DIMENSIONALITY REDUCTION AND CLUSTERING TECHNIQUES APPLIED TO SPOTIFY DATA.

STA5069Z ASSIGNMENT

Julian Albert, Fabio Fehr, James Nevin
Department of Statistical Sciences
University of Cape Town

October 7, 2019

ABSTRACT

This research paper will seek to reduce a large dataset (of Spotify music data) into a low-dimensional dataset after which cluster analysis will be performed to gain insights on musical characteristics and the popularity of music. Linear and non-linear reduction techniques shall be contrasted using Principal Component Analysis (PCA) and t-distributed Stochastic Neighbour Embedding (t-SNE) respectively. Post reduction, two types of non-hierarchical clustering algorithms will be compared, namely — K-means and Fuzzy C-Means (FCM) — using internal and external cluster validation techniques. Clusters will be analysed in an attempt to determine song popularity from the Spotify dataset. It was found that fuzzy methods were better when the data is tightly bunched as it allows for more information around point membership. The t-SNE reduction technique managed to capture the non-linear behaviour around song popularity more effectively than PCA.

Keywords Dimensionality Reduction · PCA · t-SNE · Clustering · K-Means · Fuzzy

1 Introduction

Modern multivariate techniques have become the new standard for analysing large datasets. Descriptive methodologies focus on analysing data for the purpose of identifying underlying structures within data. We assume that there exists an underlying pattern in the data, but there is no *apriori* knowledge about these patterns or relationships. Such methods include reduction and cluster analysis. Due to the acceleration of modern data volumes methods have been optimised to scale them back down. Reducing datasets is now common practice as we aim to minimise computational cost with minimal loss of information.

The poster child for dimensionality reduction is linear PCA — often the "linear" prefix is forgotten and so the method is abused — which will be contrasted to the non-linear reduction technique t-SNE. We hypothesise that t-SNE will outperform PCA due to real-world data often containing a higher degree of non-linearity. With a reduced dataset which maintains the key information we can begin to cluster observations. The goal here is to identify observations and groups, ensuring that observations in the same group are sufficiently similar and observations in different groups are sufficiently dissimilar in an attempt to determine song popularity.

This report follows with a literature review in Section 2, subsequently Section 3 will describe the dataset and perform exploratory data analysis (EDA). We then define the methodology used and apply the multivariate techniques to the Spotify dataset in Section 4. Here we contrast the Fuzzy C-Means algorithm against the K-means clustering technique for different dimension reduction methods. Lastly, in Section 6, we conclude on the methods used whilst addressing shortcomings and possible improvements to the applied methodology.

2 Literature Review

In modern times high-dimensional data is ever present and, as such, statistical methods have evolved to compensate for the curse of dimensionality — the phenomena that arise when analyzing data in high-dimensional spaces that do not occur in lower dimensions. The expression was first established by [Bellman, 1957] when exploring dynamic programming.

To combat this we consider projecting the data onto a lower-dimensional subspace using the characteristics of the original variables, without loss of important information. [Izenman, 2008]. One way of accomplishing this is by creating a reduced set of linear or nonlinear transformations of the input variables. An early example of linear methods is principal component analysis (PCA). [Fisher and Mackenzie, 1923] endorses this technique as a more suitable analysis of variance for the modelling of response data. [Hotelling, 1933] further developed the method as a technique for deriving a reduced set of orthogonal linear projections of a single collection of correlated variables [Wold et al., 1987].

Nonlinear dimensionality reduction techniques tend to be more computationally demanding than PCA. We explore the t-distributed Stochastic Neighbour Embedding (t-SNE) technique — a non-linear dimension reduction technique first introduced by [Maaten and Hinton, 2008]. This is a refinement of the Stochastic Neighbour Embedding (SNE) technique from [Roweis et al., 2002].

[Maaten and Hinton, 2008] identify two flaws with the standard SNE technique. Firstly, the cost function is difficult to optimise. This is as a result of a lack of symmetry in the terms that make up the function. A second issue is what the authors term 'crowding' — accurately modelling data points that lie close together in high-dimensions will lead to data points that lie moderately apart in high dimensions lying far apart in lower dimensions. Due to this, points tend to crowd together in the center of the map.

The adjustment to the cost function addresses these issues. The t-SNE technique displays structure more accurately at various scales. An example given by [Maaten and Hinton, 2008] are images of objects from multiple angles seen from multiple viewpoints. In terms of the improved optimisation, the authors additionally detail the use of random walks to improve the performance of the technique on large datasets (in excess of 10 000 data points). They go on to show how this technique improves the visualisation on various datasets in comparison to techniques like PCA, Sammon mapping, and Isomap.

[Maaten and Hinton, 2008] primarily focus on t-SNE dimension-reduction as a means of improving visualisation i.e. they reduce to two or three dimensions. The majority of the literature follows this approach. As a result, there is room to investigate the performance of t-SNE when the focus is less on visualisation, and more on improving tractability of datasets. After reducing the dataset to a low-dimensional approximation we can consider cluster analysis — a methodology that aims to discover groups in data [Everitt et al., 2011]. The groups are not known *apriori* making the methodology an unsupervised classification technique. Good clustering algorithms result in high intra-class (within group) and low inter-class (between group) similarity.

