What Will Be Your Favorite Song? Let Me Tell You!

Clement Ntwari Nshuti
Stanford University
cntwarin@stanford.edu

Steven Soria, Jr.
Stanford University
ssoriajr@stanford.edu

Abstract

Previous efforts at predicting song popularity have often focused on music-analytic features, such as properties of the waveform, number of beats, etc., or on NLP analytics such as lyrics, sentiment, and keyword frequency. Recent resurgence in related research has shown some success, with conferences and challenges generating interest in applying machine learning strategies. In this project, we explore combinations of both music-analytic and qualitative features, as well as unsupervised feature selection techniques, to improve upon prediction success. Can the intrinsic properties of a song, such as frequent beats and higher decibel levels, inform popularity? Or are more qualitative features, such as danceability and similarity to other popular songs, sufficient to produce a hit single? With our best feature set, the answer seems to be the latter. Most effective were features correlating current songs with other similar songs, and their respective popularities. Using a Weighted Linear Regression model against the Million Song Dataset, combined with a distortion matrix, we were able to achieve a final Root Mean Squared Error (RMSE) of 10.4%.

1. Introduction

Our motivation stems from the broader challenge of predicting hit songs for some future time period. Developing an understanding of how predictor features themselves change over time with respect to contemporary hit songs may drive models that move beyond song prediction, and into forecasting evolving genre preferences and musical tastes. The first step, however, must be to build a reliable model for predicting popular songs with features that provide insight into the notion of popularity itself. Thus, we focus on predicting song popularity through a combination of supervised and unsupervised learning techniques, to develop novel feature sets which attempt to exploit the nature of what makes a "hit song".

An example of recent prior work, *Hit Song Once Again a Science?*[4] also attempts to predict popularity, but its focus is on United Kingdom song data using primarily music analytic features.[5] Popularity ground truths are based on UK Top 40 Singles Chart data.[5] For United States-based data, we relied primarily on the Million Song Dataset (MSD). Like the prior citation, this dataset also incorporates music analytic features from The Echo Nest.[1] However, the MSD does not directly include US sales data, which is more difficult to obtain from public sources, rather it provides a "hotness" metric, which we use instead as our ground truth. Whereas the cited work predicts popularity for a specific year, our focus is on predicting popularity, period, of which year is but a single feature. Other comparisons, such as performance, are difficult due to lack of specific metric details in the cited work.

2. Methodology

2.1. Data

The Million Song Dataset specifies, for each song spanning from 1922 - 2010, such qualitative features as artist familiarity, artist popularity, danceability, energy, and song popularity, and quantitative features such as artist location, year, loudness, and music analytic array features for bars, beats, segments, sections, tatums, and so on. We developed a Matlab infrastructure to parse the one million MSD data files into a feature matrix. Since our goal was to predict song popularity, within the broader motivation of modeling temporal changes, we filtered the data to ensure that both *year* and *song_hotttnesss* (hotness) features were strictly greater than 0. The specifics of the hotness metric are opaque, with MSD providing real-valued numbers in $[0, 1]$. With only this filter applied, the number of usable samples dropped to 8,163.

The MSD has spawned a significant number of derivative projects. Last.FM provides SQLite databases which correlate MSD songs with similar artists, a similarity score, and genre tagging data.[1] We utilized the mksqlite[3] extension to read SQLite data directly from Matlab, and correlate it with MSD data. Our infrastructure allowed us to interchange models, generate features from database data as well as MSD file data, plot results, easily turn on/off features of interest, and track results from different test runs.

Just as MSD had proven to be sparse when filtering for year and hotness, many other qualitative features were also too sparse to be usable. We therefore had to eliminate features such as artist latitude and longitude, energy, and danceability. Samples were chosen to equalize representation across the hotness spectrum, into 10 bands of 0.10 increments. This was to ensure adequate learning for all gradations of hotness. The MSD did not include songs with hotness less than 0.10, and samples were sparse for hotnesses of 1.0. We allowed hotnesses of 1.0 anyway, with the intuition that we might learn strongly associated features. The resulting feature matrix was then normalized. From this, we randomly sampled our training and test sets according to an 80/20 split, with the ability to re-use the same permutation from prior runs, to fairly compare performance among subsequent runs. The final sample size for our best feature set was 5,667.

2.2. Algorithms

Weighted Linear Regression

In the early phases of our work we considered linear regression and logistic regression. That choice was motivated by the fact that these two regression methods allow to quantify the relevance of each feature through a statistical analysis of the relevance of the parameters. However, due to their poor performances (with RMSE of 0.1849), we decided to use weighted linear regression instead. During our experiments, as described in section 3 we saw that the results of weighted linear regression could be improved by introducing a distortion matrix M in the computation of the weights. Specifically for a test sample x and training sample $x^{(i)}$ we compute the weight as follows :

$$w(i) = \exp\left(-\frac{(x - x^{(i)})^T M^T M (x - x^{(i)})}{2\tau^2}\right). \quad (1)$$

By introducing the distortion matrix M, our goal is to have distorted samples $\tilde{x} = Mx$ and $\tilde{x}^{(i)} = Mx^{(i)}$ such that if $\tilde{x}$ is close to $\tilde{x}^{(i)}$ then their corresponding *hotnesses* are also close. The learning of the optimal distortion M was done using Weinberger and Tesauro's metric learning algorithm [6]. Motivated by our intuition that songs released in the same year have the same hotness model, we also tried a modified weight

$$\tilde{w}(i) = w(i)\lambda^{|x_{year} - x_{year}^{(i)}|}$$

where x_{year} is the release year of the song x and $\lambda \in (0, 1]$ is a decay parameter optimized through cross-validation.

2.3. Evaluation Method

We considered two metrics for the evaluation of our results : the root mean squared error (RMSE) and accuracy (ACC) defined as

$$ACC = 1 - P(|\hat{y} - y| > 0.1)$$

i.e. the probability of having an estimate $\hat{y}$ of the hotness that lies within 0.1 of the true value.

3. Experiments

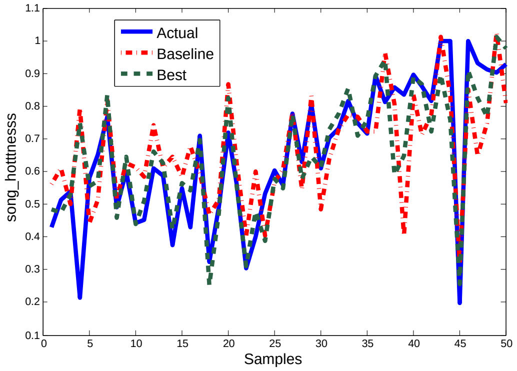
Our experiments fell into two main categories: features engineering and metric learning. For features engineering, substantial effort went into unsupervised feature selection for music analytics, but the primary driver of results improvements were from supervised feature selection for mostly qualitative features.

3.1. Features Engineering

Supervised

Our first choice of features were simply the real-valued features from the MSD for each song: *artistID, sample_rate, artist_familiarity, artist_hotttnesss, duration, end_of_fade_in, key, key_confidence, loudness, mode, mode_confidence, start_of_fade_out, tempo, time_signature, time_signature_confidence, year*. For the music analytic array features, *bars, beats, sections, segments, tatums, terms*, we simply converted them into a simple count where, for example, the beats array size represents the number of beats in a song. Following this example, even though we were unable to use the *energy* feature due to sparsity, the number of beats in a song might infer an energy level. We reasoned that more beats might imply a faster pace, and hence a more energetic song, that perhaps was more appealing to a consumer than, say, a slower song with fewer beats. As expected, this initial, naive choice of features performed poorly, with an RMSE of 0.169 per sample. We also iterated over several choices for τ and found that $\tau = 4.0$ consistently provided the best results throughout the remainder of our experiments.

Figure 1: *Plot of predicted hotnesses for 50 test samples from our baseline and best feature set. Note how our best feature set results in a curve which more closely follows the actual curve.*



The most substantial improvement came from the intuition that, at any given time, popular songs tend to be similar to each other. While song popularity can vary by genre, the most popular songs tend to be of the same genre. Using the Last.FM data set, we extract similar songs for a given song and correlated this with the MSD to get the artist, artist familiarity, and song hotness for each similar song. Typically, songs with the highest similarity scores were by the same artist. We allowed this since an artist with a hit song is likely to generate additional interest as a result, increasing the likelihood that existing songs in the artist's catalog might benefit from this exposure, as well as influencing listener acceptance of new songs by the artist. We also selected the most similar song and its hotness from a different artist, in the hope of capturing additional insight into popularity offered by this new artist. Generally, similar songs from unique artists had lower similarity scores and so these features did not improve results. However, using the song hotness from the top two similar songs improved RMSE substantially, by 5.6%, to 0.113.

We experimented with many different features, primarily qualitative, since music analytic features were largely driven by unsupervised feature selection, as explained in the next section. Such features included the number of years an artist was represented from among our usable sample set, the idea being that artists with greater staying power in the industry were likely to produce a higher number of popular songs. From this, we also considered the total number of hit songs produced by an artist, where "hit song" was iteratively defined as a hotness value greater than or equal to 60, 70, 80, and 90 percent. Similarly, we defined features that counted the number of unpopular songs as well, where hotness was iteratively less than or equal to 40, 30, and 20 percent. Among music analytic features, we considered the idea that many consumers decide whether or not they like a song from 30-second preview clips. Therefore, whereas we had primarily worked with total counts initially, we looked at first 30 second counts instead. Similarly, we looked at the sum of max loudnesses across song segments from the first 30 seconds, as well as the average loudness over that same 30 seconds. Most of these features degraded RMSE, with some resulting in changes that were negligible. Among the permutations, we also tried removing our best features, to see if the remaining features would become more important, possibly yielding lower RMSE. They did not, with RMSE returning to near baseline levels.

Unsupervised

The challenge with the array-features in the MSD dataset is that they are of varying length for each song. We focused our efforts on the *loudness,pitches and timbre* array-features because of their close relation to the musical content of the song. The loudness of each song was interpolated on 100 equally spaced time instants over the normalized song duration interval $[0, 1]$. In order to reduce the dimension of the pitches and timbre features, PCA and k-means clustering was performed for these features on each song. Specifically, we computed the 12 principal components for each of these features, tiled them by decreasing eigenvalue and used the resulting vector as a feature. The choice of 12 principal components was based on the result that, on average, 12 eigenvectors are necessary to describe more than 90% of the energy contained in these array features. Additionnally we also computed 12 centroids for the same features and tiled by decreasing order of cluster density. The choice of 12 centroids was made using Sugar and James' information theoretic approach to unsupervised k-means clustering [7].

3.2. Metric Learning

By using distored samples $Mx^{(i)}$ we get a correlation between $||M(x^{(i)} - x^{(j)}||$ and $||y^{(i)} - y^{(j)}||$ (see Figure 3) that was not as significant as in the non-distorted case (see figure 2). This resulted in a slight improvement of the performances as described in section 4.

