The Project Proposal

Brian Magee and Prasanth Prahladan

November 9, 2015

1 Introduction

In the "music-genre classification" problem, we are required to determine the genre of a given query song, based on an algorithm that builds an internal representation of different music-genres, using a fixed database of classified(tagged) songs. The purpose of the project, is to enable the student to understand how to build a rudimentary functional prototype of a music-classification system. As a design challenge, the students are required to understand the high-level process of mining for patterns in a given music database,to split the process-flow into different sub-systems(algorithms) and determine how these functional-blocks may be combined to determine a process-flow for improving the efficiency of the classification problem. The specifications of the course-project as described in the project-guide, indicates the genres we need to focus on, and the number of songs assigned to each genre.

The music-genre classification process can be summarized by a five-stage pipeline.

1. Embedding of Dataset into Euclidean Space:
   It is important to note that the representation of data-files in memory, needs to be chosen based on the application for which that data shall be utilized. A given song when represented/stored as a .mp3 or .wav file, may be processed and played using an Audio player. However, for the purpose of classification, we need to design/develop a particular vector-representation that incorporates certain "descriptive features" of the audio file. These "features" may be based on analytical notions of time-frequency domain representation or based on "perceptive/cognitive" notions. The representation of these features, as mathematical vectors embedded in a high-dimensional space, then permits us to use algorithms for processing vectorial data, for the purpose of automated classification/clustering. We intend to use the "mel-frequency cepstral coefficients", its derivatives and possibly discrete-cosine-transform coefficients as the required features representative of the songs.

2. Visual Pattern Detection of Music Database:
   The preliminary step involves the use of a Data-visualization library, to determine whether there are any patterns like clusters in the data, which can then be exploited. It would be helpful to understand the logic/ideas behind the algorithms used to guide the visualization process, which could then be incorporated into the classification engine, that we need to develop.

3. Determine Intrinsic Dimension of the Data:
   If any patterns(clusters) have been detected in the data-visualization, it helps to as-

sume that the different genres of high-dimensional data in fact is distributed along low-dimensional manifolds embedded within the high-dimensional space. We can then use algorithms for determining the intrinsic dimension of these manifolds (e.g "Correlation Dimension"). By determining the intrinsic-dimension of each of the genres, we can determine the minimum dimension of the space into which all the data can be embedded without losing much information.

4. Metric Learning:
   Ideally, when data are categorized into categories, we would like to believe that the songs/data-points that are "similar" to each other are "closer" to each other, and those that are "dissimilar" are "farther" apart. The closeness of two data-points is determined by the "distance metric" used to compute the distance between the two points. The reason such a "metric" would exist, can be deduced from the fact that the data points are distributed upon a Manifold. The measure of distance between any two points, thus becomes the distance travelled along the manifold surface between the points, rather than the shortest euclidean straight-line distance between them. The "metric-learning" problem deals with the problem of ascertaining the metric for each of the manifolds separately, or the notion of a "global-metric" for all the datapoints

5. Dimension Reduction:
   Once the minimal intrinsic dimension of the datasets has been determined, and we have a notion of "distance-metric" that helps compute the distances between the points, we then deal with the problem of reducing the dimension of the data. We note that this procedure, helps us determine how much of the information from the initial choice of vector representation of the dataset is redundant. The solution to this problem, is the design of a Map/function that projects every data-point from the high-dimensional space to the low-dimensional space. To ensure that the representation for each data-point is unique in the reduced dimension, we impose the constraint of an injective mapping, on this function.

6. Clustering:
   If we have identified patterns in the data-visualization stage, we would assume that the songs belonging to the same music-genre are distributed as clusters in the high-dimensional space. We expect that the dimension-reduction process does not hamper/destroy the clustering of the dataset, but rather reinforce/amplifies the extent of clustering - "similar" songs are brought closer together and "dissimilar" songs are pushed farther apart. It is obvious that the "distance-metric" used to measure the distances between songs in the reduced-dimensional space is very-different from that in the high-dimensional space. However, we can demand that the distance metric in the reduced-space be closer to the Euclidean space. The "metric" in the reduced-space shall then be used to identify clusters in the dataset.

