Steering Distortions to Preserve Classes and Neighbors in Supervised Dimensionality Reduction

Benoît Colange
Univ. Grenoble Alpes, INES,
F-73375, Le Bourget du Lac, France
benoit.colange@cea.fr

Jaakko Peltonen
Tampere University, Faculty of Information Technology
and Communication Sciences, Finland
jaakko.peltonen@tuni.fi

Michaël Aupetit
Qatar Computing Research Institute,
Hamad bin Khalifa University, Doha, Qatar
maupetit@hbku.edu.qa

Denys Dutykh
Univ. Grenoble Alpes, Univ. Savoie Mont Blanc,
CNRS, LAMA, 73000 Chambéry, France
denys.dutykh@univ-smb.fr

Sylvain Lespinats
Univ. Grenoble Alpes, INES,
F-73375, Le Bourget du Lac France
sylvain.lespinats@cea.fr

Abstract

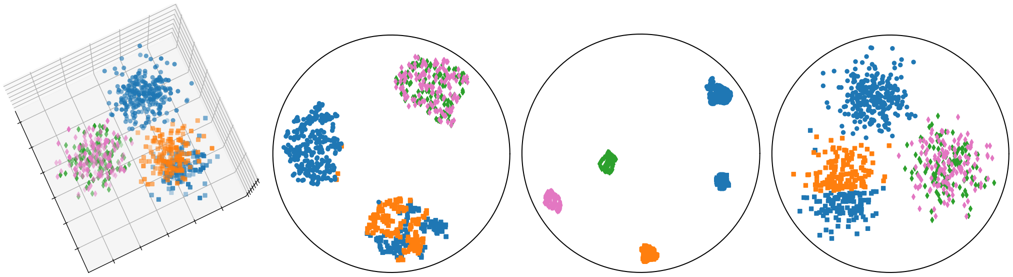
Nonlinear dimensionality reduction of high-dimensional data is challenging as the low-dimensional embedding will necessarily contain distortions, and it can be hard to determine which distortions are the most important to avoid. When annotation of data into known relevant classes is available, it can be used to guide the embedding to avoid distortions that worsen class separation. The supervised mapping method introduced in the present paper, called ClassNeRV, proposes an original stress function that takes class annotation into account and evaluates embedding quality both in terms of false neighbors and missed neighbors. ClassNeRV shares the theoretical framework of a family of methods descending from Stochastic Neighbor Embedding (SNE). Our approach has a key advantage over previous ones: in the literature supervised methods often emphasize class separation at the price of distorting the data neighbors' structure; conversely, unsupervised methods provide better preservation of structure at the price of often mixing classes. Experiments show that ClassNeRV can preserve both neighbor structure and class separation, outperforming nine state of the art alternatives.

1 Introduction

Dimensionality Reduction (DR) methods aim at mapping a high dimensional dataset as points in a lower dimensional embedding space, while preserving some similarity measure between data points. DR may be supervised by taking advantage of class information. Hence, supervised methods compute the mapping both from relative positions of data (also used by non-supervised methods) and from the class labels. DR techniques [1, 2, 3] can be used as a pre-processing step for classification or clustering, or to visualize (labeled) data as a scatterplot. When mapping labeled data, there are two contradictory objectives:

- *Classification* is typical of *supervised* DR techniques: class separation is emphasized and measured with classification accuracy in the embedding space.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.



(a) 3D dataset (b) tSNE (unsupervised) (c) S-UMAP (supervised) (d) ClassNeRV

Figure 1: ClassNeRV is designed to preserve both classes and neighbors: data are sampled from three 3D Gaussian clusters (a), a cluster of circles purely of the blue class, a cluster of squares halved by a plane separating blue and orange classes, and a cluster of diamonds with randomly distributed green and pink classes. Different planar embeddings of these data are shown: the unsupervised tSNE (b) preserves clusters well, but overlaps the orange and blue classes in the cluster of squares as it ignores the labels. The supervised S-UMAP (c) splits the clusters of squares and diamonds forcing class separation. Hence, it is misleading about the original spatial adjacency of orange and blue classes, and mixing of green and pink classes. ClassNeRV (d) is designed to better preserve the three-clusters structure as well as classes' adjacency.

- *Exploratory data analysis* is typical of *unsupervised* DR techniques which operate without knowledge of class information: data neighborhood structure is prioritized and measured as a discrepancy between data similarities in both original and embedding spaces.

These objectives derive from visual analytic tasks [3, 4]. They are contradictory unless class and data neighborhood structures match each other well in both the data and embedding spaces: each class constitutes distinct areas with no cross-class neighborhood relations. Unfortunately, this ideal case is very unlikely because the data neighborhood structure and classes do not always match in the data space, and low dimensional embeddings of high-dimensional data come with unavoidable distortions [3]: *false neighbors* which are neighboring points in the embedding but not in the data, and *missed neighbors* which are neighbors in the data but not in the embedding.

In this work, we propose ClassNeRV, a supervised DR technique to accomplish the exploratory analysis objective while taking class information into account. Our solution is similar in principle to the earlier ClassiMap [5] distance-based projection, but we derive our approach from the same well-grounded probabilistic framework as NeRV [6, 7], SNE [8], and tSNE [9]. Our solution differs notably from other previous supervised methods which tend to force class separation at the expense of neighbors' structure, *e.g.* S-Isomap [10], S-NeRV [11] and S-UMAP [12]. Figure 1 illustrates the essential characteristics of ClassNeRV.

Our contribution is two-fold: we propose **ClassNeRV** which utilizes class information to ensure a better preservation of classes when embedding high dimensional labeled data into a low dimensional space. Its stress function, derived from the unsupervised NeRV [6, 7], steers the optimization so that the unavoidable distortions of the neighborhood structure are placed where they are less harmful to the class structure. Harmful distortions are avoided by emphasizing penalization of false neighbors between classes and missed neighbors within classes. We also derive two new **class-aware quality indicators** from the standard *Trustworthiness* and *Continuity* quality indicators [13], to account specifically for the distortions affecting class preservation.

