APMA E4990 Final Project Report

Varun Ravishankar, vr2263 and Viktor Roytman, vr2262

May 10, 2013

1 Introduction

It's no surprise that people value the songs of their youth and often react with disdain at newer music. The problem with this mindset is that it is no way to judge music; it is completely subjective. What we would like is an objective way to quantify and compare songs and determine who's right. As it happens, with the Million Song Dataset we can do just that [Bertin-Mahieux et al. 2011].

We analyzed the Million Song Dataset in order to answer a few questions we had about the state of music and how it has changed over time. Our initial question was whether every song has the word "baby" in its lyrics. From this we decided to do a general analysis of word frequency, taking into account the prevalence of common words, and to highlight our favorite words. Next we considered the question of how to see trends in music over time. To do this we would need some musical qualities that are inherently quantifiable. The metrics we decided to study are song duration, tempo, and loudness, and we investigated the distribution and change over time for each one. By answering these questions, we can identify trends in the state of music and determine whether historical events have had noticeable effects on these musical characteristics.

In addition to our "baby" hypothesis, we had a few predictions for our metrics. There is a well-known phenomenon known as the "loudness wars," so we predicted the loudness of songs to increase over time. As an aside, the measure of loudness in this case is dBFS, or decibels relative to full scale. In other words, for a digital medium and some encoding scheme, there is a maximum possible amplitude denoted 0 dBFS, and amplitudes smaller than that are negative. We also predicted that the tempo of songs would increase over time (there was no metal music in the 1940s).

2 Datasets

To explore music trends over time, we used the Million Song Dataset [Bertin-Mahieux et al. 2011]. This dataset contains a collection of audio features and metadata for one million songs from 1920 until today, and is free for non-commercial use. This data was collected using the Echo Nest API and a copy of the musicbrainz server in December 2010. This dataset includes information on releases, musical qualities like tempo and loudness, and tags. This dataset is split into HDF5 files, a common format for data and scientific applications. The complete dataset is 300 GB, but we were able to focus on a smaller 300 MB dataset, msd_summary_file.h5, that contained just the metadata without any audio analysis, similar artists, or tags. We also used a SQLite database (track_metadata.db) containing most of the metadata on each track, including track ID, artist, and year. Finally, we used a list of 515,576 tracks in the dataset that also had year information, track_per_year.txt.

To explore song lyrics, we used another dataset available from the Million Song Dataset, the musiXmatch Dataset[1] [Bertin-Mahieux et al. 2011]. This dataset provides word counts per song for the most common 5,000 words in the data, but only contains the lyrics for 237,662 tracks because of copyright restrictions plus the removal of instrumentals (defined as having three words or less) and duplicate tracks. The word counts are stored in bag of words style, meaning position information is lost. Since there are so many words in the entire dataset, the authors chose to limit the dataset to just the 5,000 most common words, and also decided to run a modified Porter-Stemmer program on the lyrics to create the counts. The dataset comes as two files, mxm_dataset_train.txt and mxm_dataset_test.txt, for standardizing machine learning tasks.

[1] http://labrosa.ee.columbia.edu/millionsong/musixmatch

Since we were not making predictions in our analysis, we cleaned up the training and test datasets and then concatenated the two files together.

The data can be downloaded using the accompanying script download_msd.sh.

3 Technology and Development Environment

Our technology stack was simple. Our programs consisted of scripts meant to be run on POSIX machines. We tested and ran our scripts on Fedora 18 and OS X 10.8.3, using Python 3[2] and Bash[3]. For analyzing the lyrics dataset, we wrote scripts in Python to read in the dataset, find the word frequencies, plot the distributions, and output statistics on the dataset. We also used NumPy[4] to vectorize the computations and the Python natural language processing module NLTK[5] to find stopwords.

To analyze the Million Song Dataset, we used a combination of Python, NumPy, and pandas[6]. We also used queries on a SQLite[7] database of just the track metadata to avoid analyzing the full 300 GB dataset. Our graphs were created using matplotlib[8]. The database was optimized for counting and joining, and so we took advantage of the speedups possible when querying a relational database.