A baseline method for clustering is the K-means algorithm described by [MacQueen et al., 1967] in which we pre-specify the number of clusters we want the data to be grouped into and define the centroid as the mean of a group of points. K-means is a hard clustering algorithm which can bring about challenges when the data has ambiguity. [Xu and Wunsch, 2005] highlights some challenges with the K-means algorithm: K-means cannot guarantee convergence to a global optimum and is sensitive to outliers and noise. [Xu and Wunsch, 2005] further points out that if the real clusters are in other non-spherical clusters, k-means may no longer be effective. The techniques simplicity is a large-contributor to its status as a clustering "benchmark".

In contrast, there are soft clustering methods known as fuzzy algorithms. These techniques have been around for many years as they allow for ambiguity in the data. [Kaufman and Rousseeuw, 2009] explains that the algorithms provide more detail on the structure of the data than hard clustering. Fuzzy algorithms report membership coefficients which give a description of the group memberships with a level of certainty. The drawbacks of this technique are it's complexity, as it has many outputs to calculate making it computationally expensive. [Dunn, 1973] developed some of the earliest work in fuzzy algorithms. They were originally developed to detect the presence or absence of "compact and well separated" natural groupings relative to a given metric on the data. This sparked an infinite family of fuzzy clustering algorithms based on a least-squared errors criterion. Shortly after, [Gustafson and Kessel, 1979] worked on an algorithm using fuzzy means and fuzzy covariances which closely resembled maximum likelihood estimation of mixture densities.

[Bezdek et al., 1984] later developed a Fuzzy C-Means (FCM) implementation in FORTRAN-IV which aggregated the subsets by using a generalized least-squares objective function. This improved the algorithm substantially and also included a choice of three norms (Euclidean, Diagonal, or Mahalonobis), an adjustable weight for controlling noise sensitivity, variable number of clusters, and several measures of cluster validity. [Kaufman and Rousseeuw, 2009] did notable work writing the program for fuzzy analysis (FANNY) which implements FCM on an IBM-PC environment. In more modern literature [Huang et al., 2011] and [Zhao et al., 2013] adapted the algorithm by using kernels to converge more quickly or consider non-linear relationships in the data.

Many Kaggle competitions have required the use of multivariate dimension reduction and clustering techniques on multimedia data. [Hoven, 2015] performed an exploratory cluster analysis using user-specific Spotify data and concluded in their study that cluster analysis could be used to determine popularity or create a "top 10".

3 Data

3.1 Data Description

The dataset used is *SpotifyFeatures.csv* obtained from Kaggle. The dataset contains all the measures that the Spotify API provides, with the dataset totals $N = 232,725$ song tracks and $p = 18$ columns. Therefore we can denote our data matrix $\mathbf{X}_{N\times p}$. The developer Spotify API website offers in-depth description of the variables. The data consists of a mixture of character (e.g. track_name), categorical (e.g. genre) and numerical (e.g. tempo) variables.

3.2 Exploratory Data Analysis

An important component of dimensionality reduction is correlation amongst variables. As this report focuses on comparing technique performance rather than delivering particularly robust results we tailor the dataset to retain only those variables whose correlation is highest. Table 1 shows the correlation matrix of the variables used. Importantly, none of the correlations are extremely high — as is the case for real-world data.

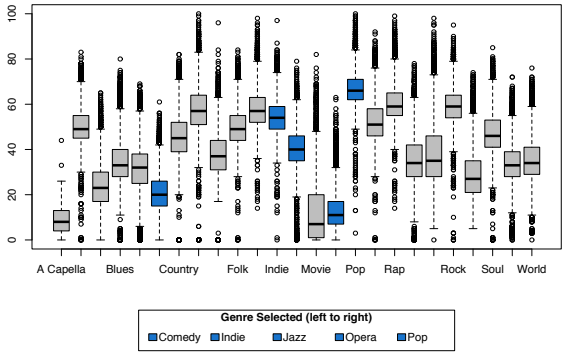


Figure 1: Boxplot of Popularity by Genre.

As we are trying to perform cluster analysis to identify popularity, we can examine the distribution of popularity by genre as in Figure 1. The dataset contains 27 genres, and we chose to reduce this to 5 genres that span the popularity space. The resulting choices are displayed by the blue boxes. We can see initially that Opera and Pop genres are vastly different in popularity and we expect that this relationship will be separable when applying the clustering techniques. For simplicity, we use a random sample of size 1000 from our original dataset for the remainder of the report. From Table 2, we can see that our sample proportions are representative of the full dataset. From the sampled set we also create an ordinal grouping of popularity which categorises the popularity into "Popular", "Mediocre" and "Unpopular".

	Comedy	Indie	Jazz	Opera	Pop
Full	9681	9543	9441	8280	9386
Sampled	216	211	207	173	193

Table 2: Frequency Table of Retained Genres.

