IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 12, 2020, accepted December 21, 2020, date of publication December 29, 2020,
date of current version January 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047910

Straightforward Working Principles Behind Modern Data Visualization Approaches

JUGURTA MONTALVÃO[1], LUIZ MIRANDA[1], AND BERNADETTE DORIZZI[2]
[1]Department of Electrical Engineering, Federal University of Sergipe, São Cristóvão 49100-000, Brazil
[2]Samovar, Télécom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France

Corresponding author: Jugurta Montalvão (jmontalvao@ufs.br)

This work supoorted in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The work of Jugurta Montalvão was supported in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under Grant 308319/2018-4.

ABSTRACT From state-of-the-art visualization algorithms, we distill six working principles which are, by hypothesis, sufficient to produce visual projections qualitatively similar to those obtained with these state-of-the-art algorithms. These working principles are presented through the geometrical reasoning of the classical Multidimensional Scaling algorithm, and their effectiveness is illustrated through a novel straightforward algorithm for data visualization. We show, using several datasets originated from various applications, that our algorithm can produce visual projections qualitatively similar to those obtained with these state-of-the-art algorithms. Besides, under the same motivation (of simplification), the problem of visualizing large datasets is tackled through a companion algorithm which is able to embed new input patterns.

INDEX TERMS Data visualization, LargeVis, multidimensional scaling, t-SNE, UMAP.

I. INTRODUCTION

As simply stated by Kruskal [1], in 1964, Multidimensional Scaling (MDS) "is the problem of representing n objects geometrically by n points, so that the inter-point distances correspond in some sense to experimental dissimilarities between objects." It is frequently assumed that the current MDS formulation was first proposed in 1952 by Torgerson [2], although previous works such as the one by M. W. Richardson, published in the Psychological Bulletin, in 1938, suggest that MDS principles predate Torgerson's paper.

Originally used to determine the dimensionality of the stimulus space from similarity analysis between stimuli, MDS quickly became also an important tool for data visualization, for it allows 2D and 3D projection and computational visualization of high dimensional data. More recently, the replacement of dissimilarities in MDS with geodesic distances imposed by weighted graphs, as in the Isometric Feature Mapping (ISOMAP) [3] and similar approaches, renewed the public interest in visualization tools, in a turning point when the flow of high dimensional data was growing through the Internet, in the form of images, sounds and a myriad of behavioral signals easily acquired with hand-held devices such as mobile phones. In that scenario, ISOMAP adapted MDS to compare geodesic distance matrices, which allowed 2D or 3D projection of points lying in possibly curved nonlinear manifolds. In ISOMAP, as in typical MDS formulation, once all pairwise distances (e.g. Euclidean, geodesic, L-metrics, Minkowski, rank-image) are computed, the choice of a convex stress cost function [1] allows the application of the *Classical MDS* efficiently, through eigenvalue decomposition of a double centered distance matrix [4].

Alternatively, whenever the cost function associated to the low dimensional projection problem is not convex, the steepest descent (or gradient) method can be used instead, through computational iterations [1], [5]. That is the same approach used in the visualization algorithm Stochastic Neighbor Embedding (SNE) [6], proposed in 2003, which can be loosely regarded as another steepest descent version of the MDS. This new method was further improved, a few years later, becoming the state-of-the-art t-distributed Stochastic Neighbor Embedding (t-SNE, where the "t" comes from the Student's t-distribution) [7], which quickly gained broad notoriety among data analysts, in part because of its visually attractive results, with many examples of labeled datasets forming self-organized clusters corresponding to

The associate editor coordinating the review of this manuscript and approving it for publication was Rashid Mehmood.

known labels. Moreover, t-SNE was made available in many versions of programming languages, such as Python and Matlab, which possibly further boosted its popularization.

More recently, new visualization algorithms such as the LargeVis, an acronym used in the paper entitled *Visualizing Large-scale and High-dimensional Data* [8], and the Uniform Manifold Approximation and Projection (UMAP) [9] were proposed with evident inspiration in t-SNE, as they approximately follow its recipe. Indeed, besides the graph based reasoning already used in ISOMAP, they also make use of probabilities instead of distances. UMAP further includes some elements of fuzzy models in its theoretical background.

It is noteworthy that much older works based on MDS also included sounding probabilistic background, such as [1], but possibly the principal aspect shared by SNE, t-SNE, LargeVis and UMAP is the joint effect of:

(a) a probabilistic perspective where distances between points are replaced either with conditional probabilities (in SNE, t-SNE and LargeVis), or with probabilistic norms (in UMAP),
(b) and an imposed constraint of (almost) uniform density of projected points.

From this perspective, in this work we claim that these effects can also be obtained with straightforward MDS, under a few changes based on six geometrically explainable working principles shared by t-SNE, LargeVis and UMAP. This claim is corroborated by experimental results obtained with an intentionally simple algorithmic implementation of the six working principles.

To expose these principles and test their effectiveness, this paper is laid out as follows. In Section II the MDS is reformulated in a broad perspective that allows it to connect to state-of-the-art algorithms. In Section III a new algorithm is proposed as a straightforward implementation of six highlighted working principles found in modern algorithms. We provide in Section IV an algorithm for estimating the underlying projection function which allows a straightforward coding of new points, paving the way to a method able to deal with a large amount of data. Both algorithms are tested on various datasets and their results are illustrated in Section V, which are discussed in Section VI.

II. FROM MDS TO STATE-OF-THE-ART VISUALIZATION APPROACHES

As explained in [10], given N objects, $A_1, A_2, \ldots, A_N$, such as "variables, categories, people, social groups, ideas, physical objects, or any other" the MDS analysis of relationships between these objects starts with the computation of pairwise similarities/dissimilarities (e.g. Euclidean distances, correlation coefficients, conditional probabilities or even psychological confusion measures). These pairwise measures, either metric or non-metric [2], are organized in an $N \times N$ matrix $\boldsymbol{P}$, where $P_{i,j}$ is the numeric comparison between objects A_i and A_j. Then a set $\mathcal{Y}$ of N real-valued D_Y-dimensional vectors, $\boldsymbol{y}_i$ $(i = 1, 2, \ldots, N)$ is adjusted in order to numerically reduce the discrepancy between $\boldsymbol{Q}$ and $\boldsymbol{P}$, where $\boldsymbol{Q}$ is another matrix with elements $Q_{i,j}$ representing similarity/dissimilarity between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ (not necessarily the same similarity/dissimilarity used to obtain $P_{i,j}$).

MDS has been used for many years in a myriad of theoretical developments and practical works on data analysis, some of them in data visualization, where the elements of $\mathcal{Y}$ are chosen to be 2D or 3D. In such cases, MDS analysis allows a visual inspection of the relationship among all N objects, as a consequence of the correspondence between the geometrical position of points representing elements of $\mathcal{Y}$ and the measures in matrix $\boldsymbol{P}$.

Reducing the discrepancy between matrices $\boldsymbol{Q}$ and $\boldsymbol{P}$ is an optimization problem where $\boldsymbol{Q}$ is adapted through changes in $\boldsymbol{y}_i$. Under certain constraints, this optimization problem becomes convex and can be efficiently solved as in classical MDS, through eigenvalue decomposition of double centered versions of squared distance matrices. But for the purpose of this work, we prefer to tackle the optimization problem through iterative adaptation of vectors $\boldsymbol{y}_i$, where, in general, a cost function $J(\boldsymbol{P}, \boldsymbol{Q})$ guides the optimization process through its negative gradient, according to (1),

$$\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \alpha \nabla_{\boldsymbol{y}_i} J(\boldsymbol{P}, \boldsymbol{Q}) \tag{1}$$

where α is an arbitrary adaptation step (or learning/adaptation rate), and $\nabla_{\boldsymbol{y}_i} J$ stands for the gradient vector of J with respect to $\boldsymbol{y}_i$. In this work, the iterative formulation of the MDS is referred to as gradient optimized MDS, as opposed to the classical MDS, where optimization is derived by eigenvalue analysis.

