What is the optimal solution? A case study of

Clustering songs using different solutions

Shervin Shahidizandi

Department of Computer Science

Sshahidizandi@albany.edu

Sarthak Gupta

Department of Computer Science

Sgupta2@cs.albany.edu

Ramya Sheetal

Department of Computer Science

Author3@cs.albany.edu

**Abstract**

Analysis of a given problem, use case and data for choosing optimal solution is the key skill no matter the environment of work (Business Company or Research lab) because of financial budget, deadlines and many other factors. In this paper, we try to examine all the possible optimal solutions to a real-life problem of Clustering of songs to label them based on their ‘moods’ and “categories”. We try to choose the best possible solution based on the performance of each studied one using the same dataset. We propose four different solutions to the given problem and evaluate the two optimal ones. The chosen clustering approach for this problem is the known popular k-means algorithm. One of specific solutions we study is Apache Spark. We try to examine how profitable Spark can be in different occasions.

**General Terms**

Algorithms, Measurement, Performance,

**Keywords**

Spark, Performance, Machine Learning, Clustering, K-means

# Introduction / Background

A great answer always would be built based on great understanding of the problem. This fact makes the role of a great analyst, no matter the discipline, crucial when a solution is to be proposed for a problematic case. In the field of computer science, in particular, choosing the most optimal and beneficial path for achieving the desired results is always a must and during the recent years there have been many works done to improve this optimizations and efficiency for the challenges being met.

One of the known cases of this happened back in the day when the notion of BigData[4] was a hot trending topic in the whole field of CS and brand-new systems and architectures were proposed and some, such as Mapreduce[2] were brought back and customized. This trend had large impacts on companies before anywhere else, although many research ideas were influenced to analyze the proposed systems but companies were somewhat forced to use. Then after a while using the challenges the current systems had (such as MapReduce being a bit slow by writing mostly everything on distributed file system instead of memory) and the rise of graph analysis along with machine learning, more advanced systems such as Apache Spark [1] were proposed and affect the business companies again before anywhere else.

But as these trends are having these days there might be cases for a business company that using the biggest most powerful system is not really the best solution. The problem the company needs to deal with may not be big enough to use fanciest BigData solutions.

The purpose of this paper is to evaluate a real-word Machine Learning problem and check if one of the most wanted fanciest systems in the field of Machine Learning, Spark, is really performing better than other possible optimal but less expensive solution solutions.

In the following chapters we first explain the machine learning problem to be solved the related works regarding the typical challenges of our problem would be discussed in chapter 3. The solutions applied to the problem is discussed in chapter 4. We then evaluate the each to be study solution using the dataset in chapter 5 and present results. Chapter 6 involves our Conclusion and Future work.

# Problem

## *Definition / Motivation*

Our problem involves categorizing and labeling a given set of songs based on their numerous features in order to achieve something very similar to Spotify [7] Application, which provides songs not only for different category of songs but also for different occasions, moods, situations and even holidays.

People listen to songs according to their moods and it would be an optimal case if we could have songs clustered together based on the moods. It would even be better if we could come up with the method by which we could have a fastest approach for clustering since there might be numerous songs and albums being added to our dataset every minute. Therefore our optimal solution should be able to function as close as possible to real-time.

## *Approach: K-means Clustering*

Category is one of the main aspect of a song, there are numerous possible categories and moods which songs can be labeled for hence in categorizing songs each feature of the song plays a main role. Furthermore, another fact is Songs in most cases cannot be a part of numerous categories at the same time. Therefore, we use clustering approaches as the main algorithm of our problem. The particular clustering approach used is k-means which is a known, easy to implement and relatively fast algorithm in clustering so it matches our case of problem and evaluation for multiple solutions very well.

K-means algorithm can be formally defined as follows:

1. The number of clusters is first initialized and accordingly the initial cluster centers are randomly selected.
2. A new partition is then generated by assigning each data to the cluster that has the closest centroid.
3. When all objects have been assigned, the positions of the K centroids are recalculated.
4. Steps 2 and 3 are repeated until the centroids no longer move any cluster.

## *Challenges (from original problem to final version)*

Our original topic of project was based on using Spark MlLib [5] library along with REST API for Spotify via a python library Spotipy [8], the Spotipy library had its own challenegs which we discuss in details in chapter 5, but regarding the solution we found out using python code directly is much faster than our Spark solution this was originated the first idea of the final topic of this project; we learned from experience that using fancy popular solutions may not be beneficial for all sort of problems.

Another challenge we had along the way was finding the useful and right number of features for every song. Our goal was to choose this right number in a way to avoid the typical bottleneck in clustering algorithms, curse of dimensionality, which treats all data points as similar having same distance, and at the same time use enough number of dimensions (features) to fully take advantage of k-means algorithm. There was a Spotify songs analysis paper [12] that inspired us to look for the little details of songs like loudness, tempo and etc. as discussed in the dataset section.

# Related Work

There are multiple works related to Spotify songs using different clustering approaches [3, 8] and also other approaches such as Deep-learning [13,9]. Additionally, there are some publications Introducing an analytical model for the distribution of the response time, a key metric in the Spotify service. [14] which predicts response times for the Spotify backend. And also some works in Understanding user behavior in Spotify [5] which deals with the fact that Spotify users have their favorite times of day, during which period they spend large fractions of their sessions.

# Methodology

All the solutions have been chosen to be implemented on python which is nowadays known as the programming language for machine learning.