4 Methodology

4.1 Dimensionality Reduction

When dealing with high-dimensional data in a multivariate setting we can compress the data to a low-dimensional subspace that captures most of the variability. This is done to improve computational efficiency and often inference, as well as aiding in visualisation.

	Acoustic	Danceability	Energy	Instrumental	Loudness	Speechiness	Valence
Acoustic	1.00	-0.46	-0.54	0.05	-0.64	0.32	-0.38
Danceability	-0.46	1.00	0.40	-0.15	0.48	0.11	0.58
Energy	-0.54	0.40	1.00	-0.23	0.75	0.34	0.37
Instrumental	0.05	-0.15	-0.23	1.00	-0.21	-0.25	-0.11
Loudness	-0.64	0.48	0.75	-0.21	1.00	-0.07	0.41
Speechiness	0.32	0.11	0.34	-0.25	-0.07	1.00	-0.01
Valence	-0.38	0.58	0.37	-0.11	0.41	-0.01	1.00

Table 1: Correlation Matrix of Retained Variable

4.1.1 Principal Component Analysis (PCA)

PCA is a technique for deriving a reduced set of orthogonal linear projections of a single collection of correlated variables $\{X_1, X_2, \ldots, X_p\}$ where the projections are ordered by decreasing variances [Izenman, 2008]. The method considers linear transformations $\mathbf{Z_j}$ for $j = 1, 2, \ldots, t$ representing the first t principal components of the form

$$Z_j = \vec{\phi}'\mathbf{X} = \phi_{j1}X_1 + \phi_{j2}X_2 + \cdots + \phi_{jp}X_p$$

and tries to minimise the "information" loss due to the transformation given by

$$\sum_{j=1}^{p} \text{var}\left(X_j\right) = \text{tr}\left(\mathbf{\Sigma}_{XX}\right) \tag{1}$$

From this we can take the singluar value decomposition of the covariance matrix to obtain eigenvectors (proxy for weights in linear combination) and the eigenvalues (proxy for information). We can take these linear combinations of the measurements and reduce the dimensions necessary for visual analysis while retaining most of the information (variation explained) present in the data. The method can be derived using either a least-squares optimality criterion, or as a variance-maximizing technique [Izenman, 2008].

Because the criterion for a good projection in PCA is a high variance for that projection, we should only retain those principal components with large variances. To determine just how many components to keep, we can calculate variation explained σ^2_{exp} using Equation 2

$$\sigma^2_{exp} = \frac{\lambda_j}{\sum_{j=1}^{t} \lambda_j} \tag{2}$$

where λ_j are the eigenvalues of the covariance matrix. We can illustrate σ^2_{exp} using a *scree plot* as in Figure 2. The left plot shows the proportion of variance explained by each principal component. It shows how the earlier components explain a higher proportion of the variance, compared to the later components. The first component explains almost half the variance, and the 5^{th}, 6^{th}, and 7^{th} components each explain less than 10%. The plot on the right shows the cumulative variance explained as principal components are added. With 2 principal components, roughly 65% of the variance is explained, while with 3 components, roughly 80% is explained. It is desirable to use 3 or less components, as this allows us to plot the data in terms of their

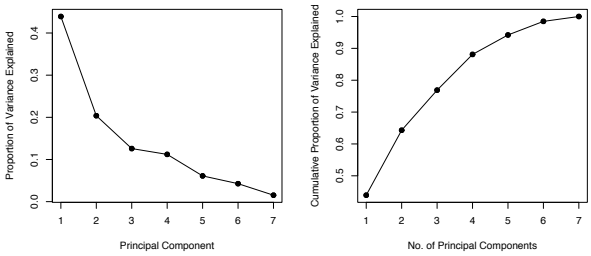


Figure 2: Variation Explained by Principal Components.

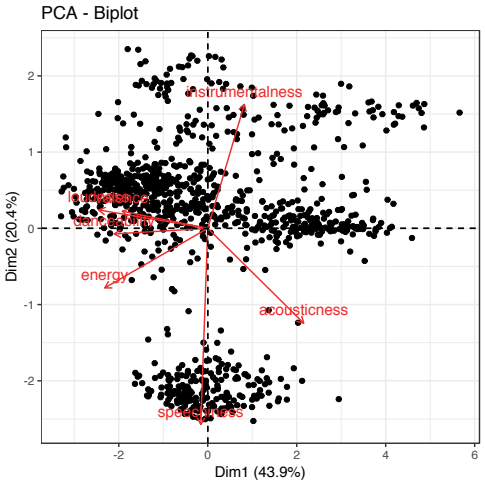
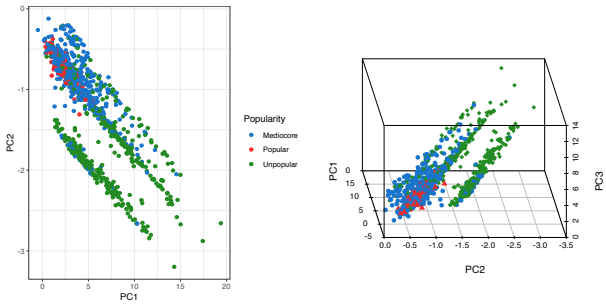


