Inheritance Relationships in Vernacular Hip Hop Music

Tim O'Brien* Spencer Salazar†

Center for Computer Research in Music and Acoustics (CCRMA), Stanford University

December 13, 2013

Introduction

All forms of music employ extensive borrowing and reappropriation of popular themes, tropes, and styles. These individual actions form of a complex network of genealogical interrelationships between individual songs, artists, and subgenres. These effects are especially inherent in hip hop music, in which sampling and borrowing are foundational elements of the musical culture itself. [1] Our work attempts to analyze these inheritance relationships quantitatively using a machine learning approach, guided by previous efforts to this end [1–3]. To date, we have focused on uncovering similarity via audio content analysis, as the pairing of a similarity metric and release dates between two songs, for example, allow us to transform the scalar similarity value into a directional influence vector.

Data set

Our final data set contained roughly 400 songs manually segmented into four hip-hop subgenres: East Coast, West Coast, Southern, and Hyphy. The selection of tracks and artists was chosen so as to be manageable yet representative of the four constituent genres. The genres chosen contain both the foundational and pervasive (most exemplified by East Coast and West Coast) as well as the more regional (most exemplified by Hyphy, a San Francisco Bay Area hip hop movement). Each song was also manually tagged with artist information.

Metadata and "ground truth"

While there is no absolute, objective ground truth for musical influence, we used the AllMusic.com/Rovi "influencers" tags[1] (freely available for non-commercial use via the Rovi API[2]) as our ground truth benchmark. This data lists artists deemed by "experts" to have influenced the artist in question, and contain a confidence/importance weighting in the form of an integer in the range of 1-10. We used a Python script to compile the influences and associated weightings of each artist in our data set via the aforementioned API. We then constructed an influence table for each artist in our data set, for comparison to results from our computational analysis of artist influence.

ML algorithm

We build upon previous machine learning applications. Collins [2, 3] employed k-means clustering on relevant feature vectors as well as relevant date data as input into a predictive (PPM) algorithm to derive an approximation to influence relations. We start out more simply with an SVM for multi-class classification, using libsvm-3.17 with a radial kernel, and the nu-SVC type of SVM. The multi-class classification is

*tsob@ccrma.stanford.edu

†spencer@ccrma.stanford.edu

[1] `http://prod-doc.rovicorp.com/mashery/index.php/Data/name-api/v1.1/name/influencers`

[2] `http://developer.rovicorp.com/docs`

accomplished by training $k(k-1)/2$ SVMs, where k is the number of classes in the system. In our case $k = 4$, so the algorithm trains 6 SVMs. Each SVM trains on two of the classes. Fig. 1 shows the decision values for each of the 6 SVMs. One can easily see that the first SVM, shown as the dark blue bars in the chart, trains East Coast vs. West Coast data. The second, shown in lighter blue, trains East Coast vs. Hyphy, and so on. Of interest are the values of the decision boundaries for the genres on which each SVM was *not* trained. For example, the yellow bar in Fig. 1 shows the decision boundary for East Coast data in the West-Coast-vs-Hyphy SVM leans toward West Coast.

