A Dimensionality Reduction Method for Data Visualization using Particle Swarm Optimization

Panagiotis C. Petrantonakis[a] and Ioannis Kompatsiaris[b]
Information Technologies Institute, Centre for Research and Technology - Hellas (CERTH), Thessaloniki, Greece

Keywords: Particle Swarm Optimization, Data Visualization, Dimensionality Reduction.

Abstract: Dimensionality reduction involves mapping of a set of high dimensional input points on a low dimensional space. Mappings in low dimensional space are expected to preserve the pairwise distances of the high dimensional inputs. In this work we present a dimensionality reduction method, called Dimensionality Reduction based on Particle Swarm Optimization (PSO-DR), where the conversion of each input to the low dimensional output does not depend on the rest of the inputs but, instead, it is based on a set of reference points (beacons). The presented approach results in a simple, fast, versatile dimensionality reduction approach with good quality of visualization and straightforward out-of-sample extension.

1 INTRODUCTION

In the era of data deluge, robust dimensionality reduction (DR) tools for visualization of large, high-dimensional data have become an imperative need. The fundamental principle of such tools is to translate high dimensional data so that similar inputs are mapped onto nearby low dimensional representations. Ultimately, DR techniques aim at preserving as much of the high dimensional structure either globally or locally to the low dimensional representation. A variety of such tools for DR have been proposed the last few decades, many of which have been reviewed by Laurens van der Maaten (Van Der Maaten et al., 2009).

DR algorithms fall mainly into two categories. The ones that preserve the distance-wise global structure of the data, such as PCA (Hotelling, 1933), Sammon mapping (Sammon, 1969), and others that preserve the structure of a confined neighborhood (local structure) such as, tSNE (Van Der Maaten and Hinton, 2008), Isomap (Tenenbaum et al., 2000), Laplacian Eigenmaps (Belkin and Niyogi, 2002; Belkin and Niyogi, 2003), and LLE (Hadsell et al., 2006). Both categories have been applied in a wide variety of applications with ever increasing data set sizes. Thus, it is crucial that DR methods are both versatile and fast, in order to cope with massive data.

In this paper we introduce a new DR technique for data visualization based on the Particle Swarm Optimization algorithm (PSO) (Eberhart and Kennedy, 1995; Shi and Eberhart, 1998; Shi and Eberhart, 1999; Shi et al., 2001). The PSO-based DR approach (PSO-DR) seeks to preserve the high dimensional structure by exploiting the fast and versatile nature of the PSO algorithm. We provide the respective algorithmic approach for PSO-DR and present its implementation with parallel, fast computation. We test PSO-DR on four different datasets, either real or artificial, and it exhibits good performance on both categories. We compare the proposed PSO-DR approach with the current state-of-the-art tSNE method along with other linear and nonlinear DR algorithms. Its performance is better or comparable with the most of the state-of-the-art DR approaches and performs faster than those with the best visualization quality, especially for larger data sets. In addition, PSO-DR allows for the mapping of new data points explicitly, in contrast with the majority of the nonlinear DR techniques where approximate estimation of out-of-sample extension leads to mapping errors of new data points. In general, through its simplicity, versatility, fast computation and straightforward out-of-sample extension PSO-DR constitutes an efficient, general purpose, DR technique.

The outline of the paper is as follows. In section 2 we present the methodological approach along with the background information, PSO-DR algorithm and implementation issues. Section 3 describes the experimental set up and presents the corresponding results.

[a] https://orcid.org/0000-0001-9631-4327
[b] https://orcid.org/0000-0001-6447-9020

Petrantonakis, P. and Kompatsiaris, I.
A Dimensionality Reduction Method for Data Visualization using Particle Swarm Optimization.
DOI: 10.5220/0010020601310138
In *Proceedings of the 12th International Joint Conference on Computational Intelligence (IJCCI 2020)*, pages 131-138
ISBN: 978-989-758-475-6

Section 4 discusses the presented outcomes and proposes future research. Finally, section 5 concludes the paper.

