application is the main challenge in building pattern recognition systems.

3.1.1. Timbral Features: Mel-Frequency Cepstral Coefficients

MFCCs are perceptually motivated features that are based on the STFT.

$$\text{MFCC}(n) = \sum_{k'=1}^{\mathcal{K}'} \log(|X'(k', n)|) \cdot \cos\left(j \cdot \left(k' - \frac{1}{2}\right) \frac{\pi}{\mathcal{K}'}\right)$$

3.1.2. Spectral Features

The Spectral Centroid (hereafter referred to as *SC*) is defined as the center of gravity of the magnitude spectrum of the STFT. The centroid is a measure of spectral shape and higher centroid values correspond to "brighter" textures with more high frequencies.

$$\text{SC}(n) = \frac{\sum_{k=0}^{\mathcal{K}/2-1} k \cdot |X(k, n)|^2}{\sum_{k=0}^{\mathcal{K}/2-1} |X(k, n)|^2}$$

The Spectral Flux (hereafter referred to as *SF*) measures the amount of change of the spectral shape. It is defined as the average difference between consecutive STFT frames: (should be an equation here).

$$\text{SF}(n, \beta) = \frac{\sqrt[\beta]{\sum_{k=0}^{\mathcal{K}/2-1} (|X(k, n)| - |X(k, n-1)|)^\beta}}{\mathcal{K}/2}$$

3.2. Dataset Creation

In this research, we selected GTZAN as our starting dataset. George Tzanetakis created GTZAN specifically for machine learning analysis of genre classification problems. This dataset has been documented to have non-negligible issues such as "album effect" with repeated audio samples which would negatively influence the accuracy of the classification [8], it still can be seen as the default choice in the research area of music genre classification. Figure 1 represents the degree of activity of GTZAN.

We used a modified version of the GTZAN dataset augmented by metadata and preview audio clips sourced from 7digital.com. Our dataset started as 1000 songs from 10 genres but many of the audio clips lacked information. Furthermore, genre groups contained duplicate songs or entire albums. In an attempt to reduce the album effect and distribute the years represented, we reduced the dataset to 7 genres with

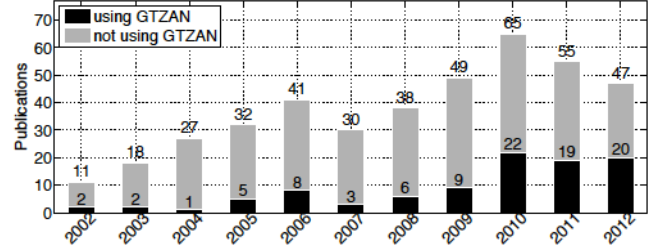


Fig. 1. Annual numbers of published works in MIR with experimental components, divided into ones that use and do not use GTZAN [8]

60 songs in each. We used the 7digital API endpoint interface to search within genres and scrape for additional preview clips to remove redundant data.

After creating the list of files with the artist and song name, metadata was then scraped from Discogs.com API using python. This metadata included artist, title, album name, genre, styles, and year of master release. Although some of the previews from 7digital represent remastered versions, we cross referenced with the Discogs master release for consistency. Ideally, the dataset would have an even distribution of songs spanning the range of all years without repeated artists. The final dataset consists of 420 audio clips in total with genres Blues, Country, Disco, Hip Hop, Metal, Pop, Rock. Any additional track samples added from 7digital were cut to 30 seconds in length using ffmpeg. All songs were 30 second long 44.1k 16bit .wav or .au files.

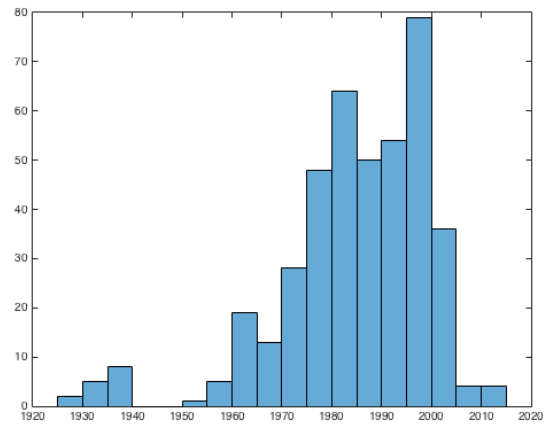


Fig. 2. Histogram displaying the distribution of years in our modified dataset

3.3. Classification

For classification, variations of two basic statistical pattern recognition (hereafter referred to as *SPR*) algorithms were

used: SVM and KNN. Essentially, an SPR is applied to estimate the probability density function for the feature vectors of each class. This paper compares the suitability of these two SPR classifiers to maximize the accuracy of computing a song's year.