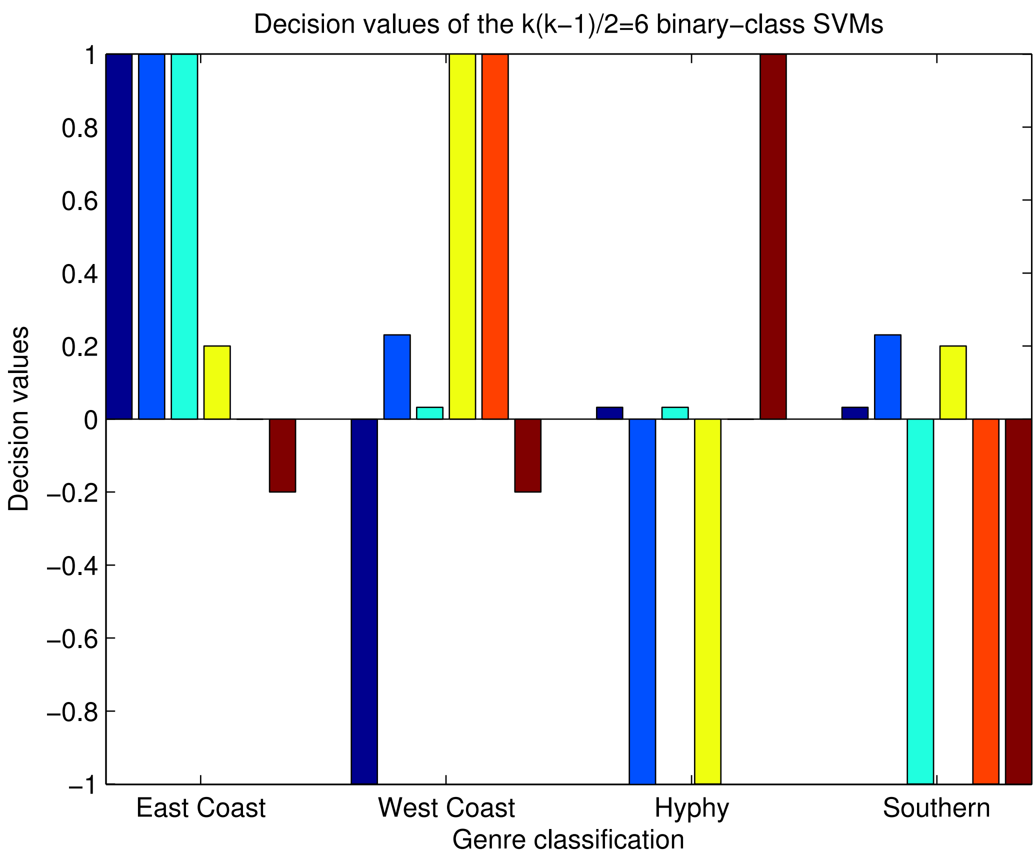

Figure 1: SVM decision values using nu-SVC and a radial kernel

Further exploration suggested the need for our own multi-class SVM classifier, using libsvm as a simple two-class classifier. In this way, we constructed a single model for each genre, using that genre's training examples as positive examples and the rest of the training examples as negative examples. The goal in this modeling is to capture the distinguishing features of a given genre, compared to its peers, in a single model, and be able to simply ascertain how well any given input song fits that model. Thus, rather than generating a label, we wish to see the specific probabilities associated with classifying a test example under a given model. Fig. 2 shows the results of this strategy. Similar to the previous algorithm, each color represents a model trained on a specific arrangement of the data set; each model uses the same data set but with different labelings. Effectively, each model is asking, "is this genre X or is this something else?" With each model, we classify every training example under it, and note the probability of that training example being a member of the modeled genre. Fig. 2 shows the mean probability of a training example being classified by a given genre-model, grouped by genre.

Here we can see additional nuances revealed by our analysis. Hyphy appears to have be the most distinctive genre under this algorithm, in that its examples have a much higher probability of being hyphy than anything else, but also have the lowest probability of self-classification, suggesting high levels of internal variation. In contrast, West Coast and Southern genres seem to be the least distinctive compared to the general training set, as the probability of these genres under each genre-model are similar.

