Comparing Sequential and "Bag of Sections" Techniques for Musical Genre Classification

Keegan Poppen

December 17, 2011

1 Introduction

A lot of genre classification work to date has gone into exploring features that demarcate genre either at the lowest (sample) level or the highest level (overall acoustic characteristics of the song). Features evaluated at the sample level are more concerned with sonic qualities like timbre and pitch, whereas higher-level features tend to focus on key, mode, tempo, and other descriptive features of the whole song.

This paper focuses instead on the space between these two strata in order to evaluate the extent to which structural-level features of music (e.g. chorus or verse level) delimit particular musical genres. The challenge with this level is that one of the most important attributes of musical composition structures is their ordering, and their various repetitions throughout a particular song. As such, this paper compares different models that variously do or do not take sequencing into account in order to quantify the importance of song structure in determining genre.

2 Related Work

Automatic genre classification is by no means a new and unexplored field, and has been tackled via a number of different methods, and to varying degrees of success. Generally speaking, prior work has fallen into two broad methodologies: using higher-level features of individual songs as features to help identify genre [1, 2] and evaluating features on small time slices of the

raw music data (e.g. 10ms sliding windows) to try and integrate the causality of attributes of very small slices of the the song in genre classification [3, 4]. While both of these areas have shown progress and fairly good (and improving) results, performance still trails behind human genre classification[5].

3 Experiment

The dataset used throughout this project is the Million Song Dataset of Bertin-Mahieux[6] et al., from which I was able to get various metadata about the song, from which I was able to derive the gene of the song (details below), and, importantly, the timing information of the various sections of each song.

In order to evaluate the comparative effect of section sequence information, I trained models that are partitioned into two groups– models that accounted for section ordering and models that did not– with the respective monikers "Sequential" and "Bag of Sections". I also trained a nave model that only considered some higher-level information about the song (duration, key, pitch, tempo, and mode).

The "Bag of Sections" models consisted of an SVM with a linear kernel (as was the baseline) and logistic regression, and the "Sequential" models evaluated were a recurrent neural network (with Long Short-Term Memory (LSTM) nodes), and a linearly-interpolated n-gram mixture language model (copied without modification from a previous CS224N assignment). The training data for the models were partitioned into an 80/20 split of training to test data.

Using the section information from the Million Song Dataset, I partitioned each input song into sections (empirically there were O(10) sections per song) that served as the input data for each model. Each section was then evaluated on 28 different features, including the magnitude of all 12 pitches in the chromatic scale, 12 different acoustic features (inspired in part by the given dataset, but evaluated on the section level, rather than the sample level provided), loudness, tempo (measured by the number of beats in the section), and estimated key (provided by dataset).

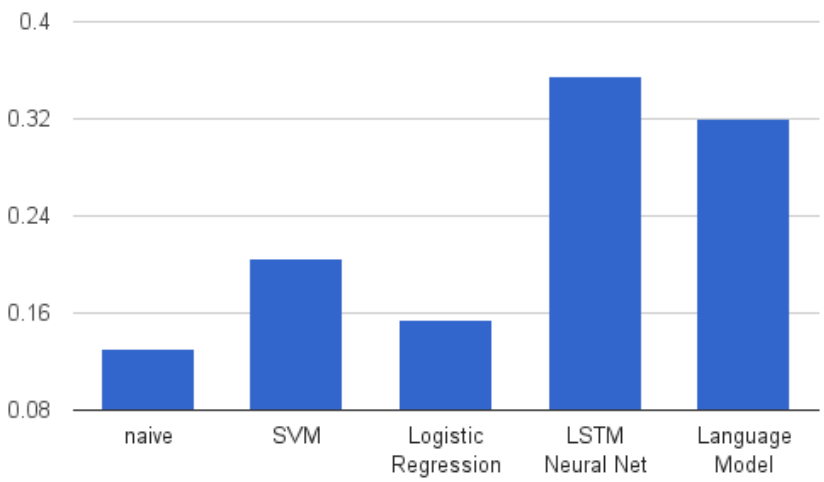
Each feature in these 28-dimensional vectors were normalized by subtracting the feature mean and normalizing to the range $[0, 1]$ for all quantitative features (e.g. not key). They were then given as input to the models. For the "bag of sections" models, each section was tagged with the genre of the song

from which it came and given directly as input; for the sequential models, the sequence of feature values for each section was tagged with the genre of the song. For the language model, a different language model was trained for each genre, and classification was done by choosing the classifier that maximizes the log probability of the input vector.