2 Related Work

Unsupervised Embeddings. Many linear or non-linear algorithms have been previously proposed including Principal Component Analysis (PCA) [14], Self Organizing Maps (SOM) [15], isometric feature mapping (Isomap) [16], Data-Driven High Dimensional Scaling (DD-HDS) [17], Local Affine Multidimensional Projection (LAMP) [18] and Uniform Manifold Approximation and Projection (UMAP) [12]. Among the wide variety of techniques, Neighborhood Embedding (NE) techniques are efficient at preserving neighborhood structures and for computing time. Their probabilistic framework also provides a theoretical background to interpret the obtained maps in terms of a neighbourhood retrieval task [7]. NE methods compute, for each pair of points i, j, the probabilistic membership of point j to the neighborhood of point i, sometimes called similarity. These membership degrees are computed both in the data space and the embedding space. The mapping is obtained by minimizing the discrepancy of membership probabilities between these two spaces. These methods include Stochastic Neighbor Embedding (SNE) [8], t-distributed SNE (t-SNE) [9], Jensen Shannon Embedding (JSE) [19] and Neighborhood Retrieval Visualizer (NeRV) [6, 7]. SNE and t-SNE differ by the kernel used to compute their neighborhood membership degrees in the embedding space. JSE and NeRV both extend SNE to control the balance between false and missed neighbors. This tunability of NeRV and JSE makes them the best-suited for introducing supervision.

Supervised Embeddings. Supervised mapping methods mainly focus on *class separation*, modifying the data neighbor structures before or during the mapping. Many of them typically decrease data similarities between classes and increase them within classes, then use a standard unsupervised mapping method on the modified similarities. They may operate either on dissimilarities (distances), using metric learning, as in Supervised Locally Linear Embedding SLLE and its variants [20, 21, 22], Supervised Isomap (S-Isomap) [10, 23], Supervised NeRV (S-NeRV) [11] or Supervised UMAP S-UMAP [12], or on similarities as in the semi-supervised version of heavy-tailed tSNE [24]. Other methods rely on a global parametric mapping and optimize its parameters to *maximize class separation*, as Linear Discriminant Analysis (LDA) [25] and its kernelized variants [26, 27, 28], Neighbourhood Component Analysis (NCA) [29] and its neural-network variant [30] or Limited Rank Matrix Learning Vector Quantization (LiRaM LVQ) [31]. Class-aware tSNE (catSNE) [32] locally adapts the size of the neighbourhood to preserve based on the distribution of classes. At last, ClassiMap [5] optimizes a stress function similar to the one of Local Multidimensional Scaling (LMDS) [33], but supports the *exploratory analysis of labeled data* by taking classes into account when penalizing false and missed neighbors. However, both LMDS and ClassiMap being distance-based are sensitive to the norm concentration phenomenon in high dimensions [34] while NE techniques like NeRV mitigate this effect using *shift-invariant* membership degrees [35]. ClassNeRV derives the same principles as ClassiMap from the NeRV stress function to get the benefits of both.

3 ClassNeRV and Class-aware Quality Indicators

3.1 NeRV Stress Function

In NeRV [6, 7], the degree of membership (conditional probability) of a point j to the neighborhood of another point i, denoted β_{ij} in the data space and b_{ij} in the embedding space, is defined as:

$$\beta_{ij} \triangleq \frac{\exp\left(-\Delta_{ij}^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\Delta_{ik}^2/2\sigma_i^2\right)} \quad \text{and} \quad b_{ij} \triangleq \frac{\exp\left(-D_{ij}^2/2s_i^2\right)}{\sum_{k\neq i}\exp\left(-D_{ik}^2/2s_i^2\right)} \tag{1}$$

with Δ_{ij} and D_{ij} the distances between points i and j in data and embedding spaces respectively. The distributions of membership degrees are denoted as $\beta_i \triangleq \{\beta_{ij}\}_{j\neq i}$ and $b_i \triangleq \{b_{ij}\}_{j\neq i}$. The membership degrees are shift-invariant, reducing norm concentration [35]. Scale parameters σ_i are set to get a fixed user-chosen perplexity p akin to a smooth or fuzzy measure of the number of neighbors of each point [36]: $H(\beta_i) = \log p$, with the entropy $H(\beta_i) \triangleq -\sum_{j\neq i}\beta_{ij}\log\beta_{ij}$. Here, we set embedding scale parameters s_i equal to σ_i.

The NeRV stress function is a linear trade-off between two sets of Kullback–Leibler (KL) divergences:

$$\zeta_{\text{NeRV}} \triangleq \sum_i \tau\mathcal{D}_{\text{KL}}(\beta_i, b_i) + (1-\tau)\mathcal{D}_{\text{KL}}(b_i, \beta_i) \triangleq \tau\sum_{i,j\neq i}\beta_{ij}\log\left(\frac{\beta_{ij}}{b_{ij}}\right) + (1-\tau)\sum_{i,j\neq i}b_{ij}\log\left(\frac{b_{ij}}{\beta_{ij}}\right) \tag{2}$$

In Equation (2), $\tau \in [0;1]$ controls the trade-off between $\sum_i \mathcal{D}_{\text{KL}}(\beta_i, b_i)$ penalizing *missed neighbors* (maximally when $\tau = 1$) and $\sum_i \mathcal{D}_{\text{KL}}(b_i, \beta_i)$ penalizing *false neighbors* (maximally when $\tau = 0$). When $\tau = 1$, NeRV reduces to SNE.

3.2 ClassNeRV Stress Function

We aim for an embedding that preserves both neighborhood structures and classes. We propose that the embedding should not *artificially separate points within the same class*, or *artificially cluster* *together points from different classes*. Hence, we need to penalize more *within-class missed neighbors* and *between-class false neighbors* respectively. To control these class-based distortions, we further split the divergence terms in Equation (2) into *within-class* and *between-class* relations. It gives two additional distinct class-based trade-off parameters $\tau^{\in}$ and $\tau^{\notin}$ (both in $[0,1]$) and leads to the ClassNeRV stress function:

$$\begin{aligned}\zeta_{\text{ClassNeRV}} &\triangleq \sum_i \tau^{\in}\mathcal{D}_{\text{B}}(\beta_i^{\in}, b_i^{\in}) + (1-\tau^{\in})\mathcal{D}_{\text{B}}(b_i^{\in}, \beta_i^{\in}) + \tau^{\notin}\mathcal{D}_{\text{B}}(\beta_i^{\notin}, b_i^{\notin}) + (1-\tau^{\notin})\mathcal{D}_{\text{B}}(b_i^{\notin}, \beta_i^{\notin}) \\ &\triangleq \tau^{\in}\sum_{i,j\in\mathcal{S}_i^{\in}}\left(\beta_{ij}\log\left(\frac{\beta_{ij}}{b_{ij}}\right)+b_{ij}-\beta_{ij}\right) + (1-\tau^{\in})\sum_{i,j\in\mathcal{S}_i^{\in}}\left(b_{ij}\log\left(\frac{b_{ij}}{\beta_{ij}}\right)+\beta_{ij}-b_{ij}\right) \\ &\quad + \tau^{\notin}\sum_{i,j\in\mathcal{S}_i^{\notin}}\left(\beta_{ij}\log\left(\frac{\beta_{ij}}{b_{ij}}\right)+b_{ij}-\beta_{ij}\right) + (1-\tau^{\notin})\sum_{i,j\in\mathcal{S}_i^{\notin}}\left(b_{ij}\log\left(\frac{b_{ij}}{\beta_{ij}}\right)+\beta_{ij}-b_{ij}\right) \quad (3)\end{aligned}$$

with the sets of within-class $\mathcal{S}_i^{\in} = \{j \neq i \mid L_i = L_j\}$ and between-class $\mathcal{S}_i^{\notin} = \{j \neq i \mid L_i \neq L_j\}$ indices defined based on the class labels L_i of points i.

