Music Genre Classification: Update 1 (4/4/2016)

Thomas Lillis

Student  
9962 Bear Creek Apartments  
Boulder, CO 80303   
+1 303 590 5985

thli2089@colorado.edu

Joseph Marylander

Student  
1025 Adams Circle, E215  
Boulder, CO 80303  
+1 303 475 8838

joma1054@colorado.edu

Connor McGuinness

Student  
805 29th St. #557  
Boulder, CO 80303  
+1 203 521 9512

comc3361@colorado.edu

**ABSTRACT**

Genres within music are culturally agreed upon categorical descriptions used to filter songs and artists. As music production has boomed with technology, the list of genres is well beyond reach for a single person to know all of them. In this work, a classification process that eventually will add tags to previously unclassified data containing artist and song information. In detail, the process will analyze certain musical elements like rhythm, time signature, ‘danceability’, or key in order to create tags such as ‘pop’ and ‘rap’ or even ‘happy’ and ‘creepy’. In order to develop such tags, the classification process will utilize methods currently being discussed in class.

Bayesian classification will be one of the first test classification methods used due to it’s easy to use structure and good fit with our project goal. For example, setting pre-defined probabilities for each genre tag and then applying a value for ‘rhythm’ which, depending on it’s amount, will have a stronger or lesser affect on the probability of the genre tag. Determining probabilities for each genre tag is where our classification process will become our own unique algorithm. Therefore, during the development of our classification process, we will work to minimize any biases towards a certain value affecting a probability. For example, just because a song has a high value for ‘beats per minute’ does not automatically give the song a tag such as ‘happy’ or ‘exciting’ since another value may reduce the probability the song would receive such a tag.

As well as testing a handful of classification methods, classification accuracy needs to be analyzed as well. Using a clustering algorithm, we will able to more easily check the accuracy of the tags of each song produced during the classification process.

Based on the classification accuracy and process, a user interface will be developed on either a web or mobile platform. This interface will allow an unfamiliar user to interact with the dataset and filter songs based on the pre-defined tags. Users will be able to select from a list of tags and once submitted, be presented with a list of songs that have one or more of those tags.

**1. INTRODUCTION**

Though how we interpret music genres as humans is a subjective classification, there are still some metrics which specifically fit genres in which most can agree. We will attempt to find and use the best attributes extracted from raw audio files to classify the audio files into commonly accepted, broad, generic, genres. This type of work could be useful for many different types of applications. It may be used for a Pandora type website for finding songs with a similar genre classification that the user might enjoy or could be used by radio stations to find more music they could play for their station. For this project we plan on using several different types of multi-class classification to see which method is most effective with our dataset. If no method is as informative as would like we will extend the project to extracting more attributes from the audio files to get more accurate classification.

# 2. LITERATURE SURVEY

In the paper [1] “Exploring different approaches for music genre classification” Antonio Jose Homsli Goulart, et al. describe how they extracted various features from a dataset of songs and how they attempted to classify them. They focused only on three genres of music: Blues, Classical, and Lounge.  In contrast to their research, we plan on classifying our data into as many genres as we can while being accurate. However, their classifications techniques are useful in determining how we should proceed with our own classification.

Similar to our project goal, this team proposed a musical genre classification system using attributes such as rhythmic structure and instrumentation [2]. Their work aims to solve the problem of the large amount of music being created without an automated genre classification system. As well, the team created two graphical interfaces to allow unfamiliar users to interact with their classification system; something our team wants to do. Understanding George Tzanetakis’ and his team’s approach to genre classification will help guide us in creating a similar project but with a different genre classification process.

Thierry Bertin-Mahieux [3] takes a simplified approach to musical genre classification, which is something our team wants to do as well. Instead of allowing tags to take multiple forms like ‘American pop’, ‘indie pop’, or ‘European pop’, they chose to include all of these tags into the tag ‘pop’. Thierry, B. M genre classification process only allows for the following 10 genres: classic pop and rock, folk, dance and electronica, jazz and blues, soul and reggae, punk, metal, classical, pop, and hip-hop. As well as only including these genres, Thierry, B. M excludes other popular genre tags within their dataset. This allows them to focus on artists or songs that are only within ‘pop’ instead of both ‘pop’ and ‘electronic’ for example. Our team’s aim is to follow a similar approach in simplifying the genres we choose, but also

# 3. DESIGN AND IMPLEMENTATION

The project will have multiple subtasks grouped into main tasks. Initially, main tasks will focus on obtaining and organizing the [Million Song Dataset](http://labrosa.ee.columbia.edu/millionsong/). Determining interesting attributes within the dataset and then filtering these attributes will be an important first step. The first phase of the project will be broken down into a collections of subtasks that aim to ensure the data is accurate and organized, and that there are enough values within each attribute to find meaningful trends.