4 Lyrics

We first calculated the frequency of common words in songs. Because each word could appear in a single song more than once and heavily used words could heavily skew the word distribution, we chose to count a word occurrence only once per song. We wrote a Python script, read_lyrics.py, to read in the lyrics dataset, parse the file, store the counts in an array, and then rank the counts. We maintained a reverse index of word to count to enable querying word frequency.

We started with our initial hypothesis that the words "baby" and "love" occurred in most songs. However, the actual frequency of "baby" and "love" was far lower than our expectations, as seen in Table 1:

Word	Rank	Frequency
baby	110	13.35%
love	37	30.89%

Table 1: Frequency of "baby" and "love"

Because these frequencies were far lower than what we expected, we investigated the overall distribution of words and their frequencies, as can be seen in Figure 1. Figure 1 is a histogram of the word frequencies, and is heavily skewed. Most words have low frequencies, as indicated by the long tail in the histogram. This is further confirmed by the summary statistics on the lyrics distribution displayed in Table 2. The mean is relatively high, but the median is only 152. This evidence is compounded by the high standard deviation, indicating a heavily skewed distribution.

[2] http://www.python.org/
[3] http://www.gnu.org/software/bash/
[4] http://www.numpy.org/
[5] http://nltk.org/
[6] http://pandas.pydata.org/
[7] http://www.sqlite.org/
[8] http://matplotlib.org/

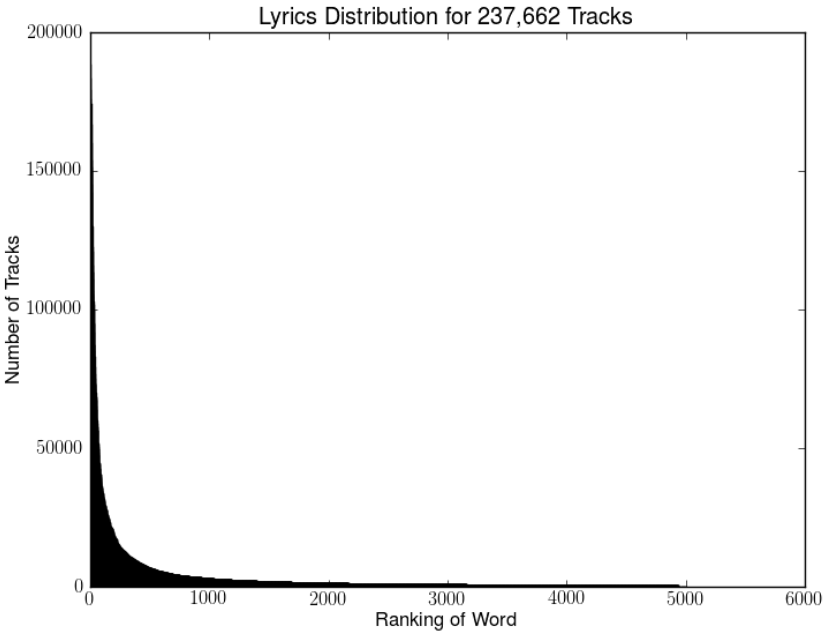

Figure 1: Distribution of Lyrics

Statistic	Measure
Count	5000
Mean	3809.07
Standard Deviation	12624.51
Min	73
25%	120.0
50%	151.99
75%	179.49
Max	188401

Table 2: Summary Statistics of Lyrics Distribution

Because of the skew in the data, we decided to look at the top 10 words to get a sense of the highest frequencies. This data, shown in Table 3, indicates that the top 10 words in the lyrics distribution are mostly stopwords, a term used in natural language processing to indicate words like articles and common prepositions that typically end a sentence and carry no special meaning. These stopwords occur in about 60%-70% of the data. Because they are so common, we decided that it was worth looking at terms that were not stopwords.

Rank	Word	Frequency
1	the	79.27%
2	a	77.19%
3	to	75.43%
4	and	74.02%
5	i	73.25%
6	you	70.17%
7	in	67.36%
8	is	64.63%
9	me	63.58%
10	it	62.91%

Table 3: Frequency of Top 10 Words

Using a list of stopwords provided by NLTK, we found the 10 most common words that were not stopwords, as seen in Table 4. This indicates that much of the top 50 words are stopwords, and that the rest of the common words only occur in about 30%-40% of the data.

