Music Artist Identification Using Linear Temporal Pyramid Matching

Thomas Kelly

***Abstract*—Herein, I show a new technique for musical artist identification using the Million Song Dataset features. The field of music identification and recognition (MIR) is a well-established field. However, extracting concrete data from art is a difficult task. I use the features from the Million Song Dataset and several techniques in order to classify if a song is by the Beatles. Using the first 60 segments of a song, I achieved 62% accuracy with C4.5 and 76% accuracy using random forest. Using recurrent neuroevolution of augmenting topologies (NEAT), I achieved 55% accuracy. Bag of Features with combined sound features classified 74.9% of songs correctly. Using Bag of Features with features split into duration, pitch, and timbre, 87.6% of songs were identified correctly. Using my new linear temporal pyramid matching (LTPM) with the split features and a one level pyramid, 87.7% of songs were correctly classified. On combined features, LTPM with two levels achieved 75.5% accuracy. As such, LTPM does not significantly improve on standard BoF.**

I. Introduction

I began this project with the intention of classifying music by genre using neuroevolution of augmenting topologies (NEAT) [1]. After reading Liang, et al. [2], I found that three graduate students using the same dataset achieved 38.6% accuracy on determining genre. From this, I concluded that the difficulty of genre classification was beyond the scope achievable during this course.

The Million Song Dataset (MSD) [3] was developed specifically to aid in music identification and recognition (MIR). However, the MSD does not provide the music directly, but instead provides heavily processed data, based on the work of Tristan Jihan [4]. Songs from the MSD are first divided into segments. Theoretically, each segment corresponds to a single beat of the music. The sound data is processed to create several features corresponding to various audio qualities, including timbre and pitch. Timbre, as defined by the 2012 Random House Dictionary, is "the characteristic quality of a sound, independent of pitch and loudness, from which its source or manner of production can be inferred." Pitch refers to the note(s) being played at a given time. For this data, pitch is divided into 12 classes, C, C#, D, ..., B. Pitch classes repeat as the pitch gets higher or lower; for example, A is at 440 Hz, 220 Hz, 880 Hz, 110 Hz, etc., doubling and halving in both directions. Pitch and timbre are each provided in the form of 12 real numbers for each segment of the song. The 12-dimensional vector corresponding to the timbre of a segment is obtained by performing a dimensional reduction by principal component analysis on mel-frequency cepstrum coefficients (MFCC) of each segment. MFCCs were invented for the purpose of speech recognition, but have been found to be useful in MIR by Beth Logan [5] and are used in other works [6], [7]. The numbers resulting from this process may theoretically be any real number. However, as can be seen

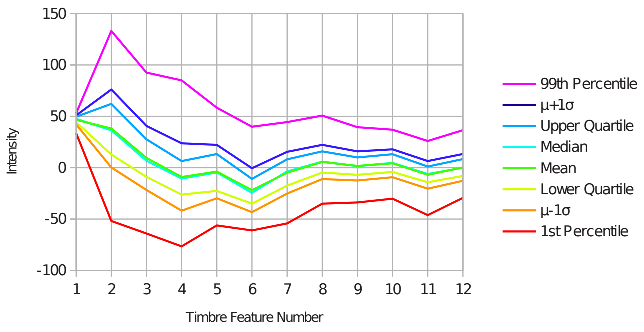

Figure 1. The statistical characteristics of the timbre vectors provided by the the MSD

Figure 1, there exist well-behaved distributions for each of the different timbre features. According to the documentation of the Echo Nest Analyze software used by the MSD [8], some of these timbre features are correlated with well-known sound characteristics. For example, timbre feature 2 is correlated with the subjective characteristic of musical "brightness." The 12-dimensional vector for pitch is obtained by finding the relative energy of each pitch class, averaged across a given segment. These energy vectors are normalized such that the most energetic pitch class of a segment is given a value of 1.0. The duration of each segment was also used for classification. This duration approximates the length of a beat at a given time. Other song data is available, but was not used.

