Music Genre Classification Using Machine Learning Techniques

Sam Clark
Danny Park
Adrien Guerard

5/9/2012

Abstract

Music is categorized into subjective categories called genres. Humans have been the primary tool in attributing genre-tags to songs. Using a machine to automate this classification process is a more complex task. Machine learning excels at deciphering patterns from complex data. We aimed to apply machine learning to the task of music genre tagging using eight summary features about each song, a growing neural gas, and a neural network. We hypothesized that the growing neural gas would improve the classification accuracy of the neural network by both reducing noise in the input data and at the same providing more input data for the network to work with. Our dataset consisted of 200 totals songs, evenly distributed across rap, reggae, classical, and country genres. Our results supported the hypothesis that the growing neural gas would assist the neural network in this task. Combining a song's two closest model vectors from the growing neural gas with a song's feature information improved the training accuracy from 77 percent to 88 percent and the testing accuracy from 66 percent to 68 percent.

1 Introduction

1.1 Background

Music is a universal language and symbol that we create,understand, and enjoy–alone and in groups. It comes in nearly infinite forms–alternatively called *genres*. There are universal features of music that all forms have, regardless of their rhythm, form, scoring or timbre. Thus we can abstract a song into a series of values and it becomes a data point that can be analyzed.

We postulate that we can determine the genre–musical category–of a song using these features. The high dimensionality of the data representing a song makes this a complicated task. Machine learning, specifically neural networks, specialize in deciphering patterns from complex datasets thus we thought that using machine learning to try to learn how to distinguish between different genres of music could be effective.

Typically, applying a genre-tag to a piece of music has been a human task. As genres are a human abstraction, it is relatively easy for us to listen to a piece of music and report what category it belongs to. It is much more difficult for a machine as the entire human listening experience is simply represented by a vector of features about the song.

1.2 Related Work on Music-Genre Classification

Classifiying musical genres is a subjective taks. Robert O. Gjerdingen and David Perrott discovered that participants correctly matched the genre of a song 70 percent of the time after hearing the song for 3 seconds.[6] These results indicate that although there may be a general agreement of what types of genre categories exist, the boundaries separating those genre categories are blurry and unique to each individual.

Although humans clearly show some measure of success, we believe that machines can assume this role with better accuracy and less effort by creating a feature model of each genre based on certain attributes of each song.

Using Gaussian mixture models and diagonal covariance matrices, George Tzanetakis and Perry Cook achieved 61 percent classification accuracy with ten genres.[5] The 3 features they used for classification were timbrel texture, rhythmic content, and pitch content.[5] Tzanetakis and Cook's results were comparable to those of human classification, although not quite as good. We believe that we can more accurately predict musical genres by exploring a combination of two machine learning methods: Growing neural gas (GNG) and Artificial Neural Networks (NNet).[5]

2 Experiment

2.1 Data Collection

We heavily debated the source of our data for this experiment. The primary focus was generating an evenly distributed data-set of songs from several different musical genres. We chose to run the experiment on four genres that logically would work well with our system–more specifically genres with distinct sounding music. For example if we chose rock, alternative-rock, country, and pop as our four genres– we predicted that the system would not work as well because those four types of music are very similar. Preliminary testing of our system with a sample from each of these genres confirmed this. To try to maximize the effectiveness of the system, we chose the following four genres: classical, country, rap, and reggae. These genres then had to be used to create an unbiased (no human interference) , well distributed (good representation of the genre) sam-

ple of songs from each genre. To do this we used **LastFM**.

LastFM is an online music streaming service where a user makes an account and then can listen to many different stations based upon categories such as genre, artist, year, special events, and many others. Conveniently, they have stations for each genre-tag such as Classical-tag radio, Rap-tag radio, etc.[2] Using the LastFM developer API we were able to extract all the tracks that a certain user has listened to. Thus we created accounts for each of the four genres, listened to those stations to accumulate songs from that genre, and then pulled the list of artist song pairs as our dataset.

