Advanced Techniques in ML - Spring 2019
"Nonlinear dimensionality reduction"

Alexandre Filiot & Clémentine Rosier

1. Introduction

Many areas of artificial intelligence are confronted to high-dimensional data: pattern recognition, information retrieval, machine learning, etc... In view of the famous curse of dimensionality, the goal of data dimensionality reduction (DR) is to learn a compact meaningful low-dimensional encoding which may be hidden in high-dimensional data. The extracted features can reproduce most of the variability of the data and eliminate redundancies in the initial variables. For classification tasks, this can drastically reduce overfitting of a broad range of classifiers. As an example, consider the character "1" from the MNIST data set. Assuming 1's are strictly vertical, the set of corresponding images roughly lie in a 2-dimensional manifold where the two dimensions correspond to rotation and thickness (see figure 1), instead of a 784 initial space (for 28×28 pixels).

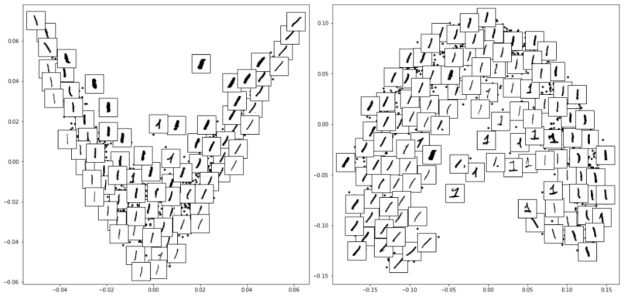

Figure 1. Projection of 1's (MNIST) onto the 2-dimensional space. Left: Locally Linear Embedding, right: PCA.

The DR problem is often known as *manifold learning*, where the notion of *manifold* has to be described. A p-dimensional manifold $\mathcal{M}$ embedded in $\mathcal{X} = \mathbb{R}^d$ is a topological space that is locally homeomorphic to an Euclidean space[1]. This means that every point has a neighbourhood for which there exists a homeomorphism mapping that neighbourhood to $\mathbb{R}^d$. This mapping is called a chart. A *smooth* manifold is a manifold where one can define directions, tangent spaces and differentiable functions on that manifold through the set of its local charts (forming an atlas). For example, the surface of the Earth (sphere) is a 2-dimensional manifold where each point can be locally embedded in a plane. Moreover, it requires at least two charts to include every point: one map for each of the North and South hemisphere. With respect to those definitions, the goal of DR is to re-embed a manifold from a high-dimensional space to a lower-dimensional one. The degrees of freedom along this submanifold correspond to the underlying variables or *latent* variable.

The most popular techniques for dimensionality reduction (DR) are the Principal Components Analysis (PCA) and/or MultiDimensional Scaling (MDS). Those techniques are meant to operate when the submanifold is embedded linearly in the observation space. In other words, when there exists a linear mapping from the high-dimensional space to a lower-dimensional subspace. Both PCA and MDS suffer from their global linearity and fail to address hard dimensionality reduction as in computer vision. Some improvements have been proposed to deal with nonlinear representations of the data. One can cite Generative Topographic Mapping (GTM), Self-Organising Maps (SOM) or neural network-based approaches which set up a nonlinear optimization problem. The main caveat of those approaches is that their solutions, obtained by gradient descent, can get stuck at local optima when the nonlinear structure does not arise from a perturbation from a linear approximation (see the "Swiss roll", figure 2, a 2D grid folded in 3D).

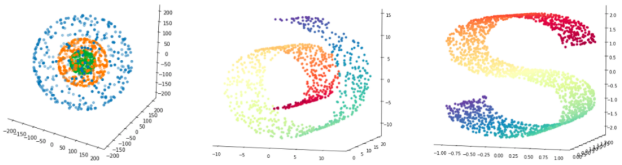

Figure 2. Three benchmark manifolds: embedded spheres (left), "Swiss roll" (middle) and "S-curve" (right).

Many recent techniques such as Kernel PCA (KPCA), Locally Linear Embedding (LLE), Laplacian Eigenmaps (LEM), Isomap or SemiDefinite Embedding (SDE) have been proposed for the past 60 years (see figure 5, appendix A.1). It is of particular interest to classify those Non Linear Dimensionality Reduction (NDRL) techniques into

[1]For the rest of this report, d and p refer to the dimensions of the initial space $\mathcal{X}$ and reduced space $\mathcal{X}'$, respectively.

2 groups: local and global. First, local approaches (LLE, LEM) aim at preserving the local geometry of the data in mapping nearby points on the manifold to nearby points in the low-dimensional submanifold. Global approches (Isomap) attempt to preserve geometry at every scale, mapping nearby points on the manifold to nearby points in low-dimensional space, and faraway points to faraway points. Kernel PCA and SDE do not fall into those two categories since it does not explicitly consider the structure of the manifold on which the data may possibly reside. The latter methods can also be separated into distance-preserving (KPCA, Isomap, SDE) and topology-preserving (LLE, LEM) techniques. Distance-preserving methods use the principle of distance (mainly spatial or graph distances) preservation in order for the submanifold to inherit the main geometric properties of data. Topology-preserving techniques rather preserve the neighborhood relationships between subregions of the manifold and the submanifold.

In this report, we focus on PCA, Kernel PCA, LEM and LLE, and give a brief overview of some other successful NLDR methods. First, we provide a theoretical overview of the main techniques along with some simple numerical experiments on synthetic manifolds. Then, we apply the latter on the MNIST data set following the 2 tasks: data vizualization and classification of the digits. The corresponding implementation can be found on Github: https://github.com/afiliot/nonlinear-dimensionality-reduction.

2. Linear techniques

Despite their intrinsic linearity, Principal Components Analysis (PCA) has to be included in the analysis in order to provide a relevant comparison with NLDR techniques. Indeed, one should not forget that those methods present some advantages that non-linear techniques attempt to combine: computational efficiency, few free parameters and non-iterative global optimisation of cost function.

