An Experimental Study of Dimensionality Reduction Methods

Almuth Meier(✉) and Oliver Kramer

Computational Intelligence Group, Department of Computing Science,
University of Oldenburg, Oldenburg, Germany
{almuth.meier,oliver.kramer}@uni-oldenburg.de

Abstract. Dimensionality reduction (DR) lowers the dimensionality of a high-dimensional data set by reducing the number of features for each pattern. The importance of DR techniques for data analysis and visualization led to the development of a large diversity of DR methods. The lack of comprehensive comparative studies makes it difficult to choose the best DR methods for a particular task based on known strengths and weaknesses. To close the gap, this paper presents an extensive experimental study comparing 29 DR methods on 13 artificial and real-world data sets. The performance assessment of the study is based on six quantitative metrics. According to our benchmark and evaluation scheme, the methods mMDS, GPLVM, and PCA turn out to outperform their competitors, although exceptions are revealed for special cases.

Keywords: Dimensionality reduction · Manifold learning · Feature extraction

1 Introduction

High-dimensional data appear in many applications, but are demanding in different ways. High dimensionalities not only challenge storage and network throughput technologies, but also complicate data analysis tasks. For humans, data with a dimensionality larger than three are difficult to understand since no intuitive visualization is possible. Even if machine learning techniques are employed to extract important information from the data, e. g., by clustering or classification, a high dimensionality is impeding as it requires a large training data set (curse of dimensionality) and extends the runtime.

DR computes a mapping $\mathbf{F} : \mathbb{R}^d \rightarrow \mathbb{R}^q$ from a d-dimensional data space to a q-dimensional latent space with $q < d$. Each data point (pattern) from the original data set is mapped to a latent point with only q features. In other words, each pattern is embedded into latent space leading to an embedding (manifold) of the whole data set. The dimensionality q of the latent space (intrinsic dimensionality) is often not determined by the DR methods, but has to be estimated with separate techniques, e. g., maximum likelihood [27]. Instead of an estimation, the user can adapt the intrinsic dimensionality to his needs. For example,

G. Kern-Isberner et al. (Eds.): KI 2017, LNAI 10505, pp. 178–192, 2017.
DOI: 10.1007/978-3-319-67190-1_14

an intrinsic dimensionality less or equal three is often chosen if DR is used to visualize data.

The way of computing mapping **F** significantly differs among the DR methods. Unlike feature selection, feature extraction methods generate completely new dimensions, which are combinations of the old ones and are therefore not directly interpretable. Our study exclusively concentrates on feature extraction methods. We include parametric methods that explicitly learn **F** and its parameters, and that are able to embed new unknown patterns. But we also concentrate on non-parametric DR methods that directly map high-dimensional patterns to latent points and thus modeling **F**. The study also comprises numerous convex and non-convex techniques. Convex methods use convex objective functions that guarantee to find the corresponding optimum, while non-convex methods might yield better mappings, in particular for non-linear data, but do not guarantee to find the best solution of their objective function. Furthermore, the methods can be grouped into families which apply similar mathematical concepts.

Due to the fact that new DR methods are often compared only against older established DR methods, like PCA [18,37], Isomap [47] or LLE [40], but not against newer ones, overall quality differences are not transparent. In most existing studies only few data sets and few quantitative measures are used deteriorating the understanding, why methods have specific strengths and weaknesses. These reasons hamper a reasonable performance evaluation of DR methods for defined applications and motivate the comparative study this paper presents.

This work is structured as follows: We review existing comparative studies in Sect. 2 and explain the setup of our experiments in Sect. 3. Afterwards, we evaluate the outcomes of our experiments in Sect. 4. A summary concludes this paper in Sect. 5.

2 Current Comparative Studies

In the current literature, various contributions giving an overview of DR techniques exist, e. g., [10,25]. They describe method design and applied mathematical concepts, but do not include empirical comparisons. Therefore, they do not give insights into differences between the methods regarding practical usage.

Other contributions, e. g., by Gisbrecht and Hammer [12], investigate the suitability of DR methods for visualization tasks. They embed high-dimensional data sets with different DR methods, visualize the resulting manifolds and assess the embedding quality with one quantitative measure. Nevertheless, these comparisons are not satisfactory as data sets, method diversity and metrics run too short.

The most extensive quantitative study is presented by van der Maaten et al. [33]. Newer methods like UNN [20], EE [5] or t-SNE [32] are not included. Mysling et al. [35] conduct a quantitative comparison that examines the dependence of four DR methods on data set properties, like data density and noise, in terms of a classification and a regression error. Also Yeh et al. [52] present a limited comparison with three DR methods on one data set with respect to one metric

that assesses the methods' suitability as pre-processing techniques before clustering. Furthermore, some studies exist that compare DR methods solely for specific applications, e. g., by Niskanen and Silvén [36].

3 Experimental Setup

Our experimental study comprises 29 DR methods, 13 data sets, and six metrics. On each data set, each DR method is executed repeatedly, each time with a separate parameter setting like grid search. Due to the non-deterministic behavior of some DR methods, each of these executions is run 25 times. Only for methods with an extensive runtime only 3 repetitions are conducted. The metrics are computed for each run and are averaged over the 25 repetitions. For each DR method we aggregate one value per data set and metric, i.e., the best one the DR method has achieved among all parameter settings.