In the specific case where objects $\{A_i\}$ are real-valued vectors $\boldsymbol{x}_i$, in $\mathbb{R}^{D_X}$, $D_X \in \mathbb{N}^+$, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are filled with pairwise Euclidean distances between vectors $\boldsymbol{x}$ and vectors $\boldsymbol{y}$, respectively, and

$$J = \sum_{i=1}^{N} \sum_{j=1}^{N} (P_{i,j} - Q_{i,j})^2 = \|\boldsymbol{P} - \boldsymbol{Q}\|_2^2, \tag{2}$$

then $\nabla_{\boldsymbol{y}_i} J(\boldsymbol{P}, \boldsymbol{Q})$ is given as in (3).

$$\nabla_{\boldsymbol{y}_i} J(\boldsymbol{P}, \boldsymbol{Q}) = -4 \sum_{j=1, j\neq i}^{N} \frac{(P_{i,j} - Q_{i,j})}{Q_{i,j}} (\boldsymbol{y}_i - \boldsymbol{y}_j), \tag{3}$$

for $Q_{i,j} \neq 0$.

In words, in each iteration vector $\boldsymbol{y}_i$ is either pushed away or attracted by the j-th vector with strength proportional to $\frac{(P_{i,j}-Q_{i,j})}{Q_{i,j}}$, and the modulus of $\boldsymbol{y}_i - \boldsymbol{y}_j$ has a multiplicative effect on this strength. Alternatively, one may note that

$$\boldsymbol{u}_{i,j} = (\boldsymbol{y}_i - \boldsymbol{y}_j)/Q_{i,j} \tag{4}$$

is a unit vector, therefore vector $\boldsymbol{y}_i$ should move according to a resultant vector, as in (5),

$$\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i + \alpha \underbrace{\sum_{j=1, j\neq i}^{N} W_{i,j} \boldsymbol{u}_{i,j}}_{\text{resultant vector}} \tag{5}$$

where $W_{i,j} = P_{i,j} - Q_{i,j}$ stands for the weight associated to the unit vector $\boldsymbol{u}_{i,j}$.

This vectorial perspective clearly shows that all neighbors of $\boldsymbol{y}_i$ have their influence on the composition of the resultant vector determined by the difference between $P_{i,j}$ and $Q_{i,j}$. For large values of D_X, this causes a well-known problem for visualization (where $D_Y = 2$ or 3), the crowding problem, where many points tend to be projected in the same spot because most weights tend to cluster around similar values. As an illustration, in Fig. 1 we consider Euclidean distances between image patterns from the emblematic MNIST dataset [11]. Each image is coded as a 784D vector (28×28 monochromatic pixels), and we randomly selected 3000 images for this illustration. The first image is taken as $\boldsymbol{x}_i$, and 2999 distances are computed, corresponding to the first row of $\boldsymbol{P}$, whose values are represented in the upper plot in Fig. 1.

Most of the 2999 Euclidean distances are clustered around the interval from 2000 to 3000, and even their minimum is greater than 1200. If corresponding entries in $\boldsymbol{Q}$ are expected to represent distances in 2D or 3D, one should expect the negative gradient to adapt the set $\mathcal{Y}$ towards a configuration where its elements are apart from each other with similar distances, around 2500. However, in 2D or 3D, this goal cannot be satisfied, which eventually induces the crowding of points as a geometric trade-off between tensions. Similar distribution of distances are expected whichever $\boldsymbol{x}_i$ is considered instead of $\boldsymbol{x}_1$, therefore, if (5) is applied to adapt projections $\boldsymbol{y}_i$ in 2D or 3D, weights $W_{i,j}$ are not sufficiently discriminating to yield good visual projections of neighboring influences.

If instead of Euclidean distances, entries in $\boldsymbol{P}$ are replaced with carefully crafted similarity measures, such as the exponential of properly scaled and squared distances, as illustrated in the lower plot in Fig. 1, the crowding of weights can be avoided. For instance, for the MNIST dataset, and again for the first row of $\boldsymbol{P}$, the division of all N distances by 200 yields about 36 non-negligible entries. Unfortunately, because the density of points is rarely the same everywhere, the same scaling factor may not be suitable for all rows of $\boldsymbol{P}$.

The use of Gaussian functions to replace Euclidean distances suggests a probabilistic reasoning where $P_{i,j}$ can be regarded as a conditional probability of picking $\boldsymbol{x}_j$ as the next sample, given that the current sample is $\boldsymbol{x}_i$. This was indeed the probabilistic framework used in [6] by Hinton and Roweis to propose the SNE. Besides, although SNE is not presented as a case of MDS, the application of the following three changes on MDS helps palliating the crowding problem, while it also makes MDS more similar to SNE:

C1 The adaptation of a specific scaling factor for each row of $\boldsymbol{P}$, thus yielding a constant effective number of relevant values per row. In [6] this number is referred to as *Perplexity*.

C2 The normalization of both $\boldsymbol{P}$ and $\boldsymbol{Q}$. More precisely, both constraints $\|\boldsymbol{P}\|_1 = 1$ and $\|\boldsymbol{Q}\|_1 = 1$ are imposed, where $\|\boldsymbol{P}\|_1 = \sum_{i=1}^{N}\sum_{j=1}^{N}|P_{i,j}|$.

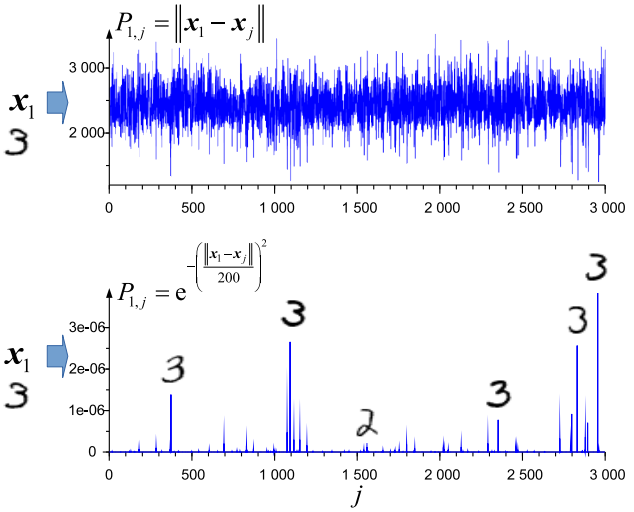

FIGURE 1. Top plot: the concentration of Euclidean distances in high dimension is illustrated with 3000 images randomly drawn from the MNIST dataset (784D). A query image of a handwritten digit '3' is compared to 2999 other images from all classes ('0' to '9'). Images associated to the highest scores are represented in the bottom plot, where the replacement of Euclidean distances with the negative exponential of properly scaled and squared distances avoids the crowding of values, thus highlighting a few near neighbors of the query image.

C3 Change (C2) allows the use of the Kullback–Leibler divergence instead of Euclidean distance as an improved criterion. Indeed,

$$J_{KL} = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{i,j}\log(P_{i,j}/Q_{i,j}), \quad Q_{i,j} \neq 0, \qquad (6)$$

takes into account the restricted matrix manifold where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are to be found.[1]

A further fourth change (C4) was added in 2008 [7] to SNE, when the the Student's t-distribution replaced the Gaussian distribution in the construction of $\boldsymbol{Q}$, whereas the Gaussian remained unchanged for the construction of $\boldsymbol{P}$. This last change yielded the t-SNE, which was shown to reduce even more the crowding effect.

