An alternative for data visualization using space-filling curve

Valentin Owczarek, Patrick Franco, Rémy Mullot

An alternative for data visualization using Space-Filling Curve

Valentin Owczarek* Patrick Franco* and Rémy Mullot*

Laboratoire L3i, EA 2118, University of La Rochelle,
Avenue Michel Crépeau, 17042 La Rochelle Cedex 1, France

firstname.lastname@univ-lr.fr

Abstract. Dimensionality reduction helps data analysts and machine learning designers to visualize in low dimension structures lying in high dimension. This is a basic but crucial operation, to discover relationship between variables, considering the difficulties to tweek machine learning algorithm. The data have not to be consider as a black-box but can be visualized, leading to better decision making. Inspired from previous works, this article proposes to create a dimensionality reduction method based on Space-Filling Curves (SFCs). Of course, the Hilbert curve was considered (guided by reflected binary gray code pattern) but also alternative high locality SFCs, recently identified. Mapping algorithms working with alternative curves are provided, and illustrated through a numerical example. Mapping a D-dimensional point to a $1-D$ index is usual but developing an algorithm for reverse mapping, i.e. from $1-D$ index to $2-D$ or $3-D$ point is more original and can allow the visualization of data. The work position is specified and justifications are given. A discussion on the choice of parameters (order of curves n and n') is led in order to guide the user to select good parameters (to define a bijection between original data space and projected space). Experiments are conducted to compare our proposition to state of the art approaches (PCA, MDS, t-SNE, UMAP) over seven dataset involving from $3-D$ to $16-D$ and covering diverse topologies. The results show interesting ability on data visualization. Compare to standard techniques, the time computing is low, which is an interesting property in regards to the amount of data today created.

Keywords: Dimension reduction, Data visualization, Space-filing curves

1 Introduction

Dimensionality reduction helps data analysts and machine learning designers to visualize in low dimension structures lying in high dimension. The basic idea is to consider the data as the seed to design machine learning techniques. The data have not to be considered as a black-box but as useful informations to design data mining application: clustering [Djouzi and Beghdad-Bey, 2019], outlier detection [Singh and Upadhyaya, 2012] and classification [Kotsiantis et al., 2007].

In order to create projection of quality, the methods have to conserve global and local structures, i.e relations between points separated with short and large distances. Moreover, the execution time criterion is important as the method must address huge mass of data. Furthermore, the complexity must not highly raise with the dimension nor the number of data points.

Recently in [Castro and Burns, 2007], J. Castro et al. shown the utility of Space-Filling Curve (SFC) to visualize high dimension data. The technique reduces the data dimension while trying to conserve as possible the neighborhood between D-dimensional points and projected points ($D' = 2,\ 3$). In particular, the paper is drawn on the capabilities of Hilbert curve [Hilbert, 1891] to visualize data compared to Principal Component Analysis (PCA), a reference method in the domain. One interesting property of SFC for data analysis is that the processing time is low, which makes SFC well designed to go across huge amount of data. This observation was confirmed in a previous paper, where high locality preserving level SFC were compared to PCA [Owczarek et al., 2020].

In other side, in [Franco et al., 2018], new space-filling curves have emerged. It was proved that selecting alternative pattern (the SFC at the first order) than the classical one - issued from Reflected Binary Gray code (RBG) - can lead to design curves with comparable (and sometimes better) locality preserving than the Hilbert curve, so far the reference [Faloutsos and Roseman, 1989,**?**,**?**].

Our proposition aims to creates a baseline for SFC dimensionality reduction, more precisely with alternative curves able to reach comparable and sometimes higher level of locality preservation than the regular Hilbert curve. Those results are compared with state of the art techniques enlighted in [Genender-Feltheimer, 2018] for their capabilities in exploratory stage: PCA [Pearson, 1901], MDS [Cox and Cox, 2008], t-SNE [van der Maaten and Hinton, 2008] and UMAP [McInnes et al., 2020].

A discussion on the choice of parameters (order of curves n and n') is led in order to guide the user to select good parameters. The reduction is then a bijection between original data space and projected one.

In order to compare those techniques, studies on the projections ($D' = 2, 3$) based on standard criteria - Sammon stress [Sammon, 1969] and topology preservation measure [Konig, 2000] - is conducted. Seven dataset reflected linear and non-linear distribution, covering wide range of dimensions (3-D to 16-D) are integrated in order to establish a credible benchmark.

The rest of this paper is organized as follows. In Section 2, the definition of standard dimension reduction methods: PCA, MDS, t-SNE is given. The motivations and the positioning of the proposed work is detailed in Section 3. In Section 4, the definition of Space-Filling Curve (SFC) is proposed with a focus on the algorithm define in [Nguyen, 2013]. Mapping algorithms working with alternative curves are provided, and illustrated through a numerical example. Moreover a study is led to optimally choose the algorithms parameters n and n' (orders of the curves). In Section 5, experiments are led to compare the performance of the proposition against PCA, MDS, t-SNE and UMAP. The experimental conditions are precised and the handled dataset are detailed. SFC

shown quality in term of visualization compared to t-SNE with execution time always lower than most of the tested algorithms.

2 Standard dimension reduction techniques

Here, some dimensionality reduction techniques are brievely introduced. This is not an exhaustive list, but usual techniques frequently used in exploratory stage [Genender-Feltheimer, 2018,**?**]. They seek useful informations within the data, depending on the selected method; covariance, nearest-neighbor, or pairwise similarity are the key of the techniques.

2.1 Principal component analysis (PCA)

Principal component analysis [Pearson, 1901,Wold et al., 1987] is a linear projection of the data in a new reference system to maximize variance. Given a set of points $\mathbf{X}$ lying in $\mathbb{R}^D$ and the mean vector $\overline{\mathbf{X}}$. PCA find a new representation $\mathbf{Y}$ of $\mathbf{X}$ where $\mathbf{Y}$ lies in dimension D', with $D' \leq D$, such as:

$$\mathbf{Y} = W(\mathbf{X} - \overline{\mathbf{X}}), \tag{1}$$

where W is a $D \times D'$ matrix formed by the D' first eigenvectors corresponding to the highest eigenvalues of the covariance matrix computed on the data $\mathbf{X}$.

2.2 Multidimensional scaling (MDS)

Multidimensional scaling [Buja et al., 2008,Cox and Cox, 2008] is a class of dimension reduction method which aim to conserve pairwise distance. In this article, the metric MDS on the euclidean distance is studied. The algorithm is based on SMACOF (Scaling by MAjorization of COmplicated Function) to minimize the stress function σ until convergence (i.e $\sigma_{t+1} - \sigma_t < \epsilon$). The σ stress function is defined as:

$$\sigma = \frac{\sum_{i<j}^{N} (d(\mathbf{X}_i, \mathbf{X}_j) - d(\mathbf{Y}_i, \mathbf{Y}_j))^2}{2}, \tag{2}$$

where d is the euclidean distance, N is the number of data point in the dataset $\mathbf{X}$ and $\mathbf{Y}$ is the resulting reduction.

2.3 t-distributed Stochastic Neighbor Embedding (t-SNE)

t-Distributed stochastic neighbor [van der Maaten, 2014,**?**] embedding is based on the conversion of distance into joint probabilities that represent similarities. In the high dimension space, the Gausian distribution P_{ij} is applied, i.e:

$$
\begin{aligned}
P_{ij} &= \frac{P_{j|i} + P_{i|j}}{2n}, \\
P_{j|i} &= \frac{\exp(-d(\mathbf{X}_i - \mathbf{X}_j)/2\sigma_i^2)}{\sum_{k \neq i} \exp(-d(\mathbf{X}_i - \mathbf{X}_k)/2\sigma_i^2)}.
\end{aligned} \tag{3}
$$

The low dimensional pairwise similarity Q_{ij} is define with a t-student distribution, i.e:

$$
Q_{ij} = \frac{(1 + d(\mathbf{Y}_i - \mathbf{Y}_j))^{-1}}{\sum_{k \neq l}(1 + d(\mathbf{Y}_k - \mathbf{Y}_l))^{-1}}. \tag{4}
$$

Then, the low dimensional projection $\mathbf{Y}$ is updated until convergence of the Kullback-Leibler divergence. This algorithm is stochastic, each run of t-SNE produce potentially different results. Reader may find complete information in [van der Maaten and Hinton, 2008].

2.4 Uniform Manifold Approximation (UMAP)

Uniform Manifold Approximation (UMAP) [McInnes et al., 2020] is based on manifold theory and topological data analysis. Similar to t-SNE, UMAP used a two step framework. First, the method creates a manifold approximation in order to access a topological representation of the high dimensional data. Then, an equivalent representation is created in low dimension minimizing the cross-entropy between the high and low topological representations. UMAP is closed to t-SNE mathematically with specific definition:

$$
P_{j|i} = \exp[(-d(x_i, xj) - \rho_i)\sigma_i], \tag{5}
$$

where $P_{j|i}$ is computed only for the n closest neighbors with $d(x_i, x_j)$ the distance between x_i and x_j and σ_i is a normalizing factor.