We used **Echonest** to retrieve the feature information about each song. Echonest is an online music library containing 30 million song from 1.5 million artists.[1] Additionally, it has a developer API that allows to both search for a song and retrieve important feature data for it. This data comes in the form of an audio_summary object. For each song in the dataset, an audio_summary object (see Table 1) was retrieved from Echonest..

2.2 Built-in Echonest Functions and Feature Normalization

Five of the eight song features we are using in classification are raw values taken directly from the song that we then normalize between 0-1. For example to normalize the time signature, the possibilities ranged between 0-7 so we took the time signature of the song and divided it by 7.

Loudness, danceability, and energy are values created by our music database, Echonest. While they do not release the exact formulas, they do give a top-level description of these song attributes. *Loudness* is a combination of decibels and beat strength. *Energy* is a combination of loudness and segment durations. *Danceability* is a synthesis of beat-strength, tempo stability, and overall temp. The inclusion of these features in our project was controversial, however, the consensus was as long as the formula remained consisten across all songs, they would add to the richness of the data-set.[7]

Feature Name	Value
Info	Song
Time Signiature	0 - 1
Energy	0 - 1
Tempo	0 - 1
Mode	0 - 1
Key	0 - 1
Duration	0 - 1
Loudness	0 - 1
Danceability	0 - 1

Table 1: Song Feature Vector: gathered from Echonest audio_summary.

2.3 Machine Learning Techniques

Growing neural gas and neural networks have been successfully implemented together in previous experiments suggesting that they perform complementary tasks. For instance, Category-Based Intrinsic Motivation, a system invented by Meeden et al. in 2009, uses a growing neural gas to perform categorization and specialized neural networks for prediction.[3] We combine the two different methods for this task in the hope that the GNG can help the Neural network categorize the dataset.

2.4 Growing Neural Gas

A Growing neural gas is a specialized form of a machine learning structure known as a self organizing map. It takes inputs in the form of a set of feature vectors of any dimensionality and starts with two initial random vectors of the same dimension. It then iterates through the input vectors, adding each to the graph structure and computing the distance between

the vector being added and the existing nodes.[4]

Depending on the type of GNG, the graph will either add a new node if the error surpasses a certain threshold or after a certain number of steps. We used an equilibrium GNG which bases node addition on error. The algorithm aims to map data into clusters based upon the distance between the feature vectors and thus we thought it would be an appropriate tool to use in this experiment.

The purpose of the GNG in the scope of our experiment is to reduce the complexity of the music space for the neural network by abstracting some of the more important features into model vectors associated with each genre of music. Furthermore a growing neural gas allows for a visualization of our music space and can show us features are important to classifying genre. It can also tell us how close a song is to a specific model vector (genre) in the GNG.

2.5 Neural Networks

Neural networks are an important tool in machine learning and have been used to perform a wide variety of tasks. The two main components of training a neural network are feed-forward computation of activations and the back-propagation of error.

A simple neural network's architecture is defined by the user and connections between nodes are acyclic. The user sets each input node's activation, and then for each non-input node it computes its own activation by computing the weighted sum of the incoming activations from the preceding nodes it is connected to and then passing that weighted sum into a function whose bounds are zero and one (hyperbolic tangent is a commonly used activation function).

Gradient descent is then used to compute the error of each node (along with each weight), which then allows for the weights to be adjusted by small amounts in order to better produce the desired output for each input. The desired output for each input is needed in order to compute the error of each output node, after which each preceding node can use its successor's error.

Once the neural network has been trained, it is tested by performing feed-forward computation of activations and then comparing its output to the desired output.

2.6 System Description

Initially our system requires t number of total songs, g genres, and s songs for each genre where $s = t/g$. We then construct song vectors using each song‘s n attributes by stacking each of the n values in a specific order into a single vector which is the songs feature vector. All the song-attributes are normalized between 0-1 as required by the GNG.