The remarkable success of t-SNE was followed by the proposal of similar approaches, such as LargeVis [8] and UMAP [9]. In LargeVis, the prohibitive practical cost of dealing with $N \times N$ matrices, for large N, was tackled through the use of efficient methods for finding near neighbors, along with the random sampling of far neighbors. As for changes (C1) to (C3), they were also used in LargeVis, although the Kullback-Leibler divergence was replaced with a likelihood function with similar effect. Only C4 was slightly disregarded in LargeVis, as some other probability density functions (PDF) beside t-Student were included.

In UMAP, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are not computed with Gaussian and t-Student distributions, but with a parametrized negative

[1]Constraint $\|\boldsymbol{M}\|_1 = 1$ defines a manifold in $\mathbb{R}^{N\times N}$, where $\boldsymbol{M}$ is an $N \times N$ matrix of positive Real numbers.

exponential of distances, and a double-parametrized generalization of the t-Student distribution, respectively, in a fuzzy set theoretic framework. But in spite of their specificities, most essential elements of t-SNE and LargeVis can find equivalences in UMAP, as succinctly presented in Appendix C of [9]. For instance, UMAP makes use of approximate nearest neighbor search, and stochastic gradient descent with negative sampling for optimization, as in LargeVis, and the cardinality of the fuzzy set of 1-simplices, in fuzzy jargon, plays the same role as the Perplexity parameter, in t-SNE.

In summary, t-SNE, LargeVis and UMAP share the following working principles (WP):

WP1 A limited amount of near neighbors are found for each $\boldsymbol{x}_i$. This corresponds to an underlying presupposition that the neighboring points lie in a *locally continuous manifold*.

WP2 Pairwise distances between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $i, j \in \{1, 2, \ldots, N\}$ are computed. UMAP does not require this distance to be Euclidean but, in most experimental results from all techniques Euclidean distance is used. This suggests an underlying presupposition that the neighboring points lie in a *locally (almost) linear manifold*.

WP3 Pairwise distances are either shrunk or expanded by a local scale factor, σ_i, so that a similarity measure $f_X(\boldsymbol{x}_i, \boldsymbol{x}_j)$ yields negligible values for far neighbors, where $f_X(\cdot)$ is a positive Radial Basis Function with maximum at $f_X(0)$. For instance, in t-SNE $f_X(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{1}{2\sigma_i^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2\right)$. As illustrated in Fig. 1, for the MNIST dataset, $\sigma_1 = 200$ is a local scale factor for $\boldsymbol{x}_1$ that retains only 36 near neighbors with similarities above 1% of the maximum, and 14 near neighbors with similarities above 5%. As expected, according to (C1), for this same scale factor the corresponding Perplexity [6] is about 25, thus in the same range.

This local distance scaling yields non-symmetrical metrics, as illustrated in Fig. 2, where points within two regions with discrepant densities highlight the need for a symmetrization strategy. The detail in Fig. 2 shows that the Euclidean distance $d_{i,j}$ is differently scaled around point $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with scale factors σ_i and σ_j, respectively. This induces a density equalization effect, also illustrated, where the resulting average density is controlled by the Perplexity parameter, in t-SNE.

On the flip side, the symmetry requirement for a metric to be a distance is violated, as $\frac{d_{i,j}}{\sigma_i} \neq \frac{d_{i,j}}{\sigma_j}$, and consequently $p_{j|i} \neq p_{i|j}$, where $p_{j|i} = e^{-\left(\frac{d_{i,j}}{\sigma_i}\right)^2}$ and, $p_{i|j} = e^{-\left(\frac{d_{i,j}}{\sigma_j}\right)^2}$. To enforce symmetry, in t-SNE pairwise similarities are set to $p_{i,j} = \frac{p_{i|j} + p_{j|i}}{2N}$.

The local scaling of distances yields density equalization of uniformly distributed regions of the space, whereas irregularly distributed regions, such as between-clusters gaps and between-densities boundaries cannot be properly handled to yield an equalized density, as illustrated in Fig. 2.

Therefore, beyond the intended dimension reduction, when $D_X > D_Y$, two remarkable effects are observed in t-SNE, namely: density equalization and density-sensitive clustering, as highlighted in Fig. 2. Note that, in this illustration there is no dimension reduction, as $D_X = D_Y = 2$, but although the original dataset has no remarkable gap, the projected data-points have a distinguishable one, d_{proj}, resulting from the projection of cross-densities distances, whereas most points are packed in almost uniform density clusters.

These visually attractive effects can be roughly induced by the *flagging* of K near neighbors (KNN) for each data point (as in WP1), followed by the normalization of the volume occupied by these KNN, thus inducing local space shrinking or expansion. This raw simplification of WP3 is used in Section III.

WP4 Matrix $\boldsymbol{P} = \{p_{i,j}\}$ is filled with symmetrized similarities between locally scaled pairwise distances in the input dataset, $\mathcal{X}$, where similarities are obtained through a given Radial Basis Function (RBF) $f_X : \mathbf{R}^{D_X} \rightarrow \mathbf{R}^+$, and elements of matrix $\boldsymbol{Q}$ are obtained

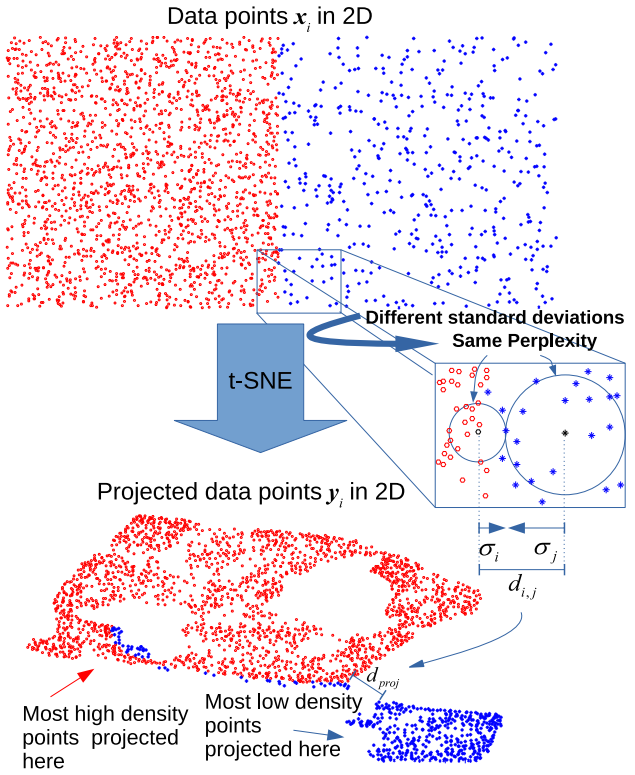

FIGURE 2. Illustration of density-sensitive clustering and density equalization yielded by the t-SNE application to a set of 2D points with two-densities of points. The between-density boundary is projected in a between-cluster gap, whereas cluster densities are equalized. Remark: no dimension reduction in this illustration, for $D_X = D_Y = 2$.

as similarities between instances of the projected low-dimensional dataset, $\mathcal{Y}$, through another RBF $f_Y : \mathbf{R}^{D_Y} \rightarrow \mathbf{R}^{+}$.

WP5 Symmetric matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ are compared, and projected points are adjusted according to rules similar to (5). The specificity of each iteration rule depends on the choice of functions f_X and f_Y, and the criterion J.