Rank	Word	Frequency
29	know	38.56%
35	like	33.34%
37	time	31.50%
38	love	30.89%
43	see	29.39%
44	come	28.83%
45	go	28.82%
46	one	28.48%
50	get	26.74%
53	never	25.18%

Table 4: Frequency of Top 10 Words that are not Stopwords

Finally, we were curious about the prevalence of swear words in music after anecdotally observing an increase in both the acceptance and occurrence of swear words in today's music. We created a list of colorful language and found their frequencies, as shown in Table 5. Surprisingly, curse words occur in only 1%-5% of all songs. However, with the knowledge that this dataset contains 5000 words, this is still surprisingly high amount. The first curse word is only Rank 294, while the last one in the table is Rank 1286.

Rank	Word	Frequency
294	fu**	5.17%
333	hell	4.47%
360	shit	4.14%
416	blow	3.51%
593	ass	2.26%
614	b*tch	2.14%
628	n**ga	2.08%
936	fu**in	1.30%
1093	sex	1.06%
1286	d**k	0.88%

Table 5: Frequency of Top 10 Swear Words

5 Duration

The track_metadata.db file includes a duration for all of the million songs and a year for about half of them. We collected the data by nesting a query for the count and average duration of songs by year in a query for the year and average duration where the count is at least 10 and the year information is present. We then created a line graph of average duration as a function of year, seen in Figure 2. There is a very clear jump in average duration around 1970.

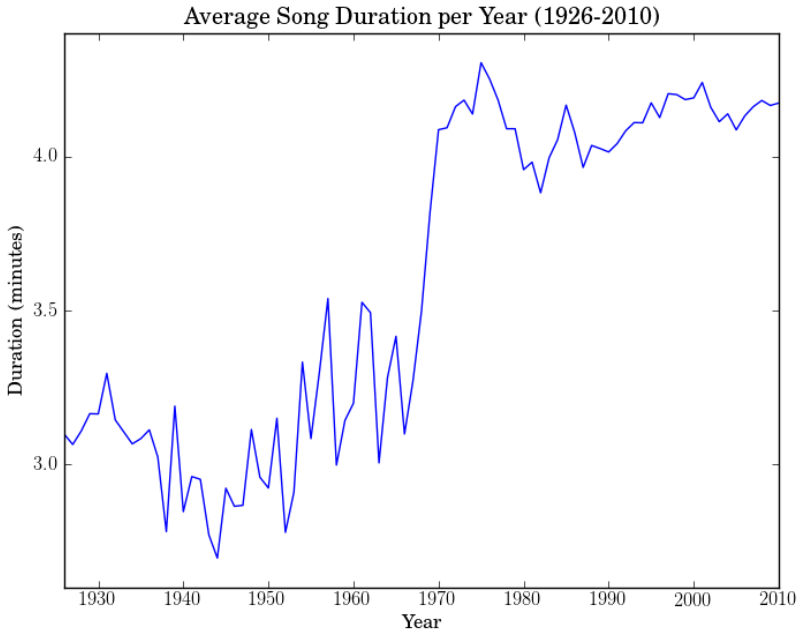

Figure 2: Average Song Duration Over Time

The creation of the duration histogram is even more straightforward: it is just the binning of a query for duration. This histogram can be seen in Figure 3. There are very few songs with exceptionally long duration, and we had no success confirming that the songs with the longest duration in the dataset were actually that long (30 - 60 minutes). The summary statistics for this distribution, displayed in Table 6, verify this and tell us that there is a high spread but little variation for most songs.