Using Weka Explorer, I tried several standard machine learning algorithms on all of the above features. C4.5 was the first algorithm used. It yields a decision tree from a dataset, which is useful because decision trees are relatively easy for humans to comprehend: a tree consists of several yes/no questions that cause the tree to branch, such as, "Is timbre dimension 3 less than 25.717?" At the leaves of each branch, a classification (e.g., the song is by The Beatles) is given. The resulting tree often gives insight into the nature of a dataset. Support vector machines (SVMs) classify data by taking a number of n-dimensional vectors and dividing the n-space, which the vectors lie in, with one or more hyperplanes, such that each region created by dividing the space best corresponds to a class that each vector falls in. SVMs are commonly used on high-dimensional data, as training them for a large n is relatively fast. The algorithm used to optimize the locations of the hyperplanes for all experiments described herein is sequential minimal optimization (SMO). Adaptive Boosting (AdaBoost) is a meta-algorithm used to improve the performance of machine learning algorithms by creating several classifiers with some other algorithm and creating a hypothetically better classifier by using a weighted sum of the results of the classifiers. Random forest operates by creating many decision trees and using the classification of a given datum by each tree as a vote for the datum being in some class.

Several algorithms not included in Weka were also used. Genetic algorithms mimic the process of evolution by starting with a random "population" of "genomes," determining the "fitness" of the genomes, selecting which genomes reproduce according to their fitness, crossing selected genomes together, "mutating" the resulting genomes, and repeating until the genomes are sufficiently "fit." Generally, genomes are any kind of data that can be used as parameters to attempt some task. The fitness of a given genome is its performance at the given task.

Neuroevolution of augmenting topologies (NEAT) [1] is a genetic algorithm for evolving neural networks. The basis of the neural network is the perceptron, which is inspired by the operation of a neuron. A perceptron takes several inputs. If a weighted sum of the inputs is above some threshold value, then the perceptron "activates," outputting a high number (e.g., 1), as opposed to a low number (e.g., 0). To make training easier, a soft activation function is often used, such as a logistic curve, which means that the output of the perceptron is simply the weighted sum fed into the logistic function. Several perceptrons can be wired together to create a neural network. Usually, neural networks use a feedforward topology, where the inputs are fed into a "layer" of several perceptrons, the outputs of that are fed to the next layer, and so forth until the last layer, which holds the outputs. The middle layers are called hidden layers. Recurrent neural networks modify this paradigm by routing the outputs of one or more of perceptrons backwards, such that a signal can loop around in the network. A specific topology called an Elman topology routes the outputs of a hidden layer to a context layer, which is sent to neurons in one of the hidden layers on the next time step. This adds a memory to the neural network. NEAT is a genetic algorithm made to generate neural networks of arbitrary topologies. This is achieved by allowing the creation and destruction of perceptrons and links between them during mutation.

Another algorithm used is the bag-of-features model, which was introduced to me by David LeBlanc. The bag-of-features (BoF) model [9] is based the bag-of-words (BoW) model, which is used for natural language processing, in which a histogram of the words in some sample of a class of documents is created; this histogram is assumed to be a characteristic distribution of the words for a given class of documents. It is then assumed every document has its words selected from a distribution of words, also known as a "bag" of words, which is characteristic to some class of document. Bayesian probability is used to determine which bag is most likely to be the "source" of the document's words. This was extended to images by considering interesting features of an image as words, forming a "visual vocabulary" of small graphical regions. However, since these features are generally vectors of several real numbers, as opposed to discretely different words, one needs to split these features among discrete groups in order to form a histogram. This is done by a clustering algorithm. Clustering algorithms operate by taking some number of points in some space and classifying them each into one of several classes, according to their proximity to other points. So, in theory, each point would correspond to an object or element found in an image. Once clustered, one may obtain a histogram of the features in the image. Since histograms are no more than vectors of real numbers, any of the earlier techniques for classification may be used on the histogram; in the original paper by Csurka, et al. [9], SVM showed the best results. BoF has been extended to music by Fu, et al. [7] and has shown promising results. However, there are some differences. In images, features of an image usually consist of many, widely-spaced, interesting points where there exist a high degree of complexity. However, there are fewer criteria for interestingness of a given moment in a song than there are for finding an interesting point in a photo, so the entire song is clustered for the histogram.