The initial chosen solutions to be compared were as follows: Spark on one Node, Spark on multiple Nodes, Python code with the known optimal machine learning libraries, Dirty Python code using no machine learning library. As our first phase of project We were able to implement and compare two of the solutions, one Node Spark and Python code with each other. These two were chosen as our initial phase for multiple reasons. Firstly, using python machine learning libraries is a popular approach when dealing with general purpose problems and this was a good test for us to see how our defined problem is specific, It also should be said that these libraries have proven their efficiency and speed which might be our fastest case in coding with python. Secondly, the two solutions being compared to each other both are being executed on one machine, the source codes are simi and the dataset chosen is really not that big that Spark would need multiple nodes for running a k-means algorithm.

*4.1 Exploration of the data*

*The exploration of data involves finding the required set of data which would give us better reading.*

*4.2 Analysis of data*

*Analyzing the data and using appropriate features to get the best and required results.*

*4.3 Choosing and Setting up models*

*Choose the appropriate model for clustering like in our case we used k-means clustering.*

*4.4 Time Complexity*

*Compare the approaches by calculating the time complexity.*

# INSTALLATION`

If you already have Python on your system you can install the library simply by downloading the distribution, unpack it and install in the usual fashion:

python setup.py install

You can also install it using a popular package manager with “pip install spotipy”

For methods that did not require authorization, we simply create a Spotify object and start making method calls [7].

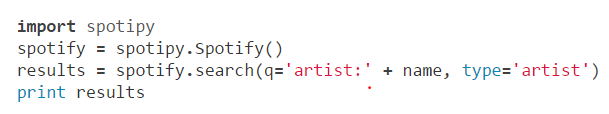


Fig 1. Installation example

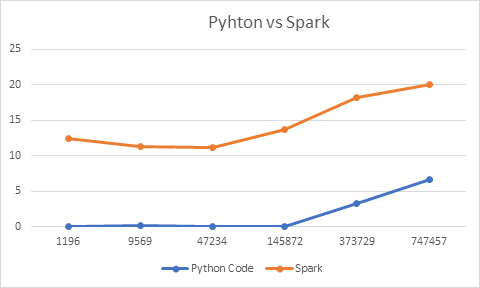
## *DRAWBACKS*

When we used this particular code: we could not get the needed track information that could help us sort the data into moods. We needed more dimensions to sort the data accurately, Information like artist, genre and title were not enough for our requirement so we had to change the dataset obtained with the above code.

The genre was the initial parameter for clustering but that was not that accurate when we performed the tests. The songs were categorized based on their genre as the possible types of genre is limited and we needed more than just one characteristic to sort the data. Hence, we selected new data set using the new code.

# FINDINGS

The Fig 2 shows that for the large set of data, python works better than spark as PyPy runs faster than spark when many cores are present.



# Conclusion

The research on dataset gave us very notable observations like we can calculate feel, loudness, acousticness of the song. This shows us the correlation between the various features in the songs and the data being very user specific gives us variations and allow us in using different models.

We concluded that performing clustering would be the best option for spotify dataset as it results in splitting the dataset in various variables. In our case, we clustered the data and tried running it on spark framework and then through python libraries. The process was done to compare the time complexity between the two and what we observed that the for the very large data it would be better to use spark and for the dataset which is not very large, it would be advisable to use python. We can conclude that Python is much faster than spark and as the dataset or core increases it will still have an advantage over spark.

# Future Work

As our research was done on single mode, we would like to extend it to using multiple nodes on the spark and check the scalability and performance. Also, the approach can be used on different dataset and that might give us more interesting observations.

### 5.1 Use different Models

Use different complex algorithm for clustering of data like Spectral Clustering, BFR, CURE, etc.

#### 5.2 Using Python from scratch

As in paper we have used many libraries of python and Spark, it would be better if we could write the code from scratch and use our own methods.

# ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# REFERENCES

1. Apache Spark. http://spark.apache.org.
2. Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." *Communications of the ACM* 51.1 (2008): 107-113.
3. King, J. and Imbrasaitė, V., 2015, March. Generating Music Playlists with Hierarchical Clustering and Q-Learning. In *European Conference on Information Retrieval* (pp. 315-326). Springer International Publishing.
4. Kreitz, G. and Niemela, F., 2010, August. Spotify--large scale, low latency, P2P music-on-demand streaming. In *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on* (pp. 1-10). IEEE.
5. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A.H., 2011. Big data: The next frontier for innovation, competition, and productivity.
6. MLlib http://spark.apache.org/mllib/
7. Osmalskyj, J., Piérard, S., Van Droogenbroeck, M. and Embrechts, J.J., 2013, May. Efficient database pruning for large-scale cover song recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* (pp. 714-718). IEEE.
8. Rajanna, A.R., Aryafar, K., Shokoufandeh, A. and Ptucha, R., 2015, December. Deep neural networks: A case study for music genre classification. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on* (pp. 655-660). IEEE.
9. Spotify <https://www.spotify.com/us/>
10. Spotipy https://spotipy.readthedocs.io/en/latest/
11. van den Hoven, J., ANALYZING SPOTIFY DATA.
12. Van den Oord, A., Dieleman, S. and Schrauwen, B., 2013. Deep content-based music recommendation. In *Advances in neural information processing systems* (pp. 2643-2651).
13. Yanggratoke, R., Kreitz, G., Goldmann, M. and Stadler, R., 2012, October. Predicting response times for the spotify backend. In *Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualiztion management (svm)* (pp. 117-125). IEEE.