Reader may find in [McInnes et al., 2020] complete information about UMAP and the diferences with t-SNE.

3 Work positioning and justifications

New methods -for dimensionality reduction- have appeared in the last decade [Lee and Verleysen, 2007,Van Der Maaten et al., 2009], but popular techniques like PCA and t-SNE are still often used, particularly for exploratory data analysis, i.e. the first stage of analysis. This trend was once again confirmed in 2018 [Genender-Feltheimer, 2018], in which a review of algorithms and challenges posed by big data was outlined. However, if PCA and t-SNE are "Ideal for initial data exploration" [Genender-Feltheimer, 2018,Wang et al., 2018], approaches able to address new challenges need to be developed. For example, to facilitate user interactive analysis and to allow on-line visualization.

By mapping a D-dimensional point to a 1-D index, the Space-Filling Curves (SFC) can be viewed as a way to reduce data dimension. If moreover the mapping computation is fast and the conservation of the topology of original data point is effective then, SFC can give rise to interesting approaches. We think that SFC have valuable properties which reasoned in the framework of dimensionality reduction and visualization:

- *locality preserving mapping* : close points in space (locality) remain close after mapping on the curve. So, SFC can capture (as much as possible) in low dimension, local topology of data points (neighborhood) observed in high dimension. That was confirmed in [Faloutsos and Roseman, 1989] where the clustering properties of SFC were studied. Logically, Castro and al. [Castro and Burns, 2007] have oriented their choices toward the Hilbert curve in order to build (attend) more effective projection, i.e. the curve that better preserves locality compared to Peano, Lebesgues and Sweep curves. But, recently in [Franco et al., 2018], alternative patterns have been identified (SFC at the first order). That contributes to design curves bringing comparable (and sometimes better) level of preservation of locality than Hilbert curve, so far the reference. Here, the utility of new curves is explored and results are compared to the initial proposition [Castro and Burns, 2007], and to dimensionality reduction standard techniques.
- *data-set independence* : the level of locality preserving reached is dedicated to a curve and it is independent of the dataset to be processed. In other words, the capability to map close multi-dimensional points to close index (and vice versa) is a property of a curve itself, and is not related to the characteristics of the data-set.
- *1 to 1 mapping* : with SFC, all the dimensions of original data point (coordinates) are integrated in the mapping operation. No arbitrage on which dimension must be removed is needed, that is why - in a certain sense - there is no loss[1] of information. Differently, PCA keeps the directions where there is the most variance (largest eigenvectors). Consequently, low variance dimensions in the data tend to be ignored.
- *fast computing* : for a fixed D-dimensional and n-order curve, off-line built, the mapping algorithms provided (cf. Section 4) run in a constant time. Precisely, the time required to map one N-D data point to an 1-D index (or 2-D, 3-D) is constant and it is no sensitive of the size of the data set ! This is an useful property for online visualization and it is well adapted to the amount of data today created. Furthermore, because understanding data is a more and more difficult task, several moving back between original data-set and embedded dimension are necessary. Practically, providing fast algorithms (cf. Section 4) for mapping and reverse mapping can help to set up an user interactive analysis schema. Results shown that is more difficult

[1] Inevitably, due to the reduction of dimensionality SFC induce topological breaks, which are taken into account in the estimation of locality preservation (via the Faloutsos criteria).

for PCA, where estimation of covariance matrix, eigenvalues and eigenvectors require high time consuming (even if incremental version of PCA exists [Artac et al., 2002]).

4 Space-filling curves applied to dimension reduction

4.1 Space-filling curve, an overview

Space-filling curves are continuous non-differentiable function. The first curve was discovered by Peano [Peano, 1890] in 1890. The function defines a bijection between a D-dimensional grid and their 1-D indexes I on the curve. The property behind theses functions is the locality-preserving level [Moon et al., 2001], the fact that if two points in the high dimensional space are close then their indexes are close too. In other words, the curve achieving the best locality-preserving level is the curve that induced less topological breaks between D-dimensional and 1-D space.

One of the best curves according to the literature is the multidimensional extension of Hilbert curve [Hilbert, 1891,**?**], based on the Reflected Binary Gray code (RBG).

Let's denote by S_n^D the D-dimensional space-filling curve at order n and the inverse function $\bar{S}_n^D$. Then, if $\mathbf{X}_i$ and $\mathbf{X}_j$ are two close D-dimensional points:

$$\begin{aligned} S_n^D(\mathbf{X}_i) = I_i,\ \bar{S}_n^D(I_i) = \mathbf{X}_i, \\ S_n^D(\mathbf{X}_j) = I_j,\ \bar{S}_n^D(I_j) = \mathbf{X}_j, \\ d(\mathbf{X}_i, \mathbf{X}_j) = \varepsilon_D,\ d(I_i, I_j) = \varepsilon_1, \end{aligned} \tag{6}$$

with I_i, I_j 1-D indexes and ε_D, ε_1 small. The curve S_n^D at order $n = 1$ (S_1^D) is called a pattern.

A standard numerical characterization of the locality-preserving level of a space-filling curve can be achieved with the parameterized Faloutsos measure [Faloutsos and Roseman, 1989]:

$$L_{\mathbf{r}}(\bar{S}_1^D) = \frac{1}{2^D} \sum_{k,l \in [2^D],\ k<l,\ d(k,l) \leq \mathbf{r}} \max\{d(\bar{S}_1^D(k), \bar{S}_1^D(l))\}, \tag{7}$$

with d a distance function. $L_{\mathbf{r}}(\bar{S}_1^D)$, is the level of locality preserving reach by a curve S at neighborhood radius r. The Faloutsos criteria verifies that two close indexes, k and l, correspond to points, $\bar{S}_1^D(k)$ and $\bar{S}_1^D(l)$, are respectively close in space. The more $L_{\mathbf{r}}(\bar{S}_1^D)$ tends to 1, the better is the level of locality preserving.

In [Castro and Burns, 2007], the preserving-locality level and the bijection properties of the Hilbert-like curve is the key to map data points from an arbitrary D-dimensional space to a lower dimensional space D' (2 or 3) for data visualization purpose. The dimension reduction is performed with two steps:

1. Map all data points $\mathbf{X}$ to their indexes I using the function S_n^D;

2. Map back the indexes to $\mathbf{Y}$ in D' using the inverse space-filling curve function $\bar{S}_{n'}^{D'}$.

The space-filling curve mapping algorithm, given in [Castro and Burns, 2007], is a variant of the algorithm provided in [Lawder and King, 2001], creating a *RBG* curve. Nevertheless recently, in [Nguyen, 2013,**?**], new algorithms emerged with better or equal performances in term of locality preserving. Thoses contributions open new kind of algorithms based on the inheritance between the $n-1$ and n order curve. Moreover, it has been proven experimentally that the locality-preserving level of the pattern induces a curve with less topological breaks.

For example, alternative patterns (S_1^D) extracted from [Franco et al., 2018] are presented in the case of $D = 3$ in Figure 1. For this specific dimension there is only 3 isometry-free patterns. This number increases with the space dimension.

For upper dimension, the locality preserving level of the RBG pattern and an alternative one ($\mathcal{P}_{RBG}^*$) is synthetised in the Table 1. $\mathcal{P}_{RBG}^*$ is an adjacency-based pattern reaching a better locality-preserving level (Faloutsos criteria, Equation 7). This result was confirmed in [Owczarek et al., 2019], where isometry-free patterns minimizing the Faloutsos critera were identified with the help of a genetic algorithm.

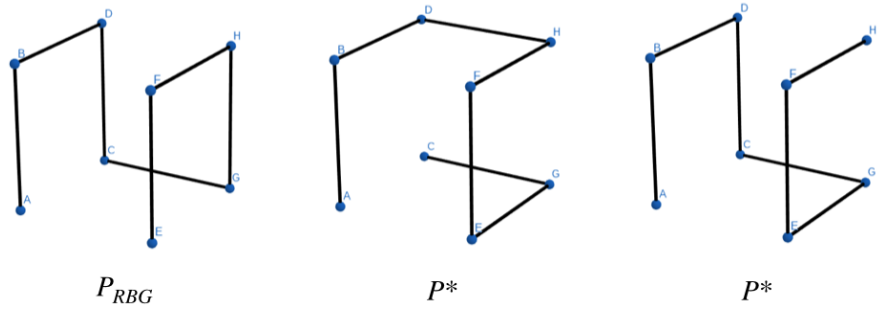

Fig. 1: Example of 3-D SFC, from left to right: the RBG pattern noted P_{RBG} and two alternative patterns called P^*.

In the next section, the space-filling curve function initialized by a pattern $\mathcal{P}$ and an order n will be denoted by $S_n^{\mathcal{P}}$ and its inverse function by $\bar{S}_n^{\mathcal{P}}$. The pattern $\mathcal{P}$ gives the information on dimension D required in the transformation.