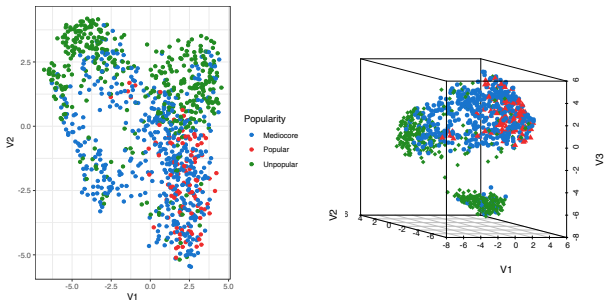
Figure 3: Biplot of selected variables.

loadings. The biplot in Figure 3 shows that these variables, although correlated, load very differently in the first two principle components. Principle component 1 along the x-axis negative loadings are being influenced by the overall energy of the music (loudness, energy, danceability, valence etc). The second component is primarily being influenced by the musical variables (speechiness & instrumentalness) which is intuitive to be orthogonal. Figure 4 shows plots of the first 2 (left) and 3 (right) principal components. The data have been coloured based on their popularity, but this information was not used in deriving the principal components. Using 2 principal components results in a plot that does not clearly split the data, with a strong overlap between mediocre and popular music.



(a) 2 PCs (b) 3 PCs

Figure 4: First 2 and 3 Principal Component Reduction.



(a) 2D (b) 3D

Figure 5: 2 and 3-Dimensional t-SNE Reduction.

There seems to be a collection of unpopular music that has generally lower loadings on the principal components, but this is still overlapping with some mediocre music. Using 3 principal components fairs better, with a more distinct grouping becoming visible. Popular music has been separated nicely from mediocre music, making distinguishing between the three popularities easier. This improvement is likely a result of the third principal component explaining a significant portion of the variance, as per Figure 2.

4.1.2 t-SNE

The methodology employed by PCA results in a linear dimension-reduction. While this can prove effective in many cases, highly non-linear data require more sophisticated techniques. One such example of this is t-SNE.

[Maaten and Hinton, 2008] explain the details of Stochastic Neighbour Embedding (SNE) and the extension introduced by t-distributed SNE (t-SNE). In the standard SNE framework, conditional probabilities that represent similarities are given by

$$p_{j|i} = \frac{\exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\left\|x_i - x_k\right\|^2 / 2\sigma_i^2\right)}. \tag{3}$$

This represents the probability that observation x_i would pick x_j as its neighbour. This is based off of a Gaussian distribution centered at observation x_i. Note that the closer points are together, the more likely that they are chosen as neighbours. These x points are in the high-dimensional space. We aim to reduce these to points y in a lower dimension, while staying close to the probabilities given by Equation 3. In other words, we want the values $p_{j|i}$ to be close to

$$q_{j|i} = \frac{\exp\left(-\left\|y_i - y_j\right\|^2\right)}{\sum_{k \neq i} \exp\left(-\left\|y_i - y_k\right\|^2\right)}. \tag{4}$$

The similarity between these values is measured by the cost function

$$C = \sum_i KL\left(P_i \| Q_i\right) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}, \tag{5}$$

which we aim to minimise. The two issues with this cost function are its difficulty in optimisation, and the crowding problem (detailed in Section 2). [Maaten and Hinton, 2008] suggest a possible solution to this: changing the cost function to

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{6}$$

where conditional probabilities have been changed to joint probabilities, and

$$q_{ij} = \frac{\exp\left(-\left\|y_i - y_j\right\|^2\right)}{\sum_{k \neq l} \exp\left(-\left\|y_k - y_l\right\|^2\right)}, \tag{7}$$

which differs from Equation 4 in the denominator (making it symmetric in i and j), and

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}. \tag{8}$$

This form for p_{ij} is symmetric, and ensures that each datapoint makes a significant contribution to the cost function. This procedure is symmetric SNE, which [Maaten and Hinton, 2008] note performs at least as well as asymmetric SNE. A final adjustment to deal with the crowding problem is changing to a Student t-distribution with 1 degree of freedom in the low-dimension map (this is a heavier tailed distribution). This means Equation 7 is changed to

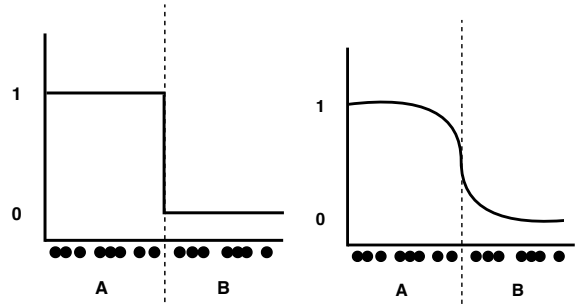
$$q_{ij} = \frac{\left(1 + \left\|y_i - y_j\right\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \left\|y_k - y_l\right\|^2\right)^{-1}}. \tag{9}$$

