Machine Learning Engineer Nanodegree

Capstone Proposal

Samy Hajal
October 3rd, 2017

Proposal

Domain Background

Music has a big role in popular culture. It has been omnipresent in society and serves as a good representation of contemporary social interactions. It can also be used both as a depiction of the masses and as a driving force to social movements. On an individual level, listening to a song can partner with one's mood to drive oneself in a particular direction. It is shown that it can both regulate and induce different moods[1].
A major part of a song that is most directly related to by individuals is the lyrical theme of a song. Recent years have seen the rise of song recommendation engines by different music providers such as Pandora, Spotify and Soundcloud. These recommendation engines typically rely on a song's meta-information (artist, year released, genre) and the user behavior (time spent listening to a song, likes/dislikes/skips, users with similar taste)[2]. In order to connect with the listeners, a recommended piece needs to also be able to relate to ones mood. One way to fulfill that requirement is to address the lyrical theme of a song in order to provide a better understanding of its context and background.

Problem Statement

Given the importance of a song's lyrical theme as it relates to one's mood, we want to be able to predict a song's meta-information feature based on its lyrics. In this project, we will be focusing on approximating a song's release year in order to gain information and be able to better refine music recommendations. Therefore, we will be performing a regression on the dataset. After our model is trained, it will expect a song lyrics as input and will output a decimal number which represents our prediction of the release year of the input song.

Datasets and Inputs

We will be using the `50 Years of Pop Music Lyrics` CSV dataset based on Billboard Magazine's Top 100 most popular songs. The dataset has been kindly built by @walkerkq and made available here: https://github.com/walkerkq/musiclyrics. A full description and samples of the dataset can be found here: https://www.kaggle.com/rakannimer/billboard-lyrics/data. In short, it contains songs ranked between 1 and 100 for each year between 1965 and 2015 along with their lyrics. Therefore, it contains a total of 5,000 songs. It also contains information about the song name and the artist. However, we will only be using the lyrics as it is what we are trying to solve in this problem; that is, the relationship between the lyrics and the release year. We will make the assumption that the 100

most popular songs for a certain year are a good representation of that release year.
Here the is full list of columns:
* **Rank**: The Billboard Magazine rank for that song in the corresponding year. Billboard has been keeping up with modern technologies to stay relevant with current trends and music platforms[3]. The rank is an integer between 1 and 100. Unused. * **Song**: Song title. Unused. (Text)
* **Artist**: Artist name. Unused. (Text) * **Year**: Release year. Our target output. The year is an integer between 1 and 100. * **Lyrics**: The song lyrics (Text). Our raw input. * **Source**: Undocumented, unused.

Solution Statement

We will be predicting a year based on song's lyrics through regression. Therefore, we will not be using a categorical output, but a continuous one. Therefore, *the output year will be a decimal number in the range 1965 to 2015*.
If we would like to incorporate our model in an application that presents the result to an end user, we can consider rounding our decimal year to the nearest integer.

Benchmark Model

We are given some extra information about the dataset. Songs length have been increasing since the 1960's until today[4]. We could use this information to build a benchmark model that linearly predicts the release year based on the number of words in the song. We will be using scikit-learn's `LinearRegression` class that uses the Ordinary Least-Squares method. We will be feeding the number of words in the song lyrics as input and predicting the release year.

Evaluation Metrics

Since we are dealing with a regression that yields an output on a continuous range, we will be using the R^2 score as a performance metrics for our algorithm. The score will help us evaluate how accurate our algorithm is at predicting that a song belongs to a certain year. We will be using 80% of our data points picked at random for training, and the remaining 20% as a testing set. Of the training set, we will be using 20% as validation data on each training pass to prevent overfitting.

Project Design

Text Preprocessing

Each song lyrics is characterized by raw text data. In order to make some sense out of the text, we will be preprocessing the data in order to generate features for each data point that we can feed into the models we choose to train. To bring up the words in the text that are more defining to a certain song, there are a few approaches we can use.
When it comes to the benchmark model, we can simply count the number of words in the corpus, by splitting on empty spaces. This will generate one-feature data points that we can then perform linear regression on.
In order to make more sense of the text, we will be using the TF-IDF for each song. `scikit-learn` provides a `TfIdfVectorizer` that we can use for this purpose. For each data point, the vectorizer

will yield a vector with length equal to the number of distinct words in all the song lyrics of our dataset.

Dimension Reduction

Since our dataset is disproportionate when it comes to the ratio of number of samples (~5k) to the number of features after TF-IDF (~30k), the dimensionality might hinder our model. We will consider removing common words like `the` and `a` which do not have much significance when predicting a song's chronological context. In addition, we will also be using the Porter Stemming algorithm[5] to reduce the dimensionality of our data. This algorithm reduces a word to its english root word which helps reduce the number of distinct words while maintaining their meaning. Another technique we will be using to reduce dimensionality is PCA. We will performance PCA on the dataset and manually choose a smaller number of features that determine enough information about the release year.

Model Training

From there, we will be feeding the data into different models and see which one is best. Our first model will be a Random Forest regressor in order to see if certain words have a major symbolism on a song's historical context. We can also apply the random subspace method to reduce the number of dimensions in the data set.
Next, we will be using a neural network to try to maximize our accuracy. We will be experimenting with the number of hidden layers and their size in order to get the best performance possible. Cross-validating these hyper-parameters will help us choose the best model, while using a validation set will help us optimize the number of epochs to use. In addition, we will use the drop-out method to minimize overfitting. It is too early to determine which model will perform better. We might need to train different models and compare the results via the R^2 measure to determine which one is best.

References

[1] Psychology Today, January 19, 2016. Music's Power Explained. Retrieved from https://www.psychologytoday.com/blog/the-truisms-wellness/201601/music-s-power-explained
[2] SlideShare, January 25, 2013. Collaborative Filtering at Spotify. https://www.slideshare.net/erikbern/collaborative-filtering-at-spotify-16182818
[3] Wikipedia. Billboard Hot 100. https://en.wikipedia.org/wiki/Billboard_Hot_100#Hot_100_policy_changes
[4] Kaggle. April 30, 2016. Dataset Analysis. https://www.kaggle.com/stansilas/love-lasts-forever/code
[5] NLTK library. Stemmers. http://www.nltk.org/howto/stem.html