7. Query Processing (Cluster Assignment/Statistical Learning):
   This forms the final stage of the Genre-Detection problem. Once the abstract model - a spatial distribution/representation of the songs in the database has been determined, we have now developed the capacity to determine the genre/family that a particular song-query might belong to. We can develop a method to determine the genre-of song, by determining the cluster that it belongs to, the proximity to its neighbours and the classification of its neighbors into the different genre.

2 Data Visualization

The t-SNE algorithm [2] is an award-winning algorithm, for visualizing high-dimensional datasets. We shall first implement this algorithm, to determine a possible visual representation of the database, that shall indicate visual patterns in the data. This visualization shall help us develop an intuition of what the data distributions might look like.

This algorithm provides us the opportunity to check the efficiency of the vector-space embedding of the selected features from the songs in our database. If the visualization does not separate our vector-space data set into a sufficient number of discrete clusters, then it is likely that our choice of features is inadequate to truly separate the different genres when implementing our own dimension reduction techniques. In this case we will augment the vector-space embedding by adding additional features from the songs. This visualization algorithm can also be used after each stage of the pipeline to check that the information in our dataset has not been adversely deformed.

Further, the visual-information could be factored into the design of the supervised genre-classifying algorithm. A potential problem with this approach is that we do not use the category or label-information of the dataset, to detect clusters. Its possible that the clusters we observe, might be aggregations of data-points from different genres. However, we believe that it would help to have some form of visual representation of the data, to identify what can be learned from the data in an unsupervised manner.

3 Intrinsic Dimension

It is assumed that the datasset associated with the different genres, can be classified into distinctively separate nodes-spaces of the abstract high-dimensional euclidean space within which each data point exists. The algorithm presented in [4] shall be implemented for the given dataset. It needs to determined, if the formulation of intrinsic dimension of the dataset is affected by the choice of the "metric" describing the distance between the nodes. The computation of the intrinsic dimension of each of the genres, shall help us determine the minimal dimension of the reduced-dimensional space within which all the data-points can be suitably represented without any significant loss of information.

4 Distance in song-space

We intend to use the Information Theoretic Metric Learning toolbox [1], for determining the appropriate "metric" that helps classify and categorize the music genres. The method involves assuming that there exists a distribution with the high dimensional space as its support, and also another distribution atop the two-dimensional plane upon which it shall be projected. The algorithm is designed as an optimization problem, which helps to reduce the KL-divergence between the two distributions, where it is expected that the data-points that are similar to each other are closer to each other, while the one's that are dissimilar are located farther away.

5 Dimension reduction

We shall create and compare multiple pipelines for the dimensional reduction process. Both linear and non-linear graph based methods shall be explored. An interesting application we intend to explore, is the extension of the work on learning multimodal similarity [3] to determining similarity between genres. The process would thus adopt a soft-categorization procedure, rather than assuming the clusters to be shaped as hard-bounds.

Linear method using Fast Johnson-Lindenstrauss transform (FJLT) algorithm. This method uses random projections to map our data from the initial high d-dimensional space to a much lower k-dimension space by allowing low distortions of the data which is controlled with a tolerance parameter (epsilon). The choice of ϵ impacts the choice of $k-$dimensional space into which the data can be suitably embedded.

Non-linear, Graph-based Refined Embedding. This method builds a similarity graph, between all the datapoints, where the edge-weights are computed using a particular kernel function and distance-metric. From the similarity graph, we derive the Graph-Laplacian and follow the procedure of spectral embedding of this graph. The eigenvalues and the eigenvectors of the Graph Laplacian shall indicate the existence of the clusters.

6 Clustering

It is assumed that in the reduced dimension space, the euclidean metric shall serve as a sufficient metric to compare the different genres of songs. We intend to use the libraries available in Python Scikit-learn [5] to perform clustering of the reduced-space data.

