LAMP: Lyrics Assisted Music Prediction

Shanjit Singh Jajmann SJAJMANN@SEAS.UPENN.EDU
Varun Sridhar VSRIDHAR@SEAS.UPENN.EDU
Jitesh Gupta JITESHG@SEAS.UPENN.EDU

Abstract

The three of us being huge music buffs had always wondered how music streaming services like Spotify, Pandora, 8Tracks etc. recommend music to a user. We were fascinated by how natural language processing and machine learning algorithms could be used to enable this feature. With this as our motivation our goal was to build LAMP, a Lyrics Assisted Music Predictor - a system which predict genres of music tracks using features extracted from lyrics as well as audio. This particular area of interest hasn't seen much work yet, atleast not on a very large scale or using complex datasets. Even though we aren't the first to try our hand at this, there are a plethora of approaches and algorithms which can be explored and our aim is to have an accuracy score better than our contemporaries. We wanted to see whether using lyrics as an addition to just audio features provides a lift and how well just a stand alone predictor based on lyrical information does. Since, there are a number of possible genres, we restrict ourselves to predicting only the 5 most popular ones in the dataset we chose- 'rock', 'pop', 'alternative', 'indie' and 'electronic'. This paper describes the information processing and machine learning techniques employed and discusses the results we obtain.

1. Literature and Related Work

The GTZAN dataset (Tzanetakis & Cook, 2002) is one of the most commonly used datasets for genre classification but it is limited in its scope and size. The Million Songs dataset was launched to remedy that and there has been a good amount of research in music information retrieval that has been conducted following the launch. To the best of our knowledge, our results are comparable if not superior to those obtained by our contemporaries, albeit that our problem statements differ to some extent. A similar problem statement to ours was researched by O'Connor, Liang & Gul at Carnegie Mellon University (Liang et al., 2011). Using a training set comprised of 143,000 tracks and testing set comprised of 10,000 tracks with 10 tags, they achieved an accuracy of 38.6% in the final model. As part of the machine learning course at Stanford in 2013, Kader and Yates (Kader & Yates, 2013) used Last.fm with MusiXmatch for genre prediction to achieve an accuracy of 29% without sentiment analysis. Ahmed Bou-Rabee, Keegan Go and Karanveer Mohan (Bou-Rabee et al., 2012) used a publicly available music lyrics dataset containing 9700 songs but with an entire dictionary of 44000 words derived and obtained an accuracy of 48%.

2. The Dataset

For the purpose of our project we our using three sources of data - Million Songs Dataset, Last.fm Dataset and the MusiXmatch Dataset (Thierry Bertin-Mahieux & Lamere, 2011). Using lyrics for genre prediction along with audio features required us to combine these datasets. The Million Song Dataset (MSD) contains audio feature analyis and metadata for one million songs which have audio characteristics like loudness, density, timbre and pitch values, energy, mode, tempo etc. The Last.fm Dataset contains tracks associated with MSD with genre tags and % values which signify how much a particular song falls into a particular genre.The MusiXmatch Dataset contains the lyrics (bag of words representation) associated with around 77% of the tracks in MSD. This is given as a dictionary of 5000 most commonly occuring words and the frequency of each word for every individual song.

2.1. Information Processing

Given the disjoint arrangement and varied locations for our data, our first step was to join the given datasets.We chose 57 relevant audio features (loudness, danceability, energy, key, mode and average and variance of pitch/timbre across 12 segments) from the Million Songs Dataset (Thierry Bertin-Mahieux & Lamere, 2011). These were extracted

University of Pennsylvania, CIS 419/519 Course Project.