Each song‘s feature vector is then used as an input to the GNG which runs for t steps to categorize the dataset.

A map of the song space is then created using the nodes from the GNG (model song-vectors) and the songs. The first and second closest model song-vectors are computed and stored for each song along with a confidence value that measures how close a song is to its closest model vector (one being close and zero far away).

At this point, we have a collection of songs along with their respective model song-vectors and confidence values. This information is then used to train a neural network using each song's data as inputs and the human-tagged genre for the target outputs. Once the neural network is done training, it is tested on songs not included in the training set.

Our system combines a GNG's ability to categorize the song-space along with a neural networks ability to be trained to classify user-defined cat-

egories. The former allows for songs to first be categorized in terms of their attributes, thus creating a map of the song-space. The information gathered from the self-organized map can then be used to train a neural network to classify user-defined genres. Through training, the machine genres created using the GNG, can be mapped to human genres, meaning our system is able to categorize a song by genre given its intrinsic attributes.

2.7 Neural Network Training

After compiling the feature vectors for each of the songs, we used them as inputs to our GNG and training data for our neural networkss. The GNG provided us with the following information about each of the songs: the song vector, the first and the second closest model vectors, and the confidence values for those mappings. With this information we decided to train our neural networks on seven different sets.

1. song attributes (8 inputs) 2. model vector (8 inputs) 3. model vector and song (16 inputs) 4. model vector and song and confidence value (17 inputs) 5. 1st and 2nd model vectors (16 inputs) 6. 1st and 2nd model vectors and song (24 inputs) 7. 1st and 2nd model vectors and song and confidence value (25 inputs).

2.8 Data-Visualization

The GNG was visualized by creating a graph object out of the GNG using a python library Networkx.[8] Then using another python library, matplotlib, we were able to create visual map of the music space as seen in Figures 1 and 2. We used matplotlib to visualize the results of the neural network as well.[9]

Genre	Country	Classical	Rap	Reggae	Total
Num Songs	50	50	50	50	200

Table 2: Complete data-set for GNG and Neural Networks

3 Results

3.1 Growing Neural Gas Results

The GNG performs slightly differently on each run as the feature vectors are passed in in a random order but the results were always similar in nature. As the chart shows, there were 12 model vectors created by the GNG and this was a consistent trend. Table 3 records the distribution of songs to model vectors. We can see a good distribution of genres: 3 reggae, 2 rap, 3 country, and 4 classical.

In many of our trials as we can see from the visualization of the GNG, we had an even distribution of genres. This spread of categorization is significant because if we had four genres in our dataset we ideally would get four model vectors, one for each genre. Our dataset is too large to alter the GNG parameters such that there would only be four model vectors thus having an even distribution of genre-model vectors shows that the GNG is successfully categorizing the dataset. Figure 1 indicates the success of the GNG in this task. We can see that generally each model represents a pretty uniform color distribution illustrating accuracy within this task.

3.2 Similar Genres More Likely to be Conflated by GNG

As shown by Figure 1 and Table 3, similar genres are more likely to be conflated with one another by the GNG. Specifically if we look at the rap and reggae nodes (green and red respectively) we can see that there is a splattering of red nodes connected to the

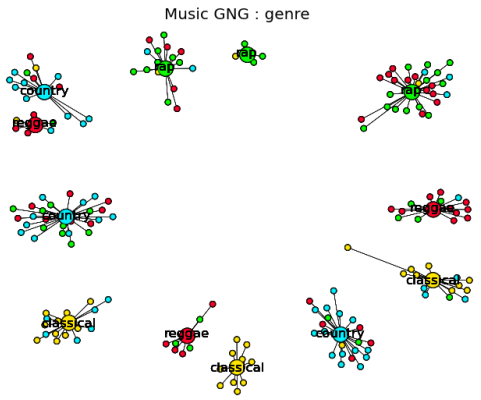

Figure 1: The GNG colored by *genre*. There is one color per genre. Each large node represents a model vector and each small node connected to the larger node by an edge represents a song that mapped to that vector

green models and similarly more teal country songs mapped to the yellow classical models.