Figure 2: *2D histogram of the pairwise distances between songs and the corresponding pairwise hotness differences. Before distortion, there is no significant correlation between $||x^{(i)} - x^{(j)}||$ and $||y^{(i)} - y^{(j)}||$.*

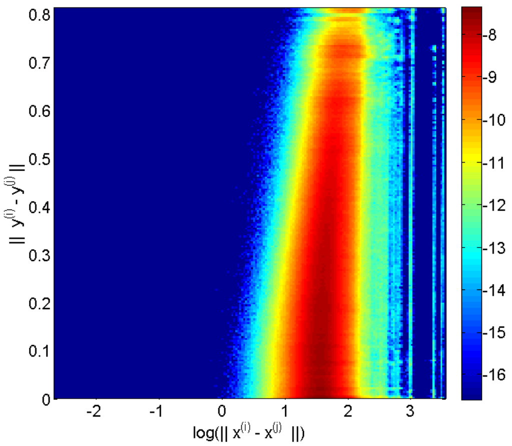
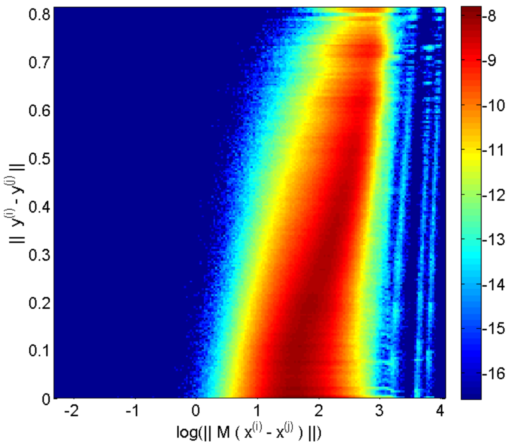


Figure 3: *Thanks to the distortion matrix M, a correlation between $||M(x^{(i)} - x^{(j)}||$ and $||y^{(i)} - y^{(j)}||$ appears. This slightly improved the performances of the WLR, as we will see in section 4.*



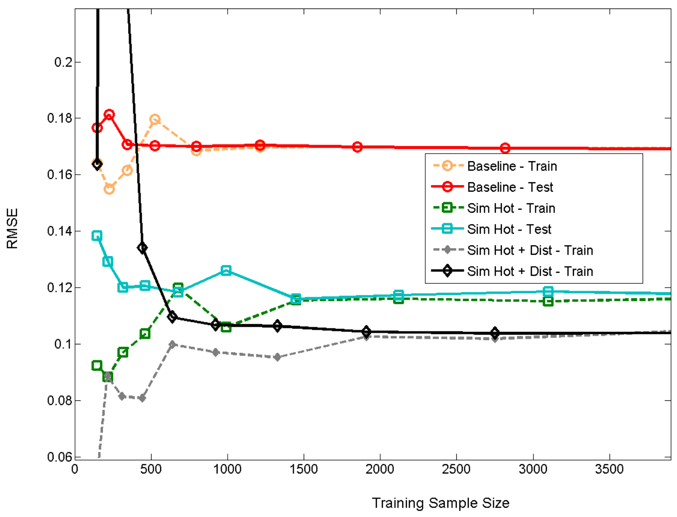
4. Results

As described in section 2, we considered two improvements from the baseline. First of all, due the high bias of our baseline (see Figure 4) the best fix was to find additionnal features. We began by adding the hotness of the two most similar songs as features for a song. This resulted in a substantial improvement as summarized in table 1 (row "Sim Hot"). Time decay ("Sim Hot + TD") and the additionnal array features were added to the model. Both resulted in worse performances.

Table 1: *To try to reduce the high bias of the baseline system, we investigated adding several features. The best improvements were seen after adding the hotness of similar songs (Sim Hot) and introducing an optimal distortion matrix (Sim Hot + Dist).*

	RMSE	ACC
Baseline	0.169	48.5%
Sim Hot	0.113	72.1%
Sim Hot + TD	0.121	71.5%
Sim Hot + Array Features	0.129	70.4%
Sim Hot + Dist	0.104	73.3%

Figure 4: *Adding the hotness of similar songs as a feature (Sim Hot), decreased the RMSE from 0.169 for the baseline to 0.113. Our interpretation is that this feature allowed us to capture some part of the decision processes of the user that are not solely based on the songs intrinsic musical properties. These could be, for example, related to how an artist markets their music. We can also see in this figure the slight improvement gained by using an optimal distortion matrix M, which allowed us to decrease the RMSE to 0.104 (Sim Hot + Dist).*