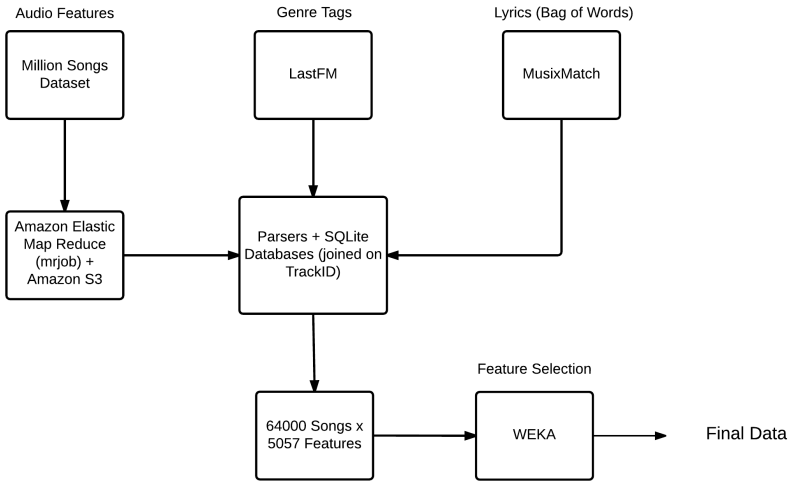

Figure 1. Steps for Information Processing

using the Amazon Elastic MapReduce (Lamere, 2011a) (Sotomayor, 2013) (EchoNest, 2011) on Amazon S3 buckets (PennMSD, 2014). For lyrical analysis, we joined the Last.fm database with the MusiXmatch database in order to get lyrics and the corresponding genre tags. We then joined this database with the Million Songs Dataset on Trackid. After all this significant data processing, we obtained a table of tracks with lyrical information and audio characteristics. All data storage was done in SQLite and all parsers were written in python.

2.2. Final Dataset

Our final dataset had a size of 1.2 GB with 64,000 unique tracks. Out of these 64000, we randomly picked 40,000 tracks to use in our experiments since the data processing for 64000 tracks was time and memory intensive. The following were the attributes used for the final data - 57 audio features (Danceability, Duration, Energy, Key, Loudness, Mode, Tempo, Density, Pitch variance and averages over 12 segments and Timbre variance and averages over 12 segments) and lyrical features (5000 word counts for each track). The 5000 top words provided in MusiXmatch are stemmed words and we decided to proceed with this since it was easier. The 40,000 tracks contained 21105 tags for 'rock', 22428 tags for 'pop', 12663 tags for 'alternative', 15729 tags for 'indie' and 5810 tags for 'electronic'. Each tag is given as a percentage for eg. a song can be 70% rock and 20% alternative or 50% pop, 30% indie, 5% alternative etc. (the percentages do not have to add to 100% since they denote how much a song belongs to a particular genre). The number of tags add up to more than 40,000 since many tracks have two or more tags. We proceeded create 3 different datasets from our data: one that just consisted of the 57 audio features, one with all the 5057 features (audio + lyrics) and one that contained just lyrical information. We used the Waikato Environment for Knowledge Analysis (Weka) to perform feature selection on our training data in the second and third case. We used 5 fold cross validation along with the Information Gain filter to select the top 3000 attributes and then applied the same filter to the testing data. Our final data consisted of 40000 tracks with 57 features in the first case and 40000 tracks with 3000 features in the second and third cases.

3. The Approach

This is a multi-class classification problem since there are 5 possible labels. The genre tags are also not exclusive to a song i.e. there are many songs with two or more tags in the dataset and this complicates matters. We use a 80%-20% training-testing split for our models.

Given the size of our data, running models like KNN or kernelized support vector machines would have taken an enormous amount of time. We decided to use a generative model - Naive Bayes, since it is a good baseline for text classification (analogous to our current problem) and is highly scalable and computationally efficient. Since the lyrics have been provided as a bag of words model, we ruled out using decision trees or random forests since one cannot split on a word. For discriminative models, we use Multinomial Logistic Regression and a Support Vector machine that accepts for an input a precomputed kernel. This precomputed kernel is computed using cosine similarity which is a commonly used metric for text/document classification. We ruled out using a radial basis or polynomial kernel since they were computationally prohibitive

and when run on a small subset of data, they actually performed worse than a cosine similarity kernel.