II. Dataset and Methodology

The dataset was obtained from the Million Song Dataset using two different queries to obtain songs by the Beatles and songs by other artists . The first query asked for 50 songs by the Beatles using the Echo Nest API. This was done by creating a dynamic playlist that searched by artist for the Beatles and retrieved the necessary number of songs from the dynamic playlist, taking care to obtain the audio summary data required for training and classifying. Unfortunately, this is not a perfect method, as there are several interviews with the Beatles in the MSD. So, after retrieving 50 songs, we replace audio tracks that have been classified to have more than 0.5 in the speechiness attribute. This is provided by the MSD for the purpose of allowing researchers to filter out non-music audio. It is not a perfect measure, as manual identification has found several interviews that slipped past the filters. In order to obtain 50 songs that are not by the Beatles, a query that searches for several song "styles" is used. The method is largely similar, but the search is done by artist/description for a style that is any of: classical, metal, pop, hip hop, rock, or jazz. This gives a diverse cross-section of music to compare against. The resulting songs are filtered to make certain that none of the songs retrieved are speechy or by the Beatles. The songs actually used are seen in Appendix A on page 5.

As noted before, several techniques were considered using Weka Explorer with the timbre, pitch, and duration of the first 60 segments of each song, concatenated into a 1 500-dimensional vector. All results were obtained using 10-fold cross-validation, including those explained in later sections. Weka's implementation of C4.5, called J48, was used with tree collapsing, a confidence factor of 0.25, a minimum of two songs per leaf, 14 folds, reduced error pruning, subtree raising, and MDL correction. Weka's implementation of SMO was used to generate SVMs using a linear kernel, complexity of 1.0, ϵ of 1×10^{-12}, training data normalization, a linear kernel, and a tolerance of 0.001. AdaBoost was used with these two algorithms with 100 folds. For AdaBoost with SMO, the same settings were used as for SMO. For AdaBoost with C4.5, settings for C4.5 were the same, except the number of folds was three and reduced error pruning was disabled. Random forest was used with unlimited depth, 100 trees, and 200 features.

NEAT was performed using the Xtructure XNet package. The resulting neural networks were tested by inserting the 25 numbers for the timbre, pitch, and duration of each segment at the input nodes, propagating signals through the network by one link, and repeating with the next segment until the end of a song. After the song, an error was calculated from a difference between the value at the output node and a value representing whether a song was by the Beatles, with 1.0 representing the Beatles and 0.0 representing other artists. The average error over all songs was used as the fitness of a given network. An Elman topology was enforced by adding 5 context nodes, which were implemented by adding an additional 5 output nodes and copying the output of those those to 5 input nodes after each time step. Additionally, 10 bias nodes were used. Mutation between generations had a probability of 50% with a further adding link mutation probability of 30%, a node adding mutation probability of 20%, a single link attribute adjustment probability of 47.5%, and a link removal probability of 4%. A population of 100 was used. Other settings were set to the package's defaults.

Bag of features (BoF) is done by performing several transformations on a song in parallel to yield several histograms of feature classes, which are then concatenated and used to train or test a classifier. The actual algorithm used is shown in Appendix B on page 7. Several design decisions were made in the building of this algorithm. In order to extract information about sequences of sounds of different lengths, the songs were divided into overlapping "frames," which consist of one to four segments. The decision to use frames of varying length is based on the fact that certain sequences of sounds may act as a signature for an artist. Since each segment is supposed to correspond to a single beat, a frame of length four should cover a single measure of music in a song in the commonly used 4/4 time. This is why the maximum frame length is four. Overlapping frames were used because it was assumed that more data is better. An exception in the histogram building process is made for durations in frames of length one because it does not make very much sense to cluster a single number. So, instead of a histogram, the statistics of the duration of segments of the song (i.e., mean and standard deviation) are produced. These statistics should be useful due to the fact that the duration of a single segment should be the inverse of tempo at that time. So, the mean of the duration of a song can be used to classify songs by tempo and the standard deviation acts to indicate how that tempo varies. Only 10% of the feature vectors from the training set are used because Fu, et al. [7] indicated that using a smaller subset of features does not affect clusterer performance, while using considerably less time. k-means clustering is used because it was the only algorithm that would run in a reasonable period of time. The k-means clusterer uses a normalized Euclidean distance to define clusters because that seemed to be the most logical metric that was included in Weka. The number of clusters used depends on the feature type and the frame length. There are 15 such numbers because there are 16 combinations of feature type (combined, timbre, pitch, duration) and frame length (one, two, three, or four), and durations for frames of length one are not clustered. The method by which these numbers were obtained will be explained later. SMO was run using the settings indicated earlier.

