A Comparative Study of Dimensionality Reduction Methods: The Case of Music Similarity

Nicolas Wack[1], Pedro Cano[1], Bram de Jong[1] and Ricard Marxer[1]

Universitat Pompeu Fabra, c/ ocata 1, 08003 Barcelona, Spain

Summary. In this paper, we investigate the performance of three unsupervised classification algorithms applied to musical data. They are first evaluated on the direct set of feature vectors that have been extracted from the original songs, and we try to highlight whether this data seems to lie on an embedded manifold or not. Furthermore, we try to enhance the obtained results by applying preprocessing transformations to the data, with encouraging results.

1 Introduction

With the advent of internet file sharing, people are becoming owners of endlessly growing music collections, and therefore must find a way to organize them, to not get lost in this profusion of new music. Most current algorithms used to organize large music databases are based on collaborative filtering, and allow you to classify tracks by using human annotations, i.e. correspondence found by using statistical analysis of human listening habits.

Computers, and the networks feeding them are constantly becoming faster and we are now able to perform very heavy computations in the blink of an eye, allowing a more "mathematical" analysis of musical properties. This can however lead to a set of descriptors of a song that is huge, and for it to be usable, it needs to be stripped down to its smallest possible size, while still retaining all the information that was contained in the original analysis.

Reducing the dimensionality allows us to have bigger databases and faster searches, which is necessary for any recommendation system to be useful. Second, it helps us to avoid the curse of dimensionality: adding more dimension to a space heavily increases the complexity of algorithms dealing with the space, and increases the sparsity of the set. In essence reducing dimensionality will give us better and more trustworthy results.

The rest of this paper is organized as follows. In section 2, we describe the methodology and the database we use for our tests. Section 3 presents the algorithms that we compare as well as the results of the evaluation. In

section 4, we strive to enhance these results by applying various preprocessing transformations to the original dataset. Section 5 introduces the use of RCA and see how it compares to unsupervised learning. Finally, in section 6 we comment on our results and give directions for future research.

2 Methodology and database used

First of all, let us define what it is that we are going to measure. The goal is to find algorithms that take a dataset in a certain original space (of dimensionality N) and transform it into a dataset of lower dimensionality (K, K<N), while preserving the similarity of the points of the dataset in the target space.

To assess similarity, which is a qualitative criteria, we need to find a quantitative description of the distances between points that corresponds to our human perception of similarity. We will do this by using two different types of measurements: genre and artist classification.

The usage of genre and artist classification for evaluation of audio similarity algorithms has been proposed and discussed numerous times by various authors (-add reference-). While it is disputable that these classifications are the "perfect" benchmark for music similarity, we can still get very meaningful results from them.

The evaluation will be done using the following algorithm:

```
for each target_dimension from 1 to 80:
  for each transformation:
    reduce dataset dimension to target_dimension using
    transformation
    for each file in dataset:
      class = class of nearest song in dataset
    ratio = correct class estimations / total number of files
```

This will allow us to:

- see how each transformation fares against the other transformations
- see how the identification results evolve with respect to the target space dimensionality. In particular, we will see to how many dimensions we can scale our target space down without losing too much information or degrading the identification results.

The dataset we use consists of 4018 tracks, is classified into 25 different genres (rock, pop, classical, ...) and represents the works of 173 different artists. Each track has been analyzed and is represented by 374 features, which describe various aspects of this song, and which are of spectral, harmonical and rhythmical nature.

3 Unsupervised transformations comparison

In this section, we are going to compare 3 different unsupervised algorithms: Principal Component Analysis (PCA), Isomap and Locally Linear Embedding (LLE). The idea is to try to discover whether musical data lies on an embedded manifold, and as such could be parameterized by fewer parameters, or if the repartition of the points in the space is uniformly spread. This is important, because we would have reduced our musical space, not only to one of lesser dimensionality, but also, and more importantly so, to one which is parameterized by a number of variables that carry direct perceptual meaning. In a way, this would help us "understand" what criteria define music at its core. This is to be compared to the problem of face similarity[1], where pictures of faces are characterized by thousands of pixels, but they lie on a 3-dimensional manifold where the dimensions represent the orientation of the face and the position of the light source (which are the defining criteria of the face picture, much more than the luminosity of any given pixel number).