Table 1: p^*_{RBG}: The pattern achieving better locality preservation (through Faloutsos criteria) instead of classical RBG often used to build the Hilbert curve.

	Faloutsos score					
Pattern Name	L_3	L_4	L_5	L_6	L_7	L_8
RBG	2.875	2.875	3	3.75	3.75	3.75
$\mathcal{P}^*_{RBG}$	2.875	2.875	3	**3.5625**	**3.5625**	3.75
p^*_{RBG} **Pattern points list**						
0 1 1 0 0 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1						
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1						
0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1						
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0						
0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0						

4.2 Dimension reduction via new Space-Filling Curve: proposition of mapping algorithms

In this section, mapping algorithms working with any patterns are provided. Mapping a D-dimensional point to a 1-D index is usual but developing an algorithm for reverse mapping, i.e. from 1-D index to 2-D or 3-D point is more original and can allow the visualization of data. In order to design such algorithms, a non-regular rotation and reflection is introduced.

$$\begin{aligned} \mathfrak{Iso} &= \mathfrak{Ref}_A \circ \mathfrak{Rot}_f, \\ \mathcal{P}'_i(k) = \mathfrak{Ref}_A(\mathcal{P}_i(k)) &= \begin{cases} 1 - \mathcal{P}_i(k) & \text{if} \quad i \in A, \\ \mathcal{P}_i(k) & \text{if} \quad i \notin A, \end{cases} \\ \mathcal{P}' &= \mathfrak{Rot}_f(\mathcal{P}) = \{\mathcal{P}(f(0)), ..., \mathcal{P}(f(D-1)\}. \end{aligned} \tag{8}$$

The Equation 8 permits to map points $\mathbf{X} \in [0, 2^n - 1]^D$ to the indices $\mathbf{I} \in [0, 2^{nD} - 1]$.

The pattern in the example is $\mathcal{P}_{RBG}$, the RBG-based pattern in $D = 2$, is.:

$$\mathcal{P}_{RBG} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

and the RBG-based pattern in $D = 3$, is:

$$\mathcal{P}_{RBG} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The attached isometry system $\mathfrak{Iso}_{RBG}$ for $D = 2$ is:

$$\mathfrak{Iso}_{RBG} = \begin{pmatrix} \mathfrak{Ref} & \emptyset & \emptyset & \emptyset & 0 & 1 \\ \mathfrak{Rot} & 1 & 0 & \emptyset & \emptyset & 1 & 0 \end{pmatrix}$$

and the $\mathfrak{Iso}_{RBG}$ for $D = 3$ is:

$$\mathfrak{Iso}_{RBG} = \begin{pmatrix} \mathfrak{Ref} & \emptyset & \emptyset & \emptyset & 1\,0 & 1\,0 & 2\,0 & 2\,0 & 1\,0 \\ \mathfrak{Rot} & 2\,0\,1\,0\,2\,1 & \emptyset & 0\,2\,1 & \emptyset & 1\,2\,0\,1\,2\,0\,0\,2\,1 \end{pmatrix}$$

where $\emptyset$ means that there is no rotation or reflection needed.

By notation, the pattern and the isometry system are zero based arrays: the first point of the RBG pattern in $D = 2$ is $\mathcal{P}_{RBG} = (0, 0)$ and the associate isometry is $\mathfrak{Iso}^0_{RBG} = \begin{pmatrix} \mathfrak{Ref} & \emptyset \\ \mathfrak{Rot} & 1\ 0 \end{pmatrix}$.

From a point to an index Let us define a point $x = (3, 2, 2)$, the question is how to compute $S_2^{\mathcal{P}_{RBG}}(x)$?

Algorithm 1 Computation of $S_2^{\mathcal{P}_{RBG}}(x)$ where $x = (3, 2, 2)$

---

1. Initialize: $\mathcal{P} = \mathcal{P}_{RBG} = \begin{pmatrix} 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0 \\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \end{pmatrix}, n = 2,\ D = 3$
2. Get $|x|_2$ the binary representation of x with n digits: $|x|_2 = (11, 10, 10)$
3. Form pp_1 with the most significant digit of $|x|$: $pp_1 = (1, 1, 1)$
4. Calculate ip_1 the index of pp_1 on $\mathcal{P} : ip_1 = 5$
5. Apply the isometry $\mathfrak{Iso}^{ip_1}_{RBG} = \begin{pmatrix} \mathfrak{Ref} & 2\ 0 \\ \mathfrak{Rot} & 1\ 2\ 0 \end{pmatrix}$ to $\mathcal{P}_{RBG}$:

$\mathcal{P} = \mathfrak{Iso}^{ip_1}_{RBG}(\mathcal{P}_{RBG}) = \begin{pmatrix} 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0 \\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0 \\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1 \end{pmatrix}$

6. Form pp_2 with the second most significant digit of $|x|_2$: $pp_2 = (1, 0, 0)$
7. Calculate ip_2 the index of pp_2 on $\mathcal{P} : ip_2 = 5$
8. Deduce $\mathcal{F}_3^{\mathcal{P}_{RBG}}(x)$ from the ip_k index: $\mathcal{F}_2^{\mathcal{P}_{RBG}}(x) = \sum_{k=1}^{n} ip_k 2^{D(n-k)} = (5 \cdot 8) + 5 = 45$

The Algortihm 1 presents the mapping computation: an index from a point, with the space-filling function: $S_2^{\mathcal{P}_{RBG}}$. The 3-D point $x = (3, 2, 2)$ has the index $i = 45$ when the space is filled via the curve $S_2^{\mathcal{P}_{RBG}}$ at order $n = 2$. $S_2^{\mathcal{P}_{RBG}}$ is built from the $\mathcal{P}_{RBG}$ pattern generator using $\mathfrak{Iso}_{RBG}$. The pattern is transformed by unregular isometric transformation (Equation 8). The reverse mapping (index to point) is always possible with the same isometry $\mathfrak{Iso}_{RBG}$.

Glossary

- D is the dimension of the space-filling curve.
- n is the order of the space filling curve.
- $\mathcal{P}$ is a pattern, a space-filling curve at $n = 1$, the seed of the space-filling curve algorithm.
- $S_n^{\mathcal{P}}(x),\ \tilde{S}_n^{\mathcal{P}}(i)$ are respectively the point to index and index to point space-filling curve functions, according to [Nguyen, 2013].
- $x,\ |x|_b$ are respectively a data point and the representation of the point in base b.
- pp_k is a point on the pattern.
- ip_k is an index on the pattern.

From an index to a point of a different dimension: reverse mapping Let us define an index $i = 45$, the question is how to compute $\bar{S}_3^{\mathcal{P}_{RBG}}(i)$?

Algorithm 2 Computation of $\bar{S}_3^{\mathcal{P}_{RBG}}(i)$ where $i = 45$

1. Initialize: $\mathcal{P} = \mathcal{P}_{RBG} = \begin{pmatrix} 0\ 0\ 1\ 1 \\ 0\ 1\ 1\ 0 \end{pmatrix}, n = 3,\ D = 2$
2. Initialize the S interval tuple: $S = ([0, 2^n - 1])^D = ([0, 7], [0, 7])$
3. Get $|i|_{2^D}$ the 2^D representation of i with n digits: $|i|_{2^D} = |i|_4 = 231$
4. Form ip_1 with the most significant digit of $|i|_4$: $ip_1 = 2$
5. Calculate pp_1 the point at index ip_1 on $\mathcal{P}$: $pp_1 = (1, 1)$
6. Update S according to pp_1:
   (a) Divide S in sub-interval $S' = ([[0, 3], [4, 7]], [[0, 3], [4, 7]])$
   (b) According to pp_1, update S: $pp_1(0) = 1$ then $S(0) = [4, 7]$
   (c) $S = ([4, 7], [4, 7])$
7. Apply the isometry $\mathfrak{Iso}_{RBG}^{ip_1} = \begin{pmatrix} \mathfrak{Ref}\ \ \emptyset \\ \mathfrak{Rot}\ 0\ 1 \end{pmatrix}$ to $\mathcal{P}$:

$\mathcal{P} = \mathfrak{Iso}_{RBG}^{ip_1}(\mathcal{P}_{RBG}^2) = \begin{pmatrix} 0\ 0\ 1\ 1 \\ 0\ 1\ 1\ 0 \end{pmatrix}$

8. Form ip_2 with the second most significant digit of $|i|_4$: $ip_2 = 3$
9. Calculate pp_2 the point at index ip_2 on $\mathcal{P}$: $pp_2 = (1, 0)$
10. Update S according to pp_2:
    (a). Divide S in sub-interval $S' = ([[4, 5], [6, 7]], [[4, 5], [6, 7]])$
    (b) According to pp_1, update S: $pp_2(0) = 1$ then $S(0) = [6, 7]$
    (c) $S = ([6, 7], [4, 5])$
11. Apply the isometry $\mathfrak{Iso}_{RBG}^{ip_1} \circ \mathfrak{Iso}_{RBG}^{ip_2} = \begin{pmatrix} \mathfrak{Ref}\ \ \emptyset \\ \mathfrak{Rot}\ 0\ 1 \end{pmatrix} \circ \begin{pmatrix} \mathfrak{Ref}\ 0\ 1 \\ \mathfrak{Rot}\ 1\ 0 \end{pmatrix}$ to $\mathcal{P}$:

$\mathcal{P} = \mathfrak{Iso}_{RBG}^{ip_1} \circ \mathfrak{Iso}_{RBG}^{ip_2}(\mathcal{P}_{RBG}^2) = \begin{pmatrix} 1\ 0\ 0\ 1 \\ 1\ 1\ 0\ 0 \end{pmatrix}$

12. Form ip_3 with the less significant digit of $|i|_4$: $ip_3 = 1$
13. Calculate pp_3 the point at index ip_3 on $\mathcal{P}$: $pp_3 = (0, 1)$
14. Update S according to pp_2: $S = ([6, 6], [5, 5])$
15. Deduce $\bar{\mathcal{F}}_2^{\mathcal{P}_{RBG}}(i)$ from S : $\bar{\mathcal{F}}_2^{\mathcal{P}_{RBG}}(i) = (6, 5)$

The Algorithm 2 presents the computation of the reverse mapping from an index to a point. The index $i = 45$ by the curve $S_2^{\mathcal{P}_{RBG}}$ corresponds to the 2-D point $y = (6, 5)$ (reverse mapping) when the space is filled by the curve $S_3^{\mathcal{P}_{RBG}}$.

The algorithms (Mapping computation and Reverse mapping computation) are used for visualisation purpose in experiments conducted in section 5. Nevertheless, the input parameters could have an impact on data projection and consequently may affect the quality of visualisation. The next section is a guide to choose right parameters in order to assure a bijection mapping.

4.3 Application to dimension reduction

In [Castro and Burns, 2007], Space-filling curves are the key of a new dimension reduction method. The main difference between this technique and the other

introduced in this paper (PCA, MDS, t-SNE, UMAP) is the non usage of the data. The idea is to project the entire D-cube induced by the space-filling curve S_n^D to the D'-cube defined by $S_{n'}^{D'}$ by computing the associate indexes I. There is no formal description of the technique in [Castro and Burns, 2007], the next section is a possible one. D is the dimension of the data to be projected and D' is the target projection dimension. n and n' are the orders (resolutions) of the curve to use.

Let $\mathbf{X}_i$ be a point in the D-dimensional space then $\mathbf{Y}_i$ the projection by $S_D^{\mathcal{P}}$ and $S_{D'}^{\mathcal{P}'}$ is:

$$\mathbf{Y}_i = \bar{S}_{n'}^{\mathcal{P}'}(S_n^{\mathcal{P}}(\mathbf{X}_i)), \tag{9}$$

with $\mathcal{P}$ and $\mathcal{P}'$ are two patterns lying respectively in dimension D and D' with $D > D'$

Nevertheless, the algorithms in Section 4.2 maps points from $[0, 2^n]^D$ to $[0, 2^{nD}]$. The data points must be transformed, $\mathbf{X}$ is rewritten as:

$$\mathbf{X}_i^t = \mathbf{X}_i^t - \min(\mathbf{X}_i^t). \tag{10}$$

This transformation translates all the points in the interval $[0, 2^{nD}]$.

The orders of the two curves are free parameters, but the user must respect the following inequality to avoid any collision, i.e. two different points $\mathbf{X}_i$ and $\mathbf{X}_j$ in high dimension D with the same coordinate in the projected space D':

$$Dn \leq D'n'. \tag{11}$$

The reduction operation is then a bijection, a data point in D corresponds to a unique data point in D'. The grid in high dimension space have less than $2^{D'n'}$ points.

Moreover, the reader may take into consideration, that the minimum order n to capture all the diversity on the data X, modified according to Equation 10, is equal to:

$$\max(\mathbf{X}_{i,j}) < 2^n, \tag{12}$$

where $\mathbf{X}_{i,j}$ denotes the j^{th} coordinate of the i^{th} point.

The Figure 2 illustrates the effect of the parameters n and n' on the 4-Dimensional Iris dataset. The orders parameters n and n' allow to modulate the collision rate between D-dimensional original data $\mathbf{X}$ and projected data $\mathbf{Y}$, evolving in D' ($D' < D$). The collision phenomena occurs when Equation 11 is not respected (cf. Figure 2 where $n = 2$ or $n' = 8$). In other cases, greater values of n and n' provide no improvement of the quality of projection but increase time complexity (cf. Figure 2, last column). This study on n and n' help to smartly choose optimal parameters for experiments conducted in Section 5.

In this section, Space-Filling Curve are introduced and application for data reduction is explained. Moreover, algorithms are presented as example of mapping ($D \to 1$ and $1 \to D'$). A discussion on the parameters n and n' is led in order to optimize the framework to a specific data set without loss in terms of quality nor execution time.

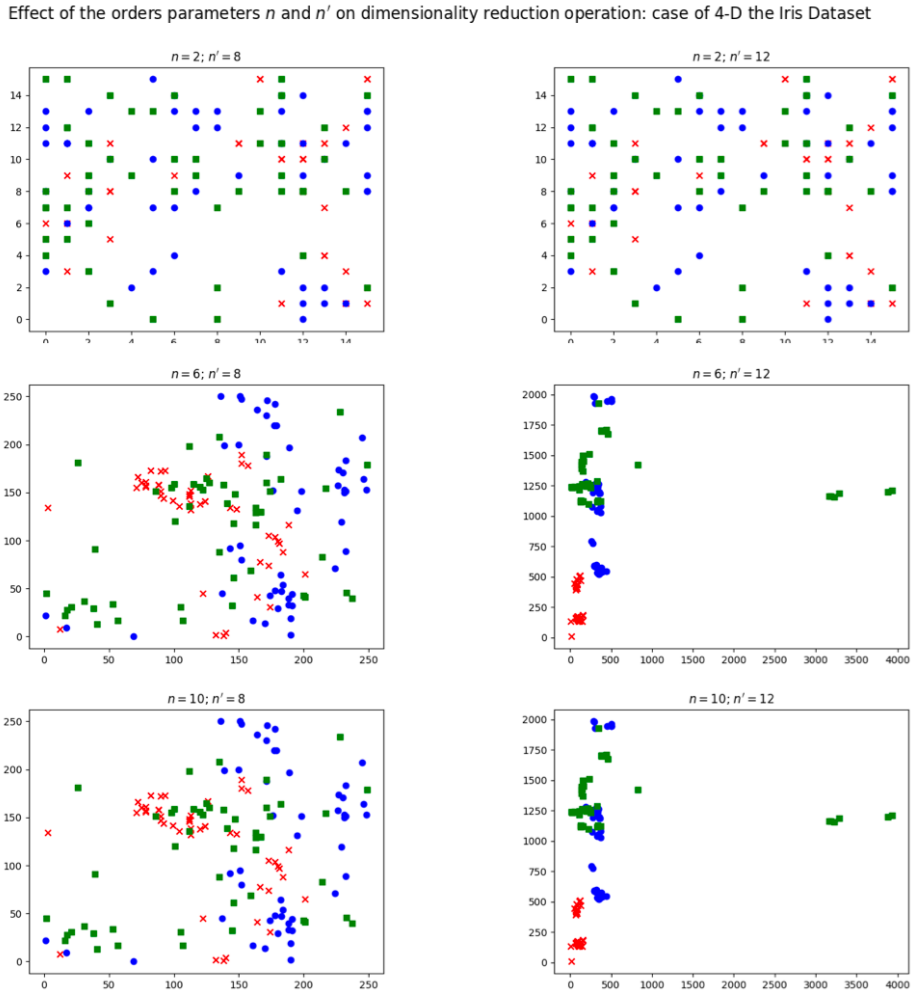

Fig. 2: Example of 2-D project ($D' = 2$) of the 4-D Iris dataset with different values of n and n'. Study on the effects of the Equation 11 on the visualization quality. When $n = 2$ or $n' = 8$ the Equation 11 is not respected, the SFC projection is then not a bijection, decreasing the quality of the visualization.

5 Experiments with standard and space-filling curves dimension reduction techniques

5.1 Experimental conditions: measure

In order to compare the different algorithms, in term of quality of dimension reduction, two standard measures are considered: the topology preservation measure and the Sammon stress.

Topology preservation measure The measure is proposed in [Konig, 2000] and it qualifies the preservation of nearest neighbor between the high dimensional point $\mathbf{X}$ and the projected one $\mathbf{Y}$. This measure is based on a credit assignment scheme:

$$qm_{ji} = \begin{cases} 3 \ NNX(j,i) = NNY(j,i), \\ 2 \ NNX(j,i) = NNY(j,t) \ \ t \in [1,n], t \neq i, \\ 1 \ NNX(j,i) = NNY(j,t) \ \ t \in]n,k], n < k, \\ 0 \qquad\qquad else, \end{cases} \tag{13}$$

where $NNX(j,i)$ and $NNY(j,i)$ correspond to the i^{th} neighbor of respectively the point $\mathbf{X}_j$ and $\mathbf{Y}_j$. The definition of the nearest neighbor is here based on the euclidean distance..

The measure is defined as the normalized sum of the $qm_{i,j}$.

$$Tpm = \frac{1}{3nN} \sum_{j=1}^{N} \sum_{i=1}^{n} qm_{ji}, \tag{14}$$

with n and k initialized to 4 and 10.

This measure indicates a perfect mapping when $Tpm = 1$.

Sammon Stress The measure is extracted from the Sammon mapping [Sammon, 1969,**?**] and can be interpreted as a distance error function. E_{SS} is equal to zero if all distances between original high dimensional points are respected in the low projected space. The original measure is sensitive to rescaling. In order to compare different techniques, which may rescaled data differently, the β term is added.

The measure is then equal to:

$$E_{SS} = \frac{1}{\sum\limits_{i=1}^{n-1} \sum\limits_{j=i+1}^{n} D(\mathbf{X}_i, \mathbf{X}_j)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{\left(D(\mathbf{X}_i, \mathbf{X}_j) - \beta D(\mathbf{Y}_i, \mathbf{Y}_j)\right)^2}{D(\mathbf{X}_i, \mathbf{X}_j)}, \tag{15}$$

with β equals to:

$$\beta = \frac{\sum\limits_{i=1}^{n-1} \sum\limits_{j=i+1}^{n} D(\mathbf{Y}_i, \mathbf{Y}_j)}{\sum\limits_{i=1}^{n-1} \sum\limits_{j=i+1}^{n} \frac{D(\mathbf{Y}_i, \mathbf{Y}_j)^2}{D(\mathbf{X}_i, \mathbf{X}_j)}}. \tag{16}$$

The distance function D is commonly the euclidean function.

The measure indicates a perfect distance preservation when $E_{SS} = 0$.

5.2 Experimental conditions

The previously described standard reduction techniques: Principal component analysis (PCA), Multidimensional Scaling (MDS), t-Stochastic Neighbor embedding (t-SNE) and Space-Filling curve dimension reduction (SFC) are tested on

seven different datasets. The main characteristics of each dataset are listed in Table 2. Practicaly, most of the datasets are extracted from OpenML[Vanschoren et al., 2014] and Scikit-learn[Pedregosa et al., 2011] Inteligence Artificial Python library.

PCA, MDS, t-SNE algorithms are taken from Scikit-learn, which proposed optimized and tested implementation. UMAP is taken from Umap-learn [McInnes et al., 2018].

Table 2: Dataset used for the experiments.

Dataset name	Dimension	Number of attributes	Number of classes
Concentric sphere	3	1000	2
Iris Database	4	150	3
Blob	5	500	3
Monks problems 2	6	601	2
Diabetes	8	768	2
Tic-Tac-Toe	9	958	2
Pendigits	16	10992	10

For the dimension reduction based on Space-Filling curves, the pattern used to map data points in high dimension is specified for each experiment. Hilbert corresponds to the RBG pattern, ($\mathcal{P}^*_{RBG}$) is a pattern achieving better performance than the RBG one according to the parameterized Faloutsos measure (Equation 7). Parameters n and n' are chosen to match the observation of Section 4.3 (Equations 11).

5.3 Results of the experiments

The five algorithms are compared with three different criteria, Sammon Stress, Topology measure, and the execution time[2]. For presentation purposes, a focus on two dataset: Blob and Tic-Tac-Toe is made. All other results are available in Appendix.

In Table 3 and 4, measures for the Blob dataset are shown. The corresponding visualization could be observed in Figure 3. Due to the nature of the data set, Gausian distribution, it is not surprising that PCA and MDS create reduction with good Sammon stress and Topology preservation measures. Nevertheless, Space-filling curve based reduction technique performed well on this dataset. For example the technique is better than PCA and MDS on dimension $D' = 2$. State of the art technique, t-SNE, reached best topology preservation measure, the scheme build on probabilities is well designed to preserved neighborhood. However, the scatter matrix (Figure 3), showed one's limits: t-SNE do not conserve the distance between cluster in dimension $D' = 3$ where SFC split them well. Similar behaviour can be observed for UMAP with comparable Sammon

[2] on a I7-7820HQ Intel® CPU with GCC 6.3.0 and Python 3.5.3

Stress and topology Preservation score. Phenomena involving in high dimension is not always conserved with t-SNE or UMAP in low dimension whereas SFC try at best to reproduce the topology.

In Table 5 and 6, results obtained for the Tic-Tac-Toe dataset are displayed, with the associate plot in Figure 4. PCA or MDS reduction technique are not adapted for this dataset, the data are placed on a grid, making it difficult to extract in dimension 2 or 3 either meaningful eigen value or distance map. This conclusion is led after reading the dataset and intermediary step. In this case t-SNE and SFC have a good advantage, enlightened by the results (Table 5 anf 6), the theoretical result are link to the scatter plot (Figure 4): t-SNE is able to create cluster of same class where SFC lacked of exactness. Umap creates good cluster with easily separable class for the Tic-Tac-Toe dataset with for example the best Sammon Stress score for $D' = 2$ but the topology preservation measure is comparable with the SFC method. Nevertheless, the SFC plot is a good representation of the data: same class point are closed from each other.

With this result in mind, the question of the execution time is to be discussed, data visualization or in more general case dimension reduction is a first step of a complex data mining process. SFC showed on this dataset good results compared to PCA in approximately the same time. The reader have to be aware of the use of LAPACK and ARPACK librairies in the PCA version of sklearn, given a real advantage for execution time.

In this section, results of state of the art techniques are discussed, a focus on two specific case is made to facilitate the understanding. SFC reached intermediary score compared to PCA, MDS, UMAP and t-SNE. But with really reasonable execution time (0.01 second and lower). Further results can be found in Table 7 and 8 with complementary plot in Figure 5 and 6. The same conclusion can be drawn on these results: SFC is well adapted to reduce the dimension of a data set while conserving the neighborhood.

Table 3: Results for the Blob dataset. The projected dimension is $\mathbf{D'} = \mathbf{2}$.

Dataset	Method	Sammon Stress	Topology Preservation	Execution Time (s)
Blob (5-D)	PCA	0.0237	0.1791	< 0.01
	MDS	**0.0159**	0.1948	0.23
	t-SNE	0.0686	**0.5127**	1.49
	UMAP	0.0500	0.3748	3
	Hilbert	0.2579	0.2195	< 0.01
	SFC ($\mathcal{P}^*_{RBG}$)	0.2439	0.2291	< 0.01

Table 4: Results for the Blob dataset. The projected dimension is $\mathbf{D'} = \mathbf{3}$.

Dataset	Method	Sammon Stress	Topology Preservation	Execution Time (s)
Blob (5-D)	PCA	0.0100	0.3243	< 0.01
	MDS	**0.0064**	0.3476	0.33
	t-SNE	0.2996	**0.4581**	3.41
	UMAP	0.0657	0.4566	3
	Hilbert	0.2406	0.2086	< 0.01
	SFC ($\mathcal{P}^*_{RBG}$)	0.1612	0.2108	< 0.01

Table 5: Results for the Tic-Tac-Toe dataset. The projected dimension is $\mathbf{D'} = \mathbf{2}$.

Dataset	Method	Sammon Stress	Topology Preservation	Execution Time (s)
Tic-Tac-Toe (9-D)	PCA	0.1488	0.0328	< 0.01
	MDS	0.1338	0.0538	19.17
	t-SNE	0.1815	**0.2638**	9.91
	UMAP	**0.0768**	0.1793	3
	Hilbert	0.1294	0.1710	< 0.01
	SFC ($\mathcal{P}_A$)	0.1926	0.1709	< 0.01

Table 6: Results for the Tic-Tac-Toe dataset. The projected dimension is $\mathbf{D'} = \mathbf{3}$.

Dataset	Method	Sammon Stress	Topology Preservation	Execution Time (s)
Tic-Tac-Toe (9-D)	PCA	0.0832	0.0578	< 0.01
	MDS	**0.0677**	0.0655	17.87
	t-SNE	0.1207	**0.3108**	30.20
	UMAP	0.4084	0.2156	4
	Hilbert	0.1355	0.1562	< 0.01
	SFC ($\mathcal{P}_A$)	0.1654	0.1529	< 0.01

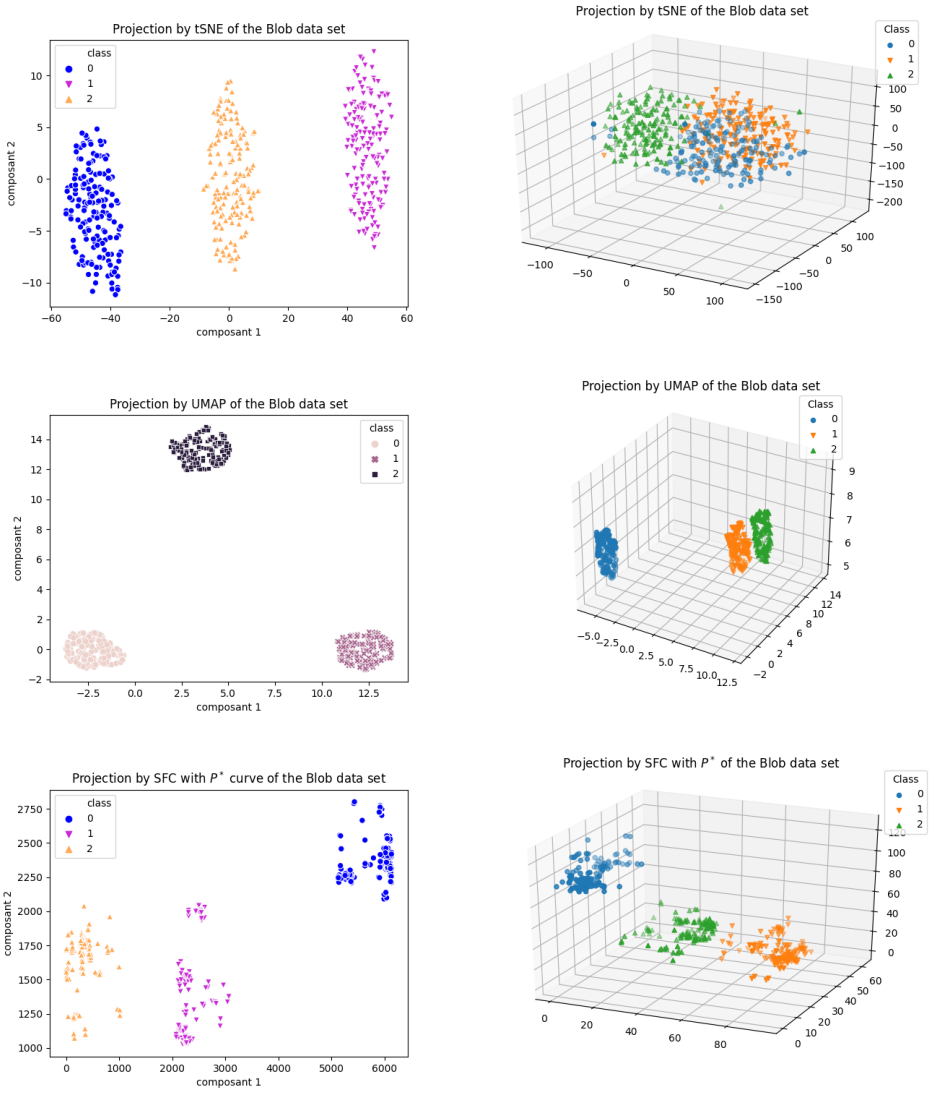

Fig. 3: Example of data projection on the 5-D Blob Dataset. Results of projection, $D' = 2, 3$, from t-SNE, UMAP and SFC based on alternative pattern $\mathcal{P}^*$. The clusters observed in original space are clearly conserved in projections guided by our proposition.

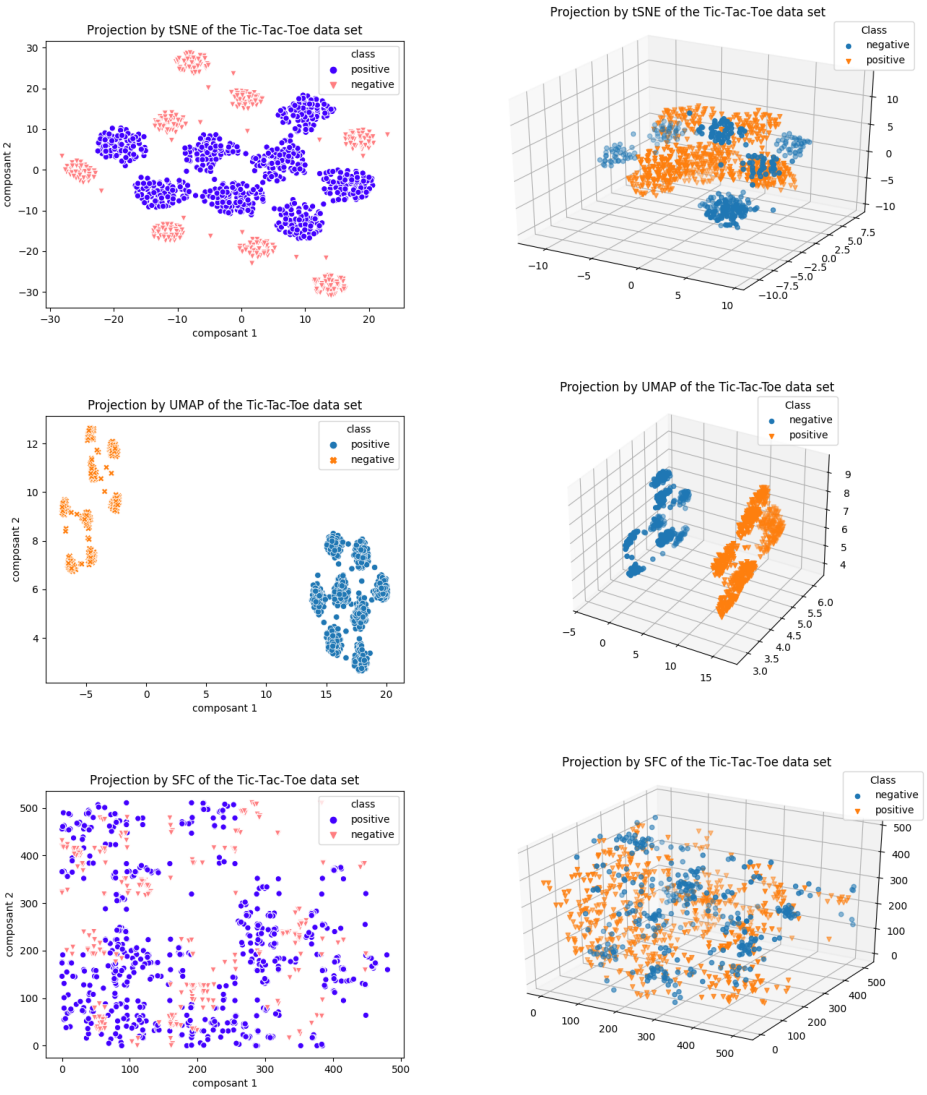

Fig. 4: Example of data projection on the 9-D Tic-Tac-Toe Dataset. Results of projection, $D' = 2, 3$, from t-SNE, UMAP and SFC. The proposed technique creates visualization where the topology of the high dimension data is relatively conserved in low dimension, but t-SNE conserves original cluster, UMAP split the cluster where SFC lackes of exactness.

6 Conclusion

Inspired from previous works, this article proposes to create a dimensionality reduction method based on space-filling curves. This proposition can be considered as an extension of Castro et al. [Castro and Burns, 2007]. Firstly, the mapping algorithms provided in Section 4 are not only able to operate with the Hilbert curve (RBG pattern) but are open to alternative patterns recently identified for theirs capabilities to conserve locality (cf. Section 4.1, Table 1). Moreover, the effects of main parameters (n, n', inputs of algorithms) on the quality of projections are analyzed in order to guide the user to make the right choice for data visualization. Secondly, our proposal is compared with four reference techniques (PCA, MDS, t-SNE), including a recent one (UMAP, [McInnes et al., 2020]), over seven datasets involving from 3-D to 16-D. So, considering the number of techniques used, the variety of topologies covered (linear and non-linear) and the range of dimensions handled, we can reasonably assume that define good experimental conditions, which helps to establish a credible assessment.

In the light of experiments, the proposed approach is fast and always faster than MDS and t-SNE. But above all, the time processing is comparable to the Scikit-Learn[Pedregosa et al., 2011] version of PCA, which is based on optimized LAPACK and ARPACK library (cf. Section 5, Table 3 and 6). This is an interesting property well adapted to the amount of data today created. Furthermore, Because understanding data is a more and more difficult, several moving back between original dataset and embedded dimension are necessary. Practically, providing fast algorithms for mapping and reverse mapping can help to set up an user interactive analysis schema.

Concerning the results of projections (data visualization), the proposed approach is low sensitive to outliers due to the point to point mapping mechanism (cf. Section 3), which is more difficult for PCA and t-SNE. For example, with PCA, outliers affect the covariance matrix and corrupt the estimation of the projected axes.

However, from quality of visualization aspect, experimental results have shown that there are cases where the proposal is competitive (Blob Dataset, cf Figure 3) but also dataset where it fails. For example, dataset Tic-Tac-Toe, topology breaks - between 9-D dataset and results of SFC mapping - appear. This is expressed by the split of the original structure while t-SNE and UMAP provide better plots (cf. Figure 4). In other side, the data analyst knows that t-SNE or UMAP are non determinist (two successive runs can lead to not exactly the same projection) and plots can sometimes be difficult to interpret (cf. mysterious plots https://distill.pub/2016/misread-tsne/)

Because the exploration of multi-dimensional data is firstly a very challenging task, and secondly, there is no way to map high-dimensional data into low dimensions and at the same time preserving the whole of original structure, thus is often more efficient to plan to use a number of techniques. In that context, our proposition with some valuable and complementary properties (refers to state-of-the-art techniques) could be helpful. (A list of properties completed by justifications can be found in Section 3).

Tracks of improvement can be investigated in future work. For example, to reduce the complexity by skipping the initial 1-D indexes projection stage. Reflections must be carried out on creating a by-pass between the original data dimension D and the target one D', substituting the $D \to 1 - D \to D'$ transformation by $D \to D'$

References

[Artac et al., 2002] Artac, M., Jogan, M., and Leonardis, A. (2002). Incremental pca or on-line visual learning and recognition. In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3 - Volume 3*, ICPR '02, pages 30781–, Washington, DC, USA. IEEE Computer Society.

[Buja et al., 2008] Buja, A., Swayne, D. F., Littman, M. L., Dean, N., Hofmann, H., and Chen, L. (2008). Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472.

[Castro and Burns, 2007] Castro, J. and Burns, S. (2007). Online Data Visualization of Multidimensional Databases Using the Hilbert Space–Filling Curve. In *Pixelization Paradigm*, pages 92–109, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Cox and Cox, 2008] Cox, M. A. A. and Cox, T. F. (2008). *Multidimensional Scaling*, pages 315–347. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Djouzi and Beghdad-Bey, 2019] Djouzi, K. and Beghdad-Bey, K. (2019). A review of clustering algorithms for big data. In *2019 International Conference on Networking and Advanced Systems (ICNAS)*, pages 1–6.

[Faloutsos and Roseman, 1989] Faloutsos, C. and Roseman, S. (1989). Fractals for Secondary Key Retrieval. In *Proceedings of the Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '89, pages 247–252. ACM.

[Franco et al., 2018] Franco, P., Nguyen, G., Mullot, R., and Ogier, J.-M. (2018). Alternative patterns of the multidimensional Hilbert curve. *Multimedia Tools and Applications*, 77(7):8419–8440.

[Genender-Feltheimer, 2018] Genender-Feltheimer, A. (2018). Visualizing high dimensional and big data. *Procedia Computer Science*, 140:112 – 121. Cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018.

[Hilbert, 1891] Hilbert, D. (1891). Ueber die stetige Abbildung einer Line auf ein Flächenstück. *Mathematische Annalen*, 38(3):459–460.

[Konig, 2000] Konig, A. (2000). Interactive visualization and analysis of hierarchical neural projections for data mining. *IEEE Transactions on Neural Networks*, 11(3):615–624.

[Kotsiantis et al., 2007] Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24.

[Lawder and King, 2001] Lawder, J. and King, P. (2001). Using state diagrams for hilbert curve mappings. *International Journal of Computer Mathematics*, 78(3):327–342.

[Lee and Verleysen, 2007] Lee, J. A. and Verleysen, M., editors (2007). *Nonlinear Dimensionality Reduction*. Springer New York.

[McInnes et al., 2020] McInnes, L., Healy, J., and Melville, J. (2020). Umap: Uniform manifold approximation and projection for dimension reduction.

[McInnes et al., 2018] McInnes, L., Healy, J., Saul, N., and Grossberger, L. (2018). Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861.

[Moon et al., 2001] Moon, B., Jagadish, H. V., Faloutsos, C., and Saltz, J. H. (2001). Analysis of the clustering properties of the Hilbert space-filling curve. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):124–141.

[Nguyen, 2013] Nguyen, G. (2013). *Courbes remplissant l'espace et leur application en traitement d'images*. PhD thesis, Université de La Rochelle.

[Owczarek et al., 2019] Owczarek, V., Franco, P., and Mullot, R. (2019). A genetic algorithm to solve a space-filling curve problem. In *SLS2019: International Workshop on Stochastic Local Search Algorithms*, pages 5–6.

[Owczarek et al., 2020] Owczarek, V., Franco, P., and Mullot, R. (2020). Space-filling curve: A robust data mining tool. In Arai, K., Bhatia, R., and Kapoor, S., editors, *Proceedings of the Future Technologies Conference (FTC) 2019*, pages 663–675, Cham. Springer International Publishing.

[Peano, 1890] Peano, G. (1890). Sur une courbe, qui remplit toute une aire plane. In *Mathematische Annalen*, volume 36, pages 157–160. Springer Vienna.

[Pearson, 1901] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Sammon, 1969] Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409.

[Singh and Upadhyaya, 2012] Singh, K. and Upadhyaya, S. (2012). Outlier detection: Applications and techniques. *International Journal of Computer Science Issues*, 9(1):307–323.

[van der Maaten, 2014] van der Maaten, L. (2014). Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245.

[van der Maaten and Hinton, 2008] van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

[Van Der Maaten et al., 2009] Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative review. *J Mach Learn Res*, 10:66–71.

[Vanschoren et al., 2014] Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. (2014). Openml: Networked science in machine learning. *SIGKDD Explor. Newsl.*, 15(2):49–60.

[Wang et al., 2018] Wang, Y., Feng, K., Chu, X., Zhang, J., Fu, C., Sedlmair, M., Yu, X., and Chen, B. (2018). A perception-driven approach to supervised dimensionality reduction for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(5):1828–1840.

[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.

Appendix

For presentation purpose, a focus is made on two dataset (Blob and Tic-Tac-Toe) in Section 5. Additional results are presented in this appendix. Table 7 and Table 8 regroups all results for Sammon Stress, Topology Preservation and Execution time. Moreover Figure 5 is a comparative plot between PCA and SFC on the Sphere Dataset and Figure 6 is between t-SNE and SFC on the Pen digits dataset.

Table 7: Results for the six datasets on two different measures. The projected dimension is $D' = 2$. For each measure, a bold font style indicates the best score.

Dataset	Method	Sammon Stress	Topology Preservation	Execution Time (s)
Concentric sphere (3-D)	PCA	0.0558	0.4624	< 0.01
	MDS	**0.0383**	0.3613	25.57
	t-SNE	0.2278	**0.7848**	10.56
	UMAP	0.2969	0.6257	4
	Hilbert	0.2739	0.3650	< 0.01
Iris Dataset (4-D)	PCA	**0.0063**	0.5844	< 0.01
	MDS	0.0064	0.5611	1.78
	t-SNE	0.1190	**0.6289**	5.10
	UMAP	0.0768	0.5372	2
	Hilbert	0.3212	0.4022	< 0.01
Blob (5-D)	PCA	0.0237	0.1791	< 0.01
	MDS	**0.0159**	0.1948	0.23
	t-SNE	0.0686	**0.5127**	1.49
	UMAP	0.0500	0.3748	4
	Hilbert	0.2579	0.2195	< 0.01
	SFC ($\mathcal{P}^*_{RBG}$)	0.2439	0.2291	< 0.01
Monks Problems 2 (6-D)	PCA	0.0985	0.2949	0.03
	MDS	**0.0872**	0.2896	13.40
	t-SNE	0.1547	**0.3707**	8.25
	UMAP	0.1558	0.2986	3
	Hilbert	0.2172	0.3219	< 0.01
Diabetes (8-D)	PCA	0.0282	0.2646	0.01
	MDS	**0.0150**	0.3433	3.44
	t-SNE	0.2349	**0.5450**	6.34
	UMAP	0.2560	0.4410	3
	Hilbert	0.9277	0.2604	< 0.01
Tic-Tac-Toe (9-D)	PCA	0.1488	0.0328	< 0.01
	MDS	0.1338	0.0538	19.17
	t-SNE	0.1815	**0.2638**	9.91
	UMAP	0.4260	0.1793	3
	Hilbert	**0.1294**	0.1710	< 0.01
	SFC ($\mathcal{P}_A$)	0.1926	0.1709	< 0.01
	SFC ($\mathcal{P}_B$)	0.2207	0.1745	< 0.01
Pendigits (16-D)	PCA	0.0942	0.0542	0.02
	MDS	**0.0730**	0.0789	6570.56
	t-SNE	0.1961	**0.4826**	221.29
	UMAP	0.2144	0.2835	13
	Hilbert	0.2157	0.1606	0.53

Table 8: Results for the six datasets on two different measures. The projected dimension is $D' = 3$. For each measure, a bold font style indicates the best score.