Note that Bregman (B) divergences replace KL divergences of Equation (2) to ensure the positivity of the stress function. Indeed, the KL divergence is only defined for probability distributions (summing to 1), while in the four terms of $\zeta_{\text{ClassNeRV}}$, the membership degrees restricted to the sets $\mathcal{S}_i^{\in}$ or $\mathcal{S}_i^{\notin}$ sum to less than 1. Hence, KL divergence is not directly applicable to such membership degrees (otherwise it would not satisfy the known properties of a divergence, namely non-negativity and identity of indiscernibles), whereas Bregman divergence is directly applicable as is. Bregman divergence terms $\mathcal{D}_{\text{B}}(\beta_i^{\in}, b_i^{\in})$ and $\mathcal{D}_{\text{B}}(b_i^{\in}, \beta_i^{\in})$ penalize *within-class* missed and false neighbours respectively, while terms $\mathcal{D}_{\text{B}}(\beta_i^{\notin}, b_i^{\notin})$ and $\mathcal{D}_{\text{B}}(b_i^{\notin}, \beta_i^{\notin})$ penalize *between-class* missed and false neighbours respectively.

Parameters $\tau^{\in}$ and $\tau^{\notin}$ define ClassNeRV mapping behavior by weighting these terms. $\tau^{\in}$ controls the balance for penalization of false neighbors and missed neighbors *within classes*, while $\tau^{\notin}$ controls a similar balancing *between classes*. Thus, **ClassNeRV is supervised** if $\tau^{\in} > \tau^{\notin}$, *i.e.* its stress function penalizes more within-class missed neighbors, and between-class false neighbors than other distortions. The greater the $\tau^{\in} - \tau^{\notin}$ difference, the higher the level of supervision with a maximum for $\tau^{\in} = 1$ and $\tau^{\notin} = 0$. Having $\tau^{\in} < \tau^{\notin}$ however would favor missed neighbors within classes, and false neighbors between classes, encouraging same-class split and distinct-class overlap, messing up class preservation. **ClassNeRV is unsupervised** if $\tau^{\in} = \tau^{\notin}$ and then reduces to the original NeRV, the combined within and between class Bregman divergences being equal to the corresponding KL divergence, with $b_{ij} - \beta_{ij}$ terms cancelling out.

We re-parameterize $\tau^{\in}$ and $\tau^{\notin}$ as $\tau^* = (\tau^{\in} + \tau^{\notin})/2$ and $\varepsilon = (\tau^{\in} - \tau^{\notin})/2$. $\tau^* \in [0,1]$ controls the average *trade-off between penalizations of false and missed neighbors* (as τ in NeRV), while $\varepsilon \in [0, 0.5]$ controls the *level of supervision* (more supervision for greater values). The reverse transformations are: $\tau^{\in} = \tau^* + \varepsilon$ and $\tau^{\notin} = \tau^* - \varepsilon$.

Given τ^* and ε, the ClassNeRV embedding is obtained by minimizing the stress $\zeta_{\text{ClassNeRV}}$ in Equation (3) with respect to the coordinates of the embedded points. The optimization is performed using a multi-scale optimization approach [37] and the quasi-Newton BFGS algorithm [38]. The current implementation has a complexity of $\mathcal{O}(N^2)$ (where N is the number of data points), but tree-based acceleration techniques [39, 40] could reduce it to $\mathcal{O}(N \log N)$.

3.3 Quality Indicators of Supervised Techniques for Exploratory Analysis

To assess the preservation of neighborhood structures, we adopt the Trustworthiness and Continuity measures [13] which are standard to evaluate unsupervised embeddings in the context of exploratory analysis [3]. These two measures separately quantify the average level of false neighbors (Trustworthiness $\mathcal{T}$) and missed neighbors (Continuity $\mathcal{C}$) for a given neighborhood size κ as:

$$\mathcal{T}(\kappa) \triangleq 1 - \frac{1}{\mathcal{T}_{\max}(\kappa)}\sum_{i,j\in\mathcal{F}_i(\kappa)}(\rho_{ij} - \kappa) \quad \text{and} \quad \mathcal{C}(\kappa) \triangleq 1 - \frac{1}{\mathcal{C}_{\max}(\kappa)}\sum_{i,j\in\mathcal{M}_i(\kappa)}(r_{ij} - \kappa), \quad (4)$$