3.1 Principal Component Analysis

PCA[2] is a linear transformation that chooses a new coordinate system for the data set such that the greatest variance by any projection of the data set comes to lie on the first axis (called the first principal component), the second greatest variance on the second axis, and so on. PCA can be used for reducing dimensionality in a dataset while retaining those characteristics of the dataset that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. The low-order components often contain the "most important" aspects of the data. It has also been shown that PCA is the optimal linear transformation for selecting a subspace that has largest variance.

3.2 Isomap

Isomap[1] is an algorithm that generalizes classic multi-dimensional scaling (MDS) to non-linear manifolds, by replacing the metric distance used in MDS with a geodesic distance defined on the manifold in the following manner:

$$D(a,b) = \min \sum_{i=0}^{n-1} d(p_i, p_{i+1}) \tag{1}$$

where d represents the original distance measure, p_i and p_{i+1} are k-nearest neighbours and p_0 is a and p_n is b. k is a parameter set at the start of the algorithm. This distance corresponds to the shortest path that connects the points a and b on the neighbourhood graph.

3.3 Locally Linear Embedding

LLE[3] is an algorithm that tries to preserve the local geometry around each data point by cutting small linear "patches" and reconstructing them in the lower-dimensional target space. It does this by first reconstructing each point as a linear combination from its neighbours. These coefficients are computed by minimizing the reconstruction error, expressed as follows:

$$\varepsilon(W) = \sum_i \left| \mathbf{X_i} - \sum_j W_{ij} \mathbf{X_j} \right|^2 \tag{2}$$

The optimal weights are found by solving a least-squares problem. Once we have these weights, we solve the same problem, but with the weights fixed and the variables being the coordinates of the points in the target space. This is done by solving a sparse eigenvalue problem, the solution of which are the new points in our target space.

3.4 Qualitative comparison

When trying to discover if there is such an embedded manifold, we will thus compare PCA against both LLE and Isomap. PCA has been proven [**?**] to converge optimally towards an embedded manifold if the latter is linear, while LLE and Isomap both have the ability to also converge towards a non-linear manifold [4]. If LLE and/or Isomap manage to converge towards this manifold, it entails that they have managed to identify the criteria that carry perceptual meaning, and thus should yield better results for the classification tests. Also, a criterion that has not yet been discussed, but which is of prime importance if this is to be applied to real-world music recommendation systems, is the complexity of these algorithms. They have to be efficient enough to be applied to databases of thousands, even millions of songs to be useful. We thus define the following variables: our original space contains N points of dimension D, and we reduce it to a d-dimensional space. With these notations, PCA has a computational complexity of $O(D^2N) + O(D^3)$, Isomap has an average complexity of $O(N^3)$, and LLE has a worst case complexity of $O(DN^2) + O(DNK^3) + O(dN^2)$ (where K is the number of nearest neigbour used for the algorithm)[5]. In our specific case, $D = 374$, $1 < d < 80$ and $N >> D$, where we consider d and D to be constant, and the parameter that we want to consider with respect to complexity is N, the number of points in the database. The complexity C for each algorithm, in this case, amounts to the following: $C(PCA) = O(N)$, $C(Isomap) = O(N^3)$, $C(LLE) = O(N^2)$. Hence PCA has linear complexity, which is good, while LLE and Isomap have polynomial complexity and do not scale that well with respect to N.

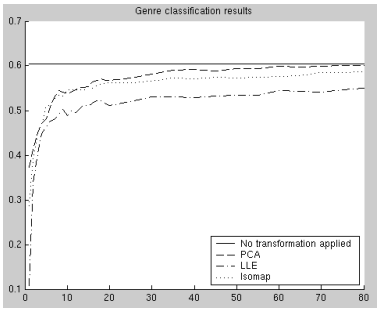

Fig. 1. Genre classification results

3.5 Results for genre classification

Figure 1 presents the results of the genre classification for these algorithms. Three things can be noticed:

1. There seems to be a target dimension ($\simeq$20-25) under which the results drop dramatically and above which the results hardly increase. This seems to indicate that all or most of the information that is relevant to genre classification is contained in the first 25 dimensions.
2. This limit is the same regardless of the algorithm used
3. PCA performs better than both Isomap and LLE, which seems to indicate that the musical descriptor space does not lie on a specific manifold, but rather is spread evenly thoughout the original space. This is also good to know, as PCA is the algorithm that scales better, and as such, seems to be the indicated choice for an algorithm to be used in music recommendation systems.