2.1. Linear PCA

Principal Components Analysis is a classical algorithm in multivariate statistics and a very popular technique for DR. PCA aims at finding a submanifold which is embedded linearly in the observation state. The linear subspace can be specified by p orthogonal vectors forming a new coordinate system, called the *principal components*. These principal components are linear transformations of the original data points onto which the variance retained under projection is maximal. Let $\mathcal{S} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be a set of input vectors in $\mathcal{X} = \mathbb{R}^d$. Assume now that the data are centred: $n^{-1}\sum_{i=1}^n \mathbf{x}_i = 0$. The orthogonal projection onto a direction $\mathbf{w} \in \mathbb{R}^d$ is the function $h_\mathbf{w} : \mathbb{R}^d \to \mathbb{R}$ defined by: $h_\mathbf{w}(\mathbf{x}) = \|\mathbf{w}\|^{-1} \cdot \mathbf{x}^\top \mathbf{w}$. Then, the empirical variance captured by $h_\mathbf{w}$ is (as the $\mathbf{x}_i$'s are centred):

$$\hat{\mathbf{V}}(h_\mathbf{w}) = \frac{1}{n}\sum_{i=1}^n h_\mathbf{w}(\mathbf{x}_i)^2 = \frac{1}{n}\sum_{i=1}^n \frac{(\mathbf{x}_i^\top \mathbf{w})^2}{\|\mathbf{w}\|^2}$$

which implies that the i-th principal direction $\mathbf{w}_i (i = 1, ..., p)$ is defined by:

$$\mathbf{w}_i = \underset{\mathbf{w} \perp \{\mathbf{w}_1, ..., \mathbf{w}_{i-1}\}}{\operatorname{argmax}} \hat{\mathbf{V}}(h_\mathbf{w}) \text{ s.t. } \|\mathbf{w}\| = 1$$

which can be rewritten in matrix form:

$$\mathbf{w}_i = \underset{\mathbf{w} \perp \{\mathbf{w}_1, ..., \mathbf{w}_{i-1}\}}{\operatorname{argmax}} \mathbf{w}^\top \mathbf{X}\mathbf{X}^\top \mathbf{w} \text{ s.t. } \|\mathbf{w}\| = 1$$

where $\mathbf{X}$ is the $\boldsymbol{d} \times \boldsymbol{n}$ data matrix. Suppose now that we wish to project the data onto a p-dimensional subspace, then the directions $\mathbf{w}_i$ (which form an orthonormal basis of this subspace), are the successive eigenvectors of $\hat{\mathbf{C}} = \mathbf{X}\mathbf{X}^\top$, i.e. the empirical covariance matrix, ranked by decreasing eigenvalues. A nice property of PCA is that it minimises the squared reconstruction error, $\sum_{i=1}^n \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$ where $\mathbf{X}' = \mathbf{U}^\top \mathbf{X}$, $\hat{\mathbf{X}} = \mathbf{U}\mathbf{X}'$ and $\mathbf{U}$'s columns are eigenvectors of $\hat{\mathbf{C}}$ corresponding to the top p eigenvalues.

2.2. MDS

MultiDimensional Scaling[2] is another classical approach to solving high-dimensional problems in a linear fashion. MDS attempts to preserve pairwise distances between the low and high-dimensional data points. When dealing with Euclidean distances, one talks about metric MDS, which produces the same solution as PCA. In this particular case, the objective to minimise (in $\mathbf{X}' = \mathbf{W}\mathbf{X}$ where $\mathbf{W}$ a $n \times p$ matrix s.t. $\mathbf{W}^\top\mathbf{W} = \mathbf{I}_p$) is $E_{\text{MDS}} = \sum_{i,j=1}^n \left(\mathbf{x}_i^\top \mathbf{x}_j - \mathbf{x'}_i^\top \mathbf{x'}_j\right)^2$. One can show that the new data $\mathbf{X}'$ can be written as $\mathbf{X}' = \mathbf{I}_{p\times n}\mathbf{\Lambda}^{1/2}\mathbf{U}^\top$ with $\mathbf{U}$ such that $\mathbf{S} = \mathbf{X'}^\top\mathbf{X}' = (\mathbf{W}\mathbf{X})^\top\mathbf{W}\mathbf{X} = \mathbf{X}^\top\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ is the eigenvalue decomposition of the Gram matrix $\mathbf{S} = \left(\mathbf{x}_i^\top \mathbf{x}_j\right)_{1\le i,j\le n}$. $\mathbf{\Lambda}$ contains the eigenvalues of $\mathbf{S}$ in decreasing order. The equivalence between the two methods is an advantage. Indeed, if $p \ll n$, PCA spends fewer memory resources than MDS since the product $\mathbf{X}\mathbf{X}^\top$ has a smaller size that $\mathbf{X}^\top\mathbf{X}$. By contrast, MDS is better when the dimensionality is very high but the number of points rather low. Exactly like PCA, the optimisation method is exact and purely algebraical: the optimal solution is obtained in closed form. Some generalisations have been made to tackle the (too) strong assumption that proximity between points

[2] see T. Cox and M. Cox. Multidimensional Scaling. *Chapman Hall, Boca Raton*, 2nd edition. 2001.

are distance measures. In particular, Shepard[3] and Kruskal[4] addressed this issue and developed a method known as nonmetric MDS. In nonmetric MDS, only the ordinal information (i.e., proximity ranks) is used for determining the spatial representation. A monotonic transformation of the proximities is calculated, yielding scaled proximities. Nonmetric MDS then consists of finding a spatial representation that minimizes the stress function

$$E_{\text{nmMDS}} = \left(c^{-1}\sum_{i,j=1}^{n} w_{ij}|f(p(\mathbf{x}_i,\mathbf{x}_j)) - \mathbf{x'}_i^\top \mathbf{x'}_j|^2\right)^{1/2}$$

where $p(\mathbf{x}_i,\mathbf{x}_j)$ is the proximity between $\mathbf{x}_i$ and $\mathbf{x}_j$, f is a monotonic transformation of the proximities, such that $f(p(\mathbf{x}_i,\mathbf{x}_j)) \approx \mathbf{x}_i^\top\mathbf{x}_j$, c is a scale factor. Quasi-Newton update rule then iteratively determines the parameters $\mathbf{x'}_i$ $(i = 1,...,p)$. The most successful version of nonmetric MDS is the Sammon's nonlinear mapping[5] which aims at minimising

$$E_{\text{sammon}} = \left(\sum_{i=1,i<j}^{n} \mathbf{x}_i^\top\mathbf{x}_j\right)^{-1} \sum_{i=1,i<j}^{n} \frac{\left(\mathbf{x}_i^\top\mathbf{x}_j - \mathbf{x'}_i^\top\mathbf{x'}_j\right)^2}{\mathbf{x'}_i^\top\mathbf{x'}_j}.$$

3. Non Linear techniques

3.1. Kernel PCA

PCA and MDS are designed to model linear variabilities in high-dimensional data but fail on some non linear dimensionality reduction tasks (see figure 3). Kernel PCA (Scholkopf et al., 1999) finds principal components which are non-linearly related to the input space by performing PCA in the space produced by the nonlinear mapping $\phi : \mathcal{X} \to \mathcal{H}$, where the low-dimensional latent structure is, hopefully, easier to discover. After centring the kernel matrix $\mathbf{K}$, $\mathbf{K}_{ij} = \langle\phi(\mathbf{x_i}),\phi(\mathbf{x_j})\rangle_{\mathcal{H}}$ (by applying the transformation $(\mathbf{I}_n - \mathbf{1}_n)\mathbf{K}(\mathbf{I}_n - \mathbf{1}_n)$ where $\mathbf{1}_n$ is a $n \times n$ matrix with all elements equal to $1/n$), we define the orthogonal projection onto a direction $f \in \mathcal{H}$ the function $h_f : \mathcal{X} \to \mathbb{R}$ such that $h_f(\mathbf{x}) = \langle\phi(\mathbf{x}), \|f\|_{\mathcal{H}}^{-1} \cdot f\rangle_{\mathcal{H}}$. Now, the i-th principal direction f_i $(i = 1,...,p)$ is given by:

$$f_i = \text{argmax}_{f\perp\{f_1,...,f_{i-1}\}} \hat{\mathbb{V}}(h_f) \text{ s.t. } \|f\|_{\mathcal{H}} = 1$$

As $\hat{\mathbb{V}}(h_f) := \frac{1}{n}\sum_{i=1}^{n} \|f\|_{\mathcal{H}}^{-2} \cdot f(\mathbf{x}_i)^2$, the i-th principal direction is in fact given by

$$f_i = \underset{f\perp\{f_1,...,f_{i-1}\}}{\text{argmax}} \sum_{i=1}^{n} f(\mathbf{x}_i)^2 \text{ s.t. } \|f\|_{\mathcal{H}} = 1$$

[3]see R.N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27(2):125140, 1962.

[4]see Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. 1964

[5]J. W. Sammon, Jr, A nonlinear mapping for data structure analysis, *IEEE Transactions on Computers*, vol.C-18, no.5, pp.401409, 1969.

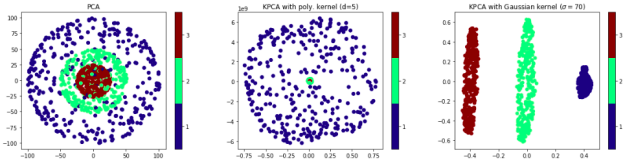

Figure 3. Projection of 3 concentric spheres onto the $2D$-plane. Left: PCA, middle: kernel PCA with polynomial kernel of degree 5, right: kernel PCA with Gaussian kernel ($\sigma = 70$).

Now, the style of RKHS theory comes into play. First, we did not mention that a key assumption for kernel PCA is that $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) and $\mathbf{K}$ (see after) is a positive definite kernel (Gram matrix is p.s.d.). It follows from the representer theorem that, for $i = 1,...,p$, $\forall\mathbf{x} \in \mathcal{X}$, $f_i(\mathbf{x}) = \sum_{j=1}^{n} \alpha_{ij}\mathbf{K}(\mathbf{x}_j,\mathbf{x})$ with $\boldsymbol{\alpha}_i = (\alpha_{i1},...,\alpha_{in})^\top \in \mathbb{R}^n$. Therefore, we have that $\|f_i\|_{\mathcal{H}}^2 = \sum_{k,l=1}^{n} \alpha_{ik}\alpha_{il}\mathbf{K}(\mathbf{x}_k,\mathbf{x}_l) = \boldsymbol{\alpha}_i^\top\mathbf{K}\boldsymbol{\alpha}_i$, $\sum_{k=1}^{n} f_i(\mathbf{x}_k)^2 = \boldsymbol{\alpha}_i^\top\mathbf{K}^2\boldsymbol{\alpha}_j$ and $\langle f_i, f_j\rangle_{\mathcal{H}} = \boldsymbol{\alpha}_i^\top\mathbf{K}\boldsymbol{\alpha}_j$. Kernel PCA thus maximises in $\boldsymbol{\alpha}$ the function $(\alpha_i =)$:

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\text{argmax}}\, \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \text{ s.t. } \begin{cases} \boldsymbol{\alpha}_i^\top\mathbf{K}\boldsymbol{\alpha}_j = 0 \text{ for } j = 1,...,i-1 \\ \boldsymbol{\alpha}_i^\top\mathbf{K}\boldsymbol{\alpha}_i = 1 \end{cases}$$

One can finally show that the solution to this optimisation problem is given by the p eigenvectors $\mathbf{u}_i$'s of the Gram matrix $\mathbf{K}$ (normalized by $1/\sqrt{\lambda_i}$) corresponding to the first p eigenvalues[6]. Kernel PCA complexity is close to MDS complexity and therefore at most $\mathcal{O}[pn^2]$ due to the computation of eigenvalues and the eigenvectors associated to the first p eigenvalues.

4. Local approaches

4.1. Locally Linear Embedding

As told in the introduction, a family of NLDR techniques concerns local approaches. Contrarily to Kernel PCA and other distance-preserving (MDS, Isomap) techniques, one considers here topology-preserving techniques. Indeed, a major drawback of metric or nonmetric MDS is that it characterises a manifold using distances only, which can be very constraining. Indeed, distances between two points are most often computed along a straight line and do not take the manifold into account. On the contrary, neighbours are exclusively inside the manifold, what makes no distinction between the manifold and the surrounding empty space. LLE or Locally Linear Embedding[7] is based on conformal mappings: a *conformal map* is a transformation that preserves local angles. Still, $\mathcal{S} = \{\mathbf{x}_1,...,\mathbf{x}_n\}$ denotes the set of input vectors in $\mathbb{R}^d$. One will first identify the nearest neighbours of each vector $\mathbf{x}_i$ using KNN methods or choos-

[6]Note that if $\langle\phi(\mathbf{x_i}),\phi(\mathbf{x_j})\rangle_{\mathcal{H}} = \mathbf{x}_i^\top\mathbf{x}_j$, we retrieve the PCA!

[7]Roweis & Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, 2000.

ing all points within a certain radius ϵ. The idea of LLE is then to replace each point $\mathbf{x}_i$ by a linear combination of its neighbours. Hence, the local geometry of the manifold can be characterised by linear coefficients which reconstruct each data point from its neighbours. Therefore, the weights are computed by minimising the following quadratic cost function:

$$W = \arg\min_W \sum_i \|\mathbf{x}_i - \sum_{j\in\mathcal{N}(i)} w_{ij}\mathbf{x}_j\|^2$$

under the constraints $\sum_{j=1}^n w_{ij} = 1$ and $w_{ij} = 0$ if $\mathbf{x}_j \notin \mathcal{N}(i)$ (the set of neighbours of $\mathbf{x}_i$). The reconstruction weights characterise intrinsic geometric properties of each neighbourhood. And then, there exists a linear mapping, consisting of a translation, rotation, and rescaling, that maps the high-dimensional coordinates of each neighbourhood to global intrinsic coordinates on the manifold. The p-dimensional output vectors, denoted $\mathbf{X}' = (\mathbf{x}'_1, ...\mathbf{x}'_n)$, are then computed using the same weights, so that each point is still a linear combination of its neighbours. Consequently, one needs to minimise the following equation:

$$\text{argmin}_{\mathbf{x}'} \sum_{i=1}^n \|\mathbf{x}'_i - \sum_{j\in\mathcal{N}(i)} w_{ij}\mathbf{x}'_j\|^2$$

which can be reformulated as:

$$\text{argmin}_{\mathbf{x}'} \text{Trace}(\mathbf{X}'\mathbf{X}'^\top\mathbf{E})$$

with $\mathbf{E} = (\mathbf{I}-\mathbf{W})^\top(\mathbf{I}-\mathbf{W})$ a sparse, symmetric, and positive semidefinite matrix and $\mathbf{W}$ the adjacency matrix. By adding the constraints that the $\mathbf{x}'_i$ are centred and have unit covariance ($\mathbf{X}'\mathbf{X}'^\top = \mathbf{I}$), we remove the possibility that $\mathbf{X}'$ can have an arbitrary orientation and origin. The final solution is given by the eigenvectors corresponding to the p nonzero lowest eigenvalues. Indeed, the last eigenvector of $\mathbf{E}$ is a scaled unit vector with all components equal; it represents a free translation mode and is associated with a zero eigenvalue.

The LLE algorithm has a relatively high computational complexity led by a n^2 term. If one uses the BallTree algorithm to identify the k ($< d$) nearest neighbours, then, the complexity can be divided into: (i) *selection of neighbours*: $\mathcal{O}[d\log(k)n\log(n)]$; (ii) *weight matrix construction*: solve $k\times k$ linear equations for each point $\mathcal{O}[dnk^3]$; (iii) *Partial Eigenvalue Decomposition*: $\mathcal{O}[pn^2]$ but can be improved. Therefore, LLE has an overall computational cost around $\mathcal{O}[d\log(k)n\log(n)] + \mathcal{O}[dnk^3] + \mathcal{O}[pn^2]$.

4.2. Laplacian Eigenmaps

Laplacian Eigenmaps (LEM) was introduced by (Belkin & Niyogi, 2003), it approximates the initial manifold by the adjacency graph with weights related to the heat equation. it relies on graph-theoretic concepts like the Laplacian operator on a graph. LEM is based on the minimisation of local distances, i.e, distances between neighbouring data points. Similarly to LLE, LEM starts by constructing an undirected weighted graph $G = (V_n, E)$ with n nodes and weights edges between points considered as neighbours (using k-nearest neighbours or defined radius), determining the weighted adjacency matrix $\mathbf{W}$. The aim of LEM is to map $\mathbf{X}$ to a set of low-dimensional points $\mathbf{X}'$ that keeps the same neighbourhood relationships. Then, the criterion to minimise is the following

$$E_{\text{LEM}} = \tfrac{1}{2}\sum_{i,j=1}^n w_{ij}\|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2$$

Here, the computation of weights differs. The authors suggest to use the heat kernel, namely, if i and j are connected, $w_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/t)$ and $w_{ij} = 0$ otherwise. t is a real parameter, which has to be tuned depending on the data set. The authors also suggest to set $t = \infty$ so that $w_{ij} = 1$ whenever the two points are connected to overcome the problem of t determination. Using such weights ensure that, with the aim to minimise E_{LEM} in $\mathbf{X}'$, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other, then $\mathbf{x}'_i$ and $\mathbf{x}'_j$ should be close as well. The weights w_{ij} act as penalties that are heavier (resp., small or null) for close (resp., faraway) data points, hence the notion of locality. Knowing that $\mathbf{W}$ is symmetric, the criterion E_{LEM} can be written in matrix form as $\text{Tr}(\mathbf{X}'\mathbf{L}\mathbf{X}'^\top)$ where $\mathbf{L}$ is the weighted Laplacian matrix of the graph G, $\mathbf{L} = \mathbf{W} - \mathbf{D}$, where $\mathbf{D}_{ii} = \sum_j w_{ij}$. As for LLE, we force the constraints $\mathbf{X}'^\top\mathbf{D}\mathbf{X}' = \mathbf{I}$ to avoid arbitrary scaling and $\mathbf{X}'^\top\mathbf{D}\mathbf{1} = 0$ to eliminate the constant mapping which would reduce all points to a single one, or to a subspace of dimension less than p. Under those constraints, minimising E_{LEM} w.r.t $\mathbf{X}'$ reduces to solving the generalized eigenvalue problem $\lambda\mathbf{D}\mathbf{u} = \mathbf{L}\mathbf{u}$, and extracting the p eigenvectors of $\mathbf{L}$ associated with the smallest nonzero eigenvalues.

The strength of LEM is its geometrical interpretation. Suppose that the initial high-dimensional manifold $\mathcal{M}$ is embedded in $\mathbb{R}^d$. Define now the Laplace Beltrami operator $\mathcal{L}$ on $\mathcal{M}$ as $\mathcal{L}f = -div\nabla(f)$ and f the twice differentiable real line mapping. The authors show, using the geodesic distance $\ell = dist_{\mathcal{M}}(\mathbf{x},\mathbf{z})$, that for any point $\mathbf{x},\mathbf{z}$, $|f(\mathbf{z}) - f(\mathbf{x})| \leq \|\nabla f(\mathbf{x})\|\|\mathbf{z}-\mathbf{x}\| + o(\|\mathbf{z}-\mathbf{x}\|)$. This implies that $\|\nabla f\|$ provides an estimate of how far apart f maps nearby points. Therefore, the mapping preserving locality minimises $\int_{\mathcal{M}}\|\nabla f\|^2 = \int_{\mathcal{M}}\mathcal{L}(f)f$. Consequently, we obtain that the optimal solution $f = (f_1, ..., f_p)$, as for the graph Laplacian, is composed of the eigenfunctions of $\mathcal{L}$ associated to the p lowest non-zero eigenvalues.

This method also shares a strong link with spectral clustering, since the weight can be considered as measures of similarity. In fact, "the process of dimensinality reduction that preserves locality reduction yields the same solution as clustering". Generally, in spectral clustering, the K main communities of a graph are related to the K smallest eigenvalues obtained from the EVD decomposition of the graph

Laplacian. If one uses the simple weight computation with $t = \infty$ and BallTree to identify neighbours, overall complexity of LEM is $\mathcal{O}[d\log(k)n\log(n)]+\mathcal{O}[dnk^3]+\mathcal{O}[dn^2]$. But we omit here the tuning of the parameter t!

5. Numerical experiments

5.1. Synthetic data

To analyse and compare the different methods, we first tested the latter on 3 different synthetic manifolds, namely the "Embedded spheres", the "Swiss Roll" and the "S-curve" (see figure 2). The challenge of the SR (and the S-curve) consists in finding a 2-dimensional embedding that "unfolds" it, in order to avoid superpositions of the successive turns of the spiral and to obtain a bijective mapping between the initial and final embeddings of the manifold. The SR is a noncompact, smooth, and connected manifold. For the spheres (compact, smooth and connected manifold), the challenge is to separate the different classes. The results are available in figures 6, 7, 8.

5.2. MNIST data set

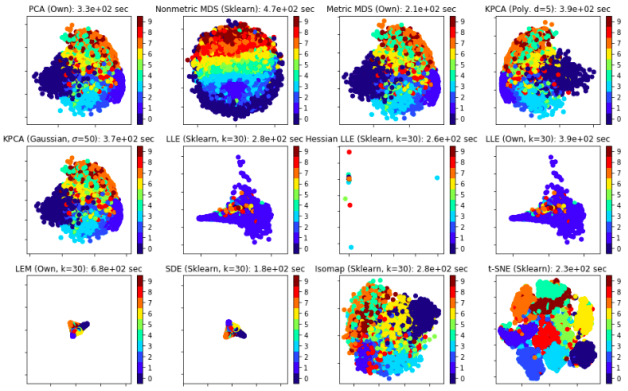

Figure 4. Projection of the MNIST data set (10000 first samples) onto the 2D-plane (zoom on LEM is available in appendix A.8).

5.3. Visual comparison

Figures 6, 7, 8 (in appendix) and 4 firstly show that our implementations of PCA, Kernel PCA, LLE and LEM seems to be in line with Sklearn[8]. It is clear that the performance of each method strongly depends on the original manifold. Nevertheless, one can draw some conclusions about the behaviour of those NLDR techniques.

First, PCA and MDS can not deal with complex data sets: the linearity assumption is too restrictive. Nonmetric MDS remains disappointing for the SC and the SR but leads to a way better clustering than its classical counterpart on the MNIST data set. It also suffers from its optimization procedure, which can be slow and inefficient for some data sets (like the SR) if the Sammon's stress function is not concave. We set the parameters `max_iter`$= 100$ and 5 initialisations as starting points for the optimization, which can be reduced in order to save computational time.

Isomap[9], considers graph distance (hence "nonlinear") rather than Euclidean ones. Consequently, it is able to perform much better than metric MDS on the SC and SR manifolds. A main drawback of Isomap is its computational time: $\mathcal{O}(d\log(k)n\log(n) + n^2(k+\log n) + dn^2)$[10]. Moreover, geodesic distances are approximated by the graph distances, whose quality depends on the data and the number of nearest neighbours in the construction of the adjacency matrix.

To conclude with distance-preserving techniques, Kernel PCA performs very well on the spheres, but very poorly on the other manifolds. Moreover, it seems to provide the same solutions as PCA on the MNIST data set. In practise, Kernel PCA is not used much in DR. The geometric interpretation of distances is not direct as, for example with the Gaussian kernels, Euclidean distances are transformed in such a way that long distances yield smaller values than short distances. Moreover, the choice of an appropriate kernel along with the right values for its parameters can be difficult, as shown in appendix A.5. In fact a particular choice of kernel can lead to increase the dimensionality of the manifold (when $p > d$ eigenvalues are similar), as done with SVM's.

Now, let's focus on the local methods: LLE and LEM. First, LLE is well unfolding the SR. The first step consisting of computing the nearest neighbours brings the nonlinear nature of LLE. The results on the MNIST data set are not as satisfying because of the incapacity for the algorithm to cluster well the digits "1". For 50000 training images, running the whole algorithm becomes untractable in terms of memory. Moreover, the embedding is very sensitive to the number of nearest neighbours k. LEM also suffers from this sensitivity but with respect to, also, the heat kernel parameter t. The Swiss-roll is partially unfolded as the third dimension of the spiral is crushed. As we saw, the locality-preserving character of LEM, combined with a clustering procedure, makes it particularly suitable for emphasizing the natural clusters in the data. Whereas PCA or Isomap can perform better on data sets without implicit clusters, LEM performs well here, in particular for the classes "0", "1" and "6" of MNIST (see A.8)). However, LEM often led to degenerate solutions, where some significant groups of points shared the same coordinates. This is observed for the spheres for a too large number of

[8] http://scikit-learn.org/stable/modules/manifold.html.

[9] see Tenenbaum & al, A global geometric framework for nonlinear dimensionality reduction, 2000.

[10] Respectively for nearest neighbour search, shortest-path graph search and partial eigenvalue decomposition.

Table 1. Best accuracies on full MNIST test obtained after grid-searching. In parentheses, the dimension of the new manifold. *No DR* refers to the absence of dimensionality reduction. Bold accuracies refer to the best ones given a classifier, italic for a given DR technique. Blue accuracy is the global best accuracy (Neural Network excluded).

