Learning the release year of a song from audio features

David Kravitz

Introduction

The digital music industry has been growing at an astonishing rate, and with it, the importance of music recommender systems. The goal of these systems is to select musical pieces that will likely be preferred by the user. One way to approach recommendations is to find songs that are similar. If we have information about the artist, producer, genre, etc, we can use these to help find similar songs. But what if we don't? Can we just use features extracted from the audio to make inferences about songs? In this project, I explore this idea by trying to predict the release year of a song from audio features.

Building the data set

Songs are mostly western, commercial tracks ranging from 1922 to 2011, with a peak in the year 2000s. The data is a subset of the Million Song Dataset (MSD), which can be found at http://labrosa.ee.columbia.edu/millionsong/.

All features come directly from the audio and do not include information about the song, e.g. musical artist or producer. The features were obtained using the following process: Each song is split into 12 segments. We take the timbre average of each segment and the timbre covariance between the segments, giving us 90 total features. In simple terms, timbre is what makes a particular musical sound different from another, even when they have the same pitch and loudness. For instance, it is the difference between a guitar and a piano playing the same note at the same loudness. Songs from any given period often have similar timbre, and thus, timbre is likely to be good predictor of the release year of a song.

There are 515,345 examples in the data set. The full data set is split as follows:
train: first 463,715 examples
test: last 51,630 examples

The split makes sure that no song from a given artist ends up in both the train and test set. The full training set is too large to run any classification problem in any reasonable amount of time, so training was completed on just 10% of the original training set.

One vs. all logistic regression

I ran one vs. all logistic regression using k-fold cross validation (with k = 3, 6, 9). Training on 50,000 examples yielded a mean training set accuracy of 8.44% and a mean test set accuracy of 7.40%.

On first glance, it looks like one vs. all logistic regression showed little success. However, note that the classifier is trying to predict the exact year a song was produced. If a song was produced in 1971, but we guessed 1972, that would marked as incorrect. But in this case, correctness is binary, so predicting 2010 is just as wrong as predicting 1972. This seems counterintuitive. It would be better to give the classifier a bit of leeway.

Next, I bucketed the years into decades, e.g. any year between 1970 and 1979 gets a 1970s output label. In this case, all vs. one logistic regression yielded a mean test set accuracy of 53.96%.

In any case, graphing the training and test set accuracy vs. the number of examples seemed to imply that there was a high bias problem. So, I mapped each feature to 2 polynomial features. This did slightly better. Training on 50,000 examples yielded a mean training set accuracy of 10.56% and a mean test set accuracy of 9.12%.

SVM with RBF kernel and PCA for dimensionality reduction

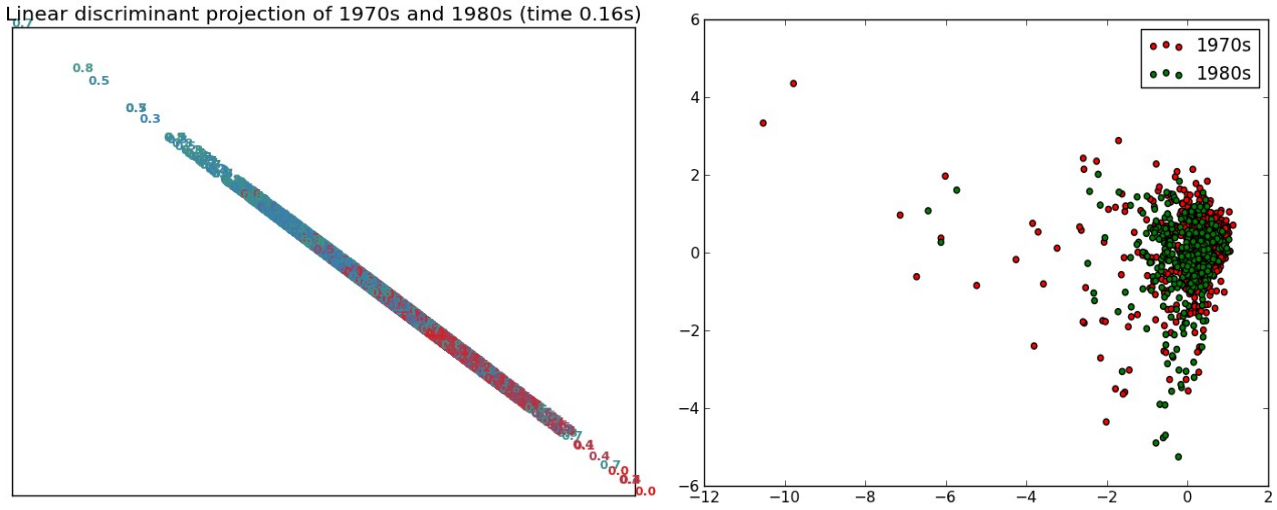
SVM with an RBF kernel was too computationally expensive given the size of our data set. So, I used PCA to reduce the data from 90 to 20 dimensions. The results were a bit better than logistic regression with polynomial features. Training on 50,00 examples yielded a mean training set accuracy of 13.44% and a test set accuracy of 12.21%.