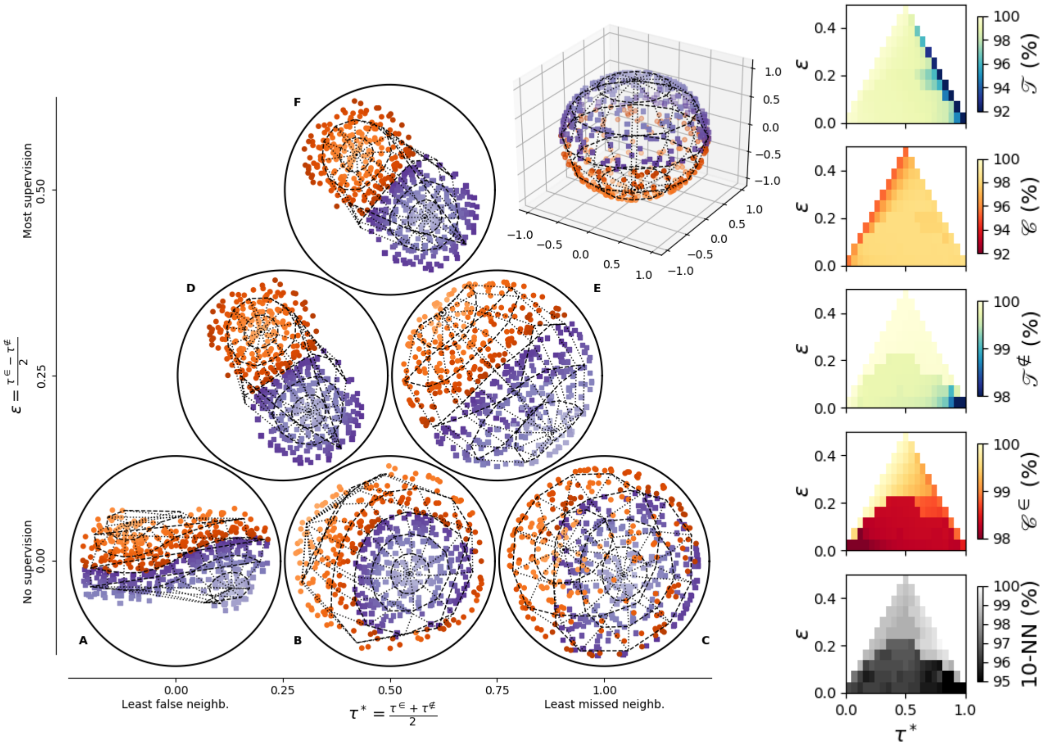
where ρ_{ij} and r_{ij} are the ranks of each point j in the neighborhood of each point i in the data and embedding spaces respectively, and $\mathcal{F}_i(\kappa)$ and $\mathcal{M}_i(\kappa)$ are the sets of indices of false and missed neighbors of point i. The normalization coefficients $\mathcal{T}_{\max}(\kappa)$ and $\mathcal{C}_{\max}(\kappa)$ are defined so that $\mathcal{T}$ and $\mathcal{C}$ range from 0 for a theoretical worst mapping to 1 for an ideal mapping. Based on a combinatorial analysis, $\mathcal{T}_{\max}(\kappa)$ and $\mathcal{C}_{\max}(\kappa)$ are equal to either $\kappa N(2N-3\kappa-1)/2$ if $\kappa < N/2$ or $N(N-\kappa)(N-\kappa-1)/2$ if $\kappa \geqslant N/2$ [41].

To assess class preservation, we also derive two new measures: Trustworthiness *restricted to between-class relations* $\mathcal{T}^{\notin}$ and Continuity *restricted to within-class relations* $\mathcal{C}^{\in}$. These *class-aware* indicators are defined as:

$$\mathcal{T}^{\notin}(\kappa) \triangleq 1 - \frac{1}{\mathcal{T}_{\max}(\kappa)} \sum_{i,j \in \mathcal{F}_i(\kappa)\cap \mathcal{S}_i^{\notin}} (\rho_{ij}-\kappa) \quad \text{and} \quad \mathcal{C}^{\in}(\kappa) \triangleq 1 - \frac{1}{\mathcal{C}_{\max}(\kappa)} \sum_{i,j\in\mathcal{M}_i(\kappa)\cap\mathcal{S}_i^{\in}} (r_{ij}-\kappa). \tag{5}$$

Notice these class-aware indicators account for only part of the distortions considered by their unsupervised counterparts, so that $\mathcal{T}^{\notin} \geqslant \mathcal{T}$ and $\mathcal{C}^{\in} \geqslant \mathcal{C}$. Thus, they may reach 1 as long as any remaining distortions do not affect between-class Trustworthiness and within-class Continuity.

Another standard quality indicator in supervised dimensionality reduction is the accuracy of a leave-one-out k-Nearest neighbors (k-NN) classifier in the embedding space [3]. This indicator makes sense when the embedding is used to perform classification, but it focuses on classes only and fails to account for neighborhood structure preservation as shown in Figure 3. Supplementary material presents extensive results for the k-NN gain [32] derived from that indicator.



(a) Maps

(b) Indicators

Figure 2: Sensitivity of ClassNeRV to the trade-off parameters while embedding the 512 data of the 3D *Globe* dataset (Section 4.2) in the $\{\tau^*, \varepsilon\}$-parameter space. The left panel (a) presents five 2-dimensional embeddings with parallels (dashed lines) and meridians (dotted lines) positioned *a posteriori*, as well as a 3D representation of the dataset (top right). The right panel (b) shows quantitative evaluation with indicators defined in Section 3.3, with from top to bottom Trustworthiness $\mathcal{T}$, Continuity $\mathcal{C}$, between-class Trustworthiness $\mathcal{T}^{\notin}$ and within-class Continuity $\mathcal{C}^{\in}$ for a given neighborhood size of $\kappa = 32$, and $10-$NN classification accuracy (99% for the original 3D data). Each pixel of the heat map corresponds to one embedding of the globe. τ^* controls the overall trade-off between false neighbors (most likely with $\tau^* = 1$) and missed neighbors (most likely with $\tau^* = 0$), while ε controls the level of supervision.

4 Experiments

4.1 Objectives, Data, Techniques

We illustrate the main characteristics of ClassNeRV compared to other unsupervised and supervised DR techniques, on a 3D toy dataset (*Globe*) and on two real high dimensional datasets (*Isolet 5* and *Digits*). The *Globe* dataset (Section 4.2) contains 512 data randomly distributed on the surface of the unit sphere in the three-dimensional Euclidean space $\mathbb{R}^3$ (Figure 2a). The two classes (blue and red points) correspond to the two hemispheres divided at the equator. These data cannot be embedded in the plane without distortions, so the final map depends on the trade-off set between the neighborhood (τ^*) and class (ε) penalizations (See Section 3.2). The *Isolet 5* dataset [42, 43] (Section 4.3) contains $1\,559$ recordings of English pronunciation of spoken letters evenly distributed in 26 classes, described by 617 features. The *Digits* dataset [44, 43] (Section 4.4) contains $3\,823$ images of handwritten digits. The 8×8 pixels images (64D data points) are separated in 10 classes (one per digit). True class labels as well as randomly generated labels are considered to evaluate the robustness to mislabeling. A random subset of 500 samples is considered to ease the readability of the maps in Figure 6.