As noted before, BoF ignores large-scale structure. In order to remedy this, I created linear temporal pyramid matching (LTPM), based very loosely on spatial pyramid matching [10], a technique developed for the same purpose in images. LTPM is similar to straight BoF, except, in addition to the histograms used previously, we also make histograms for subregions in the song and concatenate those as well. The full algorithm can be seen in Appendix C on page 7. The "pyramid" is formed because we have regions of decreasing size stacked upon one another. Level 0 is a histogram of the entire song. Level 1 consists of two histograms of each half of the song. Level 2 consists of four histograms for each quarter. This is be continued for further powers of two.

In order to find optimal values for the number of clusterers for each feature type and frame length, two genetic algorithms were used. In the first, used to calculate the best way to cluster combined features, genomes consisted of four genes. Each gene was a positive integer. Fitness was determined by running BoF with the first integer used to cluster vectors for frames of length 1, the second for frames of length 2, etc. Fitnesses were normalized with sigma-truncation with $c = 1$. Roulette selection was performed to select two parents. Single-point crossover was used to produce two children from the parents. Mutation was done to all genes with 100% chance by adding numbers taken from a discretized Gaussian distribution with a mean of 0 and a standard deviation of 10; if this drops the gene below 2, the gene is replaced with 2, as clustering into fewer than 2 clusters makes no sense. Due to a lack of available computer time, only 17 generations of a population of 4 were evaluated. Starting genes were random numbers from 2 to 200. This is due to the fact that, in earlier testing, using 100 clusters for all frame lengths worked rather well, with worse results at 50, 150, 200, and 250. In order to find optimal values for split features, the same genetic algorithm was used, with some changes. The fitness function was BoF, as seen in Appendix B on page 7, where the only the accuracy of using split features was used. Due to poor planning, the genome consisted of 11 useful genes, used to determine the numbers of clusters, as well as 5 useless genes. This was done for 30 generations with a population of 25. The exact genetic structure is further

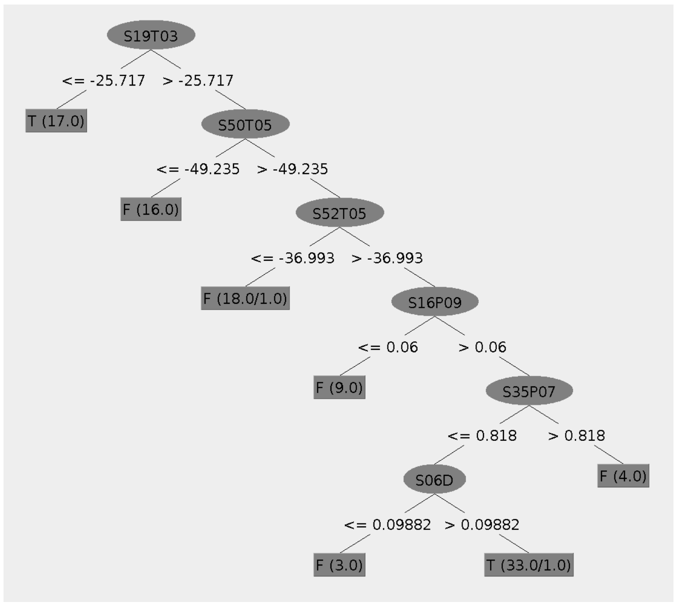

Figure 2. A decision tree generated using sound data for the first 60 segments of a song with C4.5. The ovals indicate splitting attributes for a song. Attributes are labeled as with two parts: the segment indicated with the number after "S," and a sound characteristic of a segment. Sound characteristics are either timbre, pitch, or duration, indicated with T, P or D, respectively. Since timbre and pitch have 12 numbers for each segment, these numbers are further labeled from 0 to 11. The boxes indicate how a song is classified, with T and F standing for whether it is True or False that a song is by the Beatles. The parentheses indicate how many were classified correctly by a classification and how many were classified incorrectly. Timbre feature 5 is used twice and at similar times in the song, at segments 50 and 52, as the ultimate criterion for classifying 52% of the non-Beatles songs as not being by the Beatles. A subjective interpretation of timbre feature 5 does not actually exist, according to Lehan [11], but it seems to be correlated with the Beatles. However, a lack of timbre feature 3 on segment 19 is used to classify 33% of Beatles correctly. According to Lehan and DesRoches [8], this correlates with segment 19 having a weak attack, which is the how quickly an instrument can change from silent to loud.

explained in Table I, along with the results.