This cost function given by Equation 6 has a simple closed-form gradient, and so is straightforward to solve using a gradient descent algorithm. This technique can be improved through various adjustments, which are detailed in [Maaten and Hinton, 2008]. This technique can be implemented practically in R using the Rtsne package and function. Figure 5 shows the results of applying t-SNE to the Spotify dataset, where again points have been coloured based on their popularity, but popularity has not been used to derive the lower-dimensional mappings. As was the case

with PCA, mapping to two dimensions struggles to differentiate between popular and mediocre music. Unpopular music tends to separate well, but there is a strong overlap between the other two popularities. Using three dimensions improves things significantly, giving fairly distinct clumps of popularities. The higher performance of t-SNE over PCA suggests there exist non-linearities in the data, that are not captured by PCA. Also worth noting, since popularity was not used in the derivation of the mappings, we can't expect the data to separate perfectly based on this — the measure has been used more as a proof-of-concept.

4.2 Cluster Analysis

Our project will explore a hard (represented by k-means) and soft (represented by fuzzy) clustering technique. To illustrate the difference between hard and soft clustering we consider a dataset that can be traditionally grouped into two clusters. The resulting clusters are labelled 'A' and 'B', as seen in Figure (6). With hard clustering each point is assigned 1 or 0, whilst in soft clustering the points can range from any value between 0 and 1. In this way points that lie at the inflection point are assigned to both clusters.



(a) Hard Clustering (b) Soft Clustering

Figure 6: Hard and Soft Clustering Assignment.

A key problem here is choosing the optimal number of clusters *apriori*. To do this we can use methods such as minimising the total within-clusters sum of squares objective given by Equation 10

$$\min \left(\sum_{k=1}^{K} \text{ESS}_K \right) \tag{10}$$

We can calculate the measure for different numbers of clusters to generate Figure 7 and choose a cluster configuration that corresponds to an "elbow" in the plot — in this case 3 clusters. The plot is noticeably the same for fuzzy C-Means and as such only one scree plot is displayed.

4.2.1 K-means clustering

The algorithm starts by generating random centroid(s) and assigns each data point to the closest centroid. Each collection of points assigned to the same centroid is now a cluster. The centroid of each cluster is then updated based

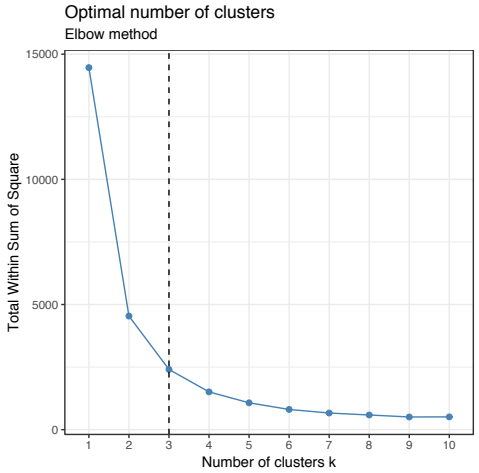


Figure 7: WSS Method for Determining Optimal Number of Cluster.

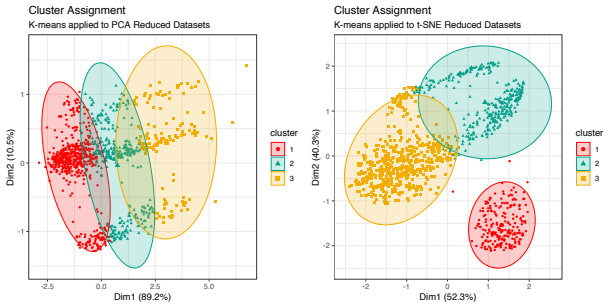
on the points assigned to the cluster. We repeat the assignment and update steps until no point changes clusters, or equivalently, until the centroids remain the same (subject to some threshold δ) [Izenman, 2008].

The solution (a configuration of items into K clusters) will typically not be unique; the algorithm will only find a local minimum of ESS. We define ESS to be the error sum of squares given by Equation 11

$$\text{ESS}_K = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left(\mathbf{x}_i - \overline{\mathbf{x}}_k\right)^{\tau} \left(\mathbf{x}_i - \overline{\mathbf{x}}_k\right) \tag{11}$$

where $\bar{x}_k$ is the $k-th$ cluster centroid and C_k is the cluster containing x_i [Izenman, 2008]. In an attempt to mitigate the problem of local minima, we run multiple K-means at different initial random assignments of the items to K clusters. We can then find the lowest minimum of ESS to determine an optimal clustering solution.