3.1 DR Methods

We selected the DR methods in our study with the objective to include at least one method from each family of unsupervised feature extraction methods (Fig. 1). The convex DR methods are based on an eigenvalue (or spectral) decomposition. They can be subdivided according to whether the eigenvalue decomposition is performed for a sparse or full matrix. Within both categories the DR methods employ different mathematical techniques. Kernels make DR methods capture non-linear structures in the data. Neighborhood graphs are used to set up a distance matrix containing the similarities of neighbored patterns, which are often measured in terms of Euclidean distances. DR methods use the distance matrix, but can only embed patterns belonging to the largest sub-graph. Since our evaluation requires an embedding for all patterns we embed the remaining patterns with a so called out-of-sample extension implemented by van der Maaten [31].

Concepts used by non-convex methods are unsupervised regression, neural networks, and probabilistic approaches. Regression methods optimize the latent points so that the patterns reconstructed from the latent points by k-nearest neighbors (kNN) regression differ as little as possible from the original patterns. In case of so called autoencoders, neural networks for DR have an odd number of layers and the middle layer of neurons represents the latent point belonging to the input pattern. During the training procedure the weights are optimized so that the output of the network, i. e., the reconstructed pattern, is similar to the input, i. e., the original pattern.

Probabilistic DR methods can also be divided into different families: methods based on the latent variable model (LVM), techniques employing a mixture model and methods that use probabilities as a measure for the similarity of patterns and latent points. Methods based on the LVM assume that the features of the observed patterns are random variables underlying a common probability distribution, which actually is based on a smaller set of unobservable random

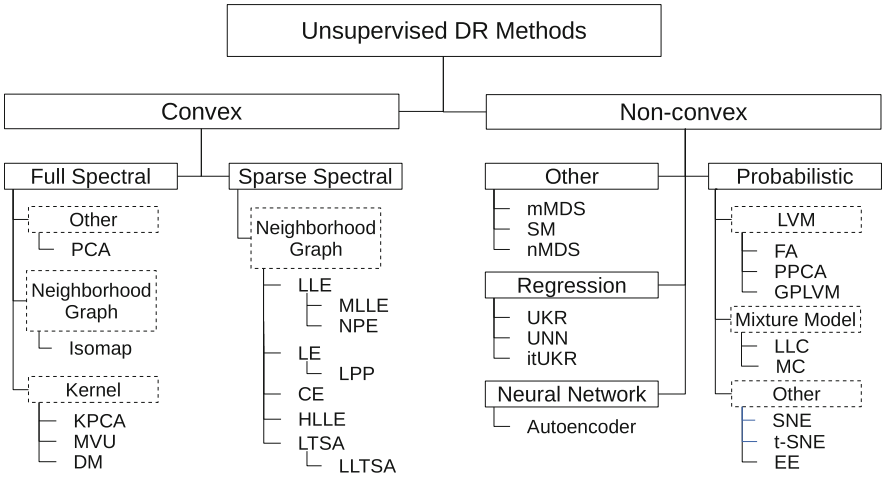


Fig. 1. A taxonomy of employed unsupervised DR methods, based on [33]

variables, so called latent variables. The values of the latent variables are the latent points. They are optimized along with the parameters of the probability distribution to maximize the likelihood for observing the patterns. A mixture model is an aggregation of multiple density estimators to one larger estimator. Its parameters and the latent points are optimized similar to the optimization procedure of the LVM.

This experimental study is based on the following methods, which we also arranged in a taxonomy (Fig. 1): CE (Conformal Eigenmaps) [44], DM (Diffusion Maps) [6,7], EE (Elastic Embedding), FA (Factor Analysis) [45], GPLVM (Gaussian Process LVM) [24], HLLE (Hessian LLE) [9], Isomap (Isometric Feature Mapping), itUKR (iterative UKR) [29], KPCA (Kernel PCA) [43], LE (Laplacian Eigenmaps) [3], LLC (Locally Linear Coordination) [46], LLE (Locally Linear Embedding), LLTSA (Linear LTSA) [53], LPP (Locality Preserving Projections) [14], LTSA (Local Tangent Space Alignment) [54], MC (Manifold Charting) [4], MDS (Multidimensional Scaling), MLLE (Modified LLE) [55], mMDS (metric MDS) [48], MVU (Maximum Variance Unfolding) [50,51], nMDS (nonmetric MDS) [23], NPE (Neighborhood Preserving Embedding) [13], PCA (Principal Component Analysis), PPCA (Probabilistic PCA) [39], SM (Sammon Mapping) [41], SNE (Stochastic Neighbor Embedding) [16], t-SNE (t-Distributed SNE), UKR (Unsupervised Kernel Regression) [34], and UNN (Unsupervised Nearest Neighbors). The autoencoder was proposed in [8,17].