In order to compensate for the stochasticity of the algorithm introduced by the k-means clustering, the algorithm was run 10 times during testing. Due to the 10-fold cross-validation, this means that the results are the result of 100 different sets of clusters.

III. Results

As noted previously, for comparison, several general techniques were tested on classifying songs based on the first 60 segments. The first of these was C4.5, which achieved 62% accuracy. A tree with analysis can be seen in Figure 2. SVMs built with SMO also achieved 62% accuracy. There does not appear to be a clear reason why both would have the same accuracy. Using AdaBoost with SMO did not improve classification over standard SMO. However, AdaBoost with C4.5 did increase accuracy to 62%. Random forest was able to get 76%.

Attempts to use NEAT made little progress on the problem as shown in Figure 3.

The evolved bags sizes for BoF on combined features were

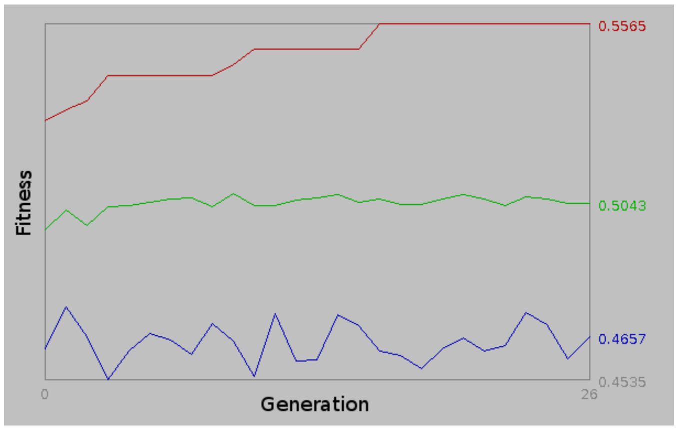

Figure 3. The fitness of NEAT over several generations. The red line is the maximum fitness of a given generation, the green is the average, and blue is the minimum. Past this point, for another 200 generations, there was no improvement.

123, 7, 38, and 100 for frames of lengths one to four. Using

Table I
THE BEST GENOME FOUND FOR SPLIT FEATURES.

Genome meaning	Clusters
1-Segment Timbre	186
2-Segment Timbre	37
3-Segment Timbre	59
4-Segment Timbre	196
1-Segment Pitch	250
2-Segment Pitch	164
3-Segment Pitch	45
4-Segment Pitch	70
Unused	67
2-Segment Duration	8
3-Segment Duration	47
4-Segment Duration	4
Unused	62
Unused	70
Unused	37
Unused	158

BoF with split features evolved a genome as seen in Table I. This was achieved after 30 generations of a populations of 25, that is, after testing 750 candidates. The last four unused genes previously corresponded to one to four segments of combined features, which were ignored while attempting to optimize bag size for split features. The single unused gene above those was the number of bags used for 1-segment durations. Single segment durations have 1-dimensional feature vectors, that is, they are single real numbers, that approximate beat length. As noted earlier, it was concluded that characterizing the distribution of these durations as a mean and standard deviation would yield easier to classify distributions than full histograms. The results for LTPM, as compared with BoF, can be seen in Table II on the next page. The false positives for the most successful trial, which achieved 90% accuracy using split features and a 1-level pyramid, were 19-2000 by the Gorillaz, Crash and Burn by Marc Pattison, and So Far by Faust. The false negatives were The Night Before, Concert

Table II
THE RESULTS OF BoF [a] COMPARED WITH LTPM TESTED ON 100 SONGS, AVERAGED OVER 10 TRIALS.

Feature division	Pyramid levels	Accuracy $\pm$ 1 Standard Deviation
Split	0	0.876$\pm$0.01
	1	0.877$\pm$0.02
	2	0.866$\pm$0.02
	3	0.856$\pm$0.01
	4	0.841$\pm$0.01
Combined	0	0.749$\pm$0.03
	1	0.750$\pm$0.03
	2	0.755$\pm$0.02
	3	0.747$\pm$0.02
	4	0.743$\pm$0.02

[a]Note that LTPM with 0 levels is the same as standard BoF.

Announcements from Paul in Melbourne, It's All Too Much, Searchin', Birthday, A Taste of Honey, and Come And Get It. All other songs, as shown in Appendix A, were classified correctly.

IV. CONCLUSIONS

It was found that LTPM performed worse than BoF. As can be seen in Table II, for split features, LTPM does not outperform BoF, with 1-level LTPM performing within a margin of statistical error of of BoF for split features. Using more than one level with LTPM actually causes worse results. When using combined features, LTPM actually showed some advantage over split features using 2 levels, though this may be a statistical anomaly, based on the lack of trend suggesting this.

In comparison with the classification accuracy found by Fu, et al. on the Beatles [7], my implementation of BoF did slightly worse, achieving an accuracy of 87.6% compared to their 91.0%. However, they had several advantages. Their segments were considerably shorter, 23 ms, compared with mine, which averaged 284$\pm$173 ms. Their timbre vectors had 20-dimensions, compared with my 12. They used a different metric (χ^2) for histogram distance (mine was Euclidean distance, with vectors normalized to fit in $[0,1]^n$), which they note to be better for comparing histograms by emphasizing relatively unique feature classes. They also use multiple sets of clusters, which eliminates much of the stochasticity of k-means. Finally, they were running several other artist classifiers simultaneously, so it is possible for another artist classifier to stop a given song from being classified as being by the Beatles. It is also possible that their selection of songs had fewer problematic songs. However, in my favor, they did not use any pitch information, while I did.

When considering the false negatives, Searchin' and A Taste of Honey were both covers, Concert Announcements from Paul in Melbourne is not a song, and Come and Get It was released on The Beatles Anthology 3 in 1996, even though the song was actually only performed by Paul McCartney as a demo tape in 1969. The remaining three, The Night Before, Birthday, and It's All Too Much all sound rather unlike The Beatles. Birthday has several uncharacteristic guitar solos and has a rather aggressive and monotonic vocal line. The Night Before has few vocals, consisting of the title sung occasionally throughout the song over McCartney playing guitar. It's All Too Much is one of only two songs by The Beatles that heavily uses feedback, the other being "I Feel Fine" [12].

Out of the false positives, only So Far by Faust sounds anything like the Beatles, with some resemblance on with the guitar, brass, and drums. The large-scale structure has few similarities to most work by The Beatles: there are no choruses, verses, or bridges. Most of the interesting parts of the song can be characterized as alien ambient noise. However, the inability for BoF to identify large-scale structure is a well known problem. 19-2000 by the Gorillaz is a hip hop song with no clear similarities, as far as I can tell. Crash and Burn by Marc Pattison is a very aggressive metal song, which is quite dissimilar from the Beatles.

While the false negatives might be rather difficult to solve, due to the fact that they are rather unusual, the false positives seem to indicate room for improvement. Furthermore, stabilizing the output could probably be done. It would be quite possible to incorporate the techniques of Fu, et al. to aid in classification with standard BoF. However, there are also clear research paths to continue with respect to LTPM. Notably, the original spatial pyramid matching technique [10] relies on a "pyramid kernel" for their SVM, which is specifically tuned to compare histograms on multiple subregions by weighting smaller regions over larger ones, creating an emphasis on structure. This is done to better identify specific objects in an image, but may work to identify large-scale song sections better than an unweighted SVM. The lack of use of this kernel is the reason why linear temporal pyramid matching is specified as linear. Other methods for incorporating song structure include the use of recurrent neural networks. Though NEAT did not work at all, using it on the considerably simpler clustered features may yield better results by simplifying the problem and the neural network necessary to solve it. Finally, there is a new technique for training recurrent neural networks called Hessian-free optimization that shows promise [13].

APPENDIX A
SONGS USED

- Janis Siegel - (If I Had) Rhythm In My Nursery Rhymes
- White Lion - Wait
- Uncle Kracker - Smile
- The Beatles - Because
- The Beatles - Tell Me Why
- Gorillaz - 19-2000
- Lenny Kravitz - American Woman
- Etta Jones - Till There Was You
- Ugly Duckling - Left Behind
- The Beatles - Julia
- The Beatles - Any Time At All
- The Beatles - I'm Happy Just to Dance With You
- The Beatles - Dizzy Miss Lizzy
- The Beatles - Every Little Thing

- Kenny Wayne Shepherd - Shame, Shame, Shame
- Marc Pattison - Crash And Burn
- The Beatles - I Should Have Known Better
- The Beatles - Sea Of Monsters
- The Beatles - The Night Before
- Gin Blossoms - Follow You Down
- The Beatles - LUCY IN THE SKY WITH DIAMONDS
- The Beatles - Let It Be
- Joe Walsh - Ordinary Average Guy
- The Beatles - FROM ME TO YOU
- The Streets - Has It Come To This?
- The Beatles - THINK FOR YOURSELF
- Mike Jones - Back Then (Explicit Version)
- The Beatles - Drive My Car
- Nelly Furtado - In God's Hands
- Toni Basil - Over My Head
- The Beatles - Words Of Love
- The Beatles - Here Comes The Sun
- The Beatles - It's All Too Much
- The Beatles - Concert Announcements, from Paul in Melbourne
- The Black Crowes - She Talks To Angels
- Maroon 5 - Won't Go Home Without You
- The Beatles - Across the Universe
- Krokus - Screaming In The Night
- Gin Blossoms - As Long As It Matters
- The Beatles - Tomorrow Never Knows
- Steve Jones - Freedom Fighter
- Nelly Furtado - All Good Things (Come To An End)
- Ashley MacIsaac - Sophia's Pipes
- The Beatles - Searchin'
- The Beatles - Doctor Robert
- Aruna Sairam - Kalinga Nartana TillAna
- GNR - Sangue Oculto
- Joe Walsh - Lucky That Way
- White Lion - Radar Love
- Soul Asylum - Runaway Train
- The Beatles - I Will
- The Beatles - Run For Your Life
- Bad English - Price Of Love
- Lenny Kravitz - Are You Gonna Go My Way
- The Beatles - Love of the Loved
- The Beatles - I Need You
- The Beatles - BIRTHDAY
- The Beatles - Hold Me Tight
- The Beatles - Not a Second Time
- Finntroll - Försvinn Du Som Lyser
- The Beatles - Cry Baby Cry
- The Beatles - Only A Northern Song
- The Beatles - I've Got A Feeling
- The Beatles - Rocky Raccoon
- The Beatles - She Came In Through The Bathroom Window
- The Beatles - Day Tripper
- The Beatles - Derek Taylor With John, Paul, George and Ringo
- The Beatles - All You Need Is Love
- Various Artists - Intro (Feat. No Doz)
- The Beatles - Yellow Submarine
- The Beatles - This Boy
- Bette Midler - I've Got My Love To Keep Me Warm
- The Beatles - YOU'VE GOT TO HIDE YOUR LOVE AWAY
- The Beatles - Your Mother Should Know
- The Beatles - Michelle
- Maroon 5 - Sunday Morning
- The Beatles - Good Day Sunshine
- The Beatles - BABY YOU'RE A RICH MAN
- Jay Gordon - Slept So Long
- Gloria - É Tudo Meu
- Steelheart - She's Gone
- The Beatles - Like Dreamers Do
- Ugly Duckling - A Little Samba
- Soul Asylum - Black Gold
- The Beatles - A Taste of Honey
- Faust - So Far
- Aruna Sairam - Baje Mrudanga-Abhang
- Kirsty MacColl - Days (2005 Remaster)
- Harvey Danger - Flagpole Sitta
- The Beatles - Maggie Mae
- Xavier Naidoo - Dieser Weg
- Kenny Wayne Shepherd - True Lies
- Harvey Danger - Carlotta Valdez
- The Beatles - Derek Taylor With John Lennon
- Eliana - Meu Cachorrinho (Chihuahua)
- Yes - I've Seen All Good People: a. Your Move, b. All Good People
- Bad English - The Time Alone With You
- Faust - It's A Bit Of A Pain (2006 Digital Remaster)
- Gin Blossoms - Not Only Numb
- The Beatles - Come and Get It

