Hotline Ng:

Applying Machine Learning Techniques to the Billboard Hot 100

Cristian Cibils, Zach Meza, Greg Ramel

Motivation

The Billboard Hot 100 has been a reliable documentation source for song popularity rankings over the past sixty years. There is an undeniable appeal to 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 understanding the success of a pop song.

We use an array of different ML algorithms to learn parameters that determine a song's path (defined as the position it holds on the charts in a given time interval) through the Billboard 100. Our problem specification can then be determined 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.

Features

The central feature we used was the song's history - either its full or partial path through the Hot 100. In path prediction, we used the preceding n values from a window to predict the following week's position, then shifted the window.

We experimented with different values of window size *n*. We found that the optimal value of *n* was 5. We also experimented with adding other extraneous features such as song year and week of year, only to find that they did not impact the results much relative to the core path information.

$$\varepsilon_{Path}(n) = \sum_{i=1}^{2n} |\hat{y}_n^{(i)} - y^{(i)}|$$

Figure 4: Error Equation

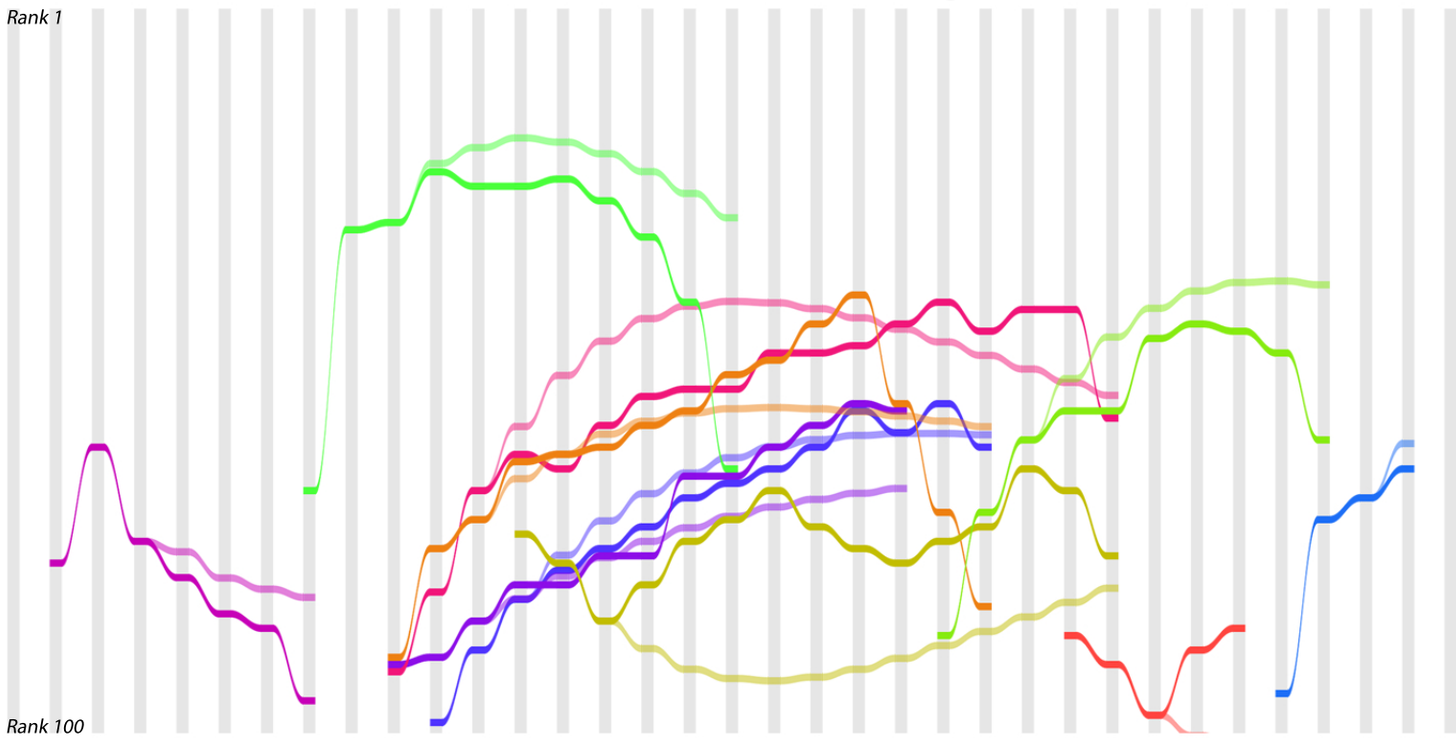

Figure 1: A selection of 2015 songs' performance week to week, both actual paths and predicted (lighter opacity). Window size = 3.

