Predicting a Song's Path through the Billboard Hot 100

Cristian Cibils, Zachary Meza, Greg Ramel

Stanford University, California

Abstract

In this project we explore the space of predicting the $(n+1)^{th}$ position in the Billboard Hot 100 charts given the first n positions and musical data. We can then use this prediction as the next entry to predict the $(n+2)^{th}$ entry, and keep repeating this regression until the song drops out of the charts. Considerable research has gone into predicting the commercial success of a song based on its innate characteristics, yet no major effort has endeavored to use previous chart success as a predictor of future success. Our results display a robust prediction engine that leverages both nonmusical and musical features to predict the full path of a song across the Billboard charts.

Keywords: Machine Learning, Music, Hit Songs

1. Introduction

The Billboard Hot 100 has been a reliable source for song popularity rankings over the past sixty years. There is an undeniable appeal for artists and record labels to be able to predict the path of their songs along the Billboard rankings – artists want to compose more popular songs, and labels want to invest in more popular artists. Even with the advent of industry-shattering changes in the world of music – namely, the introduction of digital distribution mechanisms – Billboard remains a go-to source for assessing the success of a pop song.

We trialed an array of different ML algorithms in order to most effectively learn parameters that can determine a song's path (defined as the position it holds on the charts in a given time interval) through the Billboard 100. The input to our algorithm is the preceding chart positions and qualitative features from Gracenote including mood, genre, and artist gender. We then use a linear regression algorithm, specifically Ridge regression, to output a predicted value for the song's next position in the chart. Our problem specification can then be defined as follows: given a song in the Hot 100 and its history, predict its position in the following week. What we hypothesize is that if these predictions are good enough, then we can add the results of the predictor to the history and extend it to an arbitrary number of weeks so that we can predict its whole path on the Billboard 100 chart.

2. Related Work

There has been a significant amount of effort in the field of predicting the commercial success of a song with different sources and features as input. We analyze some of the literature in the field in the following section.

Chon et al [1] use the LMS algorithm to explore how sales data affects chart position. Their findings show that the higher a song starts on the charts, the longer it remains there and the higher the likelihood of climbing to the top. One notable issue with Chon, et al's paper is that most of their data comes from the Top Jazz chart, where there might be some bias in listening patterns depending on music genre.

Koenigsten et al [2] use Peer to Peer networks (such as Napster) to predict Billboard success. Their results show a tight correlation between Peer to Peer network success and chart success, however, the direction of correlation or causality is left unclear.

Ni, et al [3] train a learner using only audio features to predict the commercial success of a song. They try to predict whether the song will be a hit (peak at a position higher than 5), or not (a position greater than 40 and less than 5). Their results show a 60% accuracy achievement.

In a study of Social Networks, Bischoff et al [4] show that it is possible to accurately predict the commercial success of a song given its the relationships formed in the graph structure of the network. Using sophisticated data mining techniques, they generate a graph from Music Social Networks such as Pandora, Last.fm, and Soundcloud.

Dewan et al [5] use a similar approach to [4], except they focus on mainstream social media such as Facebook and Twitter. They explore the effectiveness of old media and new media in trying to predict the success of a song. Their findings show a positive correlation between radio plays and commercial success, but no such correlation is found between social network activity and commercial success.

Our implementation differs fundamentally from those above in that we are trying to predict the popularity of a song as expressed primarily by the Billboard past data itself, while the research above relies largely on external data. Although we use musical features as well, most of our accuracy (as will be shown in the following sections) comes from the trend data.

3. Dataset and Features

Our dataset was drawn from the Billboard website [6], leveraging the fact that only very recently Billboard made this data public. We have data for every week starting from mid-year 1958 to the first week of December of 2015, adding up to 2,984 weeks of analysis total. Considering the presence of distinct songs across those weeks, we have a total of 24,469 song paths to analyze.

We modified our data from raw week-to-week top 100 rankings to instead represent it as each songs' paths through the Hot 100. That path took the form of a list of week entries, which contained the year, the number of weeks into the year, and the current rank. We used the number of weeks into the year instead of the absolute date to more accurately capture the position in the year generally across years. Below is an abbreviated example for Adele's 'Hello':

```
"Hello ::: Adele": [[2015, 45, 1], [2015, 46, 1]...]
```

We enhanced our collection of features to by using the Gracenote Audio database [7]. We think it is interesting to analyze the relationship between past success data and musically intrinsic data to see which best contributes to our prediction. Some of the fields used are: *Mood*, *Tempo*, *Artist Origin*, *Genre*, *Key*, *Artist Era*, *Artist Gender*. We analyze the relationship between musical and nonmusical features to see which are more fit at predicting future chart position.

In order to validate our algorithms, we use K-fold Cross Validation with $k = 10$. It was most striking to see, however, the predictions for this current year as compared to the actual chart positions.

4. Methods

In our first attempt we used Ridge Regression (as implemented by the scikit-learn library [8]). The features used were only the positions of the song in the previous n weeks. We tried this with several different values of n to see which performed better, and we extended our analysis not just to predicting the $n+1$ case, but also to predicting the whole path of the song. We also considered adding features such as year, and week of the year, but after a simple trial we realized that the features increased our error.

We then tried an array of different algorithms to see which produced the best results. The results are shown in the next section.

5. Results

5.1. Error Function

For the case of predicting the $n+1^{th}$ position, we calculate the error simply by getting the average absolute value of the difference between our prediction and the actual position. We can express this error as:

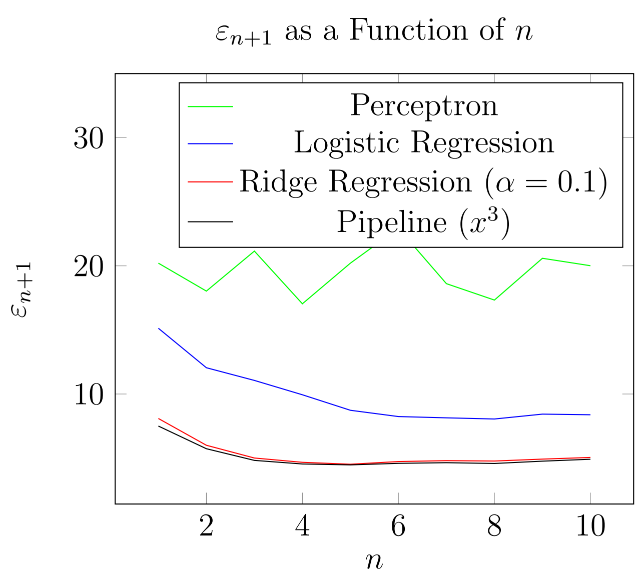
$$\varepsilon_{n+1} = \sum_{i=1}^{m} |h_{n+1}(x^{(i)}) - y_{n+1}^{(i)}|$$

Where $h_n(x^{(i)})$ is our predicted value given the i^{th} n-sized vector of previous positions ($x^{(i)}$). If we generalize our predictions to predict a path of songs with duration on the charts of at least $2n$, where again n is the number of past positions we observe, then we get the following error metric:

$$\varepsilon_{Path} = \sum_{j=1}^{m} \sum_{i=1}^{len(Path)} |h_{n+i}(x^{(j)}) - y_{n+i}^{(j)}|$$

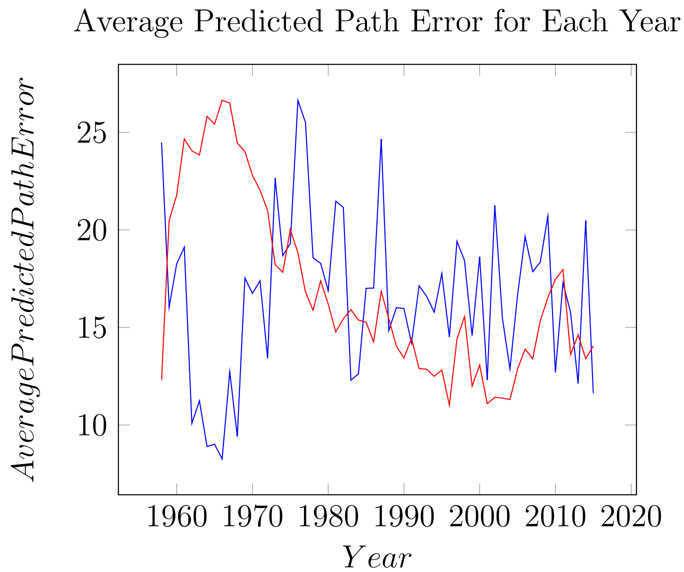
5.2. Past Data Only

Our analysis of predicting the immediate future position based on past n positions yielded the following results:



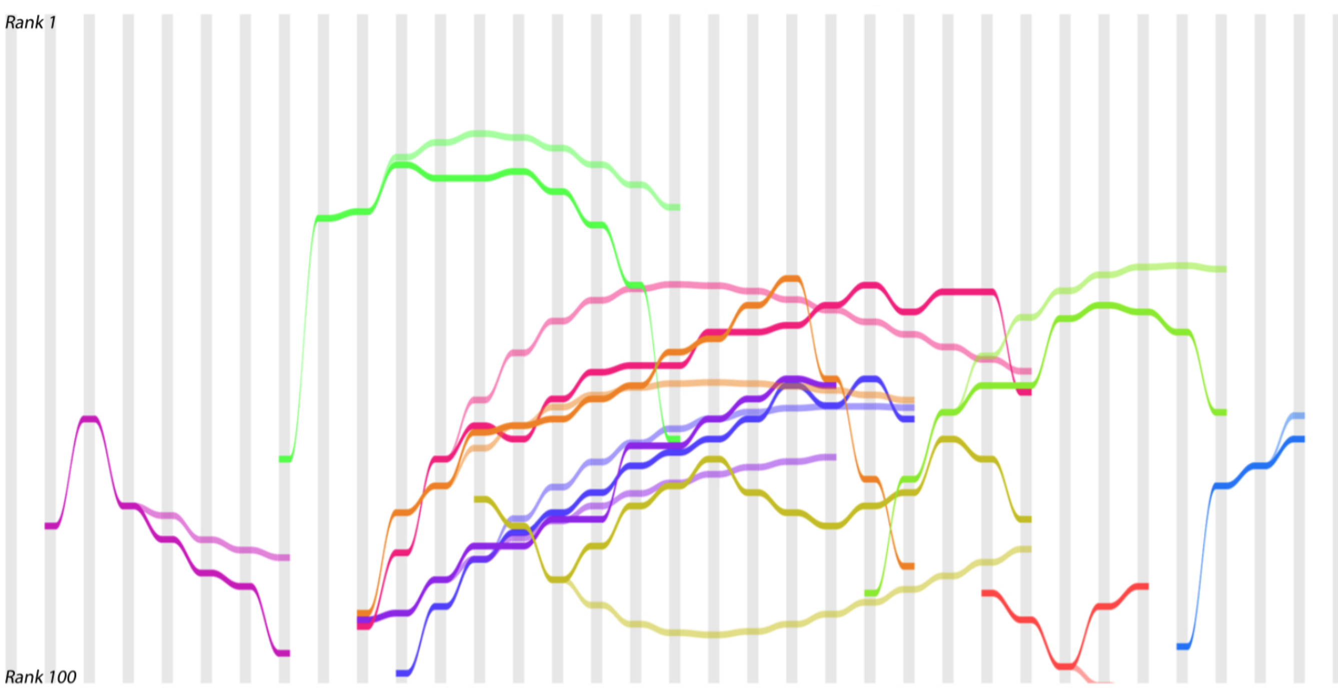
In turn, this means that our best attempt (Pipeline cubic regression, $n = 5$) was on average only 4.47 positions away from the actual result. We think that this result is remarkable given the simplicity of our model and the high accuracy it presents. With this knowledge, it becomes trivial to compute a probability for a song to reach 'hit' status given its history.

In predicting the path for a song given a window size of four in our test set, we observed some interesting trends year to year. Below is the average error for paths of length $2n = 8$ from our test set from each year. Also plotted on the chart is the number of distinct songs in each year, normalized to fit the same scale as the path error (by a factor of approximately 28).