Next, the second phase of the project will be focused on parsing the dataset and developing and trying different types of classification methods. Subtasks will consist of developing a parsing script, determining several classification methods, and implementing/testing the classification methods. This phase’s main goal is to create a well thought out classification method and then implement it. The third and final phase of the project will be analyzing the types of meta tags outputted from the classification process.  By analyzing these meta tags in we should, for example, be able to correctly predict other songs in our database that are similar thus creating a music recommendation service.

There are a number of ways we are considering classifying our data into specific genres.  Since we are also being provided labels for our data thereby allowing us to create a training set we will be determining which one of the following techniques provides the best accuracy: DAG SVM, neural network, k-means, k-nn.  However, we are also interested in running our data against a clustering algorithm in order to see what type of clusters are generated and how they compare to the actual labels we have.

We started by downloading a subset of the data that was 1% of the entire dataset. This was about 10,000 total songs to work with and was about 4.7GB of files. The data was split into one file per song and the files were “Hierarchical Data Format” files and were split into directories by a random name that was alphabetized. This required a special python library to be able to read the files which were ‘.h5’ files. This library ended up causing a lot of trouble down the line with the time of execution. We used a BASH script to traverse all of the directories that contained the files and to run the python script that extracts all of the information from each song file and aggregate all of the data with attributes into one text file that we could use for parsing. Each song has about 55 attributes associated with it and these attributes where delimited by tabs because there were commas in some of the attributes. This led to some challenges for a python parsing script down the line which was unable to parse two tabs in a row when an attribute field was empty. The BASH script took a little over an hour to run on a Intel 3rd Generation i7-3520M CPU @ 2.90GHz and 2 cores with 8GB of RAM.

To get the entire dataset it ended up being a greater challenge than we anticipated. First we attempted to access the entire dataset through a MySQL database that was provided from the million song dataset webpage. Thought this database had all the data points of the dataset it did not provide every attribute that we needed for our classification but rather mostly just meta data about the songs. Our second option was a slightly more challenging option using Amazon Web Services. Though the dataset was easily accessible through Amazon Web Services it was slightly challenging to work with. We worked with the free tier of AWS which was a single core version of Ubuntu 14.04. To get the data we had to attach a 500GB Amazon Public Data Snapshot and mount it to our filesystem on our virtual machine. Then it was traversable “Hierarchical Data Format” structure similar to the subset of the dataset.

Thought we thought it would be as easy as just running our script that we ran on the subset to aggregate our data but unfortunately this was not the case. We started running the script but the runtime of the script completed was an estimated 21 days. We believe this had to do with the power of the amazon machine. It was only a single core with a little amount of RAM. The bottleneck of this script was when the python script was called to read the file. We used a special library to read the “.h5” files which was particularly slow. It took about 1-2 seconds to read a single song out of our million songs. We were unable to to transfer the dataset off of the AWS server because we had a max of 15 GB of bandwidth. It is possible to upgrade the machine through Amazon but that would have been much more expensive. We left this script running for several days but it didn’t seem sustainable so we decided to try and parallelize the script and handle different seconds of the dataset. We made a mistake in implementation and accidently ran over 700 instances of the script in parallel. This used all the memory on the virtual machine and crashed the machine. This stopped our original script running and it only handled about 50,000 data points. We then tried running 26 instances of the script because there were 26 different head directories each a different letter of the alphabet. This did not speed up the process of the script running unfortunately. This aggregated the songs into about 10,000 songs before we realized the trajectory of the pace of the script was not as fast as even one instance of the script running. We settled on running this script with 4 instances working at once. This ran at a rate of about 50,000 songs every 12 hours. We pulled the data off the server after about 250,000 songs but left it running to be able to run on a larger dataset later. Then some simple BASH scripting was done to combine all the files and remove duplicates of all the files the where run. This made for about 300,000 songs in our aggregate file for a size of 240 MB. In this file we then used the “|” character for the delimiter because it was easier for our python script to parse.