2 PSO BASED DATA VISUALIZATION

2.1 Background

The problem of DR is to find a function that maps the high dimensional inputs to a lower dimensional space by preserving the intrinsic structure of the data. Particularly, we seek to map a high dimensional data set $X \subset \mathbb{R}^N$ to a n-D data set $Y \subset \mathbb{R}^n$ in a low dimensional space (usually $n = 2$ or 3, i.e., 2-D or 3-D space). Each low dimensional point $y_i \in Y, i = 1,...,M$ represents the mapping of a corresponding high dimensional point $x_i \in X, i = 1,...,M$. The approach followed, e.g., in classical multidimensional scaling (Torgerson, 1952) is to find all those low dimensional points $y_i, i = 1,...,M$ that minimize the sum of the differences between the pairwise distances in the high dimensional space with the pairwise distances of the low dimensional one, i.e., to minimize:

$$\phi(Y) = \sum_{ij}(d_{ij}^2 - \delta_{ij}^2) \tag{1}$$

where d_{ij} is the distance (dissimilarity measure) between x_i and x_j whereas δ_{ij} is the distance of the corresponding low dimensional points y_i and y_j.

In this work, instead of searching for an optimum solution set, Y, we seek for the optimum low dimensional points one by one. The key step of the proposed approach is to define a set of high dimensional beacons, X_b, i.e., certain, reference points in high dimensional space, and the corresponding low dimensional ones, Y_b, i.e., Y_b is the low dimensional representation of X_b. Then, map, e.g., a point x_i to a point y_i by comparing the distances of x_i from X_b with the distances of y_i from Y_b. In essence, the relative positioning of x_i with respect to X_b in high-dimensional space should be preserved in the low-dimensional space by regulating the positioning of the y_i with respect to Y_b. The following two subsections describe the PSO algorithm and the proposed PSO-DR approach, respectively.

2.2 Particle Swarm Optimization Algorithm

PSO algorithm was introduced by Kennedy and Eberhart in 1995 (Eberhart and Kennedy, 1995), and it was inspired by the social behavior of groups of, e.g., birds in order to solve optimization problems.

PSO algorithm searches the space for optimal solution based on the information shared between the particles of a group. Each particle follows a trajectory which is influenced by stochastic and deterministic components. In particular, each particle moves according to its best achieved position, in terms of the optimization problem, and the best position of the group but with a random component. In every iteration (time point t), a random particle of the group changes its position, $z_i^t, i = 1,...,P$ (P is the population of the particles in the group) according to the new velocity component, v_i^t (Eberhart and Kennedy, 1995; Koziel and Yang, 2011), i.e.:

$$v_i^t = \omega v_i^{t-1} + w_p r_1 (z_{i,p}^{t-1} - z_i^{t-1}) + w_g r_2 (z_g^{t-1} - z_i^{t-1}) \tag{2}$$

$$z_i^t = z_i^t + v_i^t, \tag{3}$$

where $z_{i,p}^{t-1}$ and z_g^{t-1} are the previous best particle and group positions, respectively, ω, w_p, w_g are constant weights and r_1, r_2 are random numbers. Usually, search space and velocity values are bounded whereas the particles are initially distributed randomly in the search space.

2.3 PSO-DR Algorithm

The proposed approach for DR is based on the PSO algorithm to find optimal positions Y that correspond one by one to the high dimensional instances X. Thus, for the search of an optimal solution y_i, a group of particles with positions $z_i, i = 1,...,P$ are moving according to the rule defined by 2 and 3. The function that is minimized is the dissimilarity between the distances of a high dimensional $x_i \in X$ from the high dimensional beacons $x_b^j \in X_b, j = 1,...,J$ and the distances of a low dimensional candidate solution y_i from the corresponding low dimensional beacons $y_b^j \in Y_b, j = 1,...,J$. In particular we seek to minimize:

$$\phi(y_i) = \sqrt{\sum_j (d_j - \delta_j)^2} \tag{4}$$

where $d_j, j = 1,...,J$ are the distances between x_i and $x_b^j, j = 1,...,J$, and $\delta_j, j = 1,...,J$ are the distances between y_i and $y_b^j, j = 1,...,J$. In essence, the optimal solution for Eq. 4, i.e., y_i, is the mapping of the high dimensional instance, x_i, in the low dimensional space based on the relative distances of x_i and y_i from the corresponding beacon sets. Thus, as soon as the high and low dimensional beacons are defined, every high dimensional data can be mapped onto the low dimensional space by minimizing Eq. 4 using PSO.