This result is intuitive to the system. If the GNG is using a distance measure between vectors, then songs with similar feature vectors will map close to each other. Consequently rap and reggae, which one could subjectively argue are more similar to each other than rap and classical, are shown to be more similar (by the GNG) in the context of our feature vectors. The next section describes in more detail how certain elements of the feature vector are particular and distinct for the different genres.

3.3 Feature-Attribute Highlighting

In addition to coloring by genre, we were able to color our visualization of the GNG based on specific song elements (the specific features inside the songs feature vector.) For example, a glance at the GNG

Model	Country	Classical	Rap	Reggae	Top
1	4	9	1	0	Classical
2	3	1	10	12	Reggae
3	0	0	4	6	Reggae
4	3	2	2	2	Country
5	11	1	0	2	Country
6	0	9	0	0	Classical
7	5	0	12	12	Rap
8	0	0	5	4	Rap
9	2	3	2	0	Classical
10	6	9	0	0	Classical
11	15	1	2	1	Country
12	1	0	6	9	Reggae

Table 3: Genre distribution for GNG Model Vectors where *top* is the genre with the highest count of songs mapped to the particular model.

colored on danceability tells us a lot about what the danceability value of each song means to its genre classification.

In Figure 1 we can notice how the classical model vectors (as indicated by the label) tend to take on a blue/teal color scheme while country hovers between green and orange and then the rap and reggae model vectors appear vastly orange and red. This illustrates that there is a distinct difference in the danceability of rap/reggae and classical songs and furthermore that danceability is important to the GNGs categorization of genre.

The reverse holds true about elements of the feature vector not as significant to classification. Notice the homogeneous appearance of the GNG colored by duration. A consistent navy-blue tint demonstrates the duration had very little if any impact on the mapping of songs to model vectors.

3.4 Song and Genre Mappings

As well as assisting the neural net in its classification of music, we can use the model vectors to create visualizations, much like music landscapes, for each

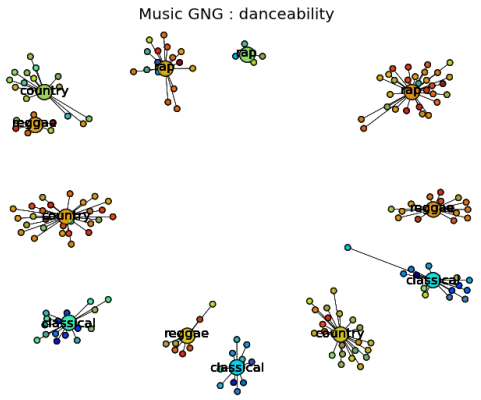

Figure 2: The GNG resulting from above data-set colored by *danceability*. The difference between danceability for different genres is clear.

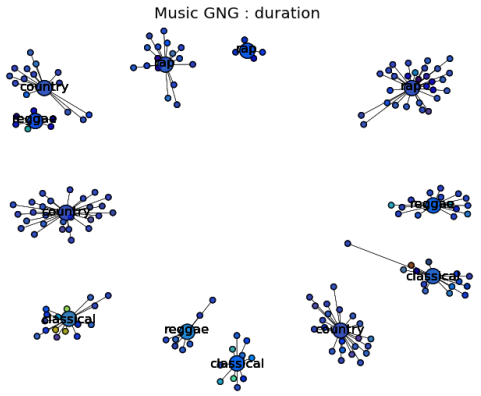

Figure 3: The GNG looks homogeneous when colored by *duration*.