We use the following implementations: SCIKIT-LEARN [38] for mMDS, nMDS, MLLE and HLLE, MATLAB code by Vladymyrov and Carreira-Perpiñán for EE according to [49], MATLAB toolbox by Klanke [19] for UKR, self-implementation of UNN in PYTHON in accordance with Kramer [21, Sect. 4.1], PYTHON code for itUKR from its author and the MATLAB toolbox by van der Maaten [31] for all other methods.

Table 1. Parameter ranges of convex DR methods

Technique	Parameter ranges
DM	$t \in \{1, 10, 30, 50, 70, 90\}, \sigma \in \{0.2, 1.0, 5.0\}$
LE, LPP	$k \in \{5, 9, 13, 50, 100\}, \sigma \in \{0.2, 1.0, 5.0\}$
KPCA	$i \in \{100, 200, 300, 400, 500\}$
Isomap, MVU, LLE, MLLE, NPE, CE, HLLE, LTSA, LLTSA	$k \in \{5, 7, 9, 11, 13, 15, 50, 100\}$

Table 2. Parameter ranges of non-convex DR methods

Technique	Parameter ranges
PPCA	$i \in \{100, 200, 300, 400, 500\}$
GPLVM	$\sigma \in \{0.2, 0.5, 1.0, 2.5, 5.0\}$
LLC	$k \in \{5, 13, 100\}, a \in \{2, 9, 20\}, i \in \{200, 400\}$
MC	$a \in \{2, 7, 12, 20\}, i \in \{200, 400\}$
SNE, t-SNE	$p \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$
EE	$h \in \{0.01, 0.1, 0.0, 10.0, 100.0\}, p \in \{5, 25, 50\}$
UKR	$kernel \in \{\text{Gaussian}, \text{Quartic}, \text{Triweight}\}$
UNN	$k \in \{5, 10, 20, 40\}, \kappa \in \{10, 30, 50\}$
itUKR	$\kappa \in \{30, 50, 70\}, bandwidth \in \{10, 20, 30, 40, 50\}$
Autoenc	$\lambda \in \{0.0, 0.2, 0.5, 1.0, 1.5, 2.5, 5.0\}$

The parameter settings we employ are listed in Tables 1 and 2; methods without parameters are not included. For a description of the parameters we refer to the documentation of the respective implementation. We executed each method for each parameter value combination except the autoencoder. Due to an extensive runtime we ran the autoencoder on the data sets CNS and ORL (Sect. 3.3) only with setting $\lambda = 0.0$. We chose value 0.0 since a larger λ often led to NaN metric values on other data sets because latent points were mapped nearly to the same point.

3.2 Metrics

No single criterion for DR methods exists, since the information to be preserved depends on the data set and the purpose of the DR task. Therefore, we assess the methods' quality with different metrics. On the one hand, we measure the topology preservation in terms of neighborhood preservation (ENX), distance preservation (EKS) and structure preservation in general (DSRE, E1NN, CRR). On the other hand, we measure the DR methods' ability for preceding a classification or regression task in terms of information preservation (EKNN, CRR). These metrics seem to be reasonable as it is assumed that most important information of a data set is encoded in its spacial properties. We adapt some

metrics so that all metrics are error measures, i. e., lower values represent better qualities.

The ENX measure becomes better the more latent points belonging to neighbored patterns are neighbors, i. e. the more neighborhoods are preserved. It is based on the co-ranking matrix $\mathbf{Q}$ [26]. Following Lueks et al. [30], we adapt ENX to

$$\mathrm{ENX}\,(k) = 1 - \frac{1}{nk} \sum_{i=1}^{k} \sum_{j=1}^{k} q_{ij}, \tag{1}$$

where q_{ij} is an entry of $\mathbf{Q}$, variable n is the number of patterns or latent points and k is the neighborhood size.

EKS [22] is the objective function of the MDS variants and is computed from the squared Frobenius norm of the distance of the normalized distance matrices of patterns ($\mathbf{D_P}$) and latent points ($\mathbf{D_L}$):

$$\mathrm{EKS} = \|\mathbf{D_P} - \mathbf{D_L}\|_F^2. \tag{2}$$

The distance matrices are normalized by dividing each value by the largest value of the respective matrix. A small EKS indicates similar distances between latent points and their corresponding patterns.

The DSRE [20] measures how well the patterns can be reconstructed from the latent points by applying the kNN regression model $\mathbf{f_L}$ with

$$\mathbf{f_L} : \mathbb{R}^q \to \mathbb{R}^d,\ \mathbf{f_L}\,(\mathbf{l}) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(\mathbf{l}, \mathbf{L})} \mathbf{p}_i. \tag{3}$$

Let $\mathbf{P} \in \mathbb{R}^{n \times d}$ denote a pattern matrix, $\mathbf{L} \in \mathbb{R}^{n \times q}$ the corresponding matrix of latent points, $\mathbf{p}_i \in \mathbb{R}^d$ the ith pattern and $\mathbf{l} \in \mathbb{R}^q$ a latent point. The set $\mathcal{N}_k\,(\mathbf{l}, \mathbf{L})$ contains the indices of the k latent points from $\mathbf{L}$ that are most similar to $\mathbf{l}$. Then the DSRE is defined as

$$\mathrm{DSRE}(\mathbf{L}, k) = \|\mathbf{P} - \mathbf{f_L}(\mathbf{L})\|_F^2, \tag{4}$$

where $\mathbf{f_L}(\mathbf{L}) \in \mathbb{R}^{n \times d}$ is a matrix containing the reconstructions of all patterns. A good DSRE is attained if latent points of neighbored patterns are neighbored.

E1NN [42] measures the overlapping of points with labels from different classes in the data and the latent space. In a w. r. t. E1NN optimal DR process latent points from different classes are clearly separated. This could be desired e. g. for visualization tasks. E1NN counts the number l^- of latent points whose next neighbor has a label from a different class and divides it by p^-, which is analogously defined for patterns:

$$\mathrm{E1NN} = \frac{l^-}{p^-}. \tag{5}$$

An E1NN larger than one indicates a worse structure of the manifold compared to the data space.

CRR [36] calculates the ratio of the number of falsely classified latent points and the number of falsely classified patterns. EKNN [22] is the counterpart of CRR for regression. It is the ratio of the regression error of the latent space and the regression error of the data space. CRR and EKNN use the kNN classification and regression model, respectively. They are useful to examine whether a classification or regression task, respectively, would be more successful on the original or reduced data set.

CRR and E1NN are only applicable to data sets with discrete labels, EKNN only to data sets with continuous labels, whereas ENX, DSRE and EKS are suitable for all data sets. We set the metrics' parameter k, representing the neighborhood size, to 15. Based on the results of Lee and Verleysen [26] this seems to be a reasonable compromise between fluctuating metric values for a very small k and smooth values for a large k.

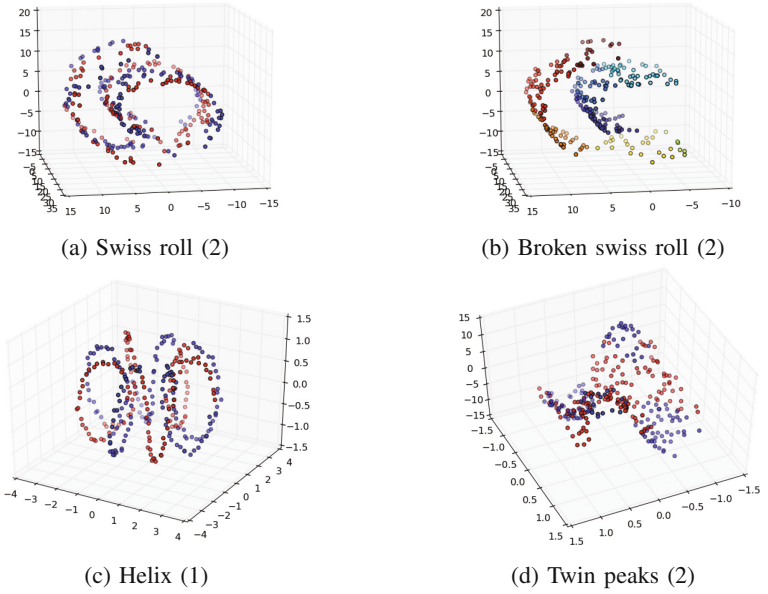
3.3 Data Sets

We apply the DR methods to five artificial and eight real-world data sets. The artificial data sets come from the comparative study of van der Maaten et al. [33]. Swiss roll, Broken swiss roll, Helix and Twin peaks are shown in Fig. 2, HD is a 10-dimensional data set with a 5-dimensional manifold. They have known manifolds and are generated with the MATLAB toolbox by van der Maaten [31].

The real-world data sets stem from different domains and employ different dimensionalities. In Table 3, their properties are listed. The intrinsic dimensionalities are estimated with the maximum likelihood estimator implementation by van der Maaten [31]. For the MNIST data set, only the GMST estimator from the same toolbox computed a reasonable dimensionality. For the experiments we randomly select 300 patterns from each data set, except for Iris and CNS since they contain less patterns.

Table 3. Properties of real-world data sets

Data set	Dim.	Intrins. dim.	Label type	#Classes	Description
Iris [38]	4	3	Discrete	3	Plant properties of iris flowers
Boston [38]	13	2	Contin.	-	House properties
Digits [38]	64	17	Discrete	10	Pictures of handwritten digits
RCT [28]	386	28	Contin.	-	CT pictures of different persons
MNIST [28]	784	2	Discrete	10	Pictures of handwritten digits
HIVA [2]	1617	15	Discrete	2	Properties of drugs
CNS [15]	7130	30	Discrete	2	Gene data of tumor patients
ORL [1]	10304	7	Discrete	40	Faces of different persons



(a) Swiss roll (2)

(b) Broken swiss roll (2)

(c) Helix (1)

(d) Twin peaks (2)

Fig. 2. Artificial data sets and intrinsic dimensionalities. Colors represent labels.