Dataset	Method	Sammon Stress	Topology Preservation	**Execution Time (s)**
Iris Dataset (4-D)	PCA	0.0006	0.7827	< 0.01
	MDS	**0.0004**	**0.7916**	4.14
	t-SNE	0.4483	0.1922	16.31
	UMAP	0.0898	0.4666	4
	Hilbert	0.3686	0.3038	< 0.01
Blob (5-D)	PCA	0.0100	0.3243	< 0.01
	MDS	**0.0064**	0.3476	0.33
	t-SNE	0.2996	**0.4581**	3.41
	UMAP	0.0657	0.4566	3
	Hilbert	0.2406	0.2086	< 0.01
	SFC ($\mathcal{P}^*_{RBG}$)	0.1612	0.2108	< 0.01
Monks Problems 2 (6-D)	PCA	0.0405	0.2950	0.03
	MDS	**0.0342**	0.3316	11.21
	t-SNE	0.0923	**0.3775**	21.10
	UMAP	0.1364	0.3193	3
	Hilbert	0.1534	0.3161	< 0.01
Diabetes (8-D)	PCA	0.0075	0.4391	< 0.01
	MDS	**0.0038**	0.4813	6.34
	t-SNE	0.4403	**0.6231**	20.13
	UMAP	0.2573	0.4771	4
	Hilbert	0.7606	0.2579	< 0.01
Tic-Tac-Toe (9-D)	PCA	0.0832	0.0578	< 0.01
	MDS	**0.0677**	0.0655	17.87
	t-SNE	0.1207	**0.3108**	30.20
	UMAP	0.4084	0.2156	4
	Hilbert	0.1355	0.1562	0.01
	SFC ($\mathcal{P}_A$)	0.1654	0.1529	< 0.01
	SFC ($\mathcal{P}_B$)	0.1606	0.1499	< 0.01
Pendigits (16-D)	PCA	0.0392	0.1806	0.02
	MDS	**0.0271**	0.1926	5856.23
	t-SNE	0.0939	**0.5273**	749.75
	UMAP	0.0748	0.3423	17
	Hilbert	0.1647	0.1574	0.53

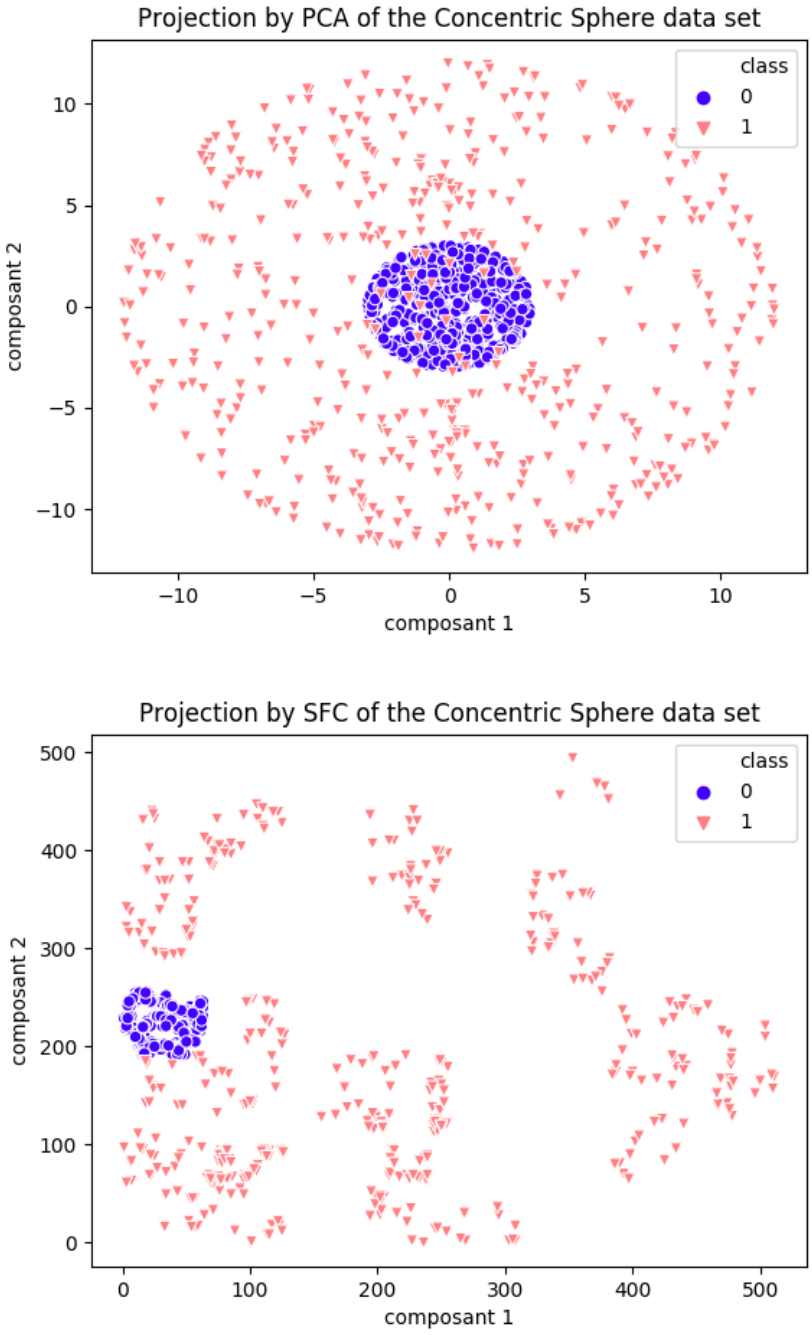

Fig. 5: Example of data projection on the 3-D Sphere Dataset. Where PCA fail to separate the two classes, SFC get a strict separation and yet PCA reached higher theoretical score (cf. Table 7).

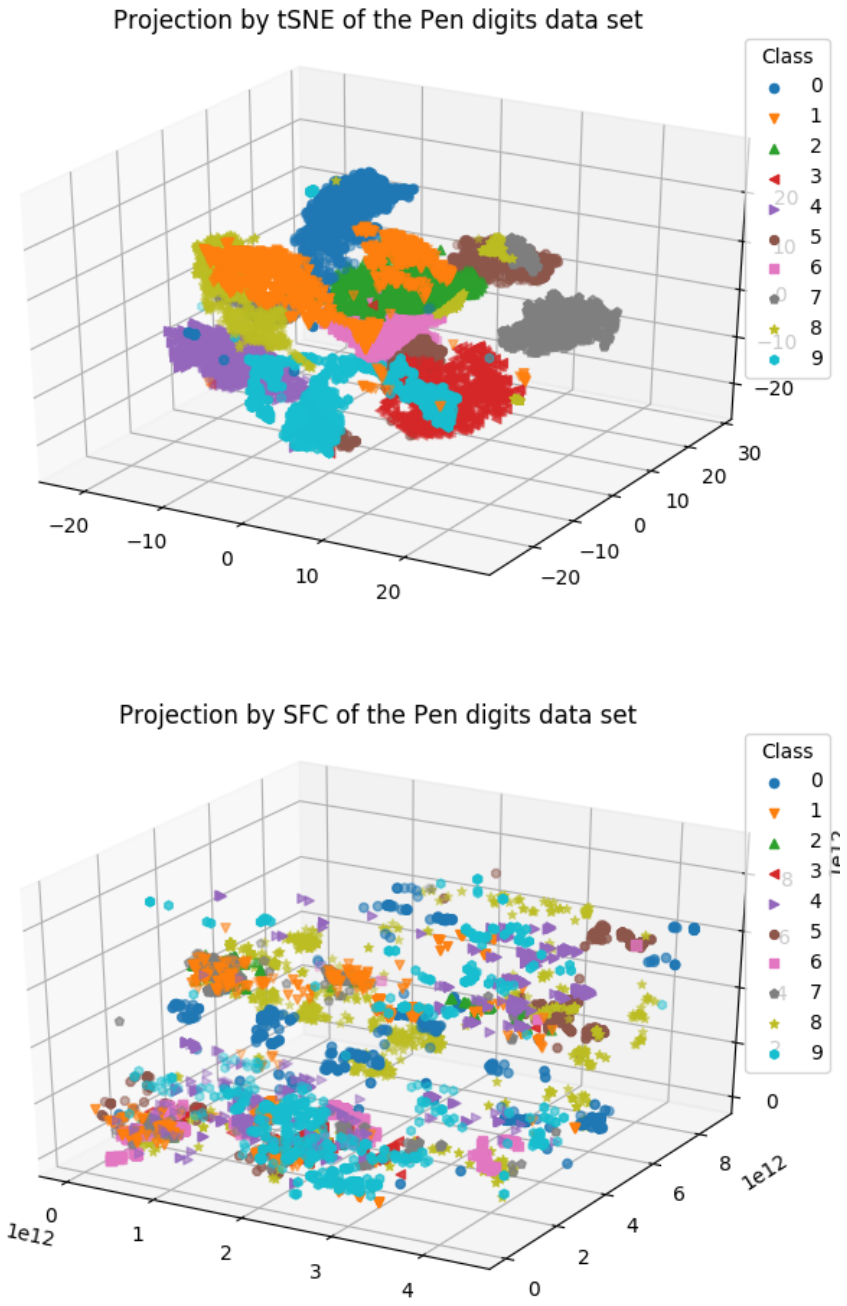

Fig. 6: Example of data projection on the 16-D Pen digits Dataset. As for the Tic-Tac-Toe dataset (cf. Figure 4), the clusters formed by t-SNE are more compact.