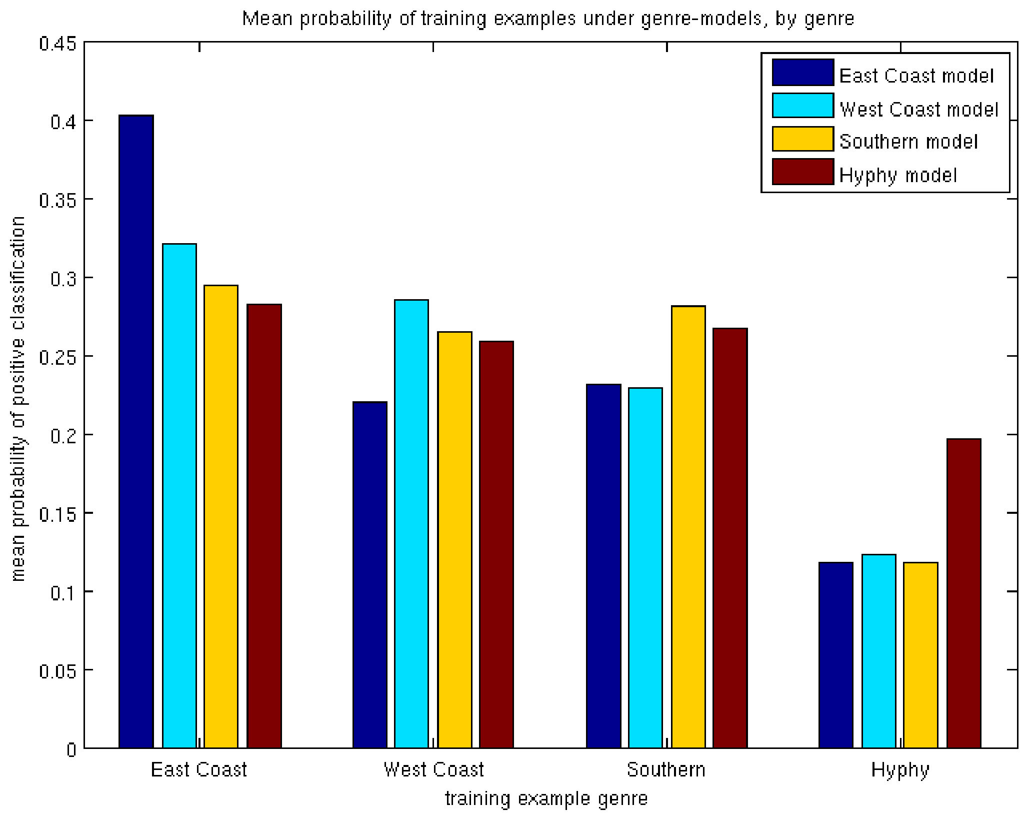

Figure 2: Genre probabilities using C-SVC and radial kernel

We also conducted a machine analysis comparing the oeuvres of individual artists. For this we selected 8 generally influential artists spanning each genre. Comparing musical influence at the level of the artist allows us to more readily verify our approach with empirical influence data, our ground truths discussed above. We utilized the exact same algorithm as with our genre analysis.

Figure 3 shows the result of this study (note all probabilities are adjusted to log-scale). Perhaps most validating of our model is the influence of Dr. Dre suggested by our results. Dr. Dre is widely regarded as one of the most influential producers and artists in the hip-hop genre, an influence accorded by both popular consensus and our ground truth data. Other notable results are the influence E-40 to Mac Dre. A relationship between these two is expected, as both are considered primary exemplars of the hyphy hip-hop genre. However, Mac Dre's primary hyphy works were released before E-40 became well-known as a hyphy artist, so one would instead expect an influence from Mac Dre to E-40 rather than vice-versa as our results suggest. This suggests an admitted flaw in our algorithm, that the direction of influence is not adequately accounted for. Though with chronological data, we can potentially correct the directionality of such misdirected influences.

Log Probability of Test-set Positively Identified by Model

test set \ model	Dr. Dre	Jay-Z	Mac Dre	Nas	Nelly	E-40	Notorious BI	T.I.
Dr. Dre	0.89534	0.56667	0.61906	0.46381	0.39175	0.62162	0.58124	0.48397
Jay-Z	0.59255	0.97393	0.57292	0.17686	0.32	0.63693	0.50443	0.46595
Mac Dre	0.57124	0.56663	0.87973	0.30465	0.34162	0.67554	0.53119	0.46749
Nas	0.58755	0.45941	0.58574	0.97612	0.38774	0.28891	0.57902	0.52309
Nelly	0.63381	0.52768	0.48189	0.46353	0.98154	0.58435	0.59426	0.37407
E-40	0.56191	0.56424	0.62193	0.081522	0.33218	0.95976	0.48573	0.48492
Notorious BIG	0.60571	0.50941	0.57666	0.50928	0.40951	0.52113	0.95363	0.44516
T.I.	0.57215	0.53792	0.54212	0.43588	0.1965	0.657	0.52615	0.96546