We compare ClassNeRV to unsupervised PCA [14], Isomap [16], UMAP [12], tSNE [9], and NeRV [6, 7], and to supervised NCA [29], S-Isomap [10], ClassiMap [5] and S-UMAP [12]. Implementations of PCA, Isomap, NCA, tSNE, UMAP and S-UMAP are from scikit-learn (version 0.22.1) [45] and umap-learn (version 0.3.10) [46] Python libraries. S-Isomap, ClassiMap, NeRV and ClassNeRV use our own implementations [47]. All off-the-shelf algorithms are set with default parameters, except for tSNE initialization, for which we used PCA instead of random (so that all methods benefit from a spectral initialization). NeRV and ClassNeRV use the optimization described in Section 3.2. The final perplexity in our multi-scale optimization is set to $p = 32$ for *Globe* and $p = 30$ for *Isolet*, for comparability with tSNE default $p = 30$. We use the Euclidean distance for all techniques and datasets. Qualitative results are given as scatterplot representations of the embeddings with circular frame following standard guidelines [3, 48]. Quantitative results are computed with unsupervised and class-aware indicators described in Section 3.3. Additional runs of stochastic methods are shown in the supplementary material showing similar results.

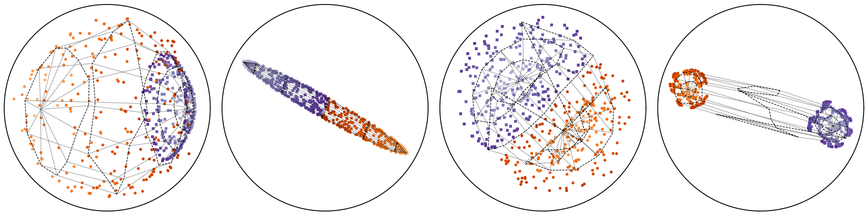
4.2 Two Hemispheres Globe Example

Figure 2a shows ClassNeRV embeddings of the *Globe* data in the (τ^*, ε)-parameter space. Maps A, B and C correspond to the original NeRV, *i.e.* without any supervision ($\varepsilon = 0$, $\tau^{\in} = \tau^{\notin}$). On map A, false neighbors are the most penalized ($\tau^{\in} = \tau^{\notin} = 0$), allowing some missed neighbors, so that the sphere is torn along a meridian and unfolded. On map C, missed neighbors are the most penalized ($\tau^{\in} = \tau^{\notin} = 1$), corresponding to the original SNE mapping [8], letting false neighbors, so that the sphere is squashed, blending the red and blue classes. Map B corresponds to NeRV with a balanced mix of missed and false neighbors ($\tau^{\in} = \tau^{\notin} = 0.5$). Adding some supervision ($\varepsilon = 0.25$), map D penalizes moderately within-class missed neighbors ($\tau^{\in} = 0.5$) and strongly all-class false neighbors ($\tau^* = 0.25$), encouraging less torn classes than in map A. Conversely, map E penalizes moderately between-class false neighbors ($\tau^{\notin} = 0.5$) and strongly all-class misses ($\tau^* = 0.75$), encouraging more class separation than in map C, but more within-class false neighbors than in map D. Finally, map F corresponds to the maximum level of supervision ($\varepsilon = 0.5$ with $\tau^{\in} = 1$ and $\tau^{\notin} = 0$), penalizing the most between-class false neighbors (the least class overlap) and within-class missed neighbors (the most class cohesion). To get a detailed idea of the individual impact of each of ClassNeRV stress sub-terms (Equation (3)), an ablation study is available in supplementary material.

Figure 2b quantitatively supports these qualitative observations. Top heat maps show *structure preservation* indicators in the same $\{\tau^*, \varepsilon\}$-parameter space: when τ^* increases, the amount of false neighbors ($\mathcal{T}$ in blue) increases, while the amount of missed neighbors ($\mathcal{C}$ in red) decreases. Bottom heat maps show *class preservation* indicators in that space: when the level of supervision ε increases, there is less class overlap ($\mathcal{T}^{\notin}$ in blue) and less class split ($\mathcal{C}^{\in}$ in red). The classification heat map also shows a greater class accuracy ($10-$NN in grey) coherent with the higher class cohesion and lower class overlap encouraged by greater ε.

Figures 3 and 4 show qualitative and quantitative comparison of the most supervised version of ClassNeRV ($\varepsilon = 0.5$) to other techniques. A qualitative analysis of supervised embeddings shows that ClassNeRV (Figure 2a, maps D, E and F), ClassiMap (Figure 3c) and NCA (Figure 3b) maps all preserve the actual 3D linear separability and adjacency of the two classes. However, S-Isomap (Figure 3a) generates a false class overlap, while S-UMAP (Figure 3d) wrongly splits classes apart.

Figure 4 (top row), shows that ClassNeRV (—) reaches similar levels of structure preservation as tSNE (—), or UMAP (—) unsupervised techniques (I), with higher trustworthiness in general (less false neighbors). It also preserves classes far better (II), as expected against unsupervised techniques. Regarding supervised techniques, ClassNeRV (—) obtains both better structure (III) and class (IV) preservation than NCA (—), ClassiMap (—) and S-Isomap (—). For S-UMAP (—), the class-independent indicators (III) are similar to ClassNeRV, except for larger neighborhood size κ, due to S-UMAP's tendency to over-separate classes (Figure 3d) actually adjacent in the *Globe* data.



(a) S-Isomap (90.6%) (b) NCA (99.2%) (c) ClassiMap (99.8%) (d) S-UMAP (100%)

Figure 3: Embeddings of the *Globe* dataset obtained with state-of-the-art dimensionality reduction techniques, to be compared with the most supervised ClassNeRV ($\varepsilon = 0.5$) in map F of Figure 2a. 10-NN accuracy is given for each technique. Analysis in Section 4.2.