We then needed to merge this data with the genre tags that we acquired from [Yajie Hu and Mitsunori Ogihara Million Song Dataset Genre Tags. To do this we wrote another python script that reads in all of the genre tags with their associated ID. It then parses the text file with all the song data in it and if there is a match on IDs then that song with all of the attributes as well as genre tag are put into a new text file. At this stage we then had a text file with all the the raw data. We did some basic cleaning of the data like removing whitespace and correcting a few formatting issues.](http://web.cs.miami.edu/home/yajiehu/resource/genre/)

After we created our merged dataset we needed to prune some features. We created a python script that allows us to remove specific column names from our dataset. Most of the features we pruned were song id’s and md5 values, neither of which are helpful to us. We then needed to convert the genre labels from text into an integer value. this was easily done by having a one to one mapping of the text labels to an integer via a dictionary. After we finished our base preprocessing we needed to run it through one of our proposed classifiers. Currently, we only have one of our classifiers working at a reasonable accuracy: a one vs rest support vector machine.

# 4. EVALUATION

In order to determine the accuracy of our classification algorithm, the team will require a set of labels that will objectively classify the genre of the song. Evaluation will be done by comparing the team’s classification labels to the labels in the dataset. We will use Yajie Hu et al.’s genre dataset[4]. This dataset is a mirror of the original “Million Song Dataset” with all the features removed except for the song name and their genre label. By using their collection of genre labels in the team’s training and evaluation set, it is guaranteed that the labels assigned to each song will match with this dataset. The ideal way to compare the accuracy for each of the proposed classification tools is to use a confusion matrix. It should be very easy to take the team’s results from each of these algorithms along with the correct labels found in this dataset and run them through sklearn’s confusion\_matrix function to gather the results.

The team also plans on classifying this dataset using an unsupervised clustering algorithm in order to see what genre clusters emerge and how they compare to the genres assigned by humans. It will be challenging to evaluate the quality of the clusters because there are not many good labels that are assigned to theses genres. The team will have to parse through theses clusters and attempt to rename the labels to have similar names to that of the dataset­ [4]. This is not an easy process because the team will have to determine what the main common features are of the set and compare that to the predefined genres in order to best rename the clusters to the appropriate label.

Furthermore, it is very possible with the team’s clustering algorithm, that some genres will be lost and as a result two or more genres are merged into a single cluster due to similarities of the song features. In this event there is no accurate way to evaluate these clusters because they do not represent any data. However, these clusters may provide insight into how genres are similar and how they could have influenced each other over the course of time. It is very likely that a clustering algorithm will choose to cluster on elements that do not directly relate to a genre, such as ‘danceability’ or the energy of a song. In this event evaluating accuracy is also very challenging but it has further applications in areas like song suggestions based on mood or “feel” of a song.

# 6. CONCLUSIONS AND FUTURE WORK

Something we will plan better for is managing the entire dataset in a better way. This would entail either upgrading the amazon server or starting working with the entire dataset earlier in our project. Which a larger dataset we could get more accurate results and possibility a better sample. For our sample of the entire dataset that we used as our final data, it is unclear if the sample was biased somehow. It could be biased by our script possibly collecting in an ordering that favors a certain genre or a certain attribute.

Another way to expand would to be extract more attributes from the data. The attributes that were given to us as part of the dataset were not the optimal attributes to do classification for genres. Extracting more attributes and being more particular about selection finding attributes that would make sense for classification.

This project could have also been expanded to work for a specific problem. For example, a music recommendation app could do genre classification on your favorite music and give you recommendations based on your preferences of things selected as similar genres. For this to work properly we would need more resolution in our genre class labels. This is because

# 6. REFERENCES

1. Goulart Homsi Jose, A., Guido Capobianco, R., and Dias Maciel, C. 2012. Exploring different approaches for music genre classification. *Egyptian Informatics Journal.* 13, 2 (July 2012), 59-63. DOI= <http://dx.doi.org/10.1016/j.eij.2012.03.001>.
2. Tzanetakis, G., Essl, G., and Cook, P. 2001. *Automatic Musical Genre Classification of Audio*. Technical Report. Princeton University.
3. Thierry, B. M., Ellis, D., Whitman, B., and Lamere, P. 2011. The Million Song Dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference* (ISMIR 2011).
4. Yajie Hu and Mitsunori Ogihara, “Genre Classification for Million Song Dataset Using Confidence-Based Classifiers Combination”, *in Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, Portland, Oregon, USA, 2012, pages 1083 – 1084

# 7. APENDIX

Honor Code Pledge:

*On my honor, as a University of Colorado at Boulder student, I have neither given nor received unauthorized assistance on this work.*

Tom Lillis: Worked on getting all of the data, aggregating the data into usable source, managing amazon web services server, appending the needed attributes to data, cleaning data, and basically everything that was not the actual machine learning.

Joey Marylander:

Connor McGuinness: