CS74 Project Milestone: Music Classification

Sebastian Bierman-Lytle, William Wang, James Drain

Project Objectives

Develop a supervised learning algorithm in order to detect characteristics of music, with a focus on EDM (electronic dance music) tracks, including genre, mood, and assorted tags.

Current Project Status

- Manually classified ~100 songs for use as a training set
- Implemented code to compute an elementary set of features for each track
- Implemented a simple forward / back-propagation neural net for classifying songs
- Conducted some preliminary runs of this neural net on the training set

To-Do List

- Expand the set of features to be used as inputs to the neural net, specifically with an emphasis on temporally-varying features rather than just using scalars
- Experiment with neural net parameters to find optimal settings
  - Consider using an adaptive learning rate to improve efficiency
  - Consider using a Nguyen-Widrow layer initialization function
- Expand the training set to encompass at least 5-10 samples for each output measure
- If time permits, wrap the engine in a more user-friendly package

Training Set & Feature Computation

Input Feature Set

Our input features are derived by preprocessing each audio track and extracting the best five second clip to use for analysis, and then by using the MIRtoolbox to extract six audio features. This procedure is described with the following steps:

1. *.wav data is analyzed for the first occurrence of an extended period (5 seconds) of low amplitude (where low amplitude is defined as below the average amplitude of the track)
2. Starting from the end point of the low amplitude window, the time of the first occurrence of a high amplitude window (5 seconds with an average amplitude above the average for the whole track) is recorded and saved
3. Seven audio analysis functions from the MIRtoolbox are run on the 5 second high energy clip and the results are compiled into our final Input Feature data set.

Feature Descriptions

Tempo is a scalar value defining the beats per minute. Our data set has a range of 70-148 BPM. In theory, this should be a good indicator of genre, as Dubstep is in the range of 70 or 140, Deep

House is in the range of 120-126, Trance is in the range of 132-150, and all other types of House are in the range of 126-132.

Pulse Clarity is a scalar value defining the regularity of a beat. This indicator should be very helpful in differentiating House, Electro House, and Dubstep.

Mode is a scalar value defining the relative pitch class of the audio sample. -1 indicates a Minor pitch, 1 indicates a Major pitch, and 0 indicates a non determinant pitch. This feature is useful for differentiating mood and tags.

Pitch is currently a scalar value indicating the average pitch throughout the sample. This will be converted to a vector value that incorporates time progression data. Pitch, especially the direction of pitch progressions, are very useful in differentiating mood and tags.

Dissonance is a scalar value describing the sonic dissonance of the waveform; i.e., how consistent / inconsistent the waveform is in shape.

Fullness is a scalar value indicating the complexity of the wave signal (ei. how many different sounds with different frequency signatures exist in close proximity to one another in the clip). This feature is very useful for determining the fullness quality, as well as many genre distinctions (Minimal House vs. Tech House vs. Trance), as well as many tags.

Output Feature Set

Our output feature set has changed slightly from the proposal, as we have dropped the vocal indicator and simplified the tagging process. This is an example of one:

```
{
                "id":22,
                "filename":"14 Trainsurfing (Original Mix)",
                "genre": "Deep House",
                "mood": "Dirty",
                "avg_pitch": "Bass Heavy",
                "fullness": "Very Minimal",
                "tags": ["Dark","Deep","Bouncy","Heavy","Grimey","Dirty"]
}
```

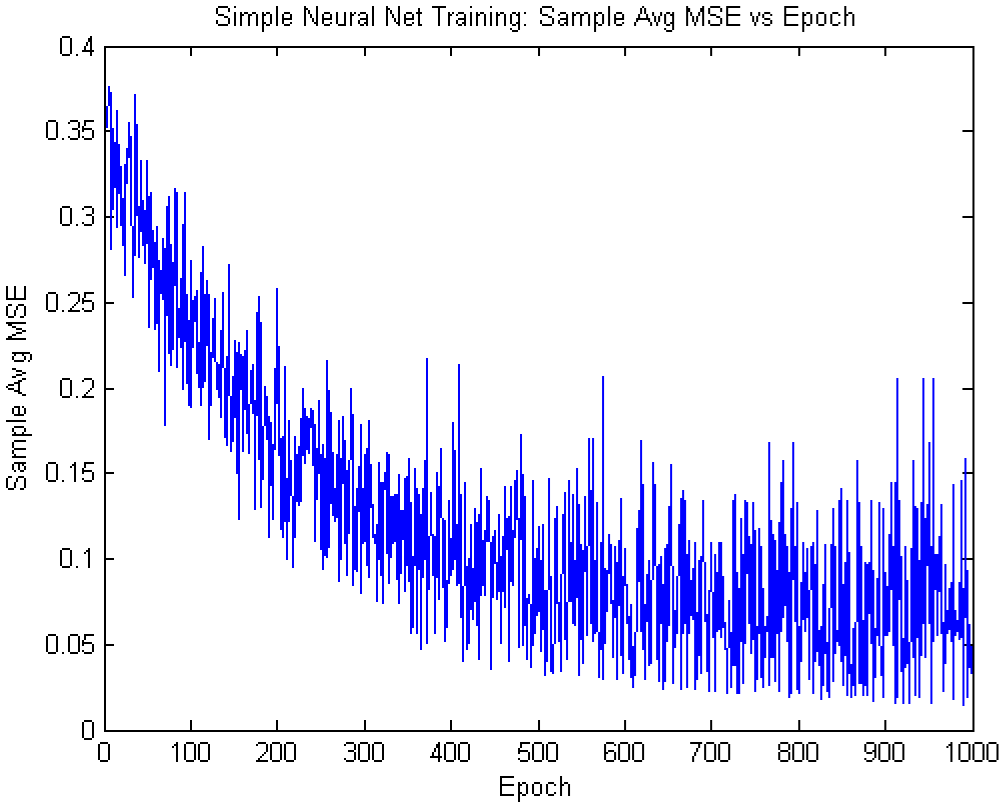
<u>Neural Net Implementation</u>

Though we initially sought to use gaussian mixture models to approach this problem, it’s become clear any such implementation would closely overlap with class material. We realized that we could view our classification problem as a clustering problem. That is, that we could cluster the training data, using Mahalanobis distance, according to the labelled outputs, and then assign each novel song probabilistic membership for each cluster. We considered soft k-means

(MacKay) and spectral clustering algorithms, and found a tutorial explaining the relaxation techniques corresponding to the different choices of Laplacian in spectral clustering (Luxburg). However, clustering is generally inappropriate for classification, especially supervised classification. The Wikipedia page for machine learning algorithms has a list of nonlinear classifying algorithms, including decision trees. We looked into the ID3 algorithm, which greedily minimizes the entropies in a binary flow-chart method, but decided it was too similar to the decision trees we will learn about in class. We rejected random forests for the same reason.

We found a paper on music genre classification that utilizes neural gas (Clark), and decided neural nets would be a powerful technique to learn about for use in our particular project. Neural gas allows for dimensionality reduction and easier visualization (Clark), but we decided to first implement a vanilla neural net, because it is both simpler and an effective baseline comparison. Our next goal for improving our neural basic neural net is to use a heuristic weight initialization to speed up training time. The initialization described in Nguyen sped up training from two days to four hours. It involves randomly distributing the weights, and then adjusting their sizes to be locally linear.

Our current neural net implementation that we've tested is quite basic. The current iteration uses forward-backward propagation through gradient descent with 20 hidden nodes in a single layer, but we are still experimenting with the specific parameters. When run over 1000 epochs in a stochastic learning process, we get the following results:



References

http://web.cs.swarthmore.edu/~meeden/cs81/s12/papers/AdrienDannySamPaper.pdf (Sam Clark, Danny Park, Adrien Guerard (5/9/2012). “Music Genre Classification Using Machine Learning Techniques.”)
https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox/MIRtoolbox1.5Guide (Oliver Lartillot. “MIRtoolbox 1.5 User’s Manual.” Accessed October 28, 2014. Finnish Center of Excellence in Interdisciplinary Music Research.)
http://www.inference.phy.cam.ac.uk/itprnn/book.pdf (David MacKay (2005). “Information Theory, Inference, and Learning Algorithms.”) Cambridge University Press 2003.
https://web.stanford.edu/class/ee373b/nninitialization.pdf (Derrik Nguyen, Bernard Widrow. “Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights.” Accessed October 28, 2014. Information Systems Laboratory.)
http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/Luxburg07_tutorial_4488%5B0%5D.pdf (Ulrike von Luxburg (2007). “A Tutorial on Spectral Clustering.” Max Planck Institute for Biological Cybernetics.)