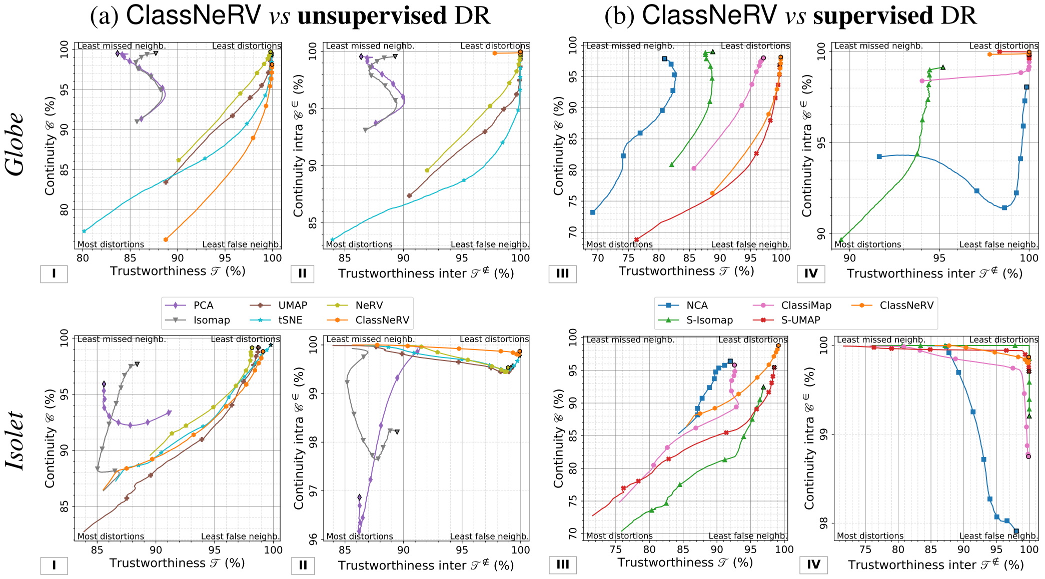
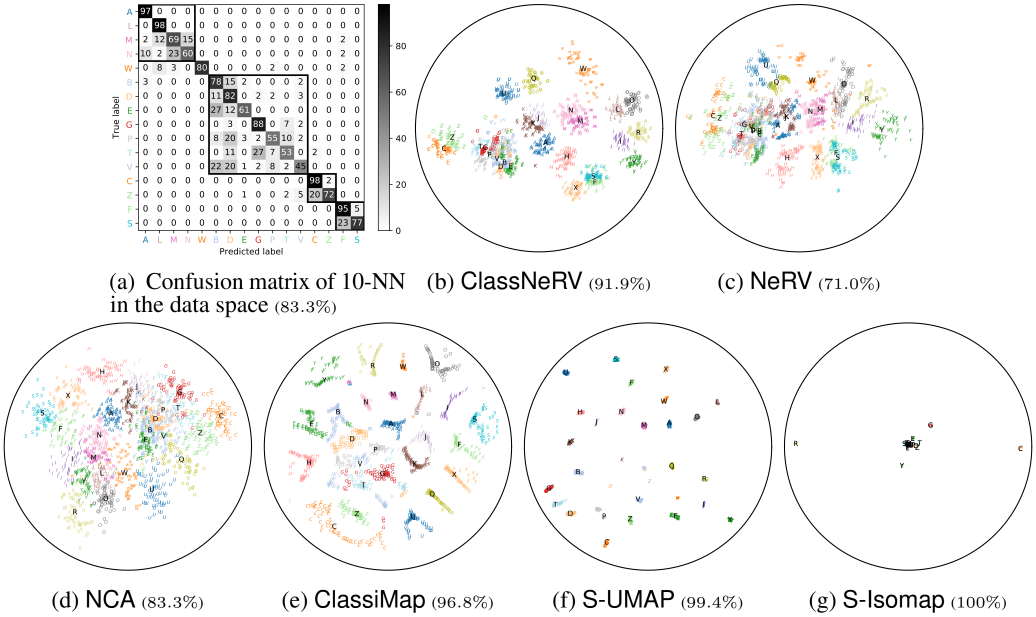


Figure 4: Quality indicators (Section 3.3) for ClassNeRV with maximum supervision ($\varepsilon = 0.5$) and state-of-the-art embeddings of *Globe* (top) and *Isolet* (bottom) datasets. Each polyline corresponds to the series of indicator values computed with κ from 2 to $\lfloor N/2 \rfloor$ (with markers indicating $\kappa \in \{2, 4, 8, 16 \ldots\}$ and $\kappa = 2$ highlighted with black outline). Unsupervised (I, II) and supervised (III, IV) DR techniques (color coded) are compared to ClassNeRV (orange) in the space of standard structure preservation indices $(\mathcal{T}, \mathcal{C})$ (I, III), and in the space of class preservation indices $(\mathcal{T}^{\notin}, \mathcal{C}^{\in})$ (II, IV). Perfect embeddings at top right corner. Analysis in sections 4.2 (*Globe*) and 4.3 (*Isolet*).

4.3 Isolet Dataset

The 10-NN confusion matrix computed in the 617D data space (Figure 5a) shows all classes with less than 90% accuracy or confused with at least 10% of another class (The full confusion matrix is given in the supplementary material). Several classes partially overlap likely due to similar sounds



(a) Confusion matrix of 10-NN in the data space (83.3%)

(b) ClassNeRV (91.9%)

(c) NeRV (71.0%)

(d) NCA (83.3%)

(e) ClassiMap (96.8%)

(f) S-UMAP (99.4%)

(g) S-Isomap (100%)

Figure 5: Confusion matrix (a) of a 10-NN classifier for the *Isolet 5* dataset showing several classes overlap in the data space. Only the most confused classes are shown in the matrix. Supervised embeddings should preserve this actual class structure. The most supervised ClassNeRV ($\varepsilon = 0.5$) (b) is compared to NeRV (c) and supervised methods (d,e,f,g). Analysis in Section 4.3. 10-NN leave-one-out accuracy score is given for the original data (a) and all the embeddings (b to g).

indicated by phonetic symbols, *e.g.* letters F [ɛf] and S [ɛs] (group FS), letters C [siː] and Z [ziː] (group CZ), letters A [eɪ], L [ɛl], M [ɛm] and N [ɛn] (group ALMN), or letters B [biː], D [diː], E [iː], G [dʒiː], P [piː], T [tiː] and V [viː] (group BDEGPTV).

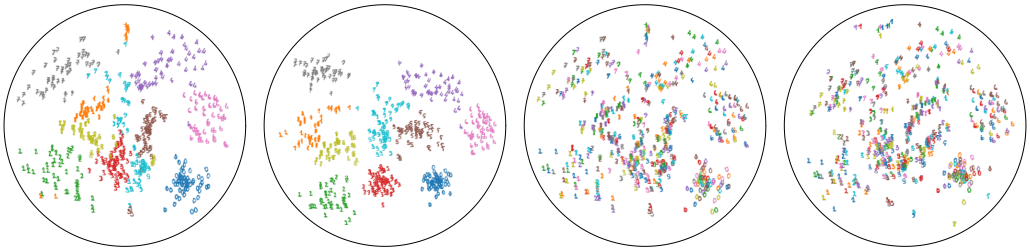
Figure 5 shows the embeddings for qualitative comparison. NCA (Figure 5d) tends to artificially overlap classes, for instance QV and JP. Conversely, S-UMAP and S-Isomap tend to over-separate classes, completely ignoring the actual overlapping of letters in groups FS, CZ, ALMN, BDEGPTV shown by the confusion matrix. ClassiMap (Figure 5e) better preserves the BDEGPTV group but fails to represent others like MN or CZ. NeRV manages to preserve groups despite being unsupervised, showing that classes of these groups actually overlap in the data space. Yet, it may artificially increase that overlap as observed in Figure 1b. Lastly, ClassNeRV (Figure 5b) provides a more trustworthy representation of the classes, better showing the stronger confusion of letters within MN, CZ, FS and BDEGPTV groups by putting them adjacent in the map, while keeping other letters separated as they actually are in the data space. Hence, the ClassNeRV map is more trustworthy to help a domain expert discover that, in this feature space, letters are strongly grouped by vowel sounds (BDEGPTV, FS or MN), with a secondary effect of the consonant.