3.1. Proposed Accuracy Metric

One of the reasons why previous work hasn't yielded very good results is because oftentimes audio features may not be vey indicative of a genre. Values for timbre and loudness may only marginally differ from genre to genre and a lot more of audio features as well as instances are probably required for better classification. Furthermore, even within the 5000 words, most songs have word counts for a lot of these as 0. Lastly and most importantly, a song can have multiple tags which is another hurdle. For the purpose of overcoming these drawbacks, we have proposed a novel way for calculating the accuracy score in this case as outlined by the Algorithm 1. Label data is available to us as percentages for all genres. We felt that the model should not be penalized if it does not return the genre with the highest percentage. What we proposed is that classification for a particular instance will be considered correct if either of the two genres with the highest percentages are returned. This accuracy metric is fair because there are a number of songs which can fall into multiple genres, especially looking at the close correlation between some of the genres. To classify a song as only one of the genres in this case is technically incorrect. We also feel that this metric is not too relaxed since there are also songs which have 3 or more genres.

3.2. Audio Features

As a first run which would serve to justify our motivation for choosing this topic, we ran the various mentioned models on the dataset comprising of just audio features. The results are tabulated and provided in Table 1.

3.3. Audio Characteristics and Lyrical Features

As can be seen from the provided results in Table 2, we obtain a significant lift by using lyrics in conjection with audio features.

3.4. Lyrical Features

Once we had observed that lyrical features added to a model comprising of just audio features yielded superior results, we wanted to evaluate how good models comprising of just lyrical features were. Our first step was to extract tf-idf vectors from the given data. We applied sublinear_tf scaling and enabled inverse-document-frequency reweighting. This reduced the impact of commonly occuring words like 'I','a' and 'the' and boosted the weights for uncommon words that would help better differentiate between genres. Following this, we perfomed dimensionality reduction via supervised and unsupervised methods in order to obtain high variance components which we would then use to transform our data. For unsupervised dimensionality reduction, we used Pincipal Component Analysis with 500 components. For supervised dimensionality reduction, we experimented with Linear Disciminant Analysis (LDA) and Canonical Correlation Analysis (CCA). While LDA and CCA yielded better training accuracies (as expected since the labels are an input while fitting), testing results using PCA yielded higher accuracies. The results run on various models are tabulated in Table 3. Since lyrical features gave us the best results out of the three experiments we ran, we decided to build an ensemble model using Naive Bayes, multinomial Logistic Regression and the SVM using a cosine similarity kernel. We used the Adaboost multi class classifier provided within sklearn and ran it for 600 boosting iterations. We used the SAMME algorithm since we were using class predictions of our learners.

3.5. Algorithms

Algorithm 1 Modified Accuracy Metric

Input: LabelData y_i, Size $\{n, d\}$
for i **in range** n **do**
  Find (indices,values[indices]) for top two genres
  Store in arrays Indices and Values of Size {n,2}
  PredictionIndex = argmax(Prediction[i,:])
  if PredictionIndex **in** Indices[i,:] **then**
    check = argmax(Indices[i,:] == PredictionIndex)
    if values[i,check]$\neq$ 0.0 **then**
      Correct
    end if
  end if
end for

4. Tables

Table 1. Classification accuracies for Audio Features Dataset

MODEL	TRAIN ACC.	TEST ACC.
NAIVE BAYES	0.24	0.22
SVM	0.35	0.31
LOGISTIC REGRESSION	0.42	0.39

Table 2. Classification accuracies for Audio & Lyrical Datasets

MODEL	TRAIN ACC.	TEST ACC.
NAIVE BAYES	0.46	0.37
SVM	0.45	0.37
LOGISTIC REGRESSION	0.51	0.42

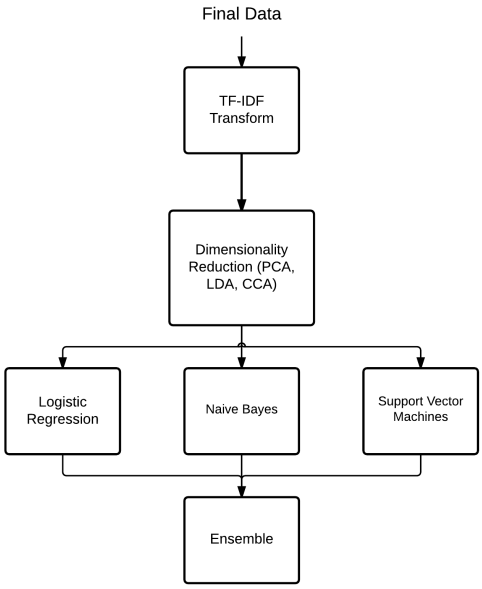

Figure 2. Approach and Models

Table 3. Classification accuracies for Lyrical Dataset

MODEL	TRAIN ACC.	TEST ACC.
NAIVE BAYES	0.62	0.56
SVM	0.69	0.50
LOGISTIC REGRESSION	0.65	0.54
ENSEMBLE(ADABOOST)	0.68	0.59