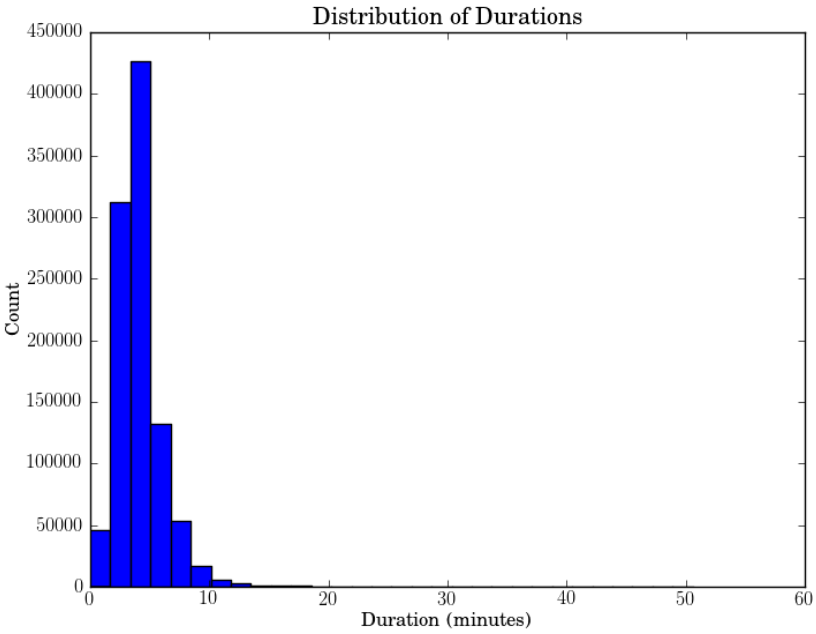

Figure 3: Song Duration Distribution

Statistic	Measure
Count	1000000
Mean	4.16 min
Standard Deviation	2.10 min
Min	0.0052 min
25%	0.30 min
50%	0.45 min
75%	0.58 min
Max	50.58 min

Table 6: Summary Statistics of Duration Distribution

6 Loudness

The loudness information is in a summary HDF5 file, msd_summary_file.h5. We pick out only the relevant columns, which are track_id and loudness. We then step through another file, tracks_per_year.txt, assembling lists of loudnesses per year. Finally, we filter out years with fewer than 10 songs, and take the average of the loudnesses. The resulting graph clearly shows an upward trend in loudness over time, and can be seen in Figure 4.

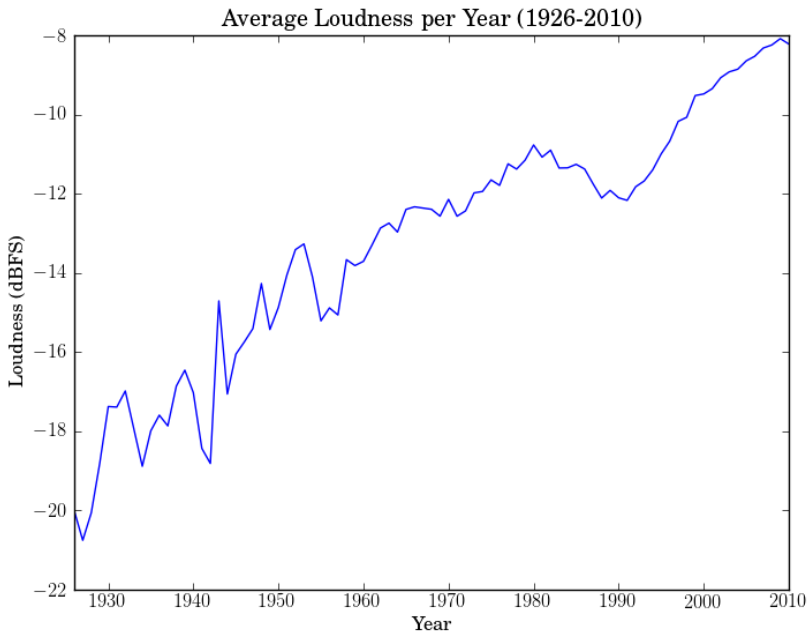

Figure 4: Average Loudness per Year

The loudness histogram, seen in Figure 5, is a simple binning of the loudness information from the summary file. It is skewed heavily toward a loudness of 0, since most of the songs in the dataset are recent, and the more recent songs tend to be louder. This is further confirmed by Table 7, which displays the summary statistics for loudness. We can see that the distribution is heavily skewed by the fact that the mean and median are so far apart, with further evidence given by the large standard deviation and large extremes given by the min and max.