4 Experimental Results

Before the actual evaluation, we examine the statistical significance of the differences between the DR methods. We conduct one Friedman test per metric leading to statistically significant p values (ENX: 4.25e−28, EKS: 1.65e−43, DSRE: 4.57e−66, E1NN: 1.94e−36, CRR: 7.58e−36). For EKNN, no test can be conducted since EKNN requires data sets with continuous labels but only two such data sets are included in our study. For the analysis of the metric values we perform two steps. First, we rank the methods and analyze their rank differences. Second, we compute quality differences that give better insight into the methods' performance. In both steps we compare the methods' performance with respect to both the metrics and the data sets.

4.1 Analysis of Rank Differences

Each DR technique is separately assigned to a rank so that each technique T employs a rank $R_T(D, M)$ for each data set D and metric M. For a more manageable comparison, we average the ranks of each technique in two ways, i. e., over all data sets resulting in one average rank per metric $\varnothing_{R_T}(M)$ and over all metrics resulting in one average rank per data set $\varnothing_{R_T}(D)$. Figures 3 and 4 show the average ranks for the convex and non-convex methods with the best

Technique	ENX	EKS	DSRE	E1NN	CRR	EKNN	Ø
PCA	10.8	7.0	12.0	16.9	15.4	11.5	12.3
Isomap	12.8	9.2	13.7	9.7	8.3	15.5	11.4
MLLE	10.8	15.9	10.3	10.9	7.6	10.5	10.9
LE	10.8	15.2	13.4	6.5	7.5	16.5	11.6
mMDS	7.4	2.7	8.2	11.5	14.1	14.5	11.1
GPLVM	10.8	6.6	11.4	15.3	14.7	10.0	11.5
t-SNE	4.9	17.2	8.2	5.4	6.1	5.5	7.8
EE	5.1	8.5	7.1	4.2	4.0	11.5	6.6
UKR	12.2	15.9	3.0	9.0	6.5	1.5	8.0

Fig. 3. Average ranks per metric for best-ranked convex (at the top) and non-convex methods (at the bottom). Column ∅ contains the row averages. The color gradient visualizes the differences of values within columns: small (yellow) values are better than large (red) ones. (Color figure online)

Technique	Swiss	Broken	Helix	Twin	HD	Iris	Boston	Digits	RCT	MNIST	HIVA	CNS	ORL	Ø(a.)	Ø(r.)	Diff.
PCA	21.4	20.6	23.0	11.2	12.4	5.6	15.3	4.2	3.8	18.6	7.8	6.2	7.0	17.7	8.6	9.2
Isomap	8.6	4.8	9.6	6.6	18.4	12.0	13.3	14.8	9.0	14.0	8.0	7.0	17.0	9.6	11.9	2.3
MLLE	10.6	6.4	8.4	12.4	11.0	17.2	10.3	7.6	17.5	10.2	7.2	14.8	13.2	9.8	12.2	2.5
LE	8.2	11.4	8.2	12.8	17.2	16.2	21.0	8.4	9.3	5.2	6.6	10.4	11.0	11.6	11.0	0.6
mMDS	9.0	11.6	20.2	5.4	8.6	8.0	11.3	5.2	4.8	11.2	5.2	6.4	6.0	11.0	7.3	3.7
GPLVM	21.0	20.2	23.0	11.0	11.0	5.6	14.5	3.8	3.3	17.4	4.6	6.2	7.0	17.2	7.8	9.4
t-SNE	6.2	5.4	7.0	9.4	18.8	11.2	6.8	9.6	7.5	1.8	11.8	6.8	7.0	9.4	7.8	1.6
EE	6.0	4.6	4.0	10.8	9.2	6.8	8.5	5.0	11.5	2.8	4.2	5.2	1.8	6.9	5.7	1.2
UKR	5.8	7.0	7.8	13.4	21.4	16.4	8.8	7.6	8.3	2.8	6.6	7.2	6.2	11.1	8.0	3.1

Fig. 4. Average ranks per data set for best-ranked convex (at the top) and non-convex methods (at the bottom). Data sets are separated into artificial (left) and real-world ones (right). Columns $\varnothing(a.)$ and $\varnothing(r.)$ contain the row averages for artificial and real-world data sets, respectively. The differences between both are listed in the last column.

average rank among all methods (i. e. with best value in column ∅ of Fig. 3), which were able to embed all data sets.[1]

Considering the average ranks per metric (Fig. 3), in particular EE, t-SNE and UKR achieve promising results, while no method performs best w. r. t. all metrics. Taking into account the average ranks per data set (Fig. 4), it can be observed that the presented methods (except Isomap and MLLE) perform better on real-world data sets, due to lower values in column $\varnothing(r.)$ than in $\varnothing(a.)$. Furthermore, on real-world data sets clearly better results of non-convex methods in contrast to convex methods can be observed. However, PCA is nearly as good as other non-convex methods.

[1] MVU, CE, NPE, LPP, HLLE, LLTSA (convex) and SM, FA, MC (non-convex) were not able to embed up to six real-world data sets due to, e. g., not computable eigenvalue problems, extensive runtime (more than four weeks) or too few patterns.