4. EVALUATION

4.1. Methodology

4.1.1. *N-fold Validation*

N-fold was applied to perform multiple iterations of classification for evaluation. The dataset contains 7 genres with 60 songs each (420 songs in total). An N value of 6 evenly distributes the genres represented in the training and test sets. This way, for each fold, the training set contains 350 songs and the test set contains 70 songs, with 10 songs from each genre.

4.1.2. *Segmentation of Data*

The data was segmented into training and testing pools using two techniques. The results were quite different and tie into some fundamental concepts of dataset creation. One approach was an interleaved distribution. The second was a randomized distribution.

For the interleaved distribution, the feature set was read in order adding one song from each genre at a time. Because there were instances of multiple songs by an artist from the same album, the distributed segmentation approach separates songs from one album occurring in both the training and test set, thus removing the risk of the "artist effect" or "album effect". In his paper, Seyerlehner argues that it is relatively easy to identify songs by one and the same artist using audio similarity algorithm [9]. This effect is known as artist effect. In some cases, even album-specific production effects are reflected in the spectral representation of songs, which is respectively called "album effect". Obviously, songs by the same artist will tend to belong to the same genre, and the ability to recognize the genre by specific production effect is not what this research intend to measure. However, this also the worst case scenario, because the training and test set will always be very different from each other.

For the randomized distribution, a seed is generated for the entire feature set and applied to all songs and metadata, completely randomizing the order. This creates a potentially even distribution of genres within each fold for the training and test sets. It greatly increases the chance of the album effect, however, because songs from the same album that were previously only in the training set now can appear in the test set as well.

4.1.3. *Feature Selection*

After attempting to select features with forward selection, we manually selected features, mimicking a backward selection approach. Testing logical combinations of feature groups proved to be more successful. In general, MFCCs and subsets of MFCCs outperformed other feature combinations. Our main groups were the mean and standard deviations of MFCCs, spectral centroid, and spectral flux as mentioned in section 3.1. We also tested mean and standard deviation of Pitch Chroma, however this never resulted in improvement.

4.2. Metrics

We tested four variations on the KNN and SVM algorithms for each of the two segmentation approaches. For the random distribution, each attempt was the mean of the same 6-fold process described in section 4.1.1. The seed and therefore the result changed with each run, so the result given is the best of 10 runs to show the lower boundary.

4.2.1. *SVM using Regression*

The SVM Regression was performed on the year data only, as genres can only be classified discretely. We scaled the years from the range of 1927-2015 to a range of 0-1. Using libsvm [10] with regression settings of a nu-SVR with a polynomial kernel, a nu value of 0.5, and a default cost parameter of 1. Combining the mean and standard deviation of MFCCs gave the most successful results. Our average mean error in calculating the year was 12.754 years for the interleaved segmentation and 10.425 years for the randomized segmentation.

4.2.2. *KNN of Year*

The second approach used only the KNN directly on the year label, with no genre influence. This was surprisingly the simplest but most effective result. We achieved our best results using a K value of 7 with only the mean of MFCCs as features. This resulted in a mean error of 8.751 years and 8.015 years for the interleaved segmentation and randomized segmentation respectively. Our KNN algorithm uses an inverse weighting function calculated by the euclidean distance as a method to give priority over nearer results, shown below.

$$w_k = (1 - (d_k / \sum_{k=1}^K d_k)) / (K - 1)$$

$$Y_E = \sum_{k=1}^K w_k * y_k$$

4.2.3. *Hierarchical Model: SVM and KNN*

Our third approach make use of both the SVM and KNN models in a hierarchical manner. First we used the the SVM to

classify the genres using both the mean and standard deviation of MFCCs and our spectral features. Prior to refactoring our dataset, we achieved results of 55.4% and 66.4% accuracy for the interleaved and randomized segmentations respectively. After refactoring, however we saw our results decrease, due to the album effect. Since we intentionally targeted repeat songs and full albums present in the GTZAN dataset, our distribution of features expanded within each genre making it more difficult for the SVM to accurately perform classification. After refactoring, the accuracy for classifying genre was 51.5% and 61.0% for the interleaved and randomized segmentations respectively. The confusion matrices in Figure 3 describe these results.