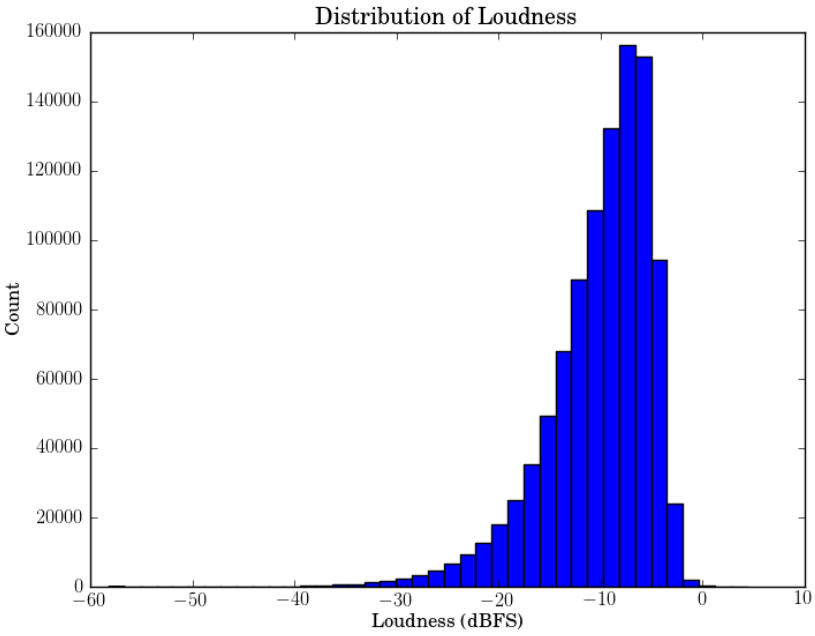

Figure 5: Song Loudness Distribution

Statistic	Measure
Count	1000000
Mean	-10.12 dBFS
Standard Deviation	5.20 dBFS
Min	-58.18 dBFS
25%	-33.03 dBFS
50%	-30.26 dBFS
75%	-28.54 dBFS
Max	4.32 dBFS

Table 7: Summary Statistics of Loudness Distribution

7 Tempo

The acquisition method of the tempo information is very similar to that of the loudness information, since it is also in the summary HDF5 file. The resulting graph, seen in Figure 6, shows wide swings in average tempo for the early years in the dataset, with a severe leveling off in recent years. It is not clear why there are such swings for the early years. It could be an artifact of the dataset or a sign of fashions in music changing quickly. In any case, generally tempo has increased with time.

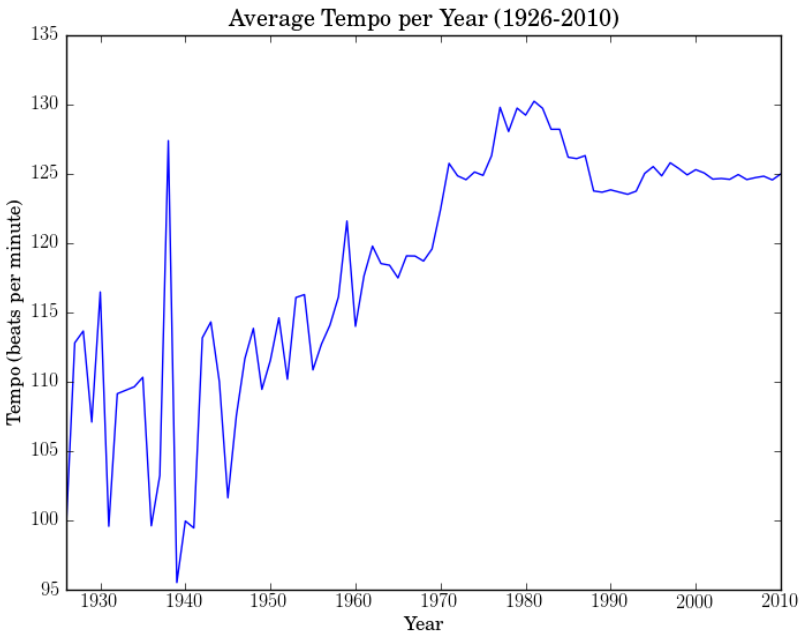

Figure 6: Average Tempo per Year

The histogram, seen in Figure 7, shows something like a normal distribution for tempo, with a longer tail of high tempo songs. We excluded songs with a tempo of 0, since we considered those particular songs to be untagged. However, like duration, we had no success confirming that the songs with the highest or lowest tempos were catalogued correctly.

