Video Game Genre Classification Using Soundtrack

Bojiong Ni, Zhiming Shi, & David Wugofski
bojiong@stanford.edu, zhimings@stanford.edu, & dwugo@stanford.edu

Problem

In this project, we attempted to identify the genre of a video through the game's soundtrack. Given a dataset of video game soundtracks labeled with the game genre, we use machine learning methods to predict the game's genre. We compared to general solutions: a 2-step solution classifying the songs directly and inferring game genre, and a 3-step solution aggregating the song features before classification.

Data

For the project, we used video game soundtracks provided by the Mirsoft database [1]. This database contains the soundtracks of 7900 genre-labeled video games, with 1349 video games containing music in formats we can easily use in our project.

Features

We compared the performance of our learning algorithms using two types of feature vectors: individual song features and aggregate features. The individual song feature vectors contained 46 features extracted from each song file relevant to genre classification. The aggregate features contained the averages and standard deviations of the features of all the songs in the dataset as well as 6 additional features designed to capture the distribution of the song vectors, for a total of 98 features.

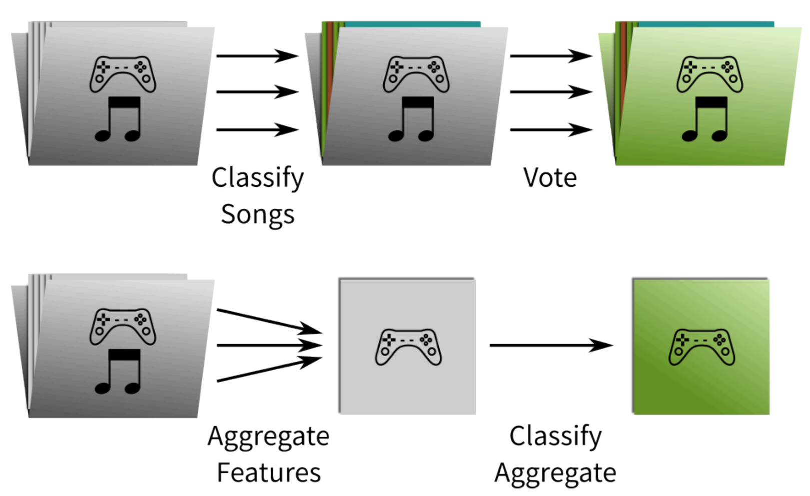

Figure 1: Overview of our 2-step voting (top) and 3-step aggregation (bottom) methods for classification

Challenges

The most immediate challenge in solving our problem is the decision of how to aggregate our song feature vectors into one feature vector for the video game. First we used the basic model applied to song analysis for genre classification laid out in [2]: create two features — mean and standard deviation — for each feature in the prime feature vector. We then modeled each game as having songs belonging to discrete clusters — where each cluster corresponds to a situation the song would be played in during the game (action sequence, menus, cutscenes, etc.) — and then added a feature denoting the probability the game would have a song in each cluster.

Learning Algorithms

We compared the results of five different learning algorithms:

- *Softmax* (Softm) assumes a multinomial distribution of feature vectors
- *Gaussian Naive Bayes* (NB) assumes our feature vectors are conditionally independent given game genre, with probabilities modeled by Gaussian distributions
- *SVM 1 vs. 1* (SVM1v1) applies SVM algorithm producing classification results comparing the probability of a feature vector belonging to a genre compared to it belonging to each other genre
- *SVM 1 vs. All* (SVM1vAll) applies SVM algorithm producing classification results comparing the probability of a feature vector belonging to a genre compared to it not belonging to the genre
- A *Deep Neural Network* (DNN) algorithm with three layers with ten, twenty, and twenty hidden units in each layer

Results

For our evaluation, we measured the accuration of different learning algorithms with our dataset. We computed the accuracy of both our 2-step approach ("Vote") and our 3-step approach ("Agg") attempting classification with 2, 3, and 5 different classes.

Learning	2 Classes		3 Classes		5 Classes	
Method	Vote	Agg	Vote	Agg	Vote	Agg
Softm	60.5%	60.5%	61.4%	61.4%	25.3%	42.1%
NB	55.3%	50.0%	49.1%	50.9%	25.3%	29.5%
SVM 1v1	50.0%	60.5%	40.4%	47.4%	22.1%	26.3%
SVM 1vAll	50.0%	60.4%	33.3%	36.8%	33.7%	22.1%
DNN*	—	—	52.6%	45.6%	20.0%	34.7%

Table 1: Test accuracy measure of the different learning algorithms for our different setups *DNN did not support binary classification

Discussion

Our results demonstrate promise in using video game soundtracks to predict video game genre using machine learning, especially using a Softmax learning algorithm. As well, using aggregated features as opposed to voting improved our results most of the time. It may be worth considering using alternative forms of aggregation than the ones we used here.

Additionally, we should note that with DNN, we occasionally observed high variance on our trainging set, as illustrated in Figure 2. It may be possible to improve the DNN performance by choosing our aggregate features more conservatively and/or increasing the size of our dataset so we can use more training samples.

Future

Given more time, we would like to work on increasing our dataset size and pursuing other ideas of feature aggregation. We could work on tracking down software that will help us process the audio files of the video games in [1] whose audio formats we did not recognize. Additionally, building on the "Discussion" section, we believe we can improve the accuracy of our project by pursuing some alternative methods of feature aggregation we have devised but were unable to implement.

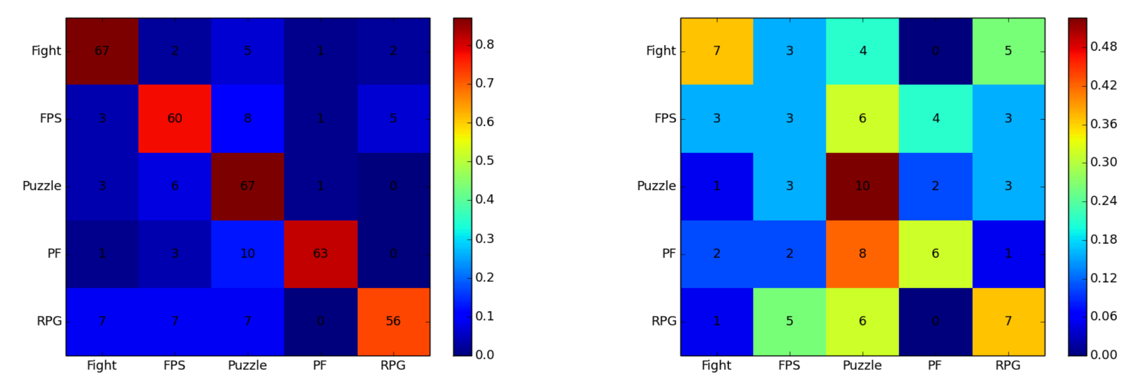

Figure 2: Confusion matrices for DNN algorithm based on aggregate features: training (left) and test (right)

References

[1] (2016). Game music database, World of game Music, [Online]. Available: http://www.mirsoft.info/gmb/index.php.
[2] C. Silla, A. Koerich, and C. Kaestner, "A machine learning approach to automatic music genre classification", Journal of the Brazilian Computer Society, vol. 14, no. 3, 2008.