A Comparative Study of Dimensionality Reduction Techniques for Labelled Datasets

EDOUARD CHAPPON ELYES KHALFALLAH

LYON 2 *LYON 2*

This study presents a comparative analysis of unsupervised dimensionality reduction techniques (MDS, Isomap, LLE, t-SNE) for labeled data across 4 3D synthetic datasets and a real-world Spotify tracks music dataset. Our investigation encompasses three main axes: 1. The presentation of each methods, 2. The presentation of the datasets. Each synthetic dataset is designed to highlight the strengths and limitations of one specific method, 3. A number of neighbors calibration for LLE and Isomap, and 4. a comprehensive comparison framework using multiple evaluation criteria and 5. Experimental results analyse.
This work provides insights into the relative advantages of different dimensionality reduction approaches and offers guidance for method selection for labeled data.
All the code and experiments are available on github : A Comparative Study of Dimensionality Reduction Techniques for Labelled Datasets.

1 Introduction

Dimension reduction aims to transform high-dimensional data into a more compact, low-dimensional representation while preserving essential structures and relationships. This process simplifies data analysis, enhances the performance of machine learning algorithms by reducing complexity, and facilitate data visualization.
This paper presents a comparative analysis of the strengths and weaknesses of four dimensionality reduction techniques (t-SNE, LLE, Isomap, and MDS) in the context of labeled data. Our study unfolds as follows: In section 2, we introduce each method and discuss its underlying principles. In section 3 we present four 3D synthetic datasets designed to reveal the strengths and limitations of each technique, followed by the introduction of a more complex real-world dataset to illustrate these methods in practice. Then in the section 4, we develop a protocol to choose the hyperparameter for Isomap and LLE. A composite metric is defined in section 5 to compare the reduction methods, the metric take in account local structure and the global one, leveraging on the labels of the data. We apply in section 6, each method to the synthetic datasets to reduce their dimensionality from 3D to 2D, analyzing the outcomes through various metrics. Following this, we extend our analysis by applying the chosen techniques to the real dataset, examining the projection results and extracting meaningful insights. The paper concludes in section 7 by summarizing our findings, discussing the comparative advantages of each method, and suggesting guidelines for selecting appropriate dimensionality reduction techniques based on dataset characteristics and analysis objectives.

2 Materials and Methods

In this section, we introduce the dimensionality reduction techniques used in this study, along with their mathematical principles. To ensure consistency, we begin by fixing the notations used throughout the article. We use, for the experimentations, the implementations given by the library Scikit-learn.

2.1 Notations

Let $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\} \subset \mathbb{R}^d$ represent a high-dimensional dataset with n data points, each having d features. The goal of dimensionality reduction is to find a corresponding low-dimensional representation $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\} \subset \mathbb{R}^l$, where $l \ll d$, while preserving specific relationships (e.g., distances or neighborhood structures).
- d_{ij}: Distance between points $\mathbf{y}_i$ and $\mathbf{y}_j$ in the high-dimensional space.
- d'_{ij}: Distance between points $\mathbf{z}_i$ and $\mathbf{z}_j$ in the low-dimensional space.
- $W = \{w_{ij}\}$: Reconstruction weights used in Locally Linear Embedding (LLE).
- σ_i: Bandwidth parameter for local scaling in t-SNE.
- P: High-dimensional similarity matrix.
- Q: Low-dimensional similarity matrix.
- k: Number of neighbors.

These notations will remain consistent across all subsections.

2.2 Torgerson Multidimensional Scaling (MDS)

MDS [2] aims to preserve the pairwise distances d_{ij} between data points in a low-dimensional space by minimizing the stress function:

$$\text{Stress}(Z) = \sqrt{\frac{\sum_{i<j} \left(d_{ij} - d'_{ij}\right)^2}{\sum_{i<j} d_{ij}^2}} \tag{1}$$

where $d'_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|$.
To compute the low-dimensional embedding $\mathbf{Z}$ from the pairwise distance matrix $D = [d_{ij}^2]$, the matrix is first converted into a centered Gram matrix $B = -\frac{1}{2}HDH$, where $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the centering matrix. The eigen decomposition $B = Q\Lambda Q^T$ is then performed, and $\mathbf{Z}$ is constructed as:

$$\mathbf{Z} = Q_l \Lambda_l^{1/2},$$

where Q_l and Λ_l are the top l eigenvectors and eigenvalues, respectively. This method provides an optimal embedding that preserves global pairwise distances. In our study, the distance metric is Euclidean; in this setting, the results would be equivalent to those obtained with PCA. Moreover, when $d \ll n$ like in this study, PCA has a lower computational complexity than MDS, which would have made it a more efficient choice if we had opted for it.

2.3 Isomap

Isomap [5] extends MDS by using geodesic distances d_{ij} instead of Euclidean distances. It constructs a neighborhood graph where

each data point is connected to its k-nearest neighbors. The geodesic distance d_{ij} is computed as the shortest path in the graph using algorithms like Dijkstra's. Classical MDS is then applied to the geodesic distance matrix D, allowing Isomap to preserve non-linear global structures. In this way, it acts as a hybrid method that integrates local information with a global optimization.

2.4 Locally Linear Embedding (LLE)

LLE is a local unlinear method [4] that reconstructs each data point $\mathbf{y}_i$ as a linear combination of its k-nearest neighbors by minimizing the reconstruction error:

$$\varepsilon(W) = \sum_i \left\| \mathbf{y}_i - \sum_j w_{ij} \mathbf{y}_j \right\|^2,$$

subject to $\sum_j w_{ij} = 1$. The low-dimensional embedding $\mathbf{Z}$ is found by minimizing:

$$\Phi(Z) = \sum_i \left\| \mathbf{z}_i - \sum_j w_{ij} \mathbf{z}_j \right\|^2.$$

This method focuses on preserving local geometric structures within neighborhoods.

2.5 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE [6] maps high-dimensional data into a low-dimensional space by minimizing the Kullback-Leibler (KL) divergence between probability distributions. For data points $\mathbf{y}_i$ and $\mathbf{y}_j$, the high-dimensional similarity is:

$$p_{j|i} = \frac{\exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_k\|^2}{2\sigma_i^2}\right)}.$$

In the low-dimensional space, the similarity is:

$$q_{j|i} = \frac{\left(1 + \|\mathbf{z}_i - \mathbf{z}_j\|^2\right)^{-1}}{\sum_{k \neq i} \left(1 + \|\mathbf{z}_i - \mathbf{z}_k\|^2\right)^{-1}}.$$

The KL divergence to minimize is:

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

where $p_{ij} = (p_{j|i} + p_{i|j})/2$. This method excels in preserving local structures.

2.6 Complexity

The table 1 summarizes the computational complexity of the methods.

Metric	LLE	Isomap	MDS	t-SNE
Complexity	$O(n\,k^3 + n^3)$	$O(n^2\,k \log n) + O(n^3)$	$O(n^3)$	$O(n^2)$

Table 1. Computational Complexity of Dimensionality Reduction Methods, where n is the number of samples and k is the number of neighbors.

For LLE, the term $O(n\,k^3)$ reflects the cost of reconstructing each point from its k nearest neighbors, while the $O(n^3)$ term arises from the eigen-decomposition of a global matrix. Isomap incurs $O(n^2\,k \log n)$ operations to compute shortest paths in the neighborhood graph (using Dijkstra's algorithm), along with an additional $O(n^3)$ cost from applying classical MDS on the resulting distance matrix. In the case of MDS, the dominant cost is the eigen-decomposition of an $n \times n$ matrix, leading to $O(n^3)$ complexity while PCA has a complexity of $O(d^3)$. Finally, t-SNE's complexity of $O(n^2)$ is due to the computation of pairwise affinities and the subsequent optimization process.

3 Synthetic and Real Datasets

In this section, we describe our synthetic and real datasets. First, we introduce four custom 3D synthetic datasets: the Swiss Roll, Helicoids, Bonhomme, and Zigzag. Each dataset is well-suited to one particular dimension reduction method. Those datasets are shown by the figure 1. We then present a more complex, a real-world dataset composed by musics features. Those datasets are defined and analysed in the notebook of the repository called data_set_analysis.

3.1 Synthetic Datasets

3.1.1 Swiss Roll. The Swiss Roll is a thousand points dataset, generated by rolling a two-dimensional strip into a spiral-like 3D shape, with each point labeled by its "unrolled" position. Because the manifold is smoothly curved, a method that preserves geodesic distances and effectively "unrolls" the spiral is required, making **Isomap** the most suitable choice. In contrast, a purely linear technique like MDS struggles with the strong curvature, leading to a distorted, flattened embedding. LLE is expected to be less effective in capturing the global structure, as it relies on linear relationships among neighbors while the manifold curves continuously in one direction at every point.

3.1.2 Helicoids. We generate two helices in 3D composed by 500 points each, stacked and shifted along a common axis. Although they share the same shape, their arrangement makes the dataset somewhat intricate and noisy. In this case, the label is binary, with one label assigned to each helix. A method that specializes in separating closely located clusters is ideal, so **t-SNE** is typically expected to achieve a distinct separation in the low-dimensional space. LLE and Isomap may merge points from both helices due to the close proximity of the two structures, while MDS cannot properly unfold this non-linear dataset.

3.1.3 Bonhomme (Stick Figure). The Bonhomme dataset is a custom 3D stick figure composed by 1163 points, labeled continuously along its symmetry axis. Due to its irregular shape (comprising a head, limbs, and body), the figure can be viewed as a direct 3D transformation of a simpler 2D stickman (with circles replaced by spheres and lines by cylinders). Consequently, a method that preserves global structure is necessary. Moreover, since the added third axis has a lower variance, a 2D linear projection like **MDS** is well-suited to capture the overall shape. LLE and Isomap are expected to struggle due to their limited global perspective, with Isomap performing better than LLE because its geodesic distance matrix provides a global view of the structure, essentially functioning as a local version of MDS.

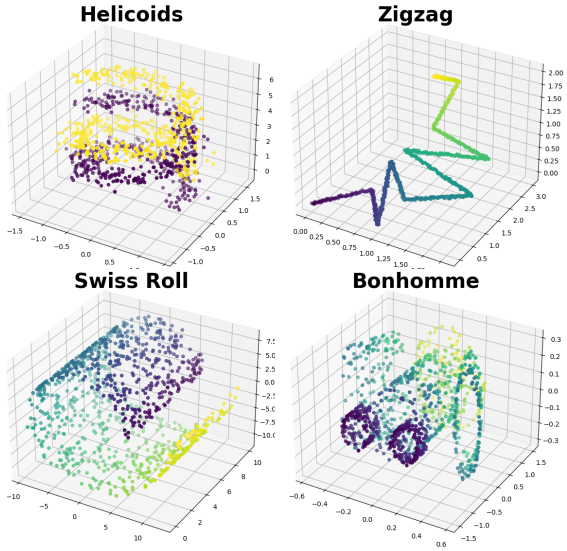

Fig. 1. Illustration of the four synthetic datasets: Helicoids, Zigzag, Swiss Roll and Bonhomme.

3.1.4 Zigzag. The Zigzag a 830 points dataset, consisting of multiple linear segments arranged along different axes, introducing piecewise-linear bends. Noise is added at the segment junctions, and the labels increase from zero to one across the manifold. Isomap should produce a reasonable result due to the simplicity of the manifold, but **LLE** is expected to perform best because it easily captures the local linearity of each segment by reconstructing each point as a linear combination of its neighbors. MDS is likely to fail due to the absence of global linearity, and t-SNE may risk aggregating some unexpected regions.

3.2 Real Dataset

The real-world dataset is sourced from the following Spotify dataset on Kaggle: https://www.kaggle.com/code/jgabrielsb/spotify-songs-music-genre-predictor-part-i/notebook. We extracted 5 357 samples representing music tracks from various Spotify playlists; each sample includes 13 numerical features and 5 textual ones. The target label, denoting the music genre, is categorized into five classes: Rap, Rock, Pop, R&B, and Latin (see Table 2). We will adopt the pretreatment describe in [1] where a more extensive analysis of this dataset is presented.

Music_genre	rap	pop	r&b	latin	rock
count	1149	1101	1086	1031	990

Table 2. Music genre repartition (tableau transposé)

The numerical features consists of variables such as tempo, danceability, liveness, energy, etc. These variables are normalized with *StandardScaler*.
The textual data come from five nearly unique categorical variables: `track_name`, `track_album_name`, `track_album_release_date`, `playlist_name` and `playlist_id`.
Since these features are almost unique, one-hot encoding is inappropriate. To address this, we concatenate the textual variables and encode them using Sentence Transformers, converting the entire textual data into an embedding vector of dimension 384 with the model `all-MiniLM-L6-v2`. This method allows for uniform processing of these variables.
We merge the 384 dimensions from the textual embeddings with the 13 numerical features to obtain a vector of dimension 397 per sample. This preprocessing strategy thus enables a unified and robust handling of heterogeneous data.

4 Neighbor Selection Protocol

The choice of the number of neighbors (k) is pivotal for local methods such as Isomap and LLE. A value of k that is too small leads to mess global structures, while an overly large k makes LLE and Isomap becoming close to PCA and MDS that are linear methods. In this section, we propose a unified procedure for selecting k, ensuring a balance between local fidelity and global structure preservation. The metric we use are not chosen to obtain the best result according to the metric we will compare the methods with. We designed one to aligned with the objective of LLE and Isomap to exacerb their behaviours. The computation of the neighbors selection process is available in the notebook of the repository called selection_of_n_neighbors.

(1) **Define a Candidate Range.**
We first choose a range of neighborhood sizes: $\{5, 6, 7, \ldots, 89\}$. The upper bound is typically set around to 10% of the total number of samples to avoid overly dense or trivial neighborhood graphs. We keep this range for the Real Dataset.

(2) **Generate Low-Dimensional Embeddings $(z_i)_i$.**
For each k in our candidate set, we produce the embedding for LLE and Isomap.

(3) **Evaluate Embedding Quality.**
Recall that the global objective is not to find the embedding that minimize a certain metric. The goal is to compare the 4 paradigms taken by the the reduction methods. This is why we define and combine two metrics (one global and one local) to select the optimal number of neighbors, that we will not use in the final comparison. Indeed that would not be fair, because we selected the k parameter to optimize those metrics.
The two metrics we use are:

- *Geodesic Stress:*
The *geodesic stress*, quantifies how well the geodesic distances in the original high-dimensional space are preserved in the embedded space. It consists to compare the pairwise geodesic distances before and after embedding. We replace d_{ij} in Equation (1) with $d_{ij}^{\mathcal{G}}$, the geodesic distance between x_i and x_j. Lower stress values indicate better preservation of these distances. This metric is global and ranges from 0 to 1, with 0 being the best value.

It is important to note that, although geodesic stress is useful for evaluating embedding fidelity, it cannot be used to fairly to compare different models in our study because Isomap explicitly minimizes this metric, giving it an inherent advantage over other methods. Therefore, even without using it as a criterion for neighbor selection, a comparison based on geodesic stress would be biased.

- *k'-Neighbor Preservation:*
  This metric is defined as the average percentage of shared neighbors between each point's high-dimensional representation and its low-dimensional embedding. The calculation depends on the number of neighbors considered; here, we use the same number as that employed by the method being evaluated : k' = k. Although this approach means that the metric varies with different hyperparameter choices. When a local method is applied with a certain value of k, it is supposed that k is the appropriate scop for capturing the data's structure, then using this k as the value of k' seems a good choice for computing this local preservation metric. This metric nature is local and takes values in [0,1], with higher values signifying better preservation of the neighborhood.

  Since methods like LLE reconstruct each point based on its local neighborhood, verifying that these local relationships are maintained in the embedding is particularly relevant. It is worth noting that this metric tends to increase with larger k. For instance, if k comprises 50% of the dataset, even a random placement of the embedded points would yield a neighbor preservation score of approximately 50%.

(4) **Identify Optimal k via Combined Criteria.**
We agregate the two metrics by taking the mean of geodesic stress and 1 − neighbor preservation. This inversion aligns both metrics with the minimization objective.
We then select the optimal k value by minimizing the combined metric for each dataset. This approach balances the trade-off between achieving low global stress and high local neighbor preservation. A plot of this metric is displayed in Appendix figures 5 (Isomap synthetic), 6 (LLE synthetic) and 7 (Real dataset).

k	Helicoids	Zigzag	Swiss Roll	Bonhomme	Real Dataset
ISOMAP	36	12	35	89	89
LLE	64	54	55	41	89

Table 3. Selected k parameter for ISOMAP and LLE.

For Bonhomme dataset, Isomap method is the best accordingly to our neighbors selection method for the highest value tested. This behaviour were expected cause the dataset has been selected to suit to MDS and more the number of neighbors is high, more it is close to MDS. We give insights to explain the saturation of the number of neighbors for Real Dataset in Section 6.2.

5 Measurement Methods

Evaluating the quality of dimensionality reduction techniques is challenging because different metrics capture complementary aspects of the embedding's performance. We propose a slight modification to the approach described in [7] that use trustworthiness, continuity and 1-NN classifier. We replace 1-NN classifier by a linear classification concistency to leverage on the labeled data. The overall measurement score is computed by averaging these three metrics [3] to derive a comparable measurement score for the embedding's quality, leveraging on the labels:

(1) **Trustworthiness (M_t)**: Measures the degree to which the local structure in the embedding avoids introducing false neighbors that were not close in the original high-dimensional space.
(2) **Continuity (M_c)**: Acts as a complementary measure to trustworthiness, capturing how well true neighbors in the original space remain preserved in the embedding, avoiding the loss of meaningful local relationships.
(3) **Classification Consistency (M_{class})**: Evaluates the preservation of label-relevant structure using a one-vs-rest linear classifier.

5.1 Trustworthiness

Trustworthiness ranges between 0 and 1, with 1 indicating a perfect preservation of local neighborhoods. For a given point i, let $U_K(i)$ denote the set of points that appear among its K nearest neighbors in the embedding but not among its K nearest neighbors in the original space. Denote by $r(i, j)$ the rank (in the original space) of point j when the distances from i are sorted in ascending order (with the closest neighbor ranked 1, excluding i itself). The trustworthiness metric is then defined as:

$$M_t = 1 - \frac{2}{nK(2n - 3K - 1)} \sum_{i=1}^{n} \sum_{j \in U_K(i)} \big(r(i, j) - K\big),$$

where n is the total number of samples and we set $K = 7$ (in line with prior studies). This formulation penalizes the inclusion of points as neighbors in the embedding when they are not among the closest in the original space.

5.2 Continuity

Continuity also takes values in $[0, 1]$ with higher values signifying better preservation of the original neighborhood structure. For each point i, let $V_K(i)$ be the set of points that are among its K nearest neighbors in the original space but are missing from its K nearest neighbors in the embedding. Let $\hat{r}(i, j)$ be the rank of point j when the distances from i in the embedding are sorted in ascending order (with the self-distance taken as the first element). The continuity measure is defined as

$$M_c = 1 - \frac{2}{nK(2n - 3K - 1)} \sum_{i=1}^{n} \sum_{j \in V_K(i)} \big(\hat{r}(i, j) - K\big),$$

again with $K = 7$. This metric penalizes cases where neighbors in the original space are not preserved in the embedding.

5.3 Classification Consistency

To assess whether the embedding retains label-relevant information, we evaluate classification consistency using a one-vs-rest linear Support Vector Classifier (SVC). The evaluation is performed as follows:

(1) **Normalization:** The low-dimensional embedding is first standardized to have zero mean and unit variance.
(2) **Label Discretization:** If the target labels are continuous, they are discretized into 10 equal-width bins; otherwise, the original categorical labels are used.
(3) **Data Partitioning:** The dataset is randomly split into a training set (70%) and a test set (30%).
(4) **Classification and Evaluation:** A linear SVC employing a one-vs-rest strategy is trained on the training data, and the resulting classification accuracy M_{class} is computed on the test data.

This metric is particularly suitable for synthetic datasets because it is straightforward to visualize how these datasets can be projected into a 2D space to achieve linear separability, given their structured nature and well-defined labels. However, its applicability to real-world datasets, such as the Spotify music dataset analyzed in this study, is limited. These limitations are discussed in subsection 6.2, and alternatives are proposed to overcome them while still using labels to evaluate dimensionality reduction methods.
A higher classification accuracy indicates that the embedding better preserves the information required to distinguish between different classes.

5.4 Aggregated Quality Score

We define the overall quality score Θ by first converting each individual metric into an error-like measure by subtracting it from 1, and then taking the arithmetic mean of the 3 metrics:

$$\Theta = \frac{1}{3}\Big((1 - M_t) + (1 - M_c) + (1 - M_{class})\Big) \tag{2}$$

This aggregated score Θ provides a comprehensive evaluation of an embedding. The aggregated score Θ lies in $[0, 1]$ and a score of 0 indicates an optimal embedding.

6 Experimental Results

We now present the experimental results obtained by applying the evaluated dimensionality reduction techniques (t-SNE, LLE, Isomap, and MDS) on both synthetic and real datasets. All 3 quality metrics defined in Section 5 were computed and aggregated to compute the overall quality score defined in (2). The experiments were conducted in the notebook located in the repository named experiment_results_synthetic.

6.1 Results on Synthetic Datasets

Table 4 summarizes the metric scores for each method. Figure 2 displays 2D projections for qualitative assessment.
Each dataset was selected using a heuristic intended to highlight the strengths of a particular embedding method. The obtained quantitative scores and visualizations largely confirm our expectations, as detailed below.

Method	Helicoids	Zigzag	Swiss Roll	Bonhomme
t-sne	**0.000**	0.029	0.039	0.112
lle	0.129	**0.028**	0.037	0.047
isomap	0.082	0.044	**0.029**	0.046
mds	0.148	0.049	0.032	**0.044**

Table 4. Aggregated quality metric scores across datasets for each dimensionality reduction method.

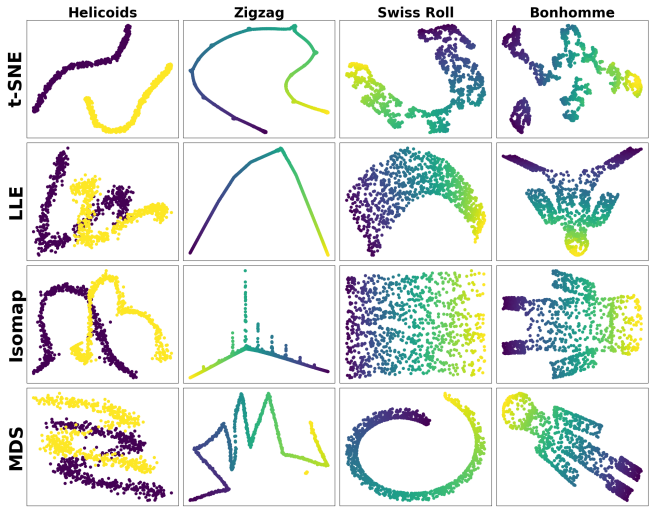

Fig. 2. Embedding of synthetic datasets for all methods

Helicoids: For the Helicoids dataset, any method successfully separated the two underlying structures, except t-SNE. **t-SNE** achieved a perfect score and produced an embedding in which the two branches were clearly delineated. In contrast, the other methods yielded embeddings in which the branches overlapped. This result underscores t-SNE's ability to unfold complex, intricate structures, a capability that is not reached with linear or local methods.

Zigzag: The Zigzag dataset, characterized by a patchwork of straight-line segments joined at distinct junctions, cannot be properly unfolded using a purely linear method like MDS. Isomap, struggles at the junctions, while t-SNE, although performing better due to the density of the points, but it produces curved embeddings that deviate from the original linear segments. In contrast, **LLE** produced both the best visual results and the highest scores; it accurately embedded the dataset by preserving the local linear relationships inherent to its structure. Nonetheless, the quantitative scores for LLE and t-SNE were quite similar, while the visual based comparison is clearly advantage LLE.

Swiss Roll: The Swiss Roll dataset presents a classic challenge for dimensionality reduction techniques. Due to its linear nature, MDS is unable to unravel the Swiss Roll, even though its score remains competitive relative to the best performance achieved by Isomap. t-SNE tends to distort the structure by aggregating certain regions

because the dataset contains holes. LLE, while producing a reasonably good embedding, compresses some regions and leaves the overall shape slightly twisted, because the method lack of a global distance preservation mechanism. In contrast, **Isomap** is specifically designed to preserve geodesic distances along the manifold, thereby maintaining the global structure and resulting in a clearly unrolled, rectangular embedding.

Bonhomme: The Bonhomme dataset is highly complex and structured. t-SNE performed poorly on this dataset, failing to capture the overall structure and aggregating some parts due to the holes. In contrast, LLE produced a better embedding than our expectations; its success can be attributed to the relatively large number of neighbors selected (41) and the near two-dimensional nature of the dataset, which makes it amenable to modeling via local linear relationships. It is noteworthy that the LLE embedding appears to discard the leg regions. This behavior is likely due to the large geodesic distances between the feet, a phenomenon that one would also expect from the isomap which is a geodesic distance preserving method. However, when Isomap was applied with a very large neighborhood size (89), it's lead to a well preserved feet distance. In this settings isomap results became similar to those of MDS. In fact, **MDS** yielded the best performances for Bonhomme, as the near linear spatial distribution of points (primarily extending from head to foot and between the hands rather than from front to back) is well-suited to this linear method. This is corroborate by the neighbors selection curve of isomap, which is decreasing accordingly to the number of neighbors. Both visualizations and quantitative scores show that MDS performing the best, even though the scores for LLE and Isomap are close.

Detailed tables with all the 3 metrics and time to compute the embeddings are given is the appendix B. We can observe that LLE as the lowest time of computation and it seems to do not change a lot with the number of neighbors, and MDS as the worst (sometimes more than 20 times longer). If PCA would be chosed instead of MDS, the computation time would be lower with the same result. Isomap is the second most efficient method while t-SNE is the third one. The aggregated quality score Θ (see Table 4) indicates a better dimensionality reduction for t-SNE.

6.2 Results on Real Datasets

Table 5 summarizes the quality metrics obtained when applying the four methods to our real-world Spotify dataset. In this experiment, t-SNE achieved the lowest aggregated score, primarily due to its excellent continuity and outstanding trustworthiness. This indicates that t-SNE excels at preserving local relationships—the neighbors identified in the low-dimensional embedding are almost always true neighbors from the original high-dimensional space.

Interpreting the linear classification consistency, is more challenging. This metric is computed using a linear classifier, which assumes that the class structure can be separated by a linear boundary. In synthetic datasets, labels are highly correlated with the spatial arrangement of the data, making linear separation straightforward. In real-world datasets, the relationship between features and labels is more complex. Weak correlations, non-linear separability, and potential labeling errors can all undermine the relevance of this metric and complicating the interpretation of the classification consistency score.

Metric	t-sne	lle	isomap	mds
M_t	**0.989**	0.784	0.801	0.790
M_c	**0.959**	0.930	0.949	0.919
M_{class}	0.297	0.273	0.252	**0.326**
Overall Score	**0.252**	0.338	0.333	0.321
Time	72.978	**46.907**	118.963	1648.607
(RBF Classification	**0.409**	0.327	0.352	0.375)
(RBF Overall Score	**0.214**	0.320	0.299	0.305)

Table 5. Quality metric scores for each dimensionality reduction method on the real dataset.

Despite these challenges, MDS yielded the highest linear classification consistency in our experiment. This suggests that, when evaluated using a linear classifier, MDS is more effective at capturing the global structure of the dataset in a way that aligns with the music genres. Moreover, this global preservation might explain why our neighbor selection process in Section 4 resulted in a higher number of selected neighbors for MDS.

To address the challenge of non-linear separability inherent in real datasets, we evaluated classification consistency using an RBF kernel and computed the RBF aggregated score, substituting M_{class} with its RBF counterpart. As visually suggested in Figure 3, t-SNE achieved the highest score, confirming that it produces the most effective embedding. The other methods yielded comparable results.