The 10-NN classification scores (Figure 5) increase with class separation observed on these maps. Yet, as S-Isomap reaches the maximal value 100% by artificially separating all classes irrespective of the actual classes' confusion in the data space, this classification indicator cannot assess the class structure preservation. Thus, it cannot be used to evaluate if an embedding is trustworthy enough to support the *exploratory analysis of labeled data*.

The quantitative comparison of ClassNeRV (—) with unsupervised techniques on Figure 4a (bottom row) shows it preserves the neighborhood structure as well as NeRV (—) and UMAP (—), but slightly worse than tSNE (—) (I). It preserves classes better than all unsupervised techniques as expected especially for small κ (II). Regarding supervised techniques, Figure 4b (bottom row) demonstrates that ClassNeRV (—) preserves the neighborhood structure of *Isolet data* far better than the other supervised methods (III). NCA (—) generates more false neighbors ($\mathcal{T} < \mathcal{C}$), while S-Isomap (—) and S-UMAP (—) generate more missed neighbors, consistently with the embeddings (Figure 5). All techniques except NCA (—) achieve similar good results for class preservation (IV), with an advantage at small and high scales κ for ClassNeRV.

4.4 Handwritten digits

Figure 6 presents maps reached by NeRV and ClassNeRV applied to the *digits* dataset with true and random labels. The case of random labels allows to study the behaviour of supervised methods when the structure and classes are decorrelated. Separating these random classes would distort the neighborhoods, which may be seen as an over-separation. We can observe on Figures 6a and 6b that ClassNeRV relying on true class-information reaches a better preservation of classes than NeRV for the *digits* dataset. Conversely, ClassNeRV with random labels (Figure 6d) does not lead to obvious over-separation of the random classes. Those observations are confirmed by Figure 7. In addition, ClassNeRV shows better preservation of both structure and true classes than other supervised methods, when fed with the true labels and even more with the random labels (Figure 7).



(a) NeRV (true) (b) ClassNeRV (true) (c) NeRV (random) (d) ClassNeRV (random)

Figure 6: NeRV and ClassNeRV maps for the *digits* dataset, points being represented by their associated image coloured based on the class supplied to the algorithm (either true or random). Given that NeRV is not supervised, maps (a) and (c) have strictly identical points position. Conversely, differences may be observed between ClassNeRV maps (b) and (d). (b) shows better class-preservation than (a) due to the valid class-information, while no obvious over-classification may be noted on (d) with completely unreliable class-information (random labelling). Analysis in Section 4.4.

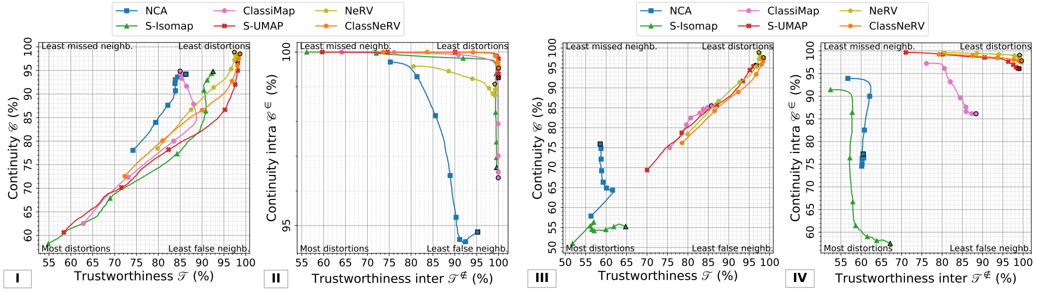


Figure 7: Quality indicators for ClassNeRV, NeRV and supervised methods on the *digits* dataset with true labels (I, II) and random labels (III, IV). Class-aware indicators $(\mathcal{T}^{\notin}, \mathcal{C}^{\in})$ (II, IV) quantify the preservation of the true classes. Note that NeRV does not account for the classes, hence its results in (III, IV) are identical to those in (I, II). Analysis in Section 4.4.

5 Conclusion

ClassNeRV allows data scientists to control class and structure preservation in low dimensional embeddings for exploratory data analysis of labeled data. Such analysis can for instance help detect whether classes are well-separated in a given feature space, which may lead to question the labels (labelling errors) or the features (feature engineering). Our experiments showed that supervised DR techniques tend to over-separate classes that are actually adjacent or overlapping in the data space, while unsupervised techniques totally ignore classes. In contrast, the fully supervised setting of ClassNeRV better preserves actual class structure than supervised DR techniques, while preserving neighborhood structure as well as state-of-the-art unsupervised DR techniques. Future work will extend this approach to other neighborhood embedding techniques, such as tSNE [9] or JSE [19], and consider the semi-supervised framework.

6 Broader impact

This work proposes improvement on a dimensionality reduction technique for exploratory data analysis. Dimensionality reduction is intended to support data scientists in analyzing multidimensional data, and can also be used to visualize high-dimensional data representing physical objects or persons in a 2D map for the lay public to get an overview of the main groups of objects/persons based on similarities in their corresponding data. It is agnostic to the nature of the objects/persons represented by the data. Better understanding of trends and variation in large-scale datasets can improve the ability of society to learn about important phenomena. However, dimensionality reduction can generate biased representation of these objects/persons, either due to the inherent bias of the data themselves (over or under represented classes of objects/persons, missing or irrelevant features artificially gathering or separating (classes of) objects/persons), or due to unavoidable projection biases, called distortions, [3] artificially gathering in the 2D representation actually separated (classes of) objects/persons, or artificially separating in the 2D representation actually similar (classes of) objects/persons. The proposed ClassNeRV method is exactly intended to reduce this second type of bias.

7 Funding disclosure

The work of Denys Dutykh has been supported by the French National Research Agency, through Investments for Future Program (ref. ANR−18−EURE−0016 — Solar Academy). Jaakko Peltonen was supported by Academy of Finland projects 313748 and 327352.

References

[1] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "Visual interaction with dimensionality reduction: A structured literature analysis," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 241–250, 2017.