In particular we intend to investigate the DBSCAN clustering algorithm as its features appear suited to our genre classification problem. In particular the notion of diagnosing the individual data points as "core" or "non-core" nodes belonging to the cluster can be used in conjunction with kNN for cluster membership detection by weighting neighbors differently for each case. Also the diagnosis of nodes as outliers not belonging to any cluster may also be helpful in "weeding out" data points for which comparisons are not particularly useful. Additionally, the fact that DBSCAN is agnostic to the shape characteristics of individual clusters is likely to be helpful.

7 Statistical learning

Every query song whose genre needs to be determined, shall undergo the same process of dimension reduction, used to determine the clusters in the songs. The probability of belonging to each of the different clusters is determined by a process of voting by the nearest neighbors, or neighbors-of-neighbors. We shall not be evaluating the performance of the cluster-membership-probability of the different voting schemes individually, but its contribution to the whole genre-classification pipeline.

8 Putting them together

We expect the different genres are comprehensive, in the sense that all songs that do not get classified into either of the specialized genres get assigned to the category "World". Thus, it is

possible that even white-noise could be classified as "World" music. Therefore, it is important to build a binary classifier that distinguishes between World and all the other families. We hope the method of Support Vector Machines can be used to discriminate between the "world" and "non-world" categories.

This is followed by a pipeline of algorithms that implement global dimension reduction, clustering algorithm, and statistical learning, to determine the probability of cluster membership of the query data.

9 Testing and Verifying the pipelines

The different pipelines are evaluated in the following manner. The entire music database is randomly split into two segments - the training(80%) and testing dataset(20%). The training data subset is used to identify the clusters of the different songs. After this, query-samples are picked from the test-dataset. The true-categorization of the test-data is known to us and is compared with the prediction of the genre-classifier. We intend to iterate the above procedure of randomly creating training-testing datasets and processing through the dimension reduction and classifier pipelines about 10 times, before we determine an average performance index of each pipeline. This error between the true-classification and the predicted cluster-membership is used to develop a confusion matrix. The "confusion-matrix" is used to quantify the performance of each of the genre-classifier pipelines, to determine which of the permutations of the cascade of pipelines, shall provide us with the best classifier.

10 Discussion

The dimensional-reduction stage of the algorithm pipeline, shall form the core component of the system. It is possible to generate myriad features from each song and thus embed the dataset into a very high dimensional space. However, the choice of the dimensional-reduction algorithm shall determine the extent of geometrical distortion, and loss of information, during the projection of points from the high-dimensional space to the reduced-dimensional space.

Each dimension reduction algorithm may have a limitation on its efficacy in reducing the dimension of the data by a particular order. We can determine this only by experimentation, for each of the alternative pipelines.

As previously mentioned, the FJLT method reduces the dimension in relation to the distortion allowance. In order to reduce to a workable dimension, excessive distortion allowance may be necessary which might lead to significant loss of information.

Though, we have described different alternatives for each stage of the data-processing pipeline, it is important to identify which permutation of the combination of algorithm sequences shall provide the best performance. These algorithms might be evaluated in isolation, and also in cascaded combinations with one another. However, it is difficult to predict which combinations shall work better and why. The better pipeline can be determined only through experimentation and a comparison of the confusion matrices.

References

[1] *Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra and Inderjit S. Dhillon.* Information Theoretic Metric Learning, ICML June 2007, 209-216, http://www.cs.utexas.edu/ pjain/itml/

[2] *L.J.P. van der Maaten and G.E. Hinton.* Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9(Nov):2579-2605, 2008, http://lvdmaaten.github.io/tsne/

[3] *McFee, B. and Lanckriet, G.R.G.*Learning multi-modal similarity, Journal of Machine Learning Research (JMLR) 2011.

[4] *M. Hein, J.-Y. Audibert.* Intrinsic dimensionality estimation of submanifolds in Euclidean space, In L. de Raedt and S. Wrobel, editors, Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), 289 - 296, ACM press, 2005, http://www.ml.uni-saarland.de/publications.htm

[5] Python Scikit-Learn Machine Learning library, http://scikit-learn.org/stable/modules/clustering.html#clustering