The summary statistics for the distribution, shown in Table 8, show a high spread for the values and show that while the distribution looks roughly Gaussian, is very skewed. A high standard deviation only confirms this.

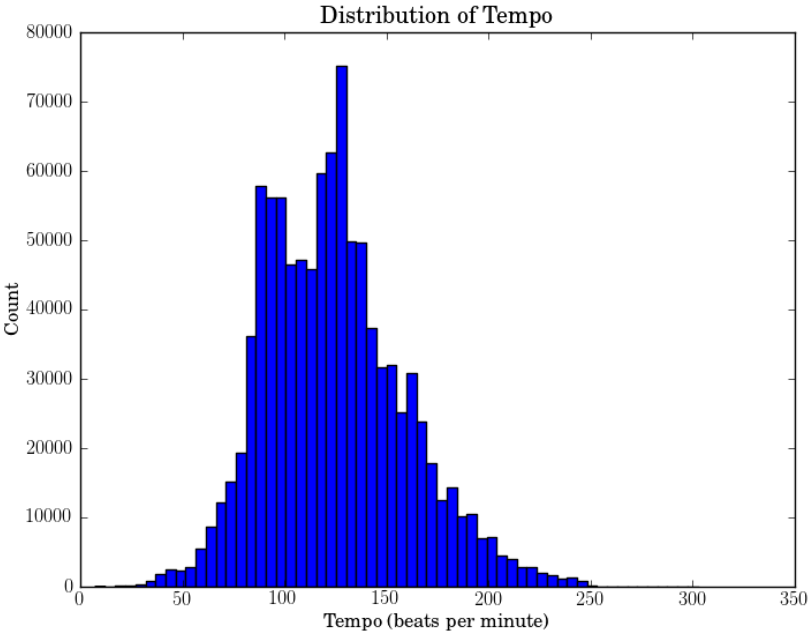

Figure 7: Song Tempo Distribution

Statistic	Measure
Count	996770
Mean	124.29 bpm
Standard Deviation	34.39 bpm
Min	7.36 bpm
25%	39.77 bpm
50%	45.02 bpm
75%	50.00 bpm
Max	302.3

Table 8: Summary Statistics of Tempo Distribution

8 Future Work

In the future, we would like to focus on genre analysis. We originally planned to use Amazon EC2, Amazon EMR, and Hadoop to explore the Million Song Dataset. We planned to find genres and then calculate duration, tempo, volume, and other features per genre. This did not work out because when we explored some of the HDF5 files in the dataset, the genres were set to empty strings. This was true even when we explored the full dataset. To further complicate matters, the dataset did not list genre as a valid field. Instead, it seems that we would have had to explore the musicbrainz tags, which are stored as arrays of strings per song. Because these tags are only a weak indicator of genre and because we tried to avoid looking at the full dataset as much as possible, we decided it would be best to explore an alternative dataset. We began to look at the Last.fm dataset[9] [Bertin-Mahieux et al. 2011], but ran out of time before we could explore this fully. We also wanted to look at lyrics distribution over time, but the format of the datasets did not make that straightforward. Given more time, we would look into the frequencies of words used over

[9] http://labrosa.ee.columbia.edu/millionsong/lastfm

time, whether the top words changed over time, and whether words such as curse words starting becoming more prevalent in the 80s and 90s.

9 Conclusion

Most of our hypotheses were somewhat off the mark. While "love," discounting the stopwords, is one of the most common words, "baby" is much further down in rank, contrary to our expectations. The most common of these words are "know," "like," and "time." In addition, fu** is a surprisingly common word in songs, occurring in over 5% of all tracks.

Our analysis of the metrics reveal some interesting discoveries. There is an obvious jump in duration around 1970, indicative of better music storage technology, and most songs don't deviate far from around 4 minutes in duration. Loudness increases more or less steadily over the years, and recently it has been getting rather close to 0 dBFS. Surprisingly, there are large swings in average tempo up until the 1960s or so, but after around 1990 the average tempo levels off. This may be indicative of a trend toward greater homogeneity in songs in recent years.

References

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.