LARGE SCALE METRIC LEARNING FOR MUSIC SIMILARITY

Daniel {Rosenberg, Erenrich}

ABSTRACT

Automatic music classification is an important problem in music analysis. In this paper, we use data from the "Million Song Dataset" to construct and evaluate music similarity metrics and metric learning techniques. While others have done metric learning on music datasets before, we evaluate which standard techniques perform best in both accuracy and time on this much larger dataset. We find that dramatically increasing the dataset size leads to larger performance gains than one might expect.

1. INTRODUCTION

Machine music analysis is a growing field in both industry and academia. One of the major difficulties with this work is that it requires large labeled corpuses which are difficult to obtain. The relatively new "Million Song Dataset" provides the music analysis community an opportunity to scale research to very large datasets with little organizational difficulty [2]. Already many papers of capitalized on this data to to solve such problems as playlist generation [4], music classification and song cover-identification [1].

Music similarity algorithms are especially useful because they can be used as subcomponents of algorithms like cover-identification and playlist generation. Previous work on music analysis was often performed on datasets as small as just 5000 songs [5]. It is now important that we reevaluate these techniques to determine how these algorithms scale in terms of accuracy and speed. It is not immediately obvious that all of our music metric-learning techniques will be able to scale. Others have already shown algorithmic techniques to speed up basic metrics of music similarity [3].

We performed metric learning across the "Million Song Dataset" in order to determine song similarity. Previous work has mainly relied upon definitions of similarity as songs appearing in the same album or being performed by the same artist. Other similarity measures have had poor performance. We take the creating artist as our ground truth for similarity.

2. DATA REPRESENTATION

2.1 Audio representation

The "Million Song Dataset" provides features and metadata for a million songs as generated by the "Echo Nest's" music analysis software. This data includes such information as volume, length and beats per minute. In order to fit the entire dataset in memory and effectively work with it we projected down to a much smaller set of features. This set of features was strongly influenced by [5] though we chose to drop several features as they are not consistantly populated across the dataset. Less than one percent of the dataset appears to be invalid or improperly populated since some values are clearly incorrect. For example: songs which are listed as having zero beats were dropped from the analysis.

We also added several features derived from the pitches contained in the songs. We chose to include the covariance matrix of pitches to get an idea of how pitches are grouped together. For instance, this should capture if a particular chord appears often. We also included the co-occurence counts so we can have a way to detect combinations that appear strongly correlated just because they occur infrequently. By adding quartiles of the max segment loudnesses, we hope to capture the song's progression over time. For instance, if a certain genre of songs tends to have a louder section towards the end, and another ends with a quiet repetition of the piece's theme, this feature will capture that.

- Song length
- Mean segment length - Echo Nest divides songs into segments of length less than a second
- Segment length variance
- Mean segment loudness
- Segment loudness variance
- Third quartile of max segment loudness - This gives us an idea of the distrubtion of loudness across segments
- First quartile of max segment loundess
- Mean segment begin loudness
- Segment begin loudness variance

- Beat interval variance
- Tatum confidence
- Mean tatum length
- Tatums per beat
- Time signature
- Time signature confidence
- Song mode
- Song mode confidence
- Pitch covariance matrix
- Pitch cooccurence counts

By eliminating all features that vary with time along the song the dimensionality of the data is reduced 308 numbers. This means that the 500GB dataset becomes just 1GB in size and so is much easier to analyze. Exactly how much information has been lost in the transformation is unclear.

2.2 Data labels

We labeled our data with the song artist. Songs are considered to be similar if they share a common artist. We used this as our ground truth to judge our performance against.

3. ALGORITHMS

3.1 Whitening

It has been shown throughout the literature that first whitening the dataset in order to ensure that the covariance matrix of the data is the identity matrix improves performance. We use KNN using euclidean distance and cosine similarity in conjunction with and without whitening. Evaluation of the distance metrics will be done by determining how well a KNN classifier is able to find songs that we have prelabeled as similar.

Note that for whitening we add in a small constant to all eigenvalues in order to diminish the impact of the smaller terms when we later take the inverse. This means that whitening here is only an approximation.

Here we show our results on a compressed version of the dataset which only includes the top 500 most frequent artists and 25000 songs. A holdout test set of 5000 songs was used. This was done to decrease artist sparsity. A table of our accuracy is shown below.

KNN	Euclidean	Whitened	Cosine	Whitened
1	12.3%	19.3%	14.8%	20.3%
3	12.1%	19.0%	14.5%	20.1%
5	11.6%	18.2%	13.5%	19.4%

While these results are worse than what was presented in [5] which showed approximately 75% accuracy for whitening we note that we are running over nearly 4 times more artists. Indeed considering the extreme differences between our datasets our results will never be directly comparable. Indeed whitening performs terribly on the full unrestricted dataset. This is to be expected as it is an entirely unsupervised technique.

3.2 RCA Metric Learning

Now we need to create a more sophisticated notion of distance than either euclidean or cosine similarity. Whitening is an unsupervised process and so is not taking advantage of all the information available to us.

To get a more useful distance metric, we turned to the Mahalanobis metric. Under this metric, the distance between two vectors is: $d(x, y) = (x - y)'M(x - y)$, where M is a positive semidefinite matrix. An alternate definition uses $d(x, y) = ||A(x - y)||_2$, where $M = A'A$. For instance, euclidean distance can be represented by $M = I$ We chose to learn a matrix A that should improve our similarity metric using relevant component analysis.

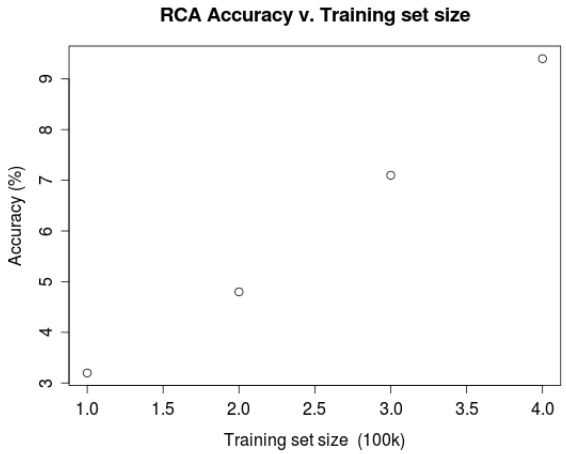
Under RCA, we divide our data into 'chucklets', and essentially perform whitening seperately on each chunk. In the end, it weights more relevant features heavier than those that don't have as much of an effect.

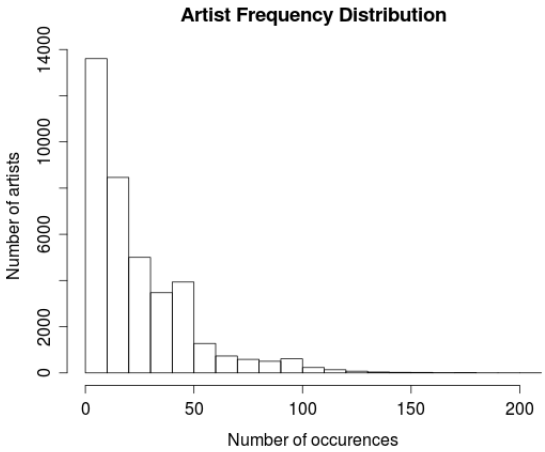
Our results over the full dataset are summarized in the following table.

Algorithm	Accuracy
Constant	0.02%
Euclidean	0.2%
Whitening	0.52%
RCA	15.7%

The constant algorithm which simply predicts the most common artist always is shown as a reference for comparison. RCA does more poorly here compared to the previous table because the number of artists has increased along with the dataset size itself.

The main focus of this paper is the importance of dataset size. Its effect on test-accuracy is shown in the following plot.





3.3 Analysis

The most striking result that we found was the large increase in performance with respect to dataset size. This was surprising because we would except a diminishing return with respect to the size of our dataset. Instead we found fairly robust gains as more data points were added. This suggests that an even larger dataset would provide even more impressive results. We though would sound a note of caution. Creating training and testing sets for this experiment is difficult. Randomly shuffling the dataset and extracting songs means that one might find that a particular artist lies only in the test set meaning that no algorithm could ever return the correct answer.

There is also a problem in the reverse. Large testing datasets result in more classes (e.g. artists) to choose from which should increase error rates. To deal with that we might want to fix the number of classes we are dealing with and increase the number of examples per class. That is not an option with this particular dataset as it follows a long tailed distribution and so large portions of the dataset would be discarded because they do not have sufficient numbers of examples.

The fact that KNN performance declines with the value of k is also interesting. What we would really should be analyzing is the fraction of the time that the correct artist appears in the top k. Again this is because of the sparsity of the artist data. Settings k to larger values means that we will never select artists which appear less frequently. Using a more dense indicator like genre should solve this problem as there are more exemplars per class.

4. CONCLUSIONS AND FUTURE WORK

We conclude that future music ML research must be done on datasets of this size in order to generate relevant results. The increase in performance is simply too dramatic to ignore.

Metric learning for music is effective and seems to be a general solution to the problem of finding similar music and creating reccomendations based on sample songs. A disadvantage of the methods used in this paper is that we base similarity on the artist that created the work. A more effective system would be to use data from a reccomendation engine to perform metric learning. Songs which are often liked together (high listerner cooccurence) would then be deemed similar. This way we do not detect songs that sound the same but songs that are liked together which in the end is the main use case for algorithms of this type.

5. REFERENCES

[1] T. Bertin-Mahieux and D.P.W. Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *Proceedings of IEEE WASPAA*, New Platz, NY, 2011. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).

[2] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[3] B. McFee and G. R. G. Lanckriet. Large-scale music similarity search with spatial trees. In *12th International Symposium for Music Information Retrieval (ISMIR2011))*, October 2011.

[4] B. McFee and G. R. G. Lanckriet. The natural language of playlists. In *12th International Symposium for Music Information Retrieval (ISMIR2011))*, October 2011.

[5] Malcolm Slaney, Kilian Weinberger, and William White. Learning a metric for music similarity.