>1 σ from mean

> mean

AllMusic.com influences data ("ground truth")

artist \ influenced by	Dr. Dre	Jay-Z	Mac Dre	Nas	Nelly	E-40	Notorious BI	T.I.
Dr. Dre	1	0	0	0	0	0	0	0
Jay-Z	0	1	0	0	0	0	0	0
Mac Dre	0.8	0	1	0	0	0	0	0
Nas	0	0	0	1	0	0	0	0
Nelly	0.5	0	0	0	1	0	0	0
E-40	0.9	0	0	0	0	1	0	0
Notorious BIG	0	0	0	0	0	0	1	0
T.I.	0	0.9	0	0	0	0	0	1

Figure 3: Artist probabilities using C-SVC and linear kernel

SVM Parameterization

How our analysis goals influenced usage and parameterization of the various SVM algorithms is worth further explanation. In particular, our goals are different from the typical design goals of an SVM. Rather than perfectly separate two classes of input data, we wish to further examine the situations in which the SVM fails, or the extent to which it comes close to misidentifying an input datum. Effectively, our results *are* the probability of misidentification. Therefore, we leaned towards SVM parameters which would increase the dynamic range of classification probabilities. We found that, in general, a radial kernel positively classified test examples within of the model-genre with near perfection, and likewise for non-model-genre test examples; such precision was suboptimal for the information we were looking to find. Ultimately we arrived at using C-SVC (i.e. the "regularized SVM" presented in lecture) with a linear kernel and a high C value, which appeared to maximize testing error of non-model-genre testing error. In this way we extracted as much information as possible from a potential classification while still maintaining low error in classifying examples from the model-genre.

Feature selection and extraction for audio content analysis

We plan to structure our features similar to Collins; by dividing features into three groups focusing on timbral, rhythmic, and harmonic audio features, respectively, we can draw more specific conclusions of similarity and influence based on these classes of features and identify the salient musical characteristics of particular sub-genres of hip-hop music.

Preliminary features were extracted in Matlab using the MIR toolbox [4]. The initial choice of features was very minimal and not yet differentiated based on musical dimension as mentioned above. Our initial features, for each song, consist of:

- Spectral centroid mean and variance
- Spectral roughness mean and variance
- Spectral flux mean and variance
- Spectral rolloff mean and variance
- Mel-frequency cepstral coefficients (MFCCs) mean and variance

Spectral centroid is simply the mean of frequency energy in a given audio segment. Spectral roughness is an approximate measure of harmonic dissonance within an audio signal. Flux is a measure of difference between the frequency content of successive frames of audio. Rolloff indicates the frequency where 85% of the spectral energy is at frequencies less than that frequency. MFCCs are the discrete-cosine transform of the mel-scale log magnitude spectrum; the mel-scale is a logarithmic frequency scale which corresponds better to human perception of frequency than the standard linear frequency scale. The MFCC is a vector quantity; typically only the first 13 values of this vector are used, as most of the information in this feature is contained in these values. Intuitively, MFCCs are a compact representation for the perceptually-relevant periodicities present in an audio signal, corresponding in a general sense to many timbral qualities. For each song, each audio feature was calculated across successive 10 millisecond frames. The audio feature was then averaged across all frames in the song to produce a single mean and variance for any given song, which across the space of audio features comprise the input features for our learning algorithms.

Initially we wished to use temporal features (particularly tempo/beats-per-minute) in addition to the features described above. However, using MIRToolbox, producing these features across our entire dataset was significantly time-consuming (approximately 20 hours per genre). This is not prohibitively expensive for a one-time batch feature extraction. However, in the context of an iterative algorithm development where we may need to calculate and re-calculate features a number of times, such a lengthy turnaround time becomes disadvantageous. Therefore temporal features were not used in this analyses.

References

[1] N. J. Bryan and G. Wang, “Musical Influence Network Analysis and Rank of Sample-Based Music.,” in *ISMIR*, pp. 329–334, 2011.

[2] N. Collins, “Influence in Early Electronic Dance Music: An Audio Content Analysis Investigation.,” in *ISMIR*, pp. 1–6, 2012.

[3] N. Collins, “Computational Analysis of Musical Influence: A Musicological Case Study Using MIR Tools.,” in *ISMIR*, pp. 177–182, 2010.

[4] O. Lartillot and P. Toiviainen, “A Matlab toolbox for musical feature extraction from audio,” in *International Conference on Digital Audio Effects*, pp. 237–244, 2007.