	LDA	QDA	KNN	SVM (LIN)	SVM (GAU.)	NEURAL NETWORK
NO DR (784)	83.79	59.04	93.25	86.88	*93.05*	99.10
PCA (50)	85.94	*95.18*	93.06	88.28	87.22	-
KPCA (POLY.)	94.31 (700)	*95.32 (50)*	94.19 (100)	93.70 (700)	**92.37 (100)**	-
KPCA (GAU.)	95.10 (700)	*95.35 (50)*	94.49 (70)	95.09 (700)	93.12 (100)	-
LLE	*94.85 (500)*	90.31 (50)	92.67 (50)	94.84 (700)	91.73 (100)	-
LEM	**95.14 (700)**	**95.44 (50)**	**94.52 (50)**	***95.82 (700)***	93.01 (100)	-
ISOMAP	91.63 (500)	*92.84 (50)*	91.17 (50)	91.67 (100)	92.21 (100)	-

points (say 2000): LEM concentrates the new points on only 3 locations!

At last, we wanted to test the t-SNE algorithm, proposed by Van der Maaten and Hinton in "Visualizing Data using t-SNE" (2008). This methods has become quite popular and very successful for visualization in a low-dimensional space of two or three dimensions. Briefly, t-SNE aims at minimizing the Kullback-Leibler divergence between two probability distributions (over the pairs of high and low-dimensional data points respectivel)w.r.t the spatial locations of the new points. The 2-dimensional embedding on MNIST is quite impressive compared to the other techniques, but extremely slow, and untractable for the full data set.

5.4. Classification of the MNIST digits

The second part of our numerical experiments consisted in training some classifiers on the MNIST data set for different NLDR strategies. We tuned the parameters of each combination of models with a brute-force approach to save computational resources. Indeed, our model selection was based on the ranking of the accuracies on the validation set, without any cross-validation. The training and validation was done on respectively 10000 (1/5) and 10000 (1/2) samples. The best accuracies on the full test set are provided in table 1. Laplacian Eigenmaps and Kernel PCA are clearly stand out from the crowd. For each classifier, linear and non-linear DR techniques systematically brings a huge benefit on the quality of predictions. Dimensionality reduction can be very drastic in some cases where the number of components kept is around 50. It's interesting to see that, even when this number is very close the original dimensionality (700), the results are way better than the *no DR* approach, meaning that each DR algorithms provide a very meaningful encoding of the data.

6. Conclusion

To conclude this project, it is clear that non linear dimensionality reduction techniques usually perform better than linear ones, for both tasks of data vizualization (clustering) and classification. But, all resides in the "generally" term. The three synthetic examples showed that PCA and MDS can't handle complex but geometrically meaningful data sets. However, computer vision tasks are often based on very complex data that do not have necessarily natural clusters or good geometrical properties. When dealing with big data and limited resources, a serious arbitration has to be made between computational time and performance gain. It turns out that local and global distance-preserving methods like LLE, LEM and Isomap suffer from their heavier complexity due to the nearest neighbours determination and EVD of $n \times n$ matrices. To some extent, Kernel PCA tackles this issue but finding the appropriate kernels is just as troublesome. Each of those NLDR techniques provides closed form solution which are nonetheless very sensitive to the parameters tuning. Thus, it is essential for any application to first try simple, linear and computational efficient techniques. With the rise of deep learning, and as highlighted in table 1, deep networks dramatically boosted performance. But, all of our NLDR techniques rely on *mathematical concepts and geometrical interpretations* which allow a deeper understanding of dimensionality reduction.

References

Belkin, M. and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

Ghodsi, A. Dimensionality reduction a short tutorial. 01 2006.

Lee, J. A. and Verleysen, M. *Nonlinear Dimensionality Reduction*. Springer Publishing Company, Incorporated, 1st edition, 2007.

Scholkopf, B., Smola, A., and Mller, K.-R. Kernel principal component analysis. In *Advances in Kernel Methods - Support Vector Learning*, pp. 327–352. MIT Press, 1999.

Vert, J.-P. and Mairal, J. Kernel methods for machine learning. *MVA MSc, ENS Paris-Saclay*, 2019.

A. Appendix

A.1. Dimensionality Reduction Timeline

ANN	DR	Method	Author(s) & reference(s)
	1901	PCA	Pearson [149]
	1933	PCA	Hotelling [92]
	1938	classical metric MDS	Young & Householder [208]
1943		formal neuron	McCulloch & Pitts [137]
	1946	PCA	Karhunen [102]
	1948	PCA	Loève [128]
	1952	MDS	Torgerson [182]
1958		Perceptron	Rosenblatt [157]
1959		Shortest paths in a graph	Dijkstra [53]
	1962	nonmetric MDS	Shepard [171]
	1964	nonmetric MDS	Kruskal [108]
1965		K-means (VQ)	Forgy [61]
1967		K-means (VQ)	MacQueen [61]
		ISODATA (VQ)	Ball & Hall [8]
	1969	PP	Kruskal [109]
		NLM (nonlinear MDS)	Sammon [109]
1969		Perceptron	Minsky & Papert's paper [138]
	1972	PP	Kruskal [110]
	1973	SOM	von der Malsburg [191]
	1974	PP	Friedman & Tukey [67]
1974		Back-propagation	Werbos [201]
1980		LBG (VQ)	Linde, Buzo & Gray [124]
	1982	SOM (VQ & NLDR)	Kohonen [104]
1982		Hopfield network	Hopfield [91]
		LLoyd (VQ)	Lloyd [127]
	1984	Principal curves	Hastie & Stuetzle [79, 80]
1985		Competitive learning (VQ)	Rumelhart & Zipser [162, 163]
1986		Back-propagation & MLP	Rumelhart, Hinton & Williams [161, 160]
		BSS/ICA	Jutten [99, 98, 100]
	1991	Autoassociative MLP	Kramer [107, 144, 183]
	1992	"Neural" PCA	Oja [145]
	1993	VQP (NLM)	Demartines & Hérault [46]
		Autoassociative ANN	DeMers & Cottrell [49]
	1994	Local PCA	Kambhatla & Leen [101]
	1995	CCA (VQP)	Demartines & Hérault [47, 48]
		NLM with ANN	Mao & Jain [134]
	1996	KPCA	Schölkopf, Smola & Müller [167]
		GTM	Bishop, Svensén & Williams [22, 23, 24]
	1997	Normalized cut (spectral clustering)	Shi & Malik [172, 199]
	1998	Isomap	Tenenbaum [179, 180]
	2000	CDA (CCA)	Lee & Verleysen [116, 120]
		LLE	Roweis & Saul [158]
	2002	Isotop (MDS)	Lee [119, 114]
		LE	Belkin & Niyogi [12, 13]
		Spectral clustering	Ng, Jordan & Weiss [143]
		Coordination of local linear models	Roweis, Saul & Hinton [159]
	2003	HLLE	Donoho & Grimes [56, 55]
	2004	LPP	He & Niyogi [81]
		SDE (MDS)	Weinberger & Saul [196]
	2005	LMDS (CCA)	Venna & Kaski [186, 187]
	2006	Autoassociative ANN	Hinton & Salakhutdinov [89]

Figure 5. Timeline of DR methods. Source: (Lee & Verleysen, 2007), p. 228, fig 7.1.