Results

After running several trials, we were able to determine the error and relative performance of several approaches. To analyze the performance of our path prediction algorithms, we took the average difference (absolute magnitude) between a predicted path and the true path. We wished to evaluate the performance of each of our potential algorithms - Figure 5 shows the Perceptron algorithm's average error compared to Ridge Regression and Logistic Regression. We disqualified the Perceptron due to erratic and overall high error, leaving us to compare Ridge and Logistic solely (Figure 6) and ultimately go forward with Ridge Regression. With Ridge Regression and a window size of 5, we were able to obtain a predicted generalization error (in k-fold validation) of ±4.52 spots.

In examining some of our predictions, we found that although our predicted paths tend to diverge and follow a smoother overall path in general, it tended to peak and descend in roughly the same vicinity as the true path. In particular, our predictions had trouble anticpating that songs would drop off so relatively more sharply after they peak.

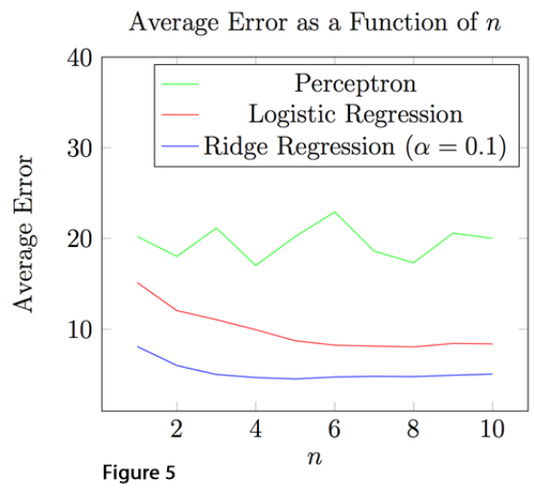

Figure 5

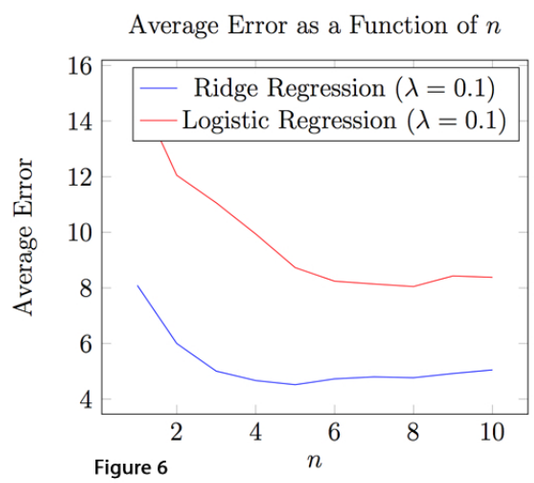

Figure 6

Dataset

With the Billboard website having recently made the entire Hot 100 chart archives available, we opted to scrape the data from the site. We have data for the full Hot 100 from every week, starting from mid-year 1958 and going through to this week (December 2015). In total, we had 2,984 weeks of analysis. Considering the presence of distinct songs across those weeks, we have a total of 24,469 song paths to analyze.

We cleaned our data into a more usable format -- dates became the number of weeks into the year in order to relay change over time, and we grouped a song's progress into discrete path entities . Further, we collected data from the Gracenote API to extract supplemental features (see below).

```
[ [2015, 45, 1], [2015, 46, 1]...]

{
    "track_title": "Hello",
    "album_artist_name": "Adele",
    "mood": {
        "1": {
            "TEXT": "Empowering",
            "ID": "42945"...
```

Figure 2: Excerpts of path data (top) and Gracenote supplemental data (bottom) for "Hello" by Adele

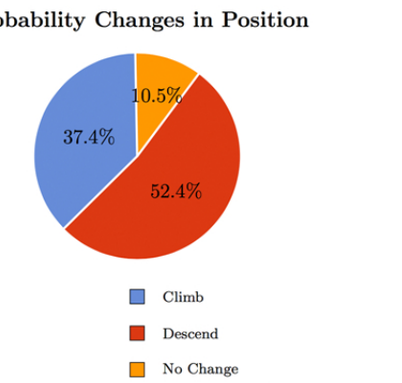

Figure 3: Probability that a song climbs, falls, or plateaus week to week

Methodology

We split our data approximately 90%-10% for a train set, randomly drawn from all the song entries. 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.