4.2 Analysis of Quality Differences

To compare the methods' quality differences, based on the percentage of the maximum accuracy employed by Fernández-Delgado et al. [11] we compute for each DR technique T the percentage of the minimum error (PME) per metric M with

$$\mathrm{PME}_T(M) = \frac{1}{13} \sum_{D \in \{\text{Swiss roll},\ldots,\text{ORL}\}} \frac{\text{value for } M \text{ achieved by } T \text{ on } D}{M^* \text{ on } D}$$

and equivalently per data set D with

$$\mathrm{PME}_T(D) = \frac{1}{6} \sum_{M \in \{\text{ENX},\ldots,\text{EKNN}\}} \frac{\text{value for } M \text{ achieved by } T \text{ on } D}{M^* \text{ on } D},$$

where M^* is the best (minimum) value for metric M that any DR technique has achieved on data set D. For example, $\mathrm{PME}_{\mathrm{PCA}}(\mathrm{ENX}) = 2.0$ denotes that the ENX values of PCA are on average two times worse than the best ENX value of any DR method. For each DR method only the data sets are included into the average for the PME measure, which the method was able to embed. This may unfairly improve the PME values of failing methods since their missing metric values for the respective data sets do not influence their PME values, whereas the possibly poor metric values of other methods deteriorate their PME values.

In the left part of Fig. 5 the nine best PME values per metric and the corresponding DR techniques are listed. It is notable that the differences of the PME values are approximately equal among the metrics except for EKS. This has two reasons. First, SM and mMDS are unfairly preferred since the EKS is their objective function. Second, the methods' absolute values for EKS differ stronger than their values for other metrics, in particular on real-word data sets. Hence, the PME values of DR techniques with failed embedding attempts are improved especially regarding the EKS. Averaging the PME values over all metrics and ignoring the failing methods, e. g., SM, CE and MVU (Sect. 4.1), shows that mMDS, GPLVM and PCA are by far the best DR methods (Fig. 5, middle part). This is surprising since mMDS and PCA belong to the earliest DR methods.

Comparing the results of the rank and the PME analysis regarding the metrics it can be observed that the methods with best ranks (EE, t-SNE, UKR) surprisingly do not always have best PME values. This is mainly caused by the poor quality of their embeddings regarding the EKS. Averaging the PME values without EKS (Fig. 5, right part) again leads to t-SNE, EE and UKR as best methods. Interestingly Isomap, MLLE and LE are among the best-ranked methods despite their poor average PME values (Fig. 5, middle part).

At last we analyze the PME values per data set. Figure 6 contains the PME values for the best convex and non-convex methods according to their average PME values listed in the middle part of Fig. 5. The symbols ### signify a failed embedding attempt; reasons for them are given in Sect. 4.1. We observe that all best convex and non-convex methods have a better average PME value

Technique	ENX	Technique	EKS	Technique	DSRE	Technique	E1NN	Technique	CRR	Technique	EKNN
SM	**1.8**	SM	**1.1**	UKR	**1.2**	t-SNE	**1.3**	Isomap	**1.6**	MVU	**1.0**
mMDS	1.9	mMDS	1.3	t-SNE	1.4	UKR	2.1	UKR	1.6	UKR	1.0
PCA	2.0	CE	1.5	EE	1.7	EE	2.7	EE	1.6	SM	1.1
GPLVM	2.0	MVU	1.6	itUKR	2.1	LE	3.0	t-SNE	1.7	NPE	1.1
MVU	2.1	GPLVM	3.2	UNN	2.3	itUKR	3.3	LE	2.0	GPLVM	1.1
CE	2.1	PCA	3.2	LE	2.6	Isomap	3.8	itUKR	2.0	PCA	1.1
t-SNE	2.4	LLTSA	15.7	MLLE	2.7	MLLE	3.9	MLLE	2.1	HLLE	1.1
EE	2.5	FA	76.8	SNE	2.7	UNN	3.9	DM	2.1	SNE	1.1
HLLE	2.8	Isomap	77.3	mMDS	2.9	LTSA	4.0	HLLE	2.1	CE	1.1

Technique	Ø
mMDS	**2.4**
SM	2.5
CE	2.7
MVU	2.8
GPLVM	2.8
PCA	2.8
LLTSA	6.1
Isomap	15.1
FA	15.9

Technique	Ø w/o EKS
t-SNE	1.6
EE	1.9
UKR	2.1
LE	2.5
mMDS	2.6
Isomap	2.6
itUKR	2.7
GPLVM	2.7
PCA	2.8

Fig. 5. PME values per metric in ascending order (left part), average PME per method (middle part), and average PME without values for EKS (right part). Smaller (yellow) values are better, convex methods are highlighted blue. (Color figure online)

Technique	Swiss	Broken	Helix	Twin	HD	Iris	Boston	Digits	RCT	MNIST	HIVA	CNS	ORL	Ø(a.)	Ø(r.)	Diff.
PCA	2.9	3.0	12.8	2.5	1.5	1.3	1.7	1.5	4.2	2.0	1.8	2.9	1.9	4.5	2.2	2.4
MVU	2.5	3.0	11.3	2.1	1.4	1.3	1.8	###	###	1.7	###	###	2.0	4.0	1.7	2.3
CE	2.5	2.8	11.6	1.9	1.4	1.3	1.7	###	###	1.6	###	###	1.7	4.1	1.6	2.5
mMDS	2.1	2.6	12.1	1.6	1.2	1.5	1.4	1.6	1.9	1.7	1.4	1.3	1.8	3.9	1.6	2.3
SM	2.2	2.5	12.9	1.8	1.1	###	1.4	1.2	1.1	1.8	1.3	###	1.7	4.1	1.4	2.7
GPLVM	2.9	3.0	12.8	2.5	1.4	1.3	1.7	1.5	4.1	2.0	1.7	2.9	1.9	4.5	2.1	2.4