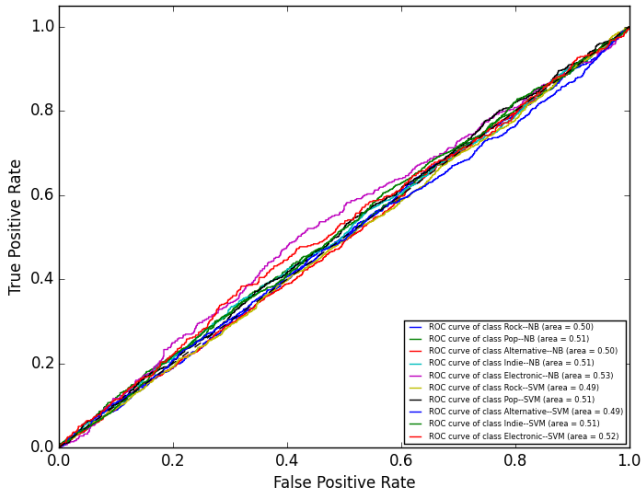

Figure 3. Genre Classification ROC for NB and SVM

5. Conclusions

Predicting randomly would result in an accuracy of about 20% and this has been bettered by all three experiments. We set out to explore whether lyrics can assist in genre prediction and the results we obtain are promising. Lyrical features provide a significant lift when added to a dataset containing just audio features and they perform even better on their own as evident in testing accuracies of 42% compared to 39% with only audio features. One of the reasons we feel this is happening is because audio features are not able to successfully differentiate between genres. Genres such as rock and alternative can oftentimes have similar pitch and timbre values while even pop and indie music these days have the full accompaniment of guitars, bass and drums. We speculate that lyrical information can help in such cases especially after tf-idf since these genres differ when it comes to song writing. An example could be that while rock bands tend to put in more of an effort into songwriting, pop songs tend to have trivial or nonsensical lyrics since the focus is on the catchiness of the tune. Using an ensemble model with just bag of words as our feature set yielded an accuracy of 59% which is the most when compared to results in our literature review. We feel that using a large dataset and our proposed accuracy metric are the primary driving factors behind this improved performance. We feel that this is a good positive step in the field of music information retrieval and specifically in genre prediction. Future work could involve extending the bag of words model to a much larger feature space than

5000, using unstemmed words instead of stemmed and increasing the number of instances in the training dataset. We feel that all these steps may further improve accuracy and increase the robustness of the models.

References

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 1st edition, 2007.

Bou-Rabee, Ahmed, Go, Keegan, and Mohan, Karanveer. Classifying the subjective: Determining genre of music from lyrics. Technical report, Stanford University, 2012.

EchoNest. Example code for processing the million song dataset and other big music datasets, 2011. URL https://github.com/echonest/msd-examples.

Kader, Hussain and Yates, Robert. Genre prediction via lyrical analysis. Technical report, Stanford University, 2013.

Lamere, Paul. How to process a million songs in 20 minutes, 2011a. URL http://musicmachinery.com/2011/09/04/how-to-process-a-million-songs-in-20-minutes/.

Lamere, Thierry Bertin-Mahieux & Daniel P.W. Ellis & Brian Whitman & Paul. Million songs dataset, 2011b. URL http://labrosa.ee.columbia.edu/millionsong/.

Liang, Dawen, Gu, Haijie, and OConnor, Brendan. Music genre classification with the million song dataset. Technical report, Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, 2011.

PennMSD. Million songs dataset mapreduce, 2014. URL http://pennmsd.s3-website-us-west-2.amazonaws.com/.

Sotomayor, Borja. mrjob and s3, 2013. URL http://www.classes.cs.uchicago.edu/archive/2013/spring/12300-1/labs/lab5/.

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman and Lamere, Paul. The million song dataset. In *In Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011.

Tzanetakis, G. and Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.*, 10 (5):293–302, 2002.