Figure 8 shows the results of applying K-means clustering to the Spotify dataset after reducing it to two dimensions using PCA (left) and t-SNE (right). The technique does not perform well on the PCA-reduced set. There is significant overlap between the three clusters — as a result, it would be impossible to classify a new observation with a first component between 0 and 2.5 and a second component between -1 and 1. With the K-means algorithm, overlaps indicate poor clustering performance, due to the aforementioned issue. The clustering on the t-SNE-reduced set is noticeably more separable, with a much smaller overlap and clear distinction between clusters. Due to the suspected non-linearity of the data, one would expect that the t-SNE clustering would be better. However, there is still some overlap, and, as explained previously, this is an issue with K-means clustering and interpreting the result. This leads into the need for a more sophisticated clustering technique.



(a) PCA (b) t-SNE

Figure 8: K-Means Clustering Assignment for Alternate Dimensionality Reduction Techniques.

4.2.2 Fuzzy clustering

[Struyf et al., 1997] contrasts the idea of "crisp" clustering techniques which assigns objects to exactly one cluster, such as K-means, against fuzzy algorithms. Fuzzy algorithms blur this assignment by spreading the objects membership over multiple clusters. For each object *i* and each cluster *v* there will be a *membership* u_{iv} which indicates the strength of membership. This is analogous to a probability. Memberships are subject to the following conditions:

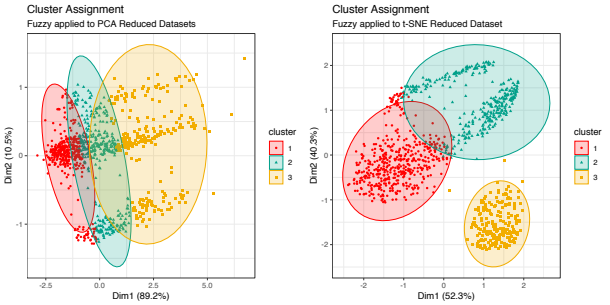
$$\begin{aligned} &u_{iv} \geq 0 \,\forall\, i = 1, \ldots, n \,\forall\, v = 1, \ldots, k \\ &\textstyle\sum_{v=1}^{k} u_{iv} = 1 \,\forall\, i = 1, \ldots, n \end{aligned} \tag{12}$$

The fuzzy clustering algorithm returns a membership score of each observation which takes on a value between zero and one. The memberships u_{iv} are defined through the minimisation of the objective function in Equation 13.

$$\sum_{v=1}^{k} \frac{\sum_{i,j=1}^{n} u_{iv}^2 u_{jv}^2 d(i,j)}{2\sum_{j=1}^{n} u_{jv}^2} \tag{13}$$

The $d(i,j)$ is a measure of dissimilarity, which are known, while the u_{iv} are unknown. A numerical optimiser combining Lagrange multipliers is used to find the optimal membership scores whilst maintaining the constraints defined by Equations 12 above. [Struyf et al., 1997]

Figure 9 is a recreation of Figure 8 above, with fuzzy clustering applied instead of K-means clustering. The points are the same, with slightly different clusters. The primary difference between the clusters created here versus with K-means is that there are heavier overlaps. This is because overlaps are allowed within the fuzzy framework, since points can belong to multiple clusters with some probability. As a result, one is able to quantify these overlaps. Despite this, it is still preferable to have less overlapping — this is why one can conclude that the t-SNE dimension reduction has been more successful. In the PCA fuzzy clustering, there is almost full overlap of cluster 2 with clusters 1 and 3, creating uncertainty about any points that lie in this region. In the t-SNE fuzzy clusters, there is only minor overlap between clusters 1 and 2, with cluster 3 lying completely separate. Again, this cluster is very similar to the one done by K-means, but the overlap poses far less of a concern in this case, leading us to conclude that fuzzy clustering is better for this dataset.



(a) PCA (b) t-SNE

Figure 9: Fuzzy Clustering Assignment for Alternate Dimensionality Reduction Techniques.

5 Cluster Validation

We can validate our methods using internal and external validation techniques. Internal validity (silhouette width, Dunn and Sum of Squares) utilises internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information. There is also external validity which consists of comparing the results of a cluster analysis to an externally known result, such as externally provided class labels (our popularity scores).

In silhouette width validation we compute $s_i\,(C_k)$ for cluster K given by Equation 14

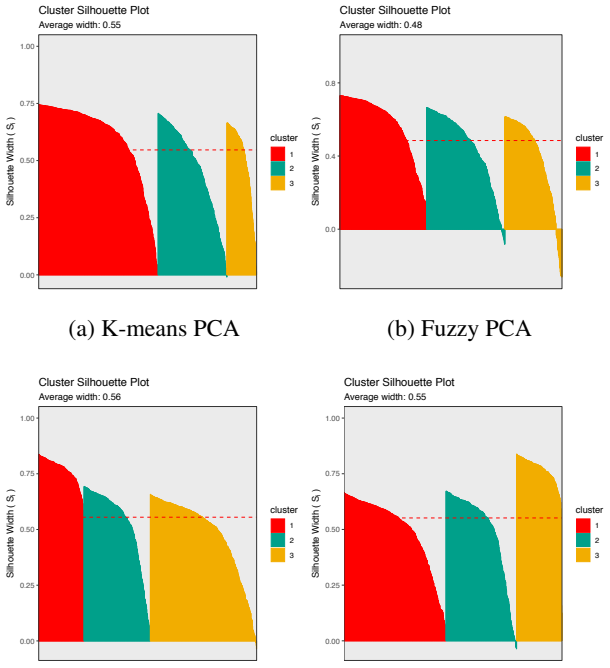
$$s_i\,(C_k) = s_{ik} = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \tag{14}$$