Fig. 6. PME values per data set for methods with best average PME (Fig. 5, middle part)

on real-world than on artificial data sets (Fig. 6, columns $\varnothing(r.)$ and $\varnothing(a.)$). This observation is consistent with the results from the rank analysis, albeit no differences between convex and non-convex methods can be observed. The suitability for practical usage of the methods listed in Fig. 6 is emphasized by the finding that all other methods except LLTSA perform much worse on real-world than on artificial data sets. In general, the PME values per data set have to be interpreted critically because they are strongly influenced by the EKS due to the large differences between best and worse EKS values among the DR methods. This strong influence not necessarily reflects the actual importance of the EKS as quality measure, which may depend on the application purpose.

In summary, mMDS, GPLVM and PCA score well in both evaluation parts. EE, t-SNE and UKR only perform well regarding the ranks, but do not have satisfying PME values due to their poor EKS values. In contrast, there are some methods (SM, CE, MVU) that seem to be quite good but fail on some real-world data sets.

5 Summary

In this paper we present a quantitative experimental study that assesses the quality of 29 DR methods on 13 artificial and real-world data sets with six metrics. It goes beyond previous comparisons by employing more and newer methods from all families of unsupervised feature extraction methods using a larger number of data sets and examining a variety of quality properties.

Based on our analysis mMDS, GPLVM and PCA are the overall best DR methods. However, depending on which metrics actually are reasonable quality measures for a specific application, other methods may be better choices, like e. g., EE, t-SNE, and UKR if distance preservation is negligible. Depending on the data set the experimental results of this paper may guide the choice of methods in real applications.

Of course, our findings can only be generalized to a certain extend due to the no free lunch theorem. But as an extensive experimental analysis is often impossible due to time and cost constraints, we recommend to choose a reasonable subset of DR methods based on our results, to perform an own evaluation and to select the best among these methods for performing the final DR task. According to the application, suitable metrics should be chosen for the analysis. For example, in a pipeline with classification, CRR and EKNN are appropriate test candidates as they assess information preservation.

Future work may concentrate on an extension and update of the experimental analysis w. r. t. the set of benchmark methods, problems, and test metrics.

References

1. The database of faces. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
2. World congress on computational intelligence (WCCI) performance prediction challenge (2006). http://www.modelselect.inf.ethz.ch/datasets.php
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
4. Brand, M.: Charting a manifold. In: Advances in Neural Information Processing Systems (NIPS), pp. 985–992. MIT Press (2003)
5. Carreira-Perpiñán, M.Á.: The elastic embedding algorithm for dimensionality reduction. In: International Conference on Machine Learning (ICML), pp. 167–174 (2010)
6. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. Natl. Acad. Sci. U. S. A. (PNAS) **102**(21), 7426–7431 (2005)
7. Coifman, R.R., Lafon, S.: Diffusion maps. Appl. Comput. Harmon. Anal. **21**(1), 5–30 (2006)
8. DeMers, D., Cottrell, G.W.: Non-linear dimensionality reduction. In: Hanson, S.J., Cowan, J., Giles, L. (eds.) Advances in Neural Information Processing Systems (NIPS), pp. 580–587. Morgan-Kaufmann, San Mateo (1992)
9. Donoho, D.L., Grimes, C.: Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. Natl. Acad. Sci. U. S. A. (PNAS) **100**(10), 5591–5596 (2003)
10. Dzemyda, G., Kurasova, O., Žilinskas, J.: Multidimensional Data Visualization: Methods and Applications. Springer, New York (2013)
11. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. (JMLR) **15**(1), 3133–3181 (2014)