WP6 For better visual results, the influence of points too far from each other in the projected space can be damped. This damping effect is presented as the advantage of t-SNE over SNE, as a result of the mismatch between a RBF f_X given by a Gaussian PDF, and another RBF f_Y corresponding to the t-Student PDF. More specifically, Equation 5 in [7] can be rewritten with the notation used in (5) as:

$$\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i + \alpha \underbrace{\sum_{j=1, j\neq i}^{N} W_{i,j} \frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2}{(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2)} \boldsymbol{u}_{i,j}}_{resultant\ \ vector} \tag{7}$$

where $\frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2}{(1+\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2)}$ plays the role of a damping factor for distances $\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2$ either near zero, or much greater than 1.

In Section III, we test the effectiveness of these WP by implementing them as simply as possible, so that if they are indeed the main engines behind state-of-the-art approaches, similar experimental results are expected from our simplified alternative.

III. PROPOSED STRAIGHTFORWARD VISUALIZATION ALGORITHM

The Straightforward Visualization Algorithm (SVA), as presented in Algorithm 1, iteratively adjust the projection of all N vectors in $\mathcal{X}$ into corresponding vectors in $\mathcal{Y}$, thus it looks for a projection $\mathcal{Y} = sva(\mathcal{X}; K, f_Y)$ where parameter K represents the arbitrary number of near neighbors (e.g. $K = 40$) and $f_Y(\cdot)$ is an arbitrary RBF.

In step 1, $N(N-1)/2$ squared Euclidean distances are computed in $\mathbb{R}^{D_x}$, thus requiring D_x scalar multiplications per distance. Likewise, in step 6.1, $N(N-1)/2$ squared Euclidean distances are computed in $\mathbb{R}^{D_y}$, were D_y is set to 2 or 3. Therefore, for a fixed number of iterations, the SVA is $O(N^2\ D_y)$. This is also the case for most state-of-the-art algorithms, and complexity reduction has been addressed since t-SNE was first proposed [7]. Besides, computational burden reduction was the main motivation behind Largevis [8], and although it is beyond the scope of this work, most techniques mentioned there and in references therein are also applicable to SVA.

IV. VISUALIZATION OF NEW DATA

The projection of N given high dimensional data points into 2D or 3D for visualization purposes, through the approaches considered in this work, is a dimension reduction obtained through complicated space contraction/expansion around each of the N observations. This projection has interesting properties in terms of density equalization and clustering, and it can be useful to represent this mapping as a function

Algorithm 1 Visualization $\mathcal{Y} = sva(\mathcal{X}; K, f_Y)$

1. Compute all $N(N-1)/2$ pairwise Euclidean distances between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j, j \neq i$.
2. Find the subset of K near neighbors (KNN) of each $\boldsymbol{x}_i$, and set $P_{i,j} = P_{j,i} = 1$ if j is in this subset. $P_{i,j} = 0$ otherwise. Therefore, each row of $\boldsymbol{P}$ plays the role of a vector of flags, indicating where the KNN are. Note that $\boldsymbol{P}$ remains symmetric, thanks to the simultaneous setting of flags at $P_{i,j}$ and $P_{j,i}$. As a consequence, each row of $\boldsymbol{P}$ may have a few more hotspots (ones) than K. The diagonal of $\boldsymbol{P}$ is kept null, since $\boldsymbol{x}_i$ in not regarded as a neighbor of itself.
3. Normalize matrix $\boldsymbol{P}$ as: $\boldsymbol{P} \leftarrow \frac{\boldsymbol{P}}{\|\boldsymbol{P}\|_1}$. Unlike t-SNE and other approaches based on probabilistic reasoning, this normalization is not mandatory, but it has a suitable consequence in terms of algorithmic convergence, as the matrix space is restricted to a unit norm matrix manifold.
4. Randomly initialize a 2D or 3D set (i.e. D_Y is either 2 or 3) of N Real valued vectors, $\mathcal{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$, typically with very small values.
5. Set a learning rate, α (around N, to compensate for the matrix normalization), a damping radius, R_η, and a damping factor, η. We successfully experimented with values of R_η from 1.5 to 3, and a fixed $\eta = 0.1$.
6. Iterate the following steps until some stopping criterion is reached (in our experiments, we used a maximum number of 2000 iterations as stop criterion, as indicated in Sec. V).
   - 6.1. Find $Q_{i,j} = f_Y(\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2)$.
   - 6.2. Set the diagonal of $\boldsymbol{Q}$ to zero and project it on the same matrix space where $\boldsymbol{P}$ is, through the following attribution: $\boldsymbol{Q} \leftarrow \frac{\boldsymbol{Q}}{\|\boldsymbol{Q}\|_1}$.
   - 6.3. For every pair of vectors in $\mathcal{Y}$, adapt $\boldsymbol{y}_i$ according to:

$$\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \alpha \sum_{j=1, j\neq i}^{N} \beta_j W_{i,j} \boldsymbol{u}_{i,j} \tag{8}$$

   where, as in (5), $W_{i,j} = P_{i,j} - Q_{i,j}$ is the weight associated to the unit vector $\boldsymbol{u}_{i,j} = (\boldsymbol{y}_i - \boldsymbol{y}_j)/\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2$, and either $\beta_j = 1$, for $||\boldsymbol{y}_i - \boldsymbol{y}_j||_2 \leq R_\eta$, or $\beta_j = \eta$, otherwise.
   These iteration steps are similar to (5), apart from the minus sign before α, which reflects the replacement of distances with similarities measures.
7. Return $\mathcal{Y}$.

The SVA has a computational burden dominated by two steps, namely:

- step 1, outside the iteration loop, and
- step 6.1, inside the loop, which is expected to be the most relevant in terms of execution time.

Algorithm 2 Encoder $\boldsymbol{y}_{new} = g(\boldsymbol{x}_{new}; M, \mathcal{X}, \mathcal{Y})$

1. Find a single near neighbor of $\boldsymbol{x}_{new}$, say $\boldsymbol{x}_k$ in $\mathcal{X}$, and take its corresponding $\boldsymbol{y}_k$ in $\mathcal{Y}$.
2. Find M near neighbors of $\boldsymbol{y}_k$ in $\mathcal{Y}$ and project back their correspondences in $\mathcal{X}$.
3. Compute the M distances, $d_i,\ i = 1, 2, \ldots, M$, between $\boldsymbol{x}_{new}$ and the M back-projected vectors selected in Step 2.
4. Compute M weights as $w_i = \exp\left(-\left(\frac{d_i}{(d_{min}+\epsilon)}\right)^2\right)$, where d_{min} is the minimum among the M distances, and ϵ is a small positive Real number which prevents division by zero.
5. Normalize weights: $w_i \leftarrow w_i / \sum_{j=1}^{M} w_j$.
6. Return the projected vector $\boldsymbol{y}_{new} = \sum_{j=1}^{M} w_j \boldsymbol{y}_j$, where $\boldsymbol{y}_j$ stands for the j-th near neighbor found in step 2.

$g_0 : \mathbb{R}^{D_X} \rightarrow \mathbb{R}^{D_Y}$, whose approximation $\boldsymbol{y} = g(\boldsymbol{x})$ can be learned from $\mathcal{X}$ and $\mathcal{Y}$, after SVA reaches its stopping criterion.

To obtain an approximation g as straightforward as SVA, a data-driven piecewise-linear approximation is proposed in Algorithm 2. It is important to highlight that this piecewise-linear projector assumes that, with Euclidean distance, finding M near-neighbors in $\mathcal{Y}$ is more trustful than in $\mathcal{X}$, for elements of $\mathcal{Y}$ are typically represented in much lower dimension than their counterparts in $\mathcal{X}$. Besides, points in $\mathcal{Y}$ tend to be density-equalized, as illustrated in Fig. 2. Therefore, in Algorithm 2, except for step 1, even near neighbours of elements in $\mathcal{X}$ are always found through Euclidean distances between corresponding (projected) elements of $\mathcal{Y}$, which is an originality of this piecewise linear interpolator. All distances in Algorithm 2 are Euclidean.