APPENDIX B
PSEUDOCODE FOR STANDARD BOF

This is the algorithm used for training and 10-fold cross-validating standard BoF. Frames are short slices of the segments of a song. As noted previously, each segment has a timbre vector, a pitch vector, and a duration. These are used to build feature vectors for each frame. Features may be combined or split. In the former case, a feature vector for a frame will consist of the timbre vector, the pitch vector, and the duration vector for each segment concatenated together. For split features, there are three feature vectors for each frame: timbre, pitch, and duration. Each of these vectors is the result of concatenating each of these across the frames. Histograms are normalized by dividing the number of features in each feature class by the total number of features in a song.

```
Depending on whether or not we are using split or combined features:
    Split songs into 10 even subsets.
    For each subset of songs:
        The test set is the subset.
        The training set constains the rest of the songs.
        For every frame length, from 1 to 4:
            Split each of the training songs into overlapping frames of the given length.
            For every feature type:
                If the feature type is duration and the frame length is 1:
                    Instead of a histogram, find the mean and standard deviation of the
                        durations in a song.
                Otherwise:
                    Turn the frames into feature vectors of the current feature type.
                    Using a random 10% of these feature vectors from the training set, and
                        train a k-means clusterer.
                    For every song:
                        Using this clusterer, build a normalized histogram of the feature
                            classes of the frame vectors.
        For each song:
            Concatenate all of the histograms and duration statistics created for a song into a
                histogram vector.
        Using the histogram vectors for the training set, create an SVM using SMO.
        Test the SVM with the histogram vectors of the test set.
        Store the accuracy statistics.
    Average the accuracy statistics.
```

APPENDIX C
PSEUDOCODE FOR LINEAR TEMPORAL PYRAMID MATCHING

This is the algorithm used for training and 10-fold cross-validating linear temporal pyramid matching. Note that this actually yields accuracies for standard BoF when the maximum pyramid level is 0.

```
For a maximum pyramid level from 0 to 4:
    Depending on whether or not we are using split or combined features:
        Split songs into 10 even subsets.
        For each subset of songs:
            The test set is the subset.
            The training set contains the rest of the songs.
            For every frame length, from 1 to 4:
                Split each of the training songs into overlapping frames of the given length.
                For every feature type:
                    If the feature type is duration and the frame length is 1:
                        Instead of a histogram, find the mean and standard deviation of the
                            durations in a song.
                    Otherwise:
                        Turn the frames into feature vectors of the current feature type.
                        Using a random 10% of these feature vectors from the training set, and
                            train a k-means clusterer, called C.
                        For every song:
                            For every pyramid level L from 0 to the maximum pyramid level:
                                For each of the 2^L regions of the song:
                                    Using C, build a normalized histogram of the feature
                                        classes of the frame vectors.
            For each song:
                Concatenate all of the histograms and duration statistics created for a song
                    into a histogram vector.
    Using the histogram vectors for the training set, create an SVM using SMO.
    Test the SVM with the histogram vectors of the test set.
    Store the accuracy statistics.
Average the accuracy statistics.
```

References

[1] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002. [Online]. Available: http://dx.doi.org/10.1162/106365602320169811

[2] D. Liang, H. Gu, and B. O'Connor, "Music genre classification with the million song dataset," 2011.

[3] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, "The million song dataset," *ISMIR 2011: Proceedings of the 12th International Society for Music Information Retrieval Conference, October 24-28, 2011, Miami, Florida*, pp. 591–596, 2011. [Online]. Available: http://academiccommons.columbia.edu/catalog/ac:148381

[4] T. Jehan, "Creating music by listening," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[5] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *In International Symposium on Music Information Retrieval*, 2000.

[6] M. I. Mandel, G. E. Poliner, and D. P. W. Ellis, "Support vector machine active learning for music retrieval," *Multimedia Systems*, vol. 12, no. 1, pp. 3–13, Apr. 2006. [Online]. Available: http://academiccommons.columbia.edu/catalog/ac%3A144461

[7] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "Music classification via the bag-of-features approach," *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1768–1777, Oct. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865511002108

[8] T. Jehan and D. DesRoches, "Analyzer documentation," Sep. 2011, analyzer Version: 3.08. [Online]. Available: http://docs.echonest.com.s3-website-us-east-1.amazonaws.com/_static/AnalyzeDocumentation.pdf

[9] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, p. 1âĂŞ22.

[10] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," vol. 2. IEEE, pp. 2169–2178. [Online]. Available: http://hal.inria.fr/inria-00548585/en/

[11] T. Lehan, "FULL list of basic function of timbre?" Aug. 2012. [Online]. Available: http://developer.echonest.com/forums/thread/794

[12] "It's all too much," Dec. 2012, page Version ID: 525539382. [Online]. Available: http://en.wikipedia.org/w/index.php?title=It%27s_All_Too_Much&oldid=525539382

[13] J. Martens, "Deep learning via hessian-free optimization," J. Fürnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 735–742. [Online]. Available: http://www.icml2010.org/papers/458.pdf