Recursive Feature Elimination for Understanding and Dimensionality Reduction

Using an SVM with a linear kernel, I used recursive feature elimination to find the most predictive features. In general, the timbre average of each segment was more important than the timbre covariance between segments. In other words, the tonal qualities at any given time in a song are more predictive than the relationship between tonal qualities at different times. This makes sense. The instruments used in a song do not change throughout the song. However, instruments used changed a lot from 1920 through 2010 (synthesizers didn't exist in until 1960, for example). Thus, the instruments themselves are the best predictors of a song year.

2-dimensional PCA and LDA for data visualization

From the 2D plots it is clear that LDA (figure on left) outperforms PCA (figure on right) in terms of class discrimination. We know that LDA deals directly with the discrimination between classes while PCA does not take into account the underlying class structure. Thus, the discriminatory information is not aligned with the direction of maximum variance.



Linear regression

Even when classifying songs by decade, it was hard to tell where things were going wrong. Was the classifier usually off by one decade, two decades, four decades? I decided to treat this as a regression problem and see the strength of the linear association between the features and the output.

Using k-fold cross-validation, I ran ordinary least squares linear regression and got a coefficient of determination $R2 = 0.25$. The coefficient of determination is a measure of how well the regression line represents the data. One way to interpret it is that 25% of the total variation in y can be explained by the linear relationship between x and y (a relationship which we found using linear regression) and the other 75% of the total variation remains unexplained.

Anomalies

I modeled each decade using a Gaussian distribution in order to find anomalies in the data.
One of the most interesting things about the data is that there was a smaller percentage of anomalies in the early to mid 1900s than there was in the late 1900s and early 2000s. This can be explained by the fact that were simply fewer types of music to be played in earlier times. For example, rock and roll didn't come into existence until the mid 1950s. But, once it was invented, it was here to stay.

Where do the errors come from?

I had some success in predicting song year, but we still suffered from large error rates. There are a few explanations for this.

(1) Music changes gradually. It doesn't just suddenly change direction. The 1970s weren't so different from the 1960s and the 1940s weren't so different from the 1930s. In general, the classifier was very easily able to predict whether a song was from the 1930s or the 2000s. However, there were some times it got confused. These usually came from examples where instrumentation was similar, despite a large difference in the year the song was produced. For example, an indie tune by Bon Iver has similar tonal qualities as a Bossa Nova tune by Joao Gilberto -- both consist of only an acoustic guitar and a tenor voice.

(2) Some genres of music have been around for a while. Most notably, jazz and classical. Louis Armstrong playing the trumpet with a jazz combo does not sound much different than Wynton Marsalis playing the trumpet with a jazz combo.

(3) Cover songs. A number of the songs in the database are cover songs. A band covering a song usually sounds pretty close to the original.

Next steps

Features based solely on timbre are not enough to guarantee an accurate prediction (see error analysis above). In future research, we can probably improve accuracy by adding other audio features such as pitch and loudness. The Million Song Dataset contains this information. The challenge will be to keep the classification task computationally feasible, since we will likely be dealing with thousands of features (average pitch and loudness, plus covariances of pitch and loudness and timbre).