4 Boosting results using preprocessing

Seeing that there exist some algorithms that have been proven to converge if the input space distribution is gaussian, or that make the assumption that their input is guassian, we decided to try to compare 3 different types of preprocessing applied to the data and see whether this would have any effect on unsupervised classification. The preprocessing transformations were the following: normalization, applying the box-cox transformation and gaussianizing. An example of these transformations, applied to a given distribution can be seen on Figure 2.

4.1 Normalization

Normalization consists in linearly scaling each dimension between 0 and 1. It is also referred in the literature as min-max normalization. By default we always apply this transformation.

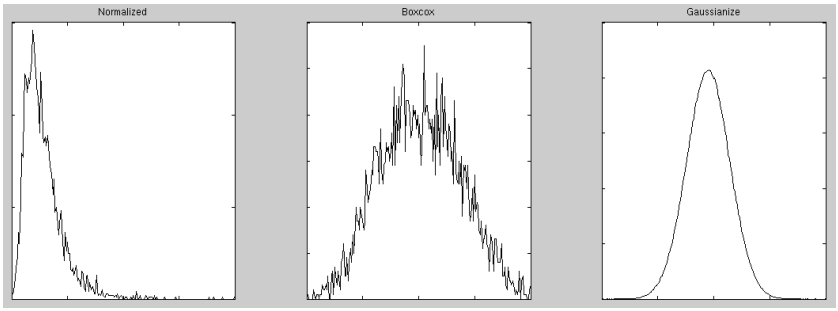

Fig. 2. Distribution example

4.2 Box-Cox Transformation

This step consists in applying the Box-Cox transformation[6] to each dimension separately. For an independent variable X and a Box-Cox parameter $\lambda \geq 0$ the transformation is defined as follows:

$$Y_\lambda = \tau(X;\lambda) = \begin{cases} (X^\lambda - 1)/\lambda \; if \; \lambda \geq 0, \\ \ln(X) \qquad\quad if \;\; \lambda = 0 \end{cases} \tag{3}$$

Y_λ will have a normal distribution if and only if $\lambda = 0$ or $1/\lambda$ is an even integer. The aim of the Box-Cox transformation is to find the value of λ which gives the best approximation to a normal distribution for the transformed data. We chose the λ_{opt} which maximizes the log-likelihood of the transformed data.

4.3 Gaussianize

Gaussianizing works in the following way: we order the distribution, and then assign to each point the value it would have if the distribution was perfectly gaussian (by applying the inverse error function $erfinv$). The method has one obvious drawback: it discards all the information about the original distribution, hence it is not revertible. After applying the gaussianizing algorithm, each dimension in the feature vector is a sample of a perfect gaussian distribution.

4.4 Results

Figure 3 represents the results of genre classification for the three aforementioned preprocessings applied before PCA, LLE and Isomap.

We can notice that even though gaussianize is a destructive transformation (in the sense that it loses information from the original distribution), it is the one that gives the best results, with as much as a 10% points boost in some cases. Box-cox also improves the results in all cases, leading us to think that gaussian variables are much more suited to the task of genre classification.

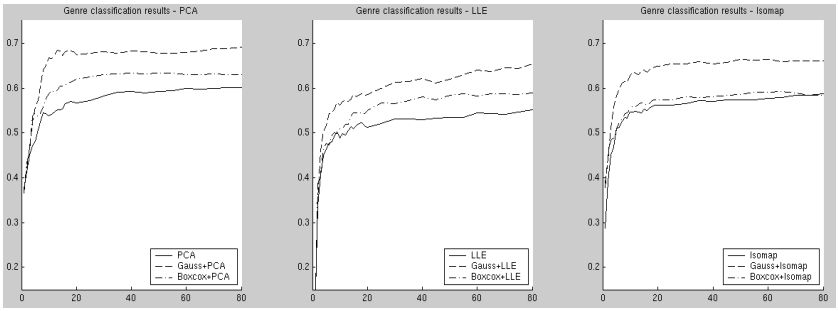

Fig. 3. Genre classification results for different preprocessings

5 Supervised test

5.1 Relevant Component Analysis (RCA)

Definition

Relevant Component Analysis (RCA) is a supervised transformation which aims at maximizing the global variance of a dataset while reducing the intra-class variance (representing unwanted variability). The algorithm is split in two parts: the first part is the dimensionality reduction that consists in applying a modified version of the Fisher Linear Discriminant (FLD) where we only use part of the classified vectors for training. This transformation amounts to resolving the following estimator:

$$\max_{A \in M_{P \times Q}} \frac{A^t S_t A}{A^t S_w A} \tag{4}$$

transforming from a space with P dimensions to a space with Q dimensions where A is the searched transformation matrix, $M_{P \times Q}$ is the space of all transformations, S_t is the total covariance matrix and S_w is the inner-class covariance matrix.