Since the original dataset only gave Last.fm tags for each song, I decided to determine the genre for each song by first picking a canonical set of genres– pop, rock, alternative, hip hop, classical, country, reggae, and folk– then searching the top 10 tags for each song (in popularity order) to see if they indicated (using a fuzzy string match with the genre name) a particular genre, in which case I assigned the matching genre to the song. If no genre was found, I just ignored it altogether and chose the next available in the song (since I was using only 10,000 of them).

4 Results + Discussion

The best results from training each model (of all configurations attempted) can be seen in the chart below:



As expected, the naive model does not do particularly well (.130), barely scraping by random chance (.125). One thing that stands out is that the "bag of sections" model far far worse than the sequential models. Although

the SVM fared better than the logistic regression, ultimately neither of them was able to find a good decision boundary. It should be pointed out, however, that both of these models also have linear decision boundaries, and given the high bias of the results (the training errors for SVM and Logistic regression were .77 and .85 respectively), it is possible that this was the fundamental problem, rather than sequential information.

The sequential models fared comparatively better, hitting 35% and 32% accuracy on the test set (40% and 35% respectively on the training set). The configuration that fared best for the neural network was having a single LSTM node in sole hidden layer, with 30 input nodes and 10 outputs into a softmax node that determined which class a particular song belongs to. Adding additional nodes in the hidden layer dramatically reduced performance (even adding one additional node took accuracy to 20%), as did, to a lesser extent, adding additional layers– two LSTM one-node layers were 5% less accurate on the test set (and increasing with more layers), although they did increasingly well on the training set, indicating that they were over fitting the training data (more on this later).

The language model performed worse than expected, which can probably be traced back to the initial vocabulary being to large. In order to make the classification tenable, I bucketed each feature value into 5 buckets, and then trained the model based on all of these possible vectors, allowing for a similarly-fuzzy matching algorithm that allowed for similar vectors to be considered to be the same. A far more robust solution would have been to cluster the sections into some number of section types, then use that more finite vocabulary to train a language model. This would help to insulate the model from outsize amounts of noise in the data.

Another observation is that none of the results are particularly good, especially compared to human genre classification (and even the current state of the art). All of the models exhibited fairly high bias, despite attempts to add additional features and additional data (such as adding additional tagged attributes to each section). As I discuss later, there are a number of potential causes for this, and areas for future investigation.

5 Conclusion

While these new methods do not achieve results comparable with the modern state of the art of automatic genre classification in their own right, signs

seem to point to them adding information to the classification task that is not redundant with current sample-level and song-level methods. As such, while not investigated too thoroughly within the scope of this paper, it seems plausible that features in this space could make meaningful contribution to genre classification accuracy (as well as potentially serving as features in other domains, including music recommendation and identifying song similarity.

Some areas that would be interesting to investigate further including finding ways to add additional features. This could be accomplished via feature learning techniques, such as Lee et al.[3], or by finding more meaningful features on the section label (e.g. if a section is a chorus or a verse). Additionally, it would be interesting to integrate additional data streams such as song lyrics and mood to see if that would improve results.

References

[1] Mandel, Michael I. and Daniel P.W. Ellis, Song-Level Features and Support Vector Machines For Music Classification. In *International Symposium on Music Information Retrieval*, 2005.

[2] McKay, Cory and Ichiro Fujinaga, Automatic Genre Classification Using Large High-Level Musical Feature Sets. In *International Symposium on Music Information Retrieval*, 2004.

[3] Lee, Honglak et al., Unsupervised Feature Learning for Audio Classification Using Convolutional Deep Belief Networks. In *NIPS*, 2009.

[4] Colonnese, Nick, Classifying Parts of Songs. *CS 229 Project*, 2009.

[5] Lippens, S et al., A Comparison of Human and Automatic Musical Genre Classification. In *ICASSP*, 2004.

[6] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In *ISMIR*, 2011.