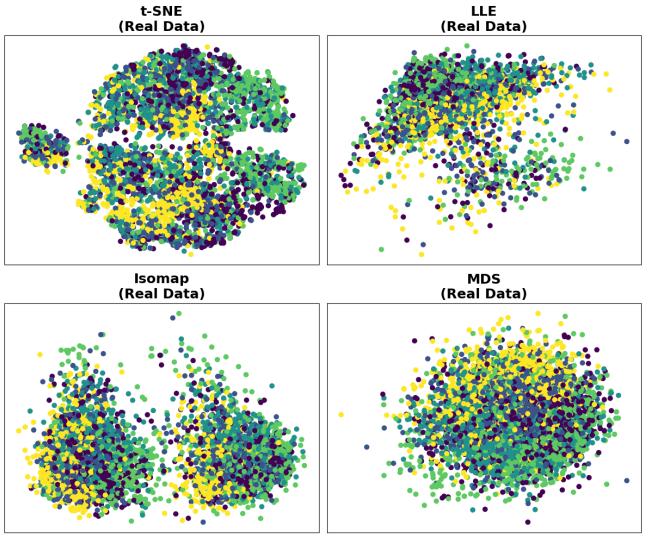

Fig. 3. Embedding of Real dataset for all methods

7 Discussion and Conclusion

This study has provided a detailed comparative analysis of four dimensionality reduction techniques: t-SNE, LLE, Isomap, and MDS. Through examination of synthetic and real-world labeled datasets, we have identified specific strengths and optimal use cases for each method. Our findings enable us to provide guidelines for selecting appropriate reduction methods based on dataset characteristics.
A Decision tree resume the following analysis in figure 4.

Guidelines for Method Selection

(1) **Datasets with Global Linear Structure**
- Recommended Method: MDS
- Suitable for data with linear relationships between features. It is effective when the variance in some dimensions is low. It preserves global structures well. This method is useful, for example, to reduce noise as pretreatment.
- MDS merges points that lie on dimensions orthogonal to the selected ones.

(2) **Datasets with Continuous Curved Manifolds**
- Recommended Method: Isomap
- Best for data lying on a smooth, continuous manifold with intrinsic lower dimensionality. It is particularly effective when geodesic distances need to be preserved. Isomap seems to be the most reliable method and can be selected by default for continuous datasets that don't fall within the scope of other methods.
- It can suffer from short-circuiting when the density of the data is too low or the number of neighbors hyperparameter is too high.

(3) **Datasets with Local Linear Patterns**
- Recommended Method: LLE
- Optimal for data that can be approximated by a collection of locally linear patches.
- Most effective when the underlying manifold consists of a collection (no necessarily linear) of linear shapes.
- Depending on the number of neighbors selected, LLE has the tendency to collapse large portions of the data very close together in the low-dimensional space, specifically when the dataset contains holes.

(4) **Datasets with Distinct Clusters (discrete labels)**
- Recommended Method: t-SNE
- Best suited for datasets where maintaining separation between different groups is requiered. It is effective when local structure preservation is more important than global geometry.
- t-SNE fails for data with inherent continuous structure, particularly when holes occur.

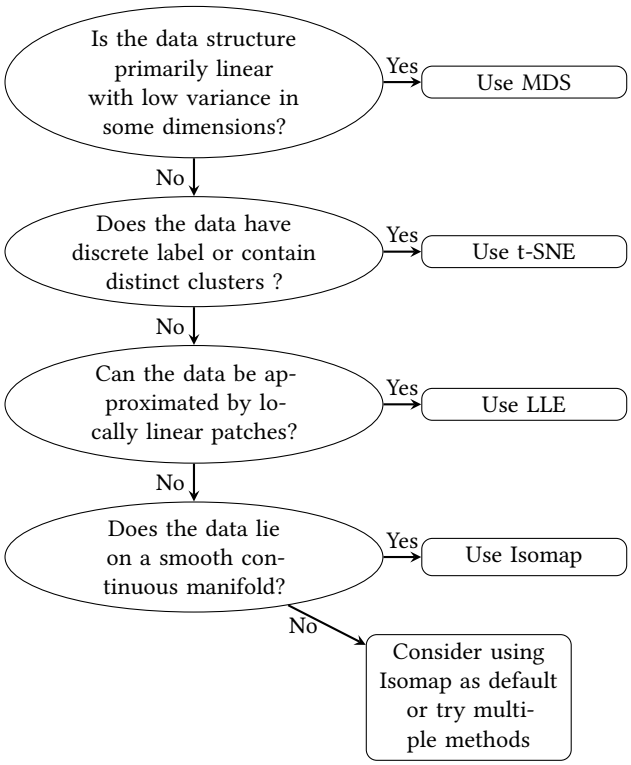

Fig. 4. Decision tree to select the appropriate dimensionality reduction method based on the data characteristics

References

[1] ChapponE. 2025. Spotify songs music genre predictor. https://github.com/ChapponE/spotify-songs-music-genre-predictor.

[2] Trevor Cox and Michael Cox. 2000. *Multidimensional Scaling* (2nd ed.). Chapman and Hall/CRC, New York. 328 pages. doi:10.1201/9780367801700 Behavioral Sciences, Mathematics & Statistics.