[2] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, S. Leman, and C. North, "Towards a systematic combination of dimension reduction and clustering in visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, pp. 131–141, Jan 2018.

[3] L. G. Nonato and M. Aupetit, "Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, pp. 2650–2673, Aug. 2019.

[4] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner, "Visualizing dimensionally-reduced data: interviews with analysts and a characterization of task sequences," in *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization, BELIV 2014, Paris, France, November 10, 2014* (H. Lam, P. Isenberg, T. Isenberg, and M. Sedlmair, eds.), pp. 1–8, ACM, 2014.

[5] S. Lespinats, M. Aupetit, and A. Meyer-Base, "ClassiMap: A New Dimension Reduction Technique for Exploratory Data Analysis of Labeled Data," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, p. 150505235857008, May 2015.

[6] J. Venna and S. Kaski, "Nonlinear dimensionality reduction as information retrieval," in *Artificial intelligence and statistics*, pp. 572–579, 2007.

[7] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 451–490, 2010.

[8] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Advances in neural information processing systems*, pp. 857–864, 2003.

[9] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[10] Xin Geng, De-Chuan Zhan, and Zhi-Hua Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, pp. 1098–1107, Dec. 2005.

[11] J. Peltonen, H. Aidos, and S. Kaski, "Supervised nonlinear dimensionality reduction by Neighbor Retrieval," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1809–1812, Apr. 2009.

[12] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv:1802.03426 [cs, stat]*, Feb. 2018. arXiv: 1802.03426.

[13] J. Venna and S. Kaski, "Neighborhood preservation in nonlinear projection methods: An experimental study," in *International Conference on Artificial Neural Networks*, pp. 485–491, Springer, 2001.

[14] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, pp. 559–572, Nov. 1901.

[15] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

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

[17] S. Lespinats, M. Verleysen, A. Giron, and B. Fertil, "DD-HDS: A method for visualization and exploration of high-dimensional data," *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1265–1279, 2007.

[18] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato, "Local Affine Multidimensional Projection," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 2563–2571, Dec. 2011.

[19] J. A. Lee, E. Renard, G. Bernard, P. Dupont, and M. Verleysen, "Type 1 and 2 mixtures of Kullback–Leibler divergences as cost functions in dimensionality reduction based on similarity preservation," *Neurocomputing*, vol. 112, pp. 92–108, July 2013.

[20] O. Kouropteva, O. Okun, and M. Pietikäinen, "Supervised locally linear embedding algorithm for pattern recognition," in *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 386–394, Springer, 2003.

[21] S.-q. Zhang, "Enhanced supervised locally linear embedding," *Pattern Recognition Letters*, vol. 30, no. 13, pp. 1208–1218, 2009.

[22] L. Zhao and Z. Zhang, "Supervised locally linear embedding with probability-based distance for classification," *Computers & Mathematics with Applications*, vol. 57, no. 6, pp. 919–926, 2009.

[23] C.-G. Li and J. Guo, "Supervised isomap with explicit mapping," in *First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06)*, vol. 3, pp. 345–348, IEEE, 2006.

[24] Z. Yang, I. King, Z. Xu, and E. Oja, "Heavy-tailed symmetric stochastic neighbor embedding," in *Advances in neural information processing systems*, pp. 2169–2177, 2009.

[25] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[26] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pp. 41–48, Ieee, 1999.

[27] D. De Ridder, M. Loog, and M. J. Reinders, "Local fisher embedding," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, pp. 295–298, IEEE, 2004.

[28] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *Proceedings of the 23rd international conference on Machine learning*, pp. 905–912, 2006.

[29] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov, "Neighbourhood Components Analysis," in *Advances in Neural Information Processing Systems 17* (L. K. Saul, Y. Weiss, and L. Bottou, eds.), pp. 513–520, MIT Press, 2005.

[30] R. Salakhutdinov and G. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Artificial Intelligence and Statistics*, pp. 412–419, 2007.

[31] K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann, and M. Biehl, “Limited Rank Matrix Learning, discriminative dimension reduction and visualization,” *Neural Networks*, vol. 26, pp. 159–173, Feb. 2012.

[32] C. de Bodt, D. Mulders, D. L. Sánchez, M. Verleysen, and J. A. Lee, “Class-aware t-SNE: cat-SNE.,” in *ESANN*, 2019.

[33] J. Venna and S. Kaski, “Local multidimensional scaling,” *Neural Networks*, vol. 19, pp. 889–899, July 2006.

[34] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, “On the surprising behavior of distance metrics in high dimensional spaces,” in *Proceedings of the 8th International Conference on Database Theory*, ICDT ’01, (Berlin, Heidelberg), p. 420–434, Springer-Verlag, 2001.

[35] J. A. Lee and M. Verleysen, “Two key properties of dimensionality reduction methods,” in *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, pp. 163–170, IEEE, 2014.

[36] M. Vladymyrov and M. A. Carreira-Perpinan, “Entropic Affinities: Properties and Efficient Numerical Computation.,” in *ICML (3)*, pp. 477–485, 2013.

[37] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen, “Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure,” *Neurocomputing*, vol. 169, pp. 246–261, Dec. 2015.

[38] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer series in operations research, New York: Springer, 1999.

[39] Z. Yang, J. Peltonen, and S. Kaski, “Scalable optimization of neighbor embedding for visualization,” in *International Conference on Machine Learning*, pp. 127–135, 2013.

[40] L. Van Der Maaten, “Accelerating t-SNE using tree-based algorithms,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[41] J. Venna, *Dimensionality reduction for visual exploration of similarity structures*. PhD thesis, Helsinki University of Technology, Espoo, 2007. OCLC: 231147068.

[42] M. Fanty and R. Cole, “Spoken letter recognition,” in *Advances in Neural Information Processing Systems*, pp. 220–226, 1991.

[43] D. Dua and E. Karra Taniskidou, “UCI machine learning repository,” 2017.

[44] E. Alpaydin and C. Kaynak, “Cascading classifiers,” *Kybernetika*, vol. 34, no. 4, pp. 369–374, 1998.

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

[46] L. McInnes, J. Healy, N. Saul, and L. Grossberger, “Umap: Uniform manifold approximation and projection,” *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[47] B. Colange, “Classnerv.” `https://doi.org/10.5281/zenodo.4094851`, Oct. 2020.

[48] F. Degret and S. Lespinats, “Circular background decreases misunderstanding of multidimensional scaling results for naive readers,” in *MATEC Web of Conferences*, vol. 189, p. 10002, EDP Sciences, 2018.