where a_i is the average dissimilarity of the ith item to all other members of the same cluster $c(i)$. Also, let $c(-i)$ be some cluster other than $c(i)$. We define $d(i,c)$ to be the average dissimilarity of the ith item to all members of $c(-i)$. Compute $d(i,c)$ for all clusters $c(-i)$ and let

$$b_i = \min_{c(-i)} d(i,c) \tag{15}$$

If $b_i = d(i,C)$, then, cluster C is called the neighbor of data point i and is regarded as the second-best cluster for the ith item. Thus b_i can be thought of as the average distance between i and the observations in the “nearest neighboring cluster".

We can plot the silhouette widths in Figure 10. Values near one mean that the observation is well placed in its cluster whilst values near zero indicate that an observation might really belong in an alternate cluster. Within each cluster, the value for this measure is displayed from smallest to largest. If the silhouette plot shows values close to one



(a) K-means PCA (b) Fuzzy PCA

(c) K-means t-SNE (d) Fuzzy t-SNE

Figure 10: Silhouette Plots for K-means and Fuzzy Clustering on PCA and t-SNE Reduced datasets.

for each observation, the fit was good; if there are many observations closer to zero or negative, it indicates a poor fit. From Figure 10 we can see that for K-Means PCA first cluster (red) is far larger than the other clusters, whereas using a Fuzzy clustering the groups are more balanced. Using t-SNE results in good clustering of one variable with similar S_i for the other two clusters, between K-means and Fuzzy there is not distinguishable difference. It appears that key difference is between dimensionality reduction techniques as they are impacting the quality of our final clusters.

Another internal validation scheme is the Dunn index, the ratio of the smallest distance between observations not in the same cluster to the largest intra-cluster distance. It is computed as per Equation 16.

$$D(C) = \frac{\min\limits_{C_k, C_l \in \mathcal{C}, C_k \neq C_l} \left(\min\limits_{i \in C_k, j \in C_l} d_{ij} \right)}{\max\limits_{C_m \in \mathcal{C}} diam(C_m)}, \qquad (16)$$

where $diam(C_m)$ is the maximum distance between observations in cluster C_m. The Dunn index has a value between zero and 1, and should be maximized [Dunn, 1974]. Table 3 shows the relative Dunn indices for the methodologies, where we are more interested in comparing the clustering techniques than the reduction techniques.

We can see in this instance that the Fuzzy C-Means has ever so slightly higher values of Dunn Indices when compared to K-means. This suggests that the fuzzy clustering has marginally outperformed the K-Means clustering — a result in-line with what we have observe from the plots of the clusters in Figures 8 and 9. Referring to Table 4 we see the Sums of Squares (SS) recorded metrics from our cluster analysis. These metrics measure the variability and spread of points within clusters and how far clusters are placed from one another. Firstly notice is that the Total-SS for t-SNE reduced clustering is substantially higher than the PCA reduction techniques. This is counter intuitive as visually you would expect t-SNE to have a lower SS as it appears to segregate the data more clearly in Figures 8 and 9. The reason for the vast difference between the reduction techniques is due to the K-means clusters overlap and tightly bunched points. Using [Kaufman and Rousseeuw, 2009] definition of between and within SS we extract the following insights.

	PCA		t-SNE	
	K-Means	Fuzzy	K-Means	Fuzzy
Dunn	0.006	0.008	0.031	0.032

Table 3: Dunn Index for Comparing Cluster Performance

The bunched points, due to PCA, would reduce the within-SS as it is the sum of the squared deviations from each observation and the cluster centroid. The overlap caused by the bunching would reduce the between-SS as this measures the squared average distance between all centroids. Having a low within-SS and between-SS means that points are densely bunched and clusters are likely to overlap which is not indicative of an effective final set of clusters. Overall this shows how SS is a counter intuitive measure for clustering as it measures the variability of points and not a fit measure as seen in a modeling context. The final clustering validation considered is external validity, shown in Table 5. What we are hoping to see is that each cluster represents a popularity category. If we consider PCA we notice that in both K-means and Fuzzy the first cluster seems to heavily represent the mediocre group, contains all popular observations and the least unpopular observations of the three clusters. The second cluster in PCA for K-means and Fuzzy appears to be balanced between unpopular and mediocre. Finally the last cluster reports a majority of unpopular observations. In Table 5, under the t-SNE dimensionality reduction, we see a similar pattern as all mediocre and popular observations are lumped together in a cluster for both K-means and fuzzy. Notice that when