[3] Mateus Espadoto, Rafael M. Martins, Andreas Kerren, Nina S. T. Hirata, and Alexandru C. Telea. 2021. Toward a Quantitative Survey of Dimension Reduction Techniques. *IEEE Transactions on Visualization and Computer Graphics* 27, 3 (2021), 2153–2173. doi:10.1109/TVCG.2019.2944182

[4] Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290, 5500 (Dec. 2000), 2323–2326. doi:10.1126/science.290.5500.2323

[5] Joshua Tenenbaum. 1997. Mapping a Manifold of Perceptual Observations. In *Advances in Neural Information Processing Systems*, M. Jordan, M. Kearns, and S. Solla (Eds.), Vol. 10. MIT Press. https://proceedings.neurips.cc/paper_files/paper/1997/file/28e209b61a52482a0ae1cb9f5959c792-Paper.pdf

[6] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[7] Laurens van der Maaten, Eric Postma, and H. Herik. 2007. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research - JMLR* 10 (01 2007).

Appendix

A Selection curves for number of neighbors with Isomap and LLE

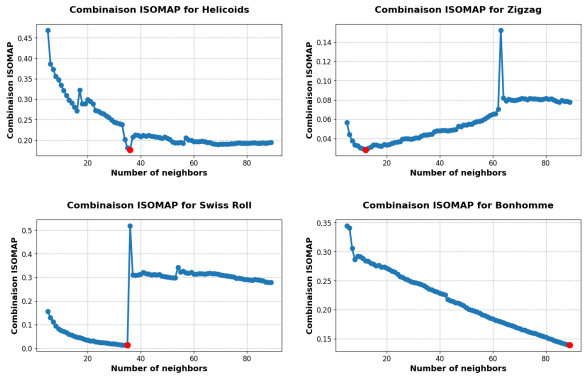

Fig. 5. Combined metrics with Isomap for the 4 datasets.

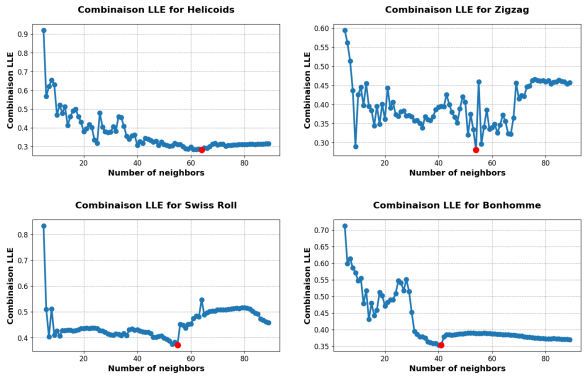

Fig. 6. Combined metrics with LLE for the 4 datasets.

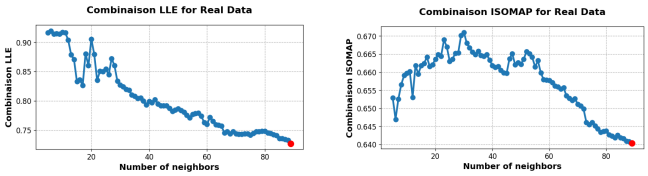

Fig. 7. Combined metrics with Isomap and LLE for the real dataset.

B Detailed tables of synthetic data results

Metric	t-sne	lle	isomap	mds
M_t	**0.999**	0.964	0.986	0.978
M_c	**1.000**	0.996	0.998	0.996
M_{class}	**1.000**	0.653	0.770	0.583
Overall Score	**0.000**	0.129	0.082	0.148
Time	3.773	**0.245**	0.817	25.803

Table 6. Quality metric scores for dataset Helicoids.

Metric	t-sne	lle	isomap	mds
M_t	**1.001**	1.000	1.001	1.001
M_c	**1.001**	1.001	1.001	1.000
M_{class}	0.912	**0.916**	0.867	0.851
Overall Score	0.029	**0.028**	0.044	0.049
Time	3.275	**0.189**	0.224	11.099

Table 7. Quality metric scores for dataset Zigzag.

Metric	t-sne	lle	isomap	mds
M_t	1.001	0.999	**1.001**	0.960
M_c	1.000	0.999	**1.001**	0.997
M_{class}	0.883	0.890	0.910	**0.947**
Overall Score	0.039	0.037	**0.029**	0.032
Time	4.097	**0.239**	0.969	8.752

Table 8. Quality metric scores for dataset Swiss Roll.

Metric	t-sne	lle	isomap	mds
M_t	**1.000**	0.981	0.980	0.981
M_c	0.998	0.998	0.998	**0.999**
M_{class}	0.665	0.880	0.883	**0.888**
Overall Score	0.112	0.047	0.046	**0.044**
Time	5.266	**0.282**	2.586	33.282

Table 9. Quality metric scores for dataset Bonhomme.