A.2. 2-dimensional embedding of the S-curve

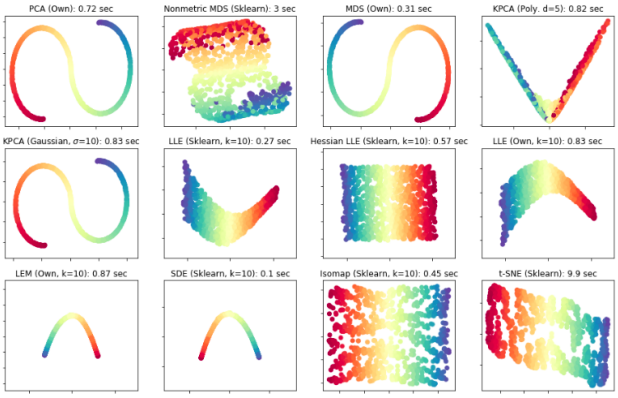

Figure 6. Projection of the S-curve manifold (2000 samples) onto the 2D-plane.

A.3. 2-dimensional embedding of the spheres

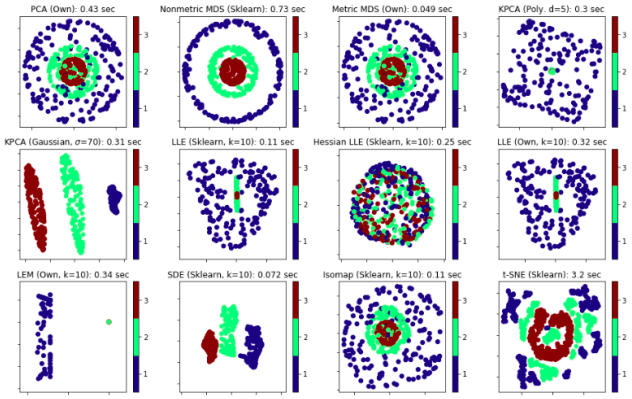

Figure 7. Projection of the embedded spheres (500 points) onto the 2D-plane.

A.4. 2-dimensional embedding of the Swiss-roll

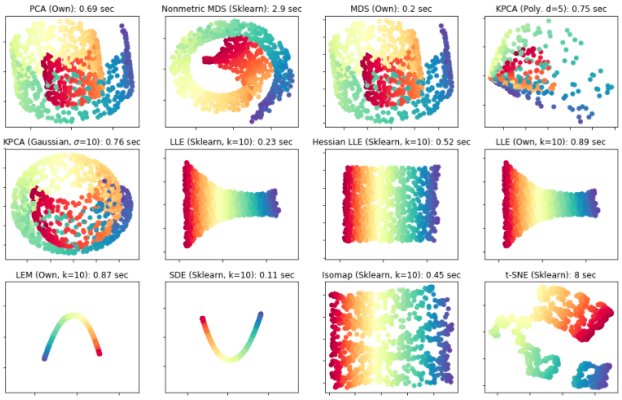

Figure 8. Projection of the swiss-roll (2000 points) onto the 2D-plane.

A.5. Kernel PCA: influence of parameters

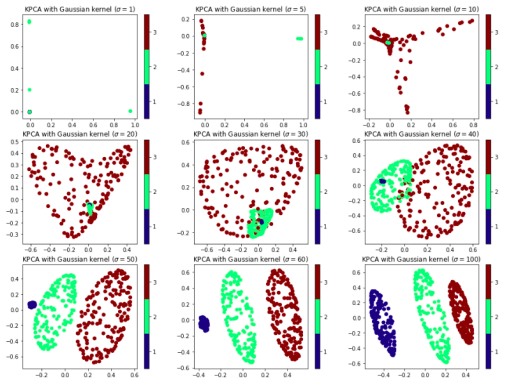

Figure 9. Influence of σ parameter in kernel PCA with Gaussian kernel ($K_{ij} = \exp\left(-0.5\|\mathbf{x_i} - \mathbf{x_j}\|^2/\sigma^2\right)$) on the spheres

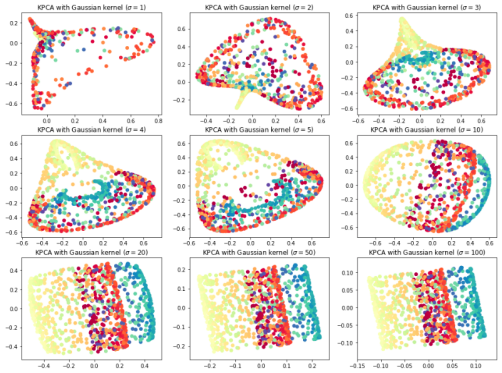

Figure 10. Influence of σ parameter in kernel PCA with Gaussian kernel ($K_{ij} = \exp\left(-0.5\|\mathbf{x_i} - \mathbf{x_j}\|^2/\sigma^2\right)$) on the S-curve manifold

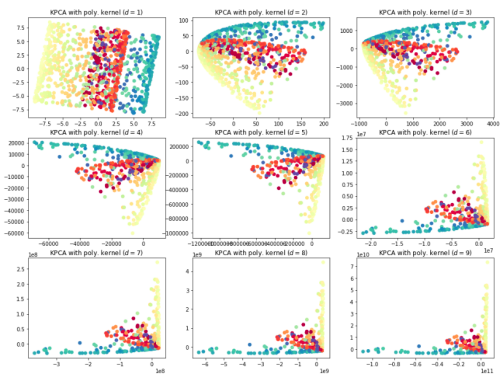

Figure 11. Influence of the degree parameter in kernel PCA with polynomial kernel ($K_{ij} = (1 + \mathbf{x_i}^\top \mathbf{x_j})^d$) on the S-curve manifold