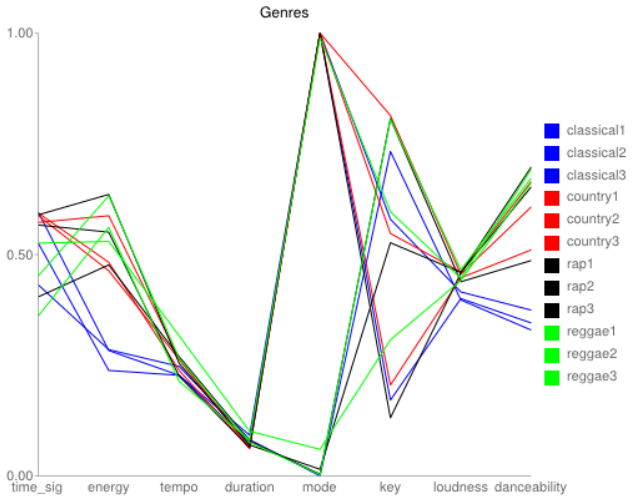

Figure 4: GNG model vectors on one graph

genre. We created a visual representation that corresponded to the values of each GNG model vector. The x-axis is labeled by the eight attributes of the feature vectors. Figure 4 shows a map of each of the four genres and their respective model vectors on top of each other. This can be seen as a landscape of our music space and illustrates the similarities and differences between genres. With these graphs we can see that the model vectors all share some similarities within their genre. Figure 5 shows a rap song *What Up Gangsta* mapped to and the rap-model vector to which that song maps to. With these graphs, we can see that the model vector is a good representative of the song mapped to it.

3.5 Neural Network Results

Our results from the neural network training are presented in Table 4 and Figure 6. The networks with the highest training accuracy rates were those that used the song vector and the two closest model vectors as inputs. Both these networks outperformed the control (just the song vectors) in terms of training ac-

Figure 5: A rap song mapped to its genre model vector.

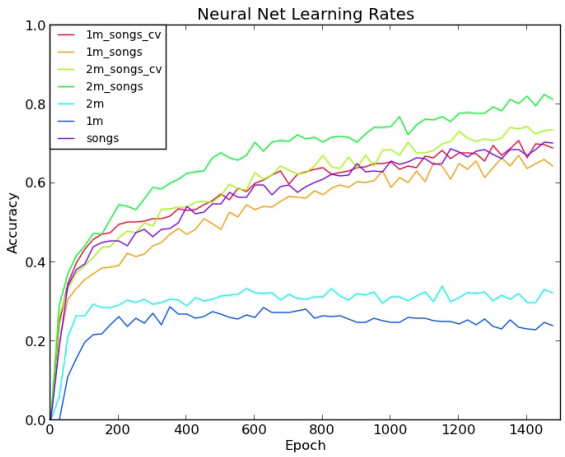

Figure 6: The training value of each Neural Networks configuration shown mapped against epoch

curacy, but only the network trained without the confidence value also outperformed the control in terms of testing accuracy (Table 5.) As demonstrated by Figure 6, the learning rate initially increases very quickly for the first 200 epochs but then the rate grows at a balanced rate for the last 1200 epochs. This is a generally common result in neural net training

From our experiment we can see that the confidence value, although helpful in some cases, ended up hindering training when in conjunction with the song and two model vectors. In terms of testing, the network trained using the confidence value always performed worse than its counterpart. Although the differences in testing accuracy are small for the top networks, the disparity in training accuracy is much larger and should not be ignored.

The results from training suggest that the neural network is better able to learn the training set when given information about the song's relationship to the GNG. These results agree with our hypothesis that a GNG would aid a neural network in categorizing songs by genre using only the song's intrinsic attributes. We found it surprising that the confidence value did not aid in the training of the best neural networks.