	PCA		t-SNE	
	K-Means	Fuzzy	K-Means	Fuzzy
Within-SS	2258	2269	6349	6350
Between-SS	12131	12564	20211	20925
Total-SS	14390	14833	26560	27275

Table 4: Sum of Squares (SS) for Comparing Cluster Performance

		K-Means			Fuzzy		
		1	2	3	1	2	3
PCA	Med	335	136	17	251	187	50
	Pop	99	5	0	85	18	1
	Unpop	114	175	119	54	148	206
t-SNE	Med	26	114	348	332	130	26
	Pop	0	5	99	99	5	0
	Unpop	182	185	41	37	189	182

Table 5: External Validity of Correct/Incorrect Cluster Assignment

using t-SNE, we can see that this cluster is more exclusive and fewer unpopular observations are within this cluster. Again there appears to be a mixed cluster with a balanced representation of mediocre and unpopular, and finally a cluster that is dominated by unpopular tracks. The clusters appear to be more clearly defined in the t-SNE dimension reduction for both K-means and Fuzzy. This would suggest that using this data we would more confidently be able to cluster an unknown observation as unpopular or mediocre/popular as opposed to see if the unknown track would be popular.

6 Conclusions

This report covers baseline methods of dimension-reduction and clustering, namely PCA and K-Means clustering. Additionally, two more sophisticated techniques in the form of t-SNE and fuzzy clustering are considered. A point of interest was to see how these advanced methodologies handle the data (in comparison to the more simple techniques). Often, the simple methods rely on $iid.$ and correlation assumptions which tend to break down for real-world applications where the data is not fully specified for the particular task. We showed how t-SNE can capture possible non-linearities in the Spotify data, and explained the necessity of utilising fuzzy clustering to understand the ambiguity that arises when using K-Means and clusters overlap. We used silhouette plots, SS measures and the Dunn index as internal methods of cluster validation and point variability. Popularity (an unused metric in the dimension-reduction and clustering) was used as an external validity test.

We conclude that the clustering techniques are marginally different with fuzzy being better when the data is tightly bunched as it allows for more information around point membership. With regards to dimension reduction, t-SNE resulted in a better spread of data. Although more variable according to SS, it separated the data more clearly with respect to popularity. The t-SNE method seemed to capture more of the non-linear behaviour shown in the data and make more defined clusters as seen in the external validity test.

A possible extension (and practical application) of this analysis would be predictions of the cluster variables (either genre or popularity) using the retained predictors in lower-dimensions. This would be useful to artists and music companies, as one may be able to predict a song's popularity and hence make decisions about marketing and product positioning. As we saw it would be easier to predict the unpopularity using this dataset rather than an unknown track's popularity. An important final note is that clusters are representative of some underlying structure in the data, but the methods only allows us to hypothesize what those relationships are. Although we observe three distinct groups, these could represent genre similarity rather than popularity as these two variables are by construct serially correlated.

References

[Bellman, 1957] Bellman, R. (1957). Dynamic programming, princeton, nj: Princeton univ. *versity Press. BellmanDynamic Programming1957*.

[Bezdek et al., 1984] Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203.

[Dunn, 1973] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

[Dunn, 1974] Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104.

[Everitt et al., 2011] Everitt, B. S., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. John Wiley & Sons, 5 edition.

[Fisher and Mackenzie, 1923] Fisher, R. A. and Mackenzie, W. A. (1923). Studies in crop variation. ii. the manurial response of different potato varieties. *The Journal of Agricultural Science*, 13(3):311–320.

[Gustafson and Kessel, 1979] Gustafson, D. E. and Kessel, W. C. (1979). Fuzzy clustering with a fuzzy covariance matrix. In *1978 IEEE conference on decision and control including the 17th symposium on adaptive processes*, pages 761–766. IEEE.

[Hotelling, 1933] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

[Hoven, 2015] Hoven, J. (2015). Analyzing spotify data.

[Huang et al., 2011] Huang, H.-C., Chuang, Y.-Y., and Chen, C.-S. (2011). Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 20(1):120–134.

[Izenman, 2008] Izenman, A. J. (2008). Modern multivariate statistical techniques: Regression, classification, and manifold learning.

[Kaufman and Rousseeuw, 2009] Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

[Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

[MacQueen et al., 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[Roweis et al., 2002] Roweis, S. T., Saul, L. K., and Hinton, G. E. (2002). Global coordination of local linear models. In *Advances in neural information processing systems*, pages 889–896.

[Struyf et al., 1997] Struyf, A., Hubert, M., Rousseeuw, P., et al. (1997). Clustering in an object-oriented environment. *Journal of Statistical Software*, 1(4):1–30.

[Wold et al., 1987] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

[Xu and Wunsch, 2005] Xu, R. and Wunsch, D. C. (2005). Survey of clustering algorithms.

[Zhao et al., 2013] Zhao, F., Jiao, L., and Liu, H. (2013). Kernel generalized fuzzy c-means clustering with spatial information for image segmentation. *Digital Signal Processing*, 23(1):184–199.