Time decay degraded the performance due to the strong, and apparently incorrect, assumption that it made on the model. As for the additionnal array features, we suspect their negative impact to be the result of the curse of dimensionality. Indeed, after adding these features, we had a total of 428 features for approximately 5000 samples. Dimensionnality reduction through PCA only allowed us to reduce the number of features to 316. The final improvement attempted was to introduce the distortion matrix described in 2.2. This resulted in only a slight improvement to the RMSE of 0.09.

Several other features were considered, as described above. However their impact was insignificant.

The results of our best performing system are summarized on Figure 4. It illustrates perfectly on one hand the reduction in bias from the introduction of the hotnesses of similar songs, and on the other hand the slight improvement due to the introduction of the distortion matrix.

5. Conclusions and Future Work

While our objective was to find a novel mix of qualitative and music analytic features that together improved song popularity prediction, it turns out that our best result is based on comparing popular songs to other popular songs. One might consider this analogous to looking for new songs by browsing the iTunes "Top" lists. Our best feature set, combined with a distortion matrix, yielded an accuracy of 73.3%, several percentage points higher than our baseline results.

Despite the lack of contribution from the music analytic features we attempted, several data sets spawned from peripheral MSD projects have far extended that variety of such data. An *energy* and *danceability* metric might yet be created based on mining results from these various data sets. Indeed, the prevalence of data sets focusing on specific areas of machine learning tasks for music often had pieces of data that might have been useful, but was infeasible given our time constraints. Other ongoing work is in sentiment classification of songs themselves, primarily based on lyrics and title, but some based on music analytic features as well. This adds a unique dimension to the notion of popularity, and how it might be informed by sentiment, or more straightforward keyword and genre tagging features. Other data sets purport to include listener play counts from sources such as iTunes and online streaming sites, though, a first glance at this data seemed too sparse to be usable at this time. Sources of error primarily stem from use of the MSD. Derived from The Echo Nest[2], the MSD was created in December 2010 and has not been updated. It's possible that mining the Echo Nest directly, through its online API set, would have reduced the sparsity problem. However, use of the API set is relatively inefficient, given our time constraints, for mass feature generation and correlation of results.

Our ability to deliver substantive improvement over initial results shows that there is still progress to be made in song popularity prediction. Given the availability of data sets from a variety of music projects, finding insightful features that exploit both the structure and semantic of music itself, have the potential to extend models like ours into more abstract and powerful machine learning systems that may someday tell us what our favorite songs will be!

References

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

[2] Tristan Jehan and Brian Whitman, et al. The Echo Nest. `http://the.echonest.com/`. November 2013.

[3] Martin Kortmann Welcome to the Project mksqlite! `http://mksqlite.berlios.de/`. August 2008.

[4] Yizhao Ni, Raul Santos-Rodriguez, Matt Mcvicar, Tijl De Bie (2011). Hit Song Science Once Again a Science? *In Proceedings of the 4th International Workshop on Machine Learning and Music: Learning from Musical Structure (MML2011).* `http://www.tijldebie.net/system/files/MML2011-final.pdf`. 2011.

[5] Yizhao Ni, Raul Santos-Rodriguez, Matt Mcvicar, Tijl De Bie. Score A Hit. `http://www.scoreahit.com/`. 2011.

[6] Kilian Q. Weinberger,Gerald Tesauro (2007). Metric Learning for Kernel Regression. *In Proceedings of the Eleventh International Workshop on Artificial Intelligence and Statistics (AISTATS-07), Puerto Rico*

[7] Catherine A. Sugar and Gareth M. James (2003). Finding the number of clusters in a data set: An information theoretic approach. *Journal of the American Statistical Association* 98 (January): 750 —763