Inputs	Hidden nodes	correct %
S	8	71
1MV	8	30
1MV+S	8	68
1MV+S+C	8	74
2MV	8	35
2MV+S	8	86
2MV+S+C	8	78
S	10	70
2MV+S	10	85
2MV+S+C	10	81
S	12	74
2MV+S	12	88
2MV+S+C	12	83
S	15	79
2MV+S	15	85
2MV+S+C	15	85
S	20	77
2MV+S	20	88
2MV+S+C	20	85

Table 4: Neural Networks Training Results: S = song, MV = closest model vector, 2MV = closest and 2nd closet model vectors C = confidence value

Inputs	Hidden nodes	% correct
S	8	61
MV	8	21
Mv+S	8	54
MV+S+C	8	51
2MV	8	28
2MV+S	8	68
2MV+S+C	8	58
S	10	64
2MV+S	10	67
2MV+S+C	10	60
S	12	66
2MV+S	12	70
2MV+S+C	12	60
S	15	65
2MV+S	15	67
2MV+S+C	15	61
S	20	66
2MV+S	20	68
2MV+S+C	20	61

Table 5: Neural Networks Testing Results: Vectors are labeled as in Figure 4

One hypothesis is that our hidden layer contained too few nodes, and so by increasing the number of hidden nodes, performance of the networks that included the confidence value would increase as well. To test this hypothesis, we reran the experiment using ten, twelve, and then fifteen hidden nodes. See Table 4 and Table 5. The results from training and testing with more hidden nodes indicate that although the training accuracy rates for the 2MV+S+C converge to 2MV+S levels, the test accuracy is consistently higher when the confidence value is not included as an input. Furthermore, networks that only included the song feature vectors as inputs never surpassed the other two networks in terms of training accuracy, but when tested with new songs, 2MV+S+C (which never went above 61% accuracy in the testing phase) was the worst. Overall the best networks (in terms of training and testing) were the ones with 2MV+S as inputs.

4 Discussion

4.1 GNG Illustrates Limited Feature Vector

The feature vector consisted of as much information about the song as we could consistently provide. What this means is that for any feature in the feature vector, we had to have a value for that feature for any song. For this experiment our feature vector was restricted to the Echonest audio_summary of a song.[1] Out of the eight components we had for each song, *time signature, energy, tempo, mode, key, duration, loudness, danceability*, we were unsure how many would actually contribute to the categorization process.

For example let us look at Figure 3. It displays the GNG with each node colored by its *duration* value. Nearly all of nodes are of a mid-ranged blue hue indicating that they have very similar duration values. The contrast between this graph and Figure 2, the GNG colored by *danceability*, is striking. Each genre has a distinct 'color' of danceability in comparison to a universal value.

When the color distribution of a graph looks homogeneous, it demonstrates that feature is not having much influence in the GNG's categorization of the song. A more sophisticated way to directly determine influence would have been a principle components analysis (PCA) , however the Networkx[8] used a built in PCA to visualize the graph. Consequently, the color distribution provides a good idea of which feature is important. Table 6 shows each of the features with a subjective influence rating. The rating is based upon how important the feature appears to be to the GNG.

We concluded that five of the features were were most important to the GNG–those with strong and medium influence values in Table 6. This reduces the ability of the system compared to a system where all eight features were strong indicators of cate-

Feature	Influence
danceability	strong
energy	strong
loudness	medium
tempo	medium
time sig	medium
mode	weak
duration	weak

Table 6: Feature Influence on System

gory. Gathering more information about each song could be a future extension of this project (see future work).

4.2 Limited Song Features Indicate Strong System

Neural networks and growing neural gases both excel at extracting patterns from highly complex datasets. Our results showed that generally the larger the input layer to the neural net, the better it performed. If we exclude the features that appear to not assist the machine learning in categorization, our input layer is relatively small.

We take this as a sign that the core principle behind the system is very strong and that by introducing more detailed information about the song, we could dramatically improve the prediction accuracy of the network. An alternate to getting a more detailed dataset would be to run the system with the current data and only provide the features that seem influential. While this might shrink the complexity of the input space, it could remove any convoluted effects that the less influential song features have on the categorization process.