After classifying genre, we used the predicted genre for each test song and ran KNN to find the nearest neighbors to compute the year. The selection used to run KNN, however, only found the nearest neighbors within the genre predicted for each song. For example, if we predicted a song to be country, KNN only selects the country songs in the current training fold. Using a value of $K=9$ gave the best results. Our average mean error in calculating the year was 12.132 and 11.602 years for interleaved segmentation and randomized segmentation respectively.

4.2.4. Hierarchical Model: KNN with Ground Truth Genre

The final approach removed genre prediction and simply used the ground truth metadata of the test set. The KNN was run again for each audio clip in the test set against using the actual genre metadata for that track. Again, it limited the selection of features from the training set to those that matched the clip's genre. For consistency, we ran this test using both the mean and standard deviation of MFCCs and our spectral features, achieving an average mean error of 12.412 and 11.926 years for interleaved segmentation and randomized segmentation respectively. Interestingly enough, the result was slightly worse than using the trained genre rather than the ground truth. Realizing that optimizing the features selected for the genre when it is not actually used, we experimented with modifying the active features to tune it to computing the year. We found that using only mean MFCCs 6 through 12 resulted in a mean error of 11.136 and 11.670 years for interleaved segmentation and randomized segmentation respectively.

5. CONCLUSION

Overall, the KNN of Year method using the mean of MFCCs as features worked best. The hierarchical model did show improvement in relation to the SVM but only slightly. It still underperformed in comparison with KNN. Our results were very dependent upon the dataset construction and any flaws within the dataset.

Forward selection did not prove successful when using MFCCs and Spectral data. The algorithm results would de-

crease and return before grouping all MFCCs. Manually tuning by finding subsets and groupings of mean and standard deviation of MFCCs with the spectral features proved most effective. Adding Pitch Chroma was ineffective and only generated worse results for both genre and year. When constructing a feature set by calculating the mean and standard deviation of pitch chroma, the temporal resolution is lost.

In each genre, many of the songs are from only a few albums. Training and testing for the genre or style classification with GTZAN is more likely doing the album classification. This "album" effect heavily influences the genre classification albeit positively, however this is actually incorrect. Essentially, by importing new songs and abandoning several songs that belong to the same album, the augmented dataset is more even distributed by year.

6. FUTURE WORK

6.1. Dataset Modification

In our research, the "album effect" and the size of the dataset are two non-negligible issues. Since the GTZAN dataset was incomplete with the basic artist and song title tags and contained duplicate songs, our resulting dataset was small. Ideally, the size of the dataset should be at least twice that of what was used. A small dataset can lead to automatic selection and neglect potentially useful features. To train a qualified year computation system, a large dataset with evenly distributed audio samples spanning many years and genres is necessary. This does take quite a long time, however the scripts provided to scrape metadata from Discogs in conjunction with the 7digital audio preview scraping may be a good solution to this problem.

6.2. Feature Extraction

In this research, timbral texture features and spectral features are extracted from the audio sample and used for training the system. However, rhythmic features like beat histogram are not employed. With the rhythmic features, the performance of system should improve.

6.3. Classification

SVM and KNN as statistical pattern recognition models are widely used in the research of MIR. Recently, deep learning methods, especially convolutional neural networks (also known as CNNs) have become more and more popular in this research area. Convolutional neural networks, characterized by a multi-layer structure, sparse connectivity, and weight sharing have already improved the performance of music genre classifiers and chord detectors. Combining CNN with audio features may help improve the accuracy of song's decade computation.