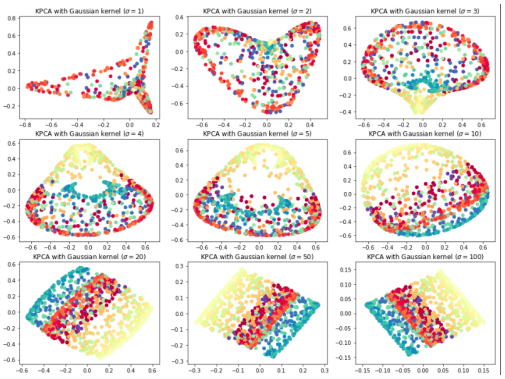

Figure 12. Influence of σ parameter in kernel PCA with Gaussian kernel ($K_{ij} = \exp\left(-0.5\|\mathbf{x_i} - \mathbf{x_j}\|^2/\sigma^2\right)$) on the Swiss-roll manifold

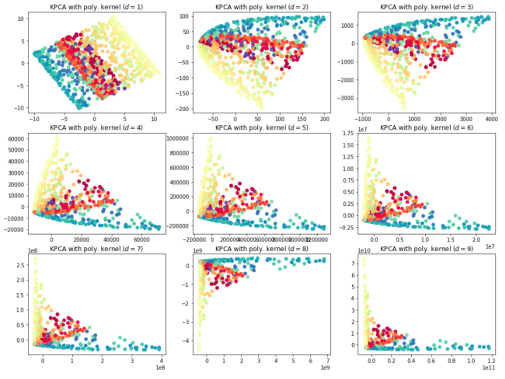

Figure 13. Influence of the degree parameter in kernel PCA with polynomial kernel ($K_{ij} = (1 + \mathbf{x_i}^\top \mathbf{x_j})^d$) on the Swiss-roll manifold

A.6. Laplacian Eigenmaps: influence of parameters

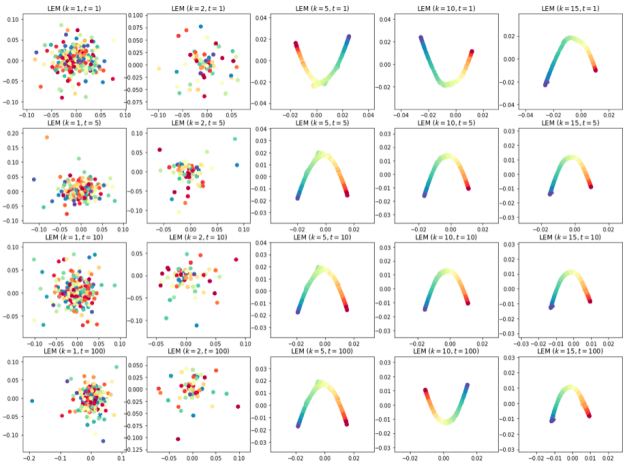

Figure 14. Influence of the heat kernel t and number of neighbours k (LEM) on the S-curve manifold

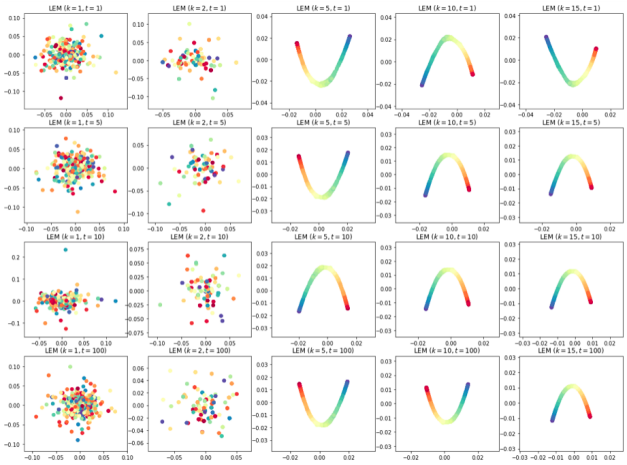

Figure 15. Influence of the heat kernel t and number of neighbours k (LEM) on the Swiss-roll manifold

A.7. Locally Linear Embedding: influence of parameters

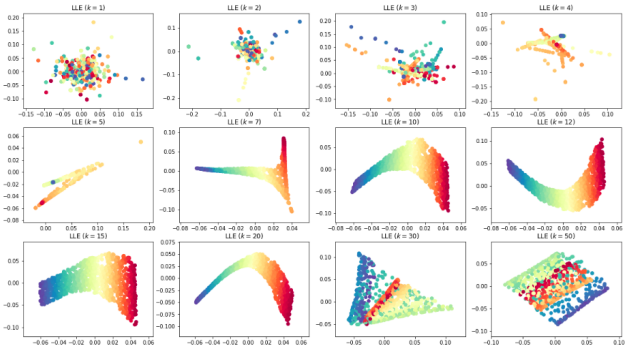

Figure 16. Influence of number of neighbours k (LLE) on the S-curve manifold

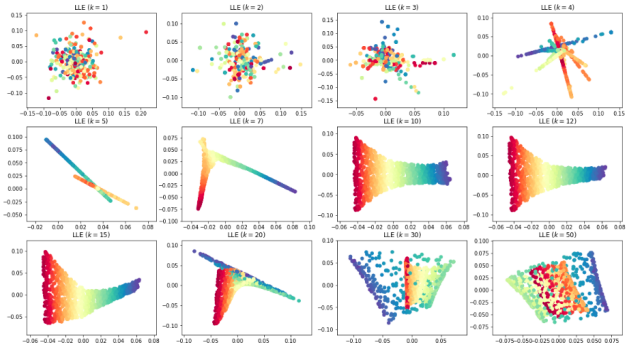

Figure 17. Influence of number of neighbours k (LLE) on the Swiss-roll manifold

A.8. Zoom on MNIST embedding with LEM

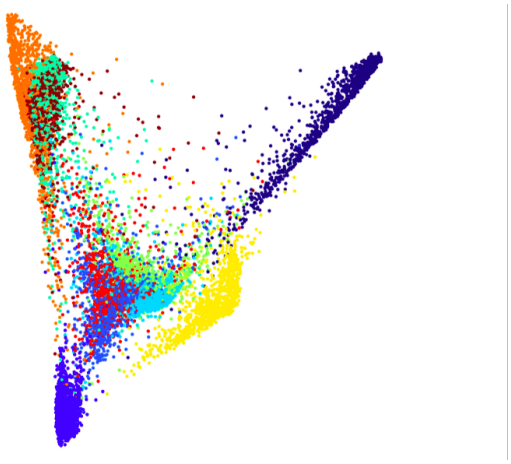

Figure 18. Closer view of the 2D distribution of the new digits from MNIST (10000 first samples) computed by LEM