The motivation for having an approximation of g_0 is two-fold: first it allows the visualization of new incoming data without any projection re-adaptation, thus g plays the role of a data compressor, or an encoder. Besides, because the current version of the SVA is not adapted to large datasets (for naive manipulation of N by N matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ may become prohibitive), Algorithm 2 can also be used to tackle large datasets, by applying SVA to a small subsample of it, and then encoding all remaining data with g.

V. EXPERIMENTAL RESULTS

Most experimental results presented here are visual evidences that, with an adequate choice of the parameters, the SVA, which is a simple implementation of the WP listed in Section II, yields results visually similar to those obtained with t-SNE (which is itself a baseline for LargeVis and UMAP, as presented in [8] and [9], respectively). We start by using two publicly available datasets, MNIST [11] comprising of of 28 × 28 grayscale 10-class handwritten digits, and Fashion-MNIST [12], a more challenging dataset than MNIST, in terms of classification, although it is also

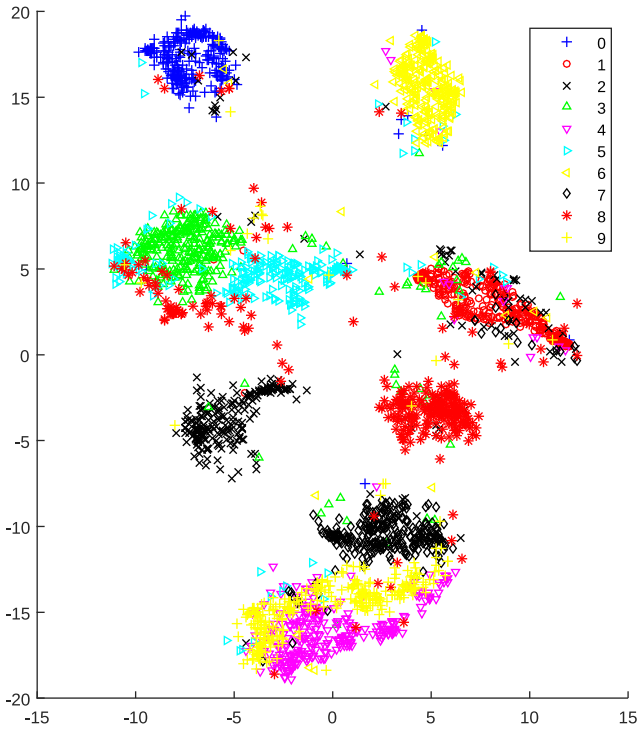

FIGURE 3. Resulting visualization with SVA for $N = 3000$ randomly drawn from the test MNIST dataset, for $K = 40$, $R_\eta = 1.5$, $\eta = 0.1$ and RBF $f_{UMAP}(r) = \frac{1}{1+ar^{2b}}$.

composed of 28 × 28 grayscale images split in 10 classes. In Fashion-MNIST each class corresponds to a fashion product category. Both databases have two non-overlapping subsets, one labeled "training", with 60,000 images, and another labeled "test', with 10,000 images. All experiments with MNIST and Fashion-MNIST used a constant step $\alpha = N$, over 2000 iteration cycles. Of course, elaborated step adaptation strategies would sensibly improve results and avoid numerical instabilities, and should be considered in practical applications of SVA. However, these additional adaptation strategies would mask the similarities between results that we want to highlight in this work.

We experimented with the negative quadratic exponential, $f_{E2}(r) = \exp(-r^2)$, successfully used in the SNE, the inverse quadratic, $f_{T2}(r) = \frac{1}{1+r^2}$, used in t-SNE and LargeVis, and the parametrized RBF $f_{UMAP}(r) = \frac{1}{1+ar^{2b}}$, with $a = 1.929$ and $b = 0.7915$, which is used in UMAP, and can be regarded as a modified version of $f_{T2}(r)$. It is noteworthy that in SVA there is not a probabilistic reasoning behind the choice of f_Y, therefore its choice is not limited to a valid PDF.

As the standard version of t-SNE (see Section 5 of [7]), the current version of SVA is not adapted to large datasets. Therefore $N = 3000$ images were randomly drawn from each test dataset, and the same set, under the same initialization of $\mathcal{Y}$ was used in all experiments in this section, to yield better visual comparison of results. Fig. 3 is to be compared to the 2D t-SNE projection shown in Fig. 4, with perplexity parameter set to 40, whereas Fig. 5 and Fig. 6 are to be

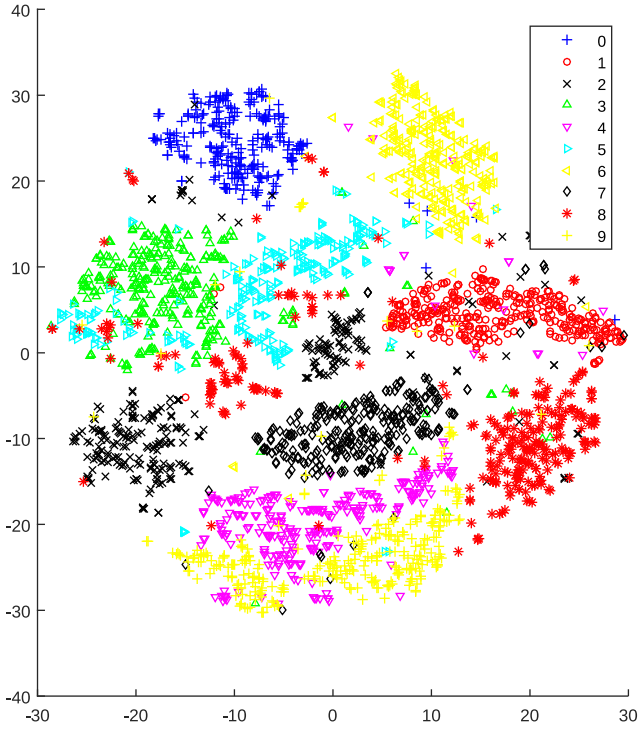

FIGURE 4. Resulting visualization with t-SNE for $N = 3000$ randomly drawn from the test MNIST dataset, under Perplexity parameter set to 40.

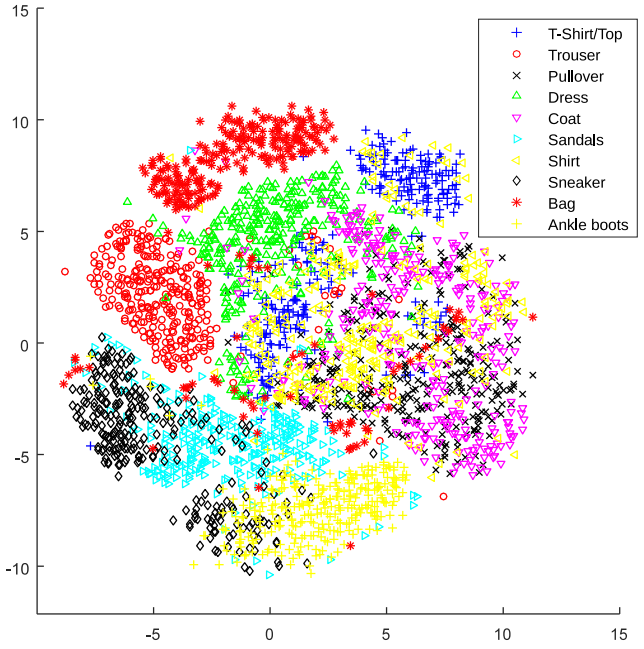

FIGURE 5. Resulting visualization with SVA for $N = 3000$ randomly drawn from the test Fashion-MNIST dataset,, for $K = 40$, $R_\eta = 1.5$, $\eta = 0.1$ and RBF $f_{E2}(r) = \exp(-r^2)$.