From this plot, we can see that there is a roughly proportional relationship between the number of distinct songs in a year. That is, the more songs there were, the less error there was. The obvious takeaway is that there are more examples for training in years with more songs. However, we also note that for a year in which there are more songs, the songs tend to take on shorter paths. We hypothesize that these shorter paths are more direct in their representation of the general trends. For instance, shorter song paths will have less of a tendency to rise slowly over numerous weeks and then drop sharply - the rise and fall will be more evenly distributed.

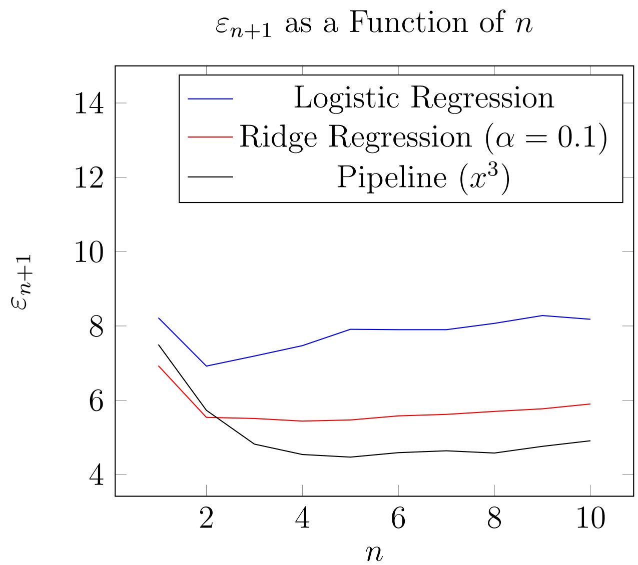
We can see our performance for path-wide prediction more specifically in the following visualization of our algorithm evaluated on test set songs from 2015:



Here, each grey bar represents a week of the year (our visualization shows 2015). The lighter colors represent our predictions, and the darker colors represent the actual positions. Our average ε_{Path} for 2015 was 12.03, using $n = 3$. We find that our algorithm is good at predicting climbs and peaks, but fails to accurately predict descents from the charts.

5.3. *Enhanced Past Data*

In this section, we experimented adding the features from the Gracenote database to our preexisting features. Here are some of our results:



A simple comparison lets us see that the features added from Gracenote, namely *Mood* and *WeeksInChart*, only clutters the same classifiers that would produce better results without it. The best performing algorithm in this case was pipeline with $n = 4$

6. Conclusions

Overall, we found that using previous positions to predict the future is a useful and reasonable feature set. Our results are surprising given the simplicity of our model, and perhaps even more surprising, given that if we add musical data, our results become worse. There are Billboard mechanics that can be captured better, namely the rapid drop from the charts, yet it is remarkable to see how accurately our algorithm predicts peaks.

7. References

[1] S. H. Chon, M. Slaney, J. Berger, Predicting success from music sales data: a statistical and adaptive approach, in: Proceedings of the 1st ACM workshop on Audio and music computing multimedia, ACM, pp. 83–88.

[2] N. Koenigstein, Y. Shavitt, N. Zilberman, Predicting billboard success using data-mining in p2p networks, in: Multimedia, 2009. ISM'09. 11th IEEE International Symposium on, IEEE, pp. 465–470.

[3] Y. Ni, R. Santos-Rodriguez, M. Mcvicar, T. De Bie, Hit song science once again a science, in: 4th International Workshop on Machine Learning and Music, Spain, Citeseer.

[4] K. Bischoff, C. S. Firan, M. Georgescu, W. Nejdl, R. Paiu, Social knowledge-driven music hit prediction, in: Advanced Data Mining and Applications, Springer, 2009, pp. 43–54.

[5] S. Dewan, J. Ramaprasad, Social media, traditional media, and music sales, Mis Quarterly 38 (2014) 101–121.

[6] Music: Top 100 songs — billboard hot 100 chart, `http://www.billboard.com/charts/hot-100`, ????. (Visited on 12/08/2015).

[7] Music recognition (music) — gracenote, `http://www.gracenote.com/music/music-recognition/`, ????. (Visited on 12/08/2015).

[8] scikit-learn: machine learning in python scikit-learn 0.17 documentation, `http://scikit-learn.org/stable/`, ???? (Visited on 12/08/2015).