Confusion Matrix							
Blues	42.86	8.57	5.71	12.86	4.29	0.00	11.43
Country	25.71	35.71	0.00	4.29	1.43	7.14	11.43
Disco	7.14	8.57	34.29	10.00	4.29	10.00	11.43
Hip Hop	12.86	2.86	5.71	42.86	5.71	14.29	1.43
Metal	7.14	1.43	1.43	4.29	67.14	0.00	4.29
Pop	0.00	11.43	1.43	7.14	0.00	62.86	2.86
Rock	5.71	5.71	15.71	10.00	15.71	2.86	30.00
	Blues	Country	Disco	Hip Hop	Metal	Pop	Rock

Confusion Matrix							
Disco	48.57	5.71	8.57	5.71	5.71	2.86	8.57
Country	4.29	55.71	7.14	7.14	1.43	7.14	2.86
Pop	1.43	10.00	65.71	4.29	0.00	0.00	4.29
Rock	8.57	7.14	0.00	48.57	8.57	2.86	10.00
Metal	2.86	0.00	0.00	4.29	71.43	4.29	2.86
Blues	2.86	5.71	0.00	4.29	5.71	61.43	5.71
Hip Hop	5.71	4.29	15.71	1.43	2.86	8.57	47.14
	Disco	Country	Pop	Rock	Metal	Blues	Hip Hop

Confusion Matrix							
Blues	40.00	27.14	2.86	7.14	4.29	1.43	2.86
Country	7.14	41.43	10.00	10.00	0.00	8.57	8.57
Disco	1.43	7.14	38.57	8.57	2.86	10.00	17.14
Hip Hop	2.86	11.43	8.57	38.57	8.57	12.86	2.86
Metal	4.29	0.00	2.86	14.29	54.29	0.00	10.00
Pop	0.00	7.14	2.86	5.71	0.00	60.00	10.00
Rock	2.86	11.43	22.86	7.14	5.71	11.43	24.29
	Blues	Country	Disco	Hip Hop	Metal	Pop	Rock

Confusion Matrix							
Rock	20.00	14.29	22.86	10.00	10.00	4.29	4.29
Country	7.14	54.29	7.14	11.43	2.86	2.86	0.00
Disco	14.29	4.29	42.86	10.00	10.00	1.43	2.86
Hip Hop	4.29	8.57	12.86	34.29	12.86	2.86	10.00
Pop	7.14	7.14	4.29	8.57	58.57	0.00	0.00
Blues	4.29	15.71	2.86	4.29	1.43	54.29	2.86
Metal	7.14	1.43	2.86	10.00	0.00	7.14	57.14
	Rock	Country	Disco	Hip Hop	Pop	Blues	Metal

Fig. 3. Confusion matrices resulting from the SVM. *Left to right:* Interleaved vs Randomized segmentation. *Top to bottom:* Original GTZAN dataset vs modified dataset

7. REFERENCES

- [1] Omar Diab, Anthony Manero, and Reid Watson, “Musical genre tag classification with curated and crowd-sourced datasets,” .
- [2] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whithman, and Paul Lamere, “The million song dataset,” in *ISMIR 2011: Proceedings of the 12th International Society for Music Information Retrieval Conference, October 24-28, 2011, Miami, Florida*. University of Miami, 2011, pp. 591–596.
- [3] Hugo Leichtentritt, “Aesthetic ideas as the basis of musical styles,” *The Journal of Aesthetics and Art Criticism*, vol. 4, no. 2, pp. 65–73, 1945.
- [4] Roger B Dannenberg, “Style in music,” in *The structure of style*, pp. 45–57. Springer, 2010.
- [5] George Tzanetakis and Perry Cook, “Musical genre classification of audio signals,” *Speech and Audio Processing, IEEE transactions on*, vol. 10, no. 5, pp. 293–302, 2002.
- [6] Michael I Mandel and Daniel PW Ellis, “Song-level features and support vector machines for music classification,” in *ISMIR 2005: 6th International Conference on Music Information Retrieval: Proceedings: Variation 2: Queen Mary, University of London & Goldsmiths College, University of London, 11-15 September, 2005*. Queen Mary, University of London, 2005, pp. 594–599.
- [7] Tao Li, Mitsunori Ogihara, and Qi Li, “A comparative study on content-based music genre classification,” in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2003, pp. 282–289.
- [8] Bob L Sturm, “The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use,” *arXiv preprint arXiv:1306.1461*, 2013.
- [9] Klaus Seyerlehner, Gerhard Widmer, and Tim Pohle, “Fusing block-level features for music similarity estimation,” in *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, 2010, pp. 225–232.
- [10] Chih-Chung Chang and Chih-Jen Lin, “Libsvm: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27, 2011.