4.3 Neural Net

Networks trained using the information from the GNG were able to achieve higher levels of training accuracy. We believe that the additional information abstracted from the GNG provides more context to the songs being trained, but is only useful when given in conjunction with the song feature vectors themselves.

Testing accuracy was only slightly better for the best network two model vectors and song vectors compared to the second best S. The confidence value seemed to hinder testing accuracy, suggesting it is a misleading attribute, and should not be used in classification of genre. However, by looking at the data we can clearly see that euclidean distance to a node has little correlation with the genre, and instead the direction of the song relative to the model vector might have been the more useful.

Our results from training these neural networks suggest that a song map with model vectors generated using a GNG can be used to increase the accuracy of genre classification. This suggests that there exists some quantifiable relationship between the location of a song in the song-space and its genre. Although the difference in testing accuracy between the network trained on 2MV+S and the one using just songs was small, the difference in training accuracy makes it clear that the GNG is providing relevant information to the network.

5 Future Work

5.1 NEAT

Neuro Evolution of Augmenting Topologies (NEAT) is a modification of a neural net that has a dynamic hidden layer. More specifically the system only introduces nodes to the hidden layer when they are needed. NEAT has relevance to our project in the

sense that we used a static hidden layer that was determined by a human programmer.

A future extension of this music classification task would be to modify our neural network to use NEAT. A dynamic hidden layer would be extremely useful to the system. Our system is designed to work with any sized set and having the programmer modify the size of the hidden layer based upon the size of the dataset is far from optimal. A dynamic network topology would allow for varying sets with different amount of songs and any number of music genres.

5.2 More Detailed Song Information

We mentioned that more detailed song information could dramatically improve the performance of our system. There are several different paths we could take to acquire more information. Many music parsing packages include detailed segmented analysis of a song. Currently the audio_summary we are using provides data that describes the song as a whole. A summary of segment of the song would be significantly more detailed and could help our system.

5.3 More Songs and Genres

Similarly to how more detailed feature data could help improve the accuracy of the system, simply using more songs could too. We chose to used 200 total songs and four genres because that was the most data we could collect with the time and resources we have. We do have code in place, however, to get every audio_summaries for every single song any artists has created and are hoping to make a much larger (thousands of songs) dataset using this method. With this method of retrieving songs, we could also quickly create datasets for many other genres and see how our system scales to this increased input size.

6 Acknowledgements

Thanks to Lisa and Jeff for all their time and help.

References

[1] 'The Echo Nest - Powering Intelligent Digital Music Applications.'
The Intelligent Music Application Platform. Web. 07 May 2012. http://the.echonest.com/

[2] 'Last.fm.' Last.fm. Web. 07 May 2012. http://www.last.fm/

[3] Category-based intrinsic motivation,Proceedings of the Ninth International Conference on Epigenetic Robotics,Rachel Lee, Ryan Walker, Lisa Meeden, and James Marshall (2009)

[4] A growing neural gas learns topologies Advances in Neural Information Processing Systems 7 , Bernd Fritzke (1995)

[5] 'Musical Genre Classification of Audio Signals',George Tzanetakis and Perry Cook,IEEE Transactions on Speech and Audio Processing, 10(5), July 2002

[6] 'Scanning the Dial: The Rapid Recognition of Music Genres', Robert o. Gjerdingen and David Perrot Journal of New Music Research, Vol. 37, No. 2. (2008), pp. 93-100

[7] 'Running with Data.' Danceability and Energy: Introducing Echo Nest Attributes. Web. 07 May 2012. http://runningwithdata.com/post/1321504427/danceability-and-energy.

[8] 'Overview NetworkX 1.6 Documentation.' Overview NetworkX 1.6 Documentation. Web. 09 May 2012. http://networkx.lanl.gov/.

[9] ‘Intro.’ Matplotlib: Python Plotting Matplotlib V1.1.0 Documentation. Web. 09 May 2012. http://matplotlib.sourceforge.net/.