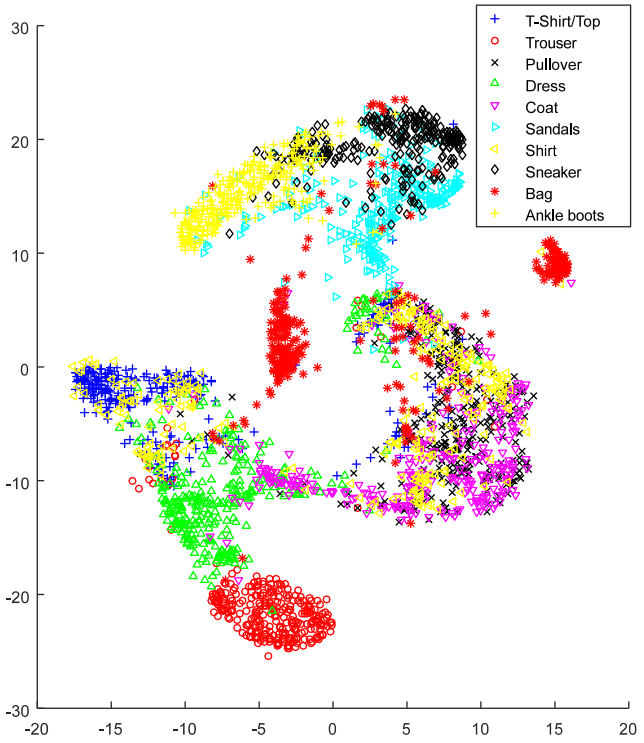

FIGURE 6. Resulting visualization with SVA for $N = 3000$ randomly drawn from the test Fashion-MNIST dataset,, for $K = 40$, $R_\eta = 1.5$, $\eta = 0.1$ and RBF $f_{T2}(r) = \frac{1}{1+r^2}$.

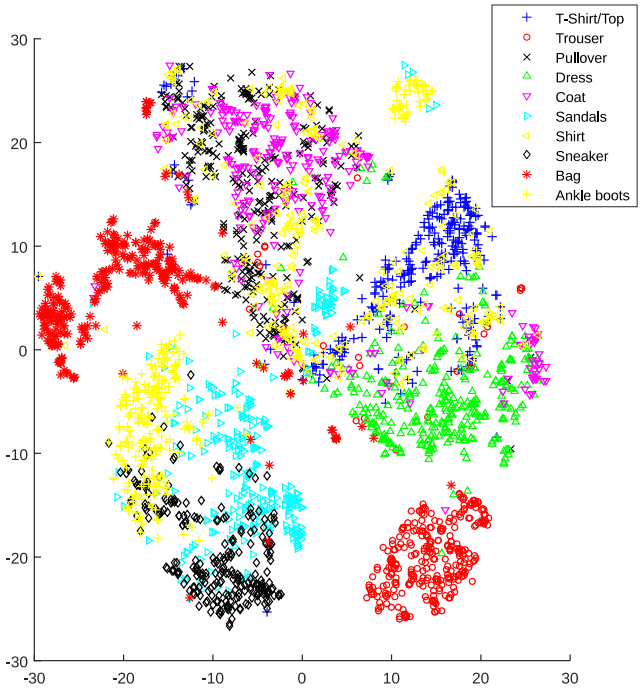

FIGURE 7. Resulting visualization with t-SNE for $N = 3000$ randomly drawn from the test Fashion-MNIST dataset, under Perplexity parameter set to 40.

compared to the 2D t-SNE projection shown in Fig. 7, also with perplexity parameter set to 40.

Regarding Algorithm 2, Fig. 8 illustrates its use, where just 3000 images sampled from the "test" MNIST dataset, along with their projections were used as parameters $\mathcal{X}$ and $\mathcal{Y}$ of g. Then, all 60,000 new images from the "training" dataset were projected (without further adaptation of the visualization projection).

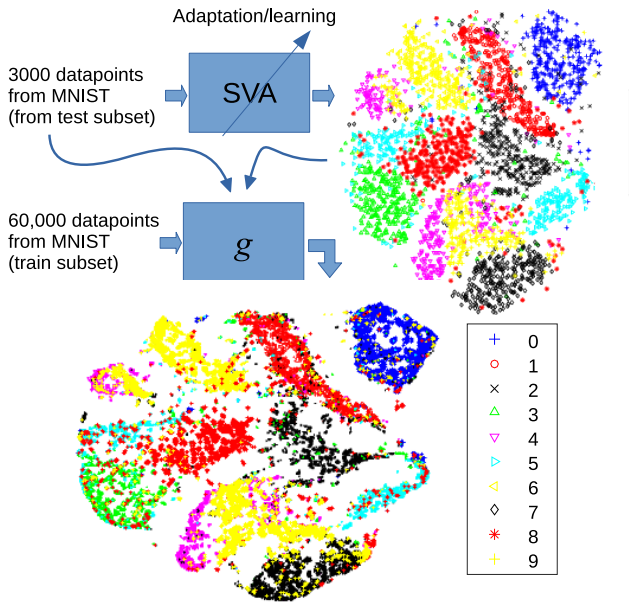

FIGURE 8. Visualization of new data after SVA was adapted to 3000 images from the "test" MNIST dataset, with $K = 40$, $R_\eta = 1.5$, $\eta = 0.1$ and RBF $f_{E2}(r) = \exp(-r^2)$, yielding $\mathcal{X}$ and $\mathcal{Y}$. Then the encoder g is applied to the 60,000 new images from the "train" MNIST dataset, with $M = 40$.

The experimental results presented in this paper were chosen as visually representative of experiments done so far. Some more results, along with suggestions of implementations of the SVA and the encoder g in some usual computer languages can be found as supplemental material posted on IEEE Xplore. https://ieee-dataport.org/documents/supplementary-material-paper-straightforward-working-principles-behind-modern-data

Visual evaluation is obviously the most usual approach for comparisons between visualization algorithms, insofar as visualization experiments are typically concerned with subjective (visual) aspects of classes and clusters dispersion, hardly replaced with any objective index. One may even conjecture that visualization algorithms are popular because there is not yet an objective index capable of replacing human cognition.

Nevertheless, by considering that the goal of all visualization algorithms considered in this work (including the SVA) is to preserve as much as possible the local neighboring structure of points before and after projection, and knowing that it can be a very difficult goal for points lying in manifolds with local dimensions much higher than 2 or 3, we crafted a simple index, namely, the Near-Neighbors Coincidence Rate (NNCR), which is computed as in (9).

$$C(\mathcal{X}, \mathcal{Y}; V) = \frac{1}{N \times V} \sum_{n=1}^{N} |\mathcal{X}_n \cap \mathcal{Y}_n| \tag{9}$$

where V is an index parameter representing the number of near neighbors to be considered, $\mathcal{X}_n$ stands for a subset of $\mathcal{X}$ whose elements are the V near neighbors of $\boldsymbol{x}_n$. Likewise, $\mathcal{Y}_n$ stands for a subset of $\mathcal{Y}$ whose elements are the V near neighbors of $\boldsymbol{y}_n$, and $|\mathcal{X}_n \cap \mathcal{Y}_n|$ is the cardinality of the intersection set. Thus, if most V near neighbors of each point are preserved after projection of $\mathcal{X}$ into $\mathcal{Y}$, C is expected to yield values close to one. By contrast, near zero values of C indicate disruption of local neighboring structures.