The second part consists in applying the actual RCA transformation, which scales down those dimensions that have great variability within our classes by whitening the resulting feature space. We first calculate the covariance for all the centered data-points in the chunklets:

$$\hat{C} = \frac{1}{p} \sum_{j=1}^{k} \sum_{i=1}^{n_j} (x_{ji} - \overline{x}_j)(x_{ji} - \overline{x}_j)^t \tag{5}$$

where p is the total number of points in the chunklets and $\overline{x}_j$ is the mean of the data-points of the chunklet j. Finally we obtain the whitening matrix:

$$W = \hat{C}^{-\frac{1}{2}} \tag{6}$$

so the new feature space is given by:

$$x_{new} = Wx \tag{7}$$

5.2 Why RCA and not Linear Discriminant Analysis (LDA)

One standard and acknowledged method of doing statistical supervised learning is to use Linear Discriminant Analysis (LDA), which also uses the FLD. However this method has not been retained here, for two reasons:

1. LDA requires you to know the class of each and every single instance in your training set. This is unrealistic when dealing with databases of millions of songs. RCA uses the points with known classes to reduce the intra-class variance and uses the entire set of points to estimate the global variance.
2. RCA only uses positive constraints, i.e. it only uses the fact that two points pertain to the same class to try to reduce their distance. It does not assume that if two points do not pertain to the same class, they are different. This applies well to similarity: omitting to say that two songs are similar does not necessarily imply that they are dissimilar.

5.3 Results

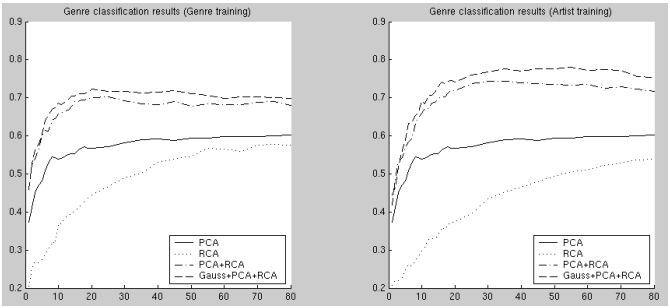

Fig. 4. Genre classification results with genre training and artist training

Figure 4 represents the results of the genre classification task. For one instance we trained the RCA algorithm with the artist classes, for the other we used the genre classes. We can observe that the RCA instance trained with artist classes performs better than the other one by nearly 10%. We attribute this to the fact that "artist" can be seen as a subdivision of the "genre"-space, thus supplying more information to the RCA algorithm.

6 Conclusion

There are some conclusions that we can draw from this analysis. First, we can see that be it unsupervised or supervised, each of the algorithms we tried was enhanced by the preprocessing of the values. This was originally done to get closer to certain optimality conditions for the algorithms, however the huge benefit we reaped from doing this (up to 10% better classification rates) hints that there may be a discrepancy between the scale of each descriptor as it is given and a perceptual scale that would better account for similarity. Applying a pre-processing transformation, such as gaussianize, helped us smooth this discrepancy to get better results.

Second, when using an algorithm such as RCA, we also realized that the more precise the information we had, the better (obviously), but that the quantity of information, if not directly quantifiable, could be estimated by looking for the inflexion point of the result curve. The results topped at dimension $\simeq$20 with the genre training set, but needed to go up to dimension $\simeq$30 to top.

References

1. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.
2. I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
3. S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.
4. Mira Bernstein, Vin De Silva, John C. Langford, and Joshua B. Tenenbaum. Graph approximations to geodesics on embedded manifolds, December 21 2000.
5. Olga Kayo. *Locally linear embedding algorithm Extensions and applications*. PhD Thesis, University of Oulu, Oulu, April 2006.
6. G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of Royal Statistical Society*, 26(B):211–246, 1964.