12. Gisbrecht, A., Hammer, B.: Data visualization by nonlinear dimensionality reduction. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **5**(2), 51–73 (2015)
13. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: International Conference on Computer Vision (ICCV), pp. 1208–1213 (2005)
14. He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing Systems (NIPS), pp. 153–160. MIT Press (2004)
15. Henschel, S., Hoyer, P.O., Ong, C.S., Sonnenburg, S., Braun, M.L.: Machine learning data set repository. http://mldata.org/repository/data/viewslug/central-nervous-system/
16. Hinton, G.E., Roweis, S.T.: Stochastic neighbor embedding. In: Advances in Neural Information Processing Systems (NIPS), pp. 857–864. MIT Press (2002)
17. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
18. Hotelling, H.: Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. **24**(6), 417 (1933)
19. Klanke, S.: http://www.sklanke.de/
20. Kramer, O.: Dimensionality Reduction with Unsupervised Nearest Neighbors. Springer, Heidelberg (2013)
21. Kramer, O.: Unsupervised nearest neighbor regression for dimensionality reduction. Springer Soft Comput. **19**(6), 1647–1661 (2015)
22. Kramer, O.: Machine Learning for Evolution Strategies. Springer, Heidelberg (2016)
23. Kruskal, J.B.: Nonmetric multidimensional scaling: a numerical method. Psychometrika **29**(2), 115–129 (1964)
24. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems (NIPS), pp. 329–336. MIT Press, Cambridge (2003)
25. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Springer, New York (2007)
26. Lee, J.A., Verleysen, M.: Quality assessment of dimensionality reduction: rank-based criteria. Neurocomputing **72**(7), 1431–1443 (2009)
27. Levina, E., Bickel, P.J.: Maximum likelihood estimation of intrinsic dimension. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems (NIPS), pp. 777–784. MIT Press, Cambridge (2005)
28. Lichman, M.: UCI machine learning repository. http://archive.ics.uci.edu/ml
29. Lückehe, D., Kramer, O.: Leaving local optima in unsupervised kernel regression. In: Wermter, S., Weber, C., Duch, W., Honkela, T., Koprinkova-Hristova, P., Magg, S., Palm, G., Villa, A.E.P. (eds.) ICANN 2014. LNCS, vol. 8681, pp. 137–144. Springer, Cham (2014). doi:10.1007/978-3-319-11179-7_18
30. Lueks, W., Mokbel, B., Biehl, M., Hammer, B.: How to evaluate dimensionality reduction? - Improving the co-ranking matrix. CoRR abs/1110.3917 (2011)
31. van der Maaten, L.: Matlab toolbox for dimensionality reduction. https://lvdmaaten.github.io/drtoolbox/
32. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. (JMLR) **9**(2579–2605), 85 (2008)
33. van der Maaten, L., Postma, E., van den Herik, H.: Dimensionality reduction: A comparative review. Technical report TiCC-TR 2009-005, Tilburg University (2009)

34. Meinicke, P., Klanke, S., Memisevic, R., Ritter, H.: Principal surfaces from unsupervised kernel regression. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **27**(9), 1379–1391 (2005)
35. Mysling, P., Hauberg, S., Pedersen, K.S.: An empirical study on the performance of spectral manifold learning techniques. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) ICANN 2011. LNCS, vol. 6791, pp. 347–354. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21735-7_43
36. Niskanen, M., Silvén, O.: Comparison of dimensionality reduction methods for wood surface inspection. In: Quality Control by Artificial Vision, pp. 178–188 (2003)
37. Pearson, K.: On lines and planes of closest fit to systems of points in space. Lond. Edinb. Dublin Philos. Mag. J. Sci. **2**(11), 559–572 (1901)
38. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. (JMLR) **12**, 2825–2830 (2011)
39. Roweis, S.T.: EM algorithms for PCA and SPCA. In: Advances in Neural Information Processing Systems (NIPS), pp. 626–632. MIT Press (1997)
40. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
41. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Trans. Comput. **18**(5), 401–409 (1969)
42. Sanguinetti, G.: Dimensionality reduction of clustered data sets. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **30**(3), 535–540 (2008)
43. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998)
44. Sha, F., Saul, L.K.: Analysis and extension of spectral methods for nonlinear dimensionality reduction. In: International Conference on Machine learning (ICML), pp. 784–791 (2005)
45. Spearman, C.: "General Intelligence", objectively determined and measured. Am. J. Psychol. **15**(2), 201–292 (1904)
46. Teh, Y.W., Roweis, S.T.: Automatic alignment of local representations. In: Advances in Neural Information Processing Systems (NIPS), pp. 865–872. MIT Press (2002)
47. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
48. Torgerson, W.S.: Multidimensional scaling: I. theory and method. Psychometrika **17**(4), 401–419 (1952)
49. Vladymyrov, M., Carreira-Perpiñán, M.Á.: Partial-hessian strategies for fast learning of nonlinear embeddings. In: International Conference on Machine Learning (ICML), pp. 345–352 (2012)
50. Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. In: Computer Vision and Pattern Recognition (CVPR), pp. 988–995 (2004)
51. Weinberger, K.Q., Sha, F., Saul, L.K.: Learning a kernel matrix for nonlinear dimensionality reduction. In: International Conference on Machine Learning (ICML), pp. 839–846 (2004)
52. Yeh, M.C., Lee, I.H., Wu, G., Wu, Y., Chang, E.Y.: Manifold learning, a promised land or work in progress?. In: International Conference on Multimedia and Expo (ICME). IEEE (2005)

53. Zhang, T., Yang, J., Zhao, D., Ge, X.: Linear local tangent space alignment and application to face recognition. Neurocomputing **70**(7), 1547–1553 (2007)
54. Zhang, Z.Y., Zha, H.Y.: Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. J. Shanghai Univ. (Engl. Ed.) **8**(4), 406–424 (2004)
55. Zhang, Z., Wang, J.: MLLE: modified locally linear embedding using multiple weights. In: Advances in Neural Information Processing Systems (NIPS), pp. 1593–1600. MIT Press (2006)