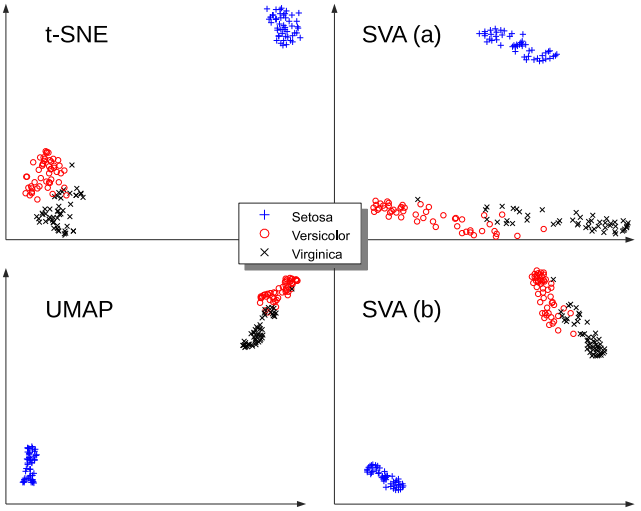

FIGURE 9. One projection per algorithm of the 150 4D vectors of the *Iris* dataset. Algorithms are indicated in the subplots.

To test this new measure, we consider four public datasets whose points represent very diverse signaling phenomena. We used the following constant parametrization of methods across all next experiments, to allow better comparisons of results:

- t-SNE: Perplexity = 30.
- UMAP: $n = 30$ (the number of neighbors), $min - dist = 0.1$ (desired separation between close points in the embedding space) and $n - epochs = 200$ (number of training epochs to use when optimizing the low dimensional representation).
- SVA(a): $K = 30$, $R_\eta = 3$, $\eta = 0.1$ and $f_{T2}(r) = \frac{1}{1+r^2}$.
- SVA(b): $K = 30$, $R_\eta = 3$, $\eta = 0.1$ and RBF $f_{UMAP}(r) = \frac{1}{1+ar^{2b}}$.

In all experiments, points were projected into $\mathbb{R}^2$, and the number of iteration for t-SNE, SVA(a) and SVA(b) was set to 1000.

Fig. 9 illustrates one projection with each algorithm of the emblematic *Iris* dataset used in [13], with 150 4D vectors numerically representing sepal and petal measurements (length and width) for flowers from 3 species, namely: Iris setosa, Iris versicolor and Iris virginica. Thus, data points were labeled here with *Setosa*, *Versicolor* or *Virginica*.

For the second set of comparative experiments, we took the recently published dataset explained in [14], here referred to as *Meat volatiles*, where an array of 10 sensors (8 Metal Oxide gas sensors plus temperature and humidity sensors), i.e. an e-nose was used to acquire multivariate signals through time from 7 controlled mixtures of beef and pork, (always

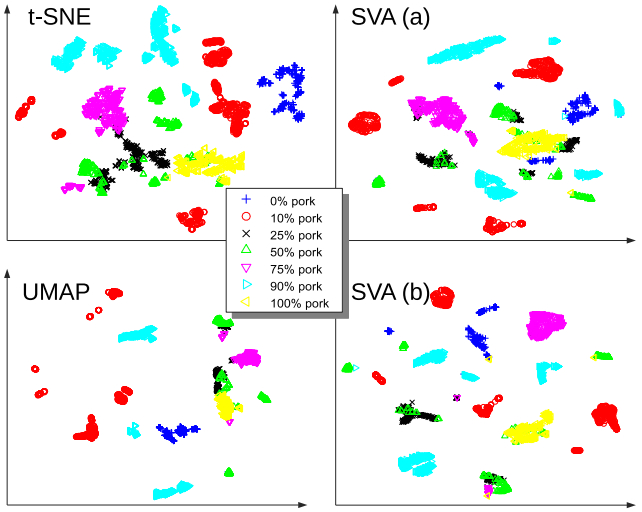

FIGURE 10. One projection per algorithm of the 3000 randomly drawn 10D vectors of the *Meat volatiles* dataset. Point labels correspond to proportions of pork in 100 g of mixed meet (pork and beef). Algorithms are indicated in the subplots.

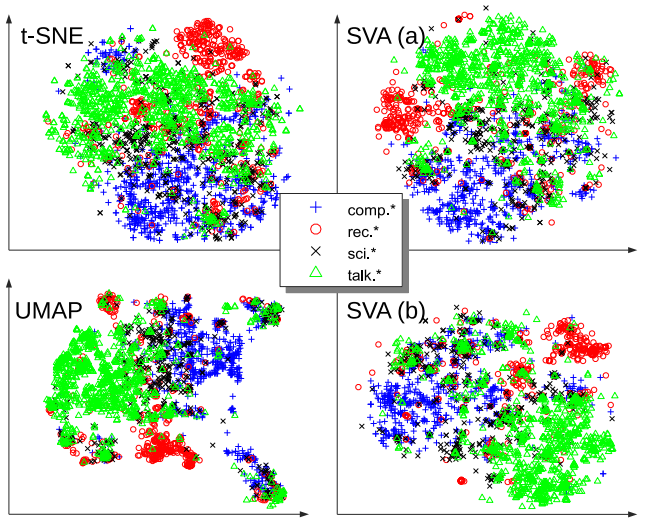

FIGURE 11. One projection per algorithm of the 3000 randomly drawn 100D binary vectors of the *Newsgroups* dataset. Algorithms are indicated in the subplots.

100 g of fresh ground meat per acquisitions sessions of 120 s). In this dataset, only 60 instances of each mixture are available, but each instance corresponds to a sequence of 60 10D measurements vectors taken every 2 seconds. Thus, from this dataset we randomly drawn 3000 measurement vectors and projected them from 10D into 2D, as illustrated in Fig. 10.

A fine analysis of each dataset is beyond the scope of this paper, where the comparison between 2D projections across methods is the main concern. However, it is worth noticing that all projections in Fig. 10 seem to present the same inconsistency, namely: that similar proportions of beef and pork are not projected in near clusters. As for this matter, one should be aware that, for e-noses based on Metal Oxide sensors, robust feature extraction from raw signals is yet a relevant research subject. In any case, in spite of these apparent raw signal inconsistencies, all four projection in Fig. 10 are in agreement with each other.

Fig. 11 illustrates one projection per algorithm for the dataset here referret to as *Newsgroups*,[2] where each of 16242 postings (texts in natural language) is encoded as a 100D binary vector, where the occurrence for 100 relevant words (e.g. cancer, baseball, car, children) is flagged with 1. Thus, each binary vector is labeled with *comp.**, *rec.**, *sci.** or *talk.**, corresponding to the group name in which it was posted. For the projections presented in Fig. 11, 3000 vectors were randomly drawn along with their corresponding labels.

Finally, a public dataset of male and female clean-speech utterances in Brazilian Portuguese [15], was used to yield 19 Mel-Frequency Cepstral Coefficients (MFCC) per short speech frames of 25 ms, taken from all utterances, from

[2]Also labeled as "20 Newsgroups," and publicly available at https://cs.nyu.edu/roweis/data.html.

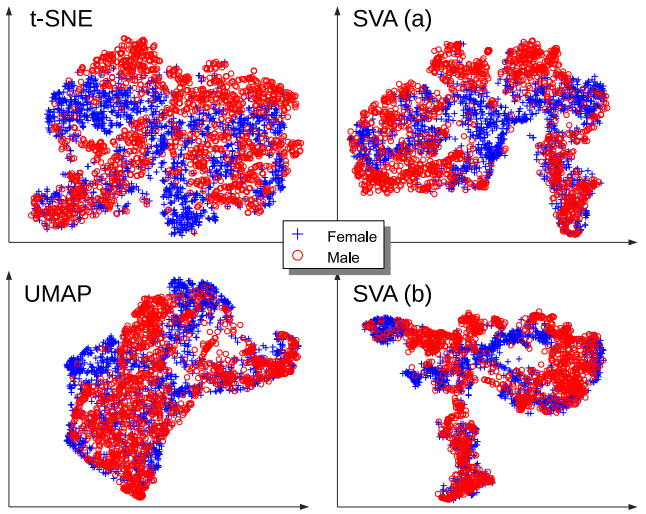

FIGURE 12. One projection per algorithm of the 3000 randomly drawn 18D MFCC vectors of the *Speaker gender* dataset. Algorithms are indicated in the subplots, and point labels correspond to the gender of the speaker who uttered it.

all speakers. The first element of each MFCC vector was systematically discarded (for it does not carry relevant acoustic information), thus yielding around 200,000 18D MFCC vectors, from which only 3000 were randomly drawn and labeled according to the corresponding speaker gender. In this paper, this dataset is referred to as *Speaker gender*, and its projection in 2D is presented in Fig. 12.

The NNCR for all 16 projections (4 per dataset) are gathered in Table 1. As compared to the corresponding visual aspects, the NNCR seems to yield meaningful comparative values. For instance, for the datasets *Iris* and *Meat volatile*, most clusters concentrate unmixed classes, and NNCR values above 0.7 confirm that, on average, more than 70 % of the local structures in the corresponding original datasets were

TABLE 1. Quantitative comparisons.

Dataset	Method	Near-neighbors coincidence rate
Iris specie	t-SNE	0.85
	SVA(a)	0.82
	UMAP	0.82
	SVA(b)	0.82
Meat volatiles	t-SNE	0.76
	SVA(a)	0.74
	UMAP	0.72
	SVA(b)	0.72
Newsgroups	t-SNE	0.33
	SVA(a)	0.33
	UMAP	0.30
	SVA(b)	0.32
Speaker gender	t-SNE	0.40
	SVA(a)	0.43
	UMAP	0.35
	SVA(b)	0.41

preserved. This further suggests that the corresponding underlying manifolds in *Iris* and *Meat volatile* datasets are more easily projected into 2D than the ones in the *Newsgroup* and the *Speaker gender* datasets, where only less than 45% of the local neighboring structures were preserved.

From the standing point proposed in this paper, what is perhaps more important than measuring the difficulty of keeping local neighboring structures, is to notice that, for each dataset, the NNCR also yields an objective index for comparing algorithm projections. Indeed, as much as the visual qualitative comparisons, this quantitative measure seems to confirm that t-SNE, UMAP and the two versions of SVA are almost equivalent in projecting high-dimensional data into 2D.

VI. CONCLUSION

State-of-the-art visualization approaches are based on elaborated probabilistic and fuzzy models. By contrast, in this work we assume that a few simple working principles would be sufficient to yield similar results, which was corroborated by experiments done with an algorithm where these principles were straightforwardly implemented. This algorithm was applied to several public datasets corresponding to various application domains.

The proposed reduction to simple principles has a first useful aspect in terms of potential boosting of new developments in visualization tools, because the simplicity of the six listed working principles allows for new contributions and improvements from researchers with a broad range of different backgrounds. For instance, we observed that the choice of an RBF, f_Y, is not restrained to probability distributions, and the knowledge of the exact cost function (and its gradient) is not imperative for a visualization result qualitatively similar to that yielded with t-SNE. These simplifications allow the experimentation with a virtually unlimited set of RBF, that can be heuristically selected and easily tested for suitable (subjective) visualization effects, without the need for (potentially laborious) algebraic manipulations of gradient cost function.

Besides, the replacement of the Perplexity, in t-SNE, with a simpler parameter K representing a fixed number of near neighbors also yielded a simple piecewise linear encoder, which was used for projection of new incoming observations, after a visualization projection was adjusted. This is a useful companion algorithm for SVA, for it allows the visualization of an unlimited amount of data, whereas SVA itself is kept simple, in terms of implementation. Indeed, the usual representation of N by N matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ is a limiting aspect that can be tackled, for instance, with efficient KNN graph construction (see [8] and references therein). On the other hand, the approach to solve the same problem implemented in this work was to split the task into two parts, namely: first a small subsample of N points is projected with SVA (e.g. $N = 3000$), then the encoder g takes the N projected points as parameters and is ready to project any amount of new incoming data. We believe that this choice is algorithmically simpler than modifications on the SVA to cope with large datasets. Moreover it allows for applications beyond data visualization, in projected dimensions higher than 3D, where SVA and g can be jointly used to yield auto-encoding structures, as a matter for the follow-up of this work.

REFERENCES

[1] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, Mar. 1964.

[2] A. Mead, "Review of the development of multidimensional scaling methods," *J. Roy. Stat. Soc., D (Statistician)*, vol. 41, no. 1, pp. 27–39, 1992.

[3] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[4] R. S. Bennett, "Representation and analysis of signals part XXI. The intrinsic dimensionality of signal collections," Dept. Elect. Comput. Eng., Johns Hopkins Univ. Baltimore, MD, USA, Tech. Rep. TR/163, 1965.

[5] J. B. Kruskal, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, vol. 29, no. 2, pp. 115–129, Jun. 1964.

[6] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 857–864.

[7] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[8] J. Tang, J. Liu, M. Zhang, and Q. Mei, "Visualizing large-scale and high-dimensional data," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 287–297.

[9] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*. [Online]. Available: http://arxiv.org/abs/1802.03426

[10] L. Guttman, "A general nonmetric technique for finding the smallest coordinate space for a configuration of points," *Psychometrika*, vol. 33, no. 4, pp. 469–506, Dec. 1968.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[12] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*. [Online]. Available: http://arxiv.org/abs/1708.07747

[13] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annu. Eugenics*, vol. 7, pp. 179–188, 1936.

[14] R. Sarno, S. I. Sabilla, D. R. Wijaya, D. Sunaryono, and C. Fatichah, "Electronic nose dataset for pork adulteration in beef," *Data Brief*, vol. 32, Oct. 2020, Art. no. 106139.

[15] C. A. Ynoguti and F. Violaro, "A Brazilian Portuguese speech database," in *Proc. 26th Simpósio Brasileiro de telecomuniçacões*, 2008, pp. 1–4.

JUGURTA MONTALVÃO was born in Aracaju, Brazil, in 1968. He received the degree in electrical engineering from the University of Campina Grande (UFPB II), in 1992, the master's degree in electrical engineering from the University of Campinas (UNICAMP), in 1995, and the doctor's degree in automatique et traitement du signal from University Paris-Sud XI, in 2000. He joined the Department of Electrical Engineering, Federal University of Sergipe (UFS), in 2005. His research interests include pattern recognition and signal processing, whereas his applied research is mostly concerned with behavioral biometrics.

LUIZ MIRANDA was born in Recife, Brazil, in 1991. He received the degree in electrical engineering and the M.Sc. degree from the Federal University of Sergipe (UFS), in 2014 and 2017, respectively. His research interests include pattern recognition and signal processing.

BERNADETTE DORIZZI received the Ph.D. (Thèse d'état) degree in theoretical physics from the University of Orsay (Paris XI-France), in 1983, with a focus on integrability of dynamical systems. She led the Electronics and Physics Department from 1995 to 2009. She has been a Professor with Telecom SudParis (ex INT) since 1989, and the Dean of Research, since 2013. She has coordinated the Biosecure Network of Excellence, and is currently the Chairwoman of the Biosecure Foundation. She is in charge of the Intermedia (Interaction for Multimedia) Research Team. She has authored over 300 research articles and has supervised over 20 Ph.D. thesis. Her research interests include pattern recognition and machine learning applied to activity detection, surveillance-video, and biometrics.