As high dimensional beacons, X_b, are used as reference points, we do not use any particular selection criterion to define them. Instead, we randomly choose J instances from data set X to use as the beacon set X_b. In order to estimate the corresponding low dimensional beacons, Y_b, we define as y_b^1 (associated with the x_b^1) a zero vector, i.e., $y_b^1 = [0...0] \in \mathbb{R}^n$. Then we proceed with the definition of the rest of the beacons $y_b^j, j = 2, ..., J$ as follows: y_b^2 is estimated by minimizing $\phi(y_b^2) = d_1 - \delta_1$ where d_1 is the distance between x_b^2 and x_b^1, and δ_1 is the distance between the candidate y_b^2 and y_b^1. In accordance, y_b^3 is estimated by minimizing $\phi(y_b^3) = \sqrt{\sum_j (d_j - \delta_j)^2}$ where $d_j, j = 1, 2$ are the respective distances between x_b^3 and x_b^1, x_b^2, and $\delta_j, j = 1, 2$ are the respective distances between the candidate y_b^3 and y_b^1, y_b^2. The rest of the Y_b set is estimated with the same procedure. When the whole Y_b set is defined the rest of the X data set is mapped by minimizing Eq. 4 with respect to the beacons Y_b.

2.4 Implementation Issues

The PSO-DR algorithm was implemented in Matlab 2019a. For the PSO algorithm the SwarmOps-Numerical and Heuristic Optimization toolbox For Matlab was used (toolbox available at: http://www.hvass-labs.org/projects/swarmops/matlab/).

We ran all the experiments on a desktop PC, with Intel Core(TM) i5-9600K at 3.70GHz, and 16 GB of RAM.

The parameters concerning the PSO algorithm was chosen according to the best parameters list presented in (Pedersen, 2010). In particular for the number of particles (swarm-size, P), number of iterations of the PSO algorithm (stopping criterion), ω (inertia weight), w_p (particle's-best weight), w_g (swarm's-best weight) we used $25, 400, 0.3925, 2.5586$, and 1.3358, respectively. These values were used for all experiments conducted in this work.

For comparison reasons, other DR methods were also used. In particular, PCA, tSNE, Isomap, Sammon mapping, LLE, and Laplacian Eigenmaps were compared with PSO-DR. For all these methods the Matlab Toolbox for Dimensionality Reduction by Laurens van der Maaten (Van Der Maaten et al., 2009) was used. For each method the default values of the parameters provided by the toolbox were used.

The number of beacons for each data set was defined to be a quarter of the number of instances x_i in each data set X, i.e., $J = \frac{1}{4}M$ except for the cases where J was larger than 1000; then we set $J = 1000$ irrespectively of the data set size.

3 EXPERIMENTS

The experiments conducted in this work are presented here. We first describe the data sets used for DR and subsequently elaborate on the experimental setup. Finally, we present the respective results.

3.1 Data Sets

Four different data sets were used to evaluate the performance of the PSO-DR algorithm. In particular, the MNIST data set (The MNIST data set is publicly available from http://yann.lecun.com/exdb/mnist/index.html), the COIL-20 data set (Nene et al., 1996), the FMNIST data set (Xiao et al., 2017) and the Swiss Roll data set with $M = 3000$.

The MNIST data set contains $60,000$, 28×28-pixel (i.e., $N = 784$), grayscale images of hand written digits $(0, ..., 9)$. In this work we choose randomly $M = 6,000$ images (600 per class) to perform the experiments. FMNIST data set has the same format as MNIST except that each class represents fashion items. Again, for FMNIST we choose randomly $M = 6,000$ images (600 per class). The COIL-20 data set, contains, 32×32 (i.e., $N = 1,024$) images of 20 different objects which are viewed from 72 orientations, i.e., resulting in $M = 1,440$ images.

3.2 Experimental Setup

For the MNIST, COIL-20, and FMNIST data sets we use the PSO-DR, PCA, tSNE, Isomap, and Sammon mapping techniques to transform the high dimensional representations to a two-dimensional ($n = 2$) map. For the Swiss Roll data set we use the PSO-DR, PCA, Laplacian Eigenmaps, Isomap, and LLE techniques to map to the 2-D space. We substituted tSNE and Sammon mapping with Laplacian Eigenmaps and LLE as techniques that do not employ neighborhood graphs perform poorly on the Swiss Roll dataset (Van Der Maaten et al., 2009).

The resulting maps in each one of the DR task is shown as a scatter plot. The coloring in the scatter plots is used to provide a way of evaluation for the performance of the DR techniques. Moreover, for each one of the DR methods the time needed to map the respective data set is depicted.

For the proposed PSO-DR method, as soon as the estimation of the beacons Y_b is completed, the mapping of, e.g., a high dimensional input x_i to the low dimensional space is independent of the mapping of any other input $x_j, j \neq i$. Thus, it is possible to map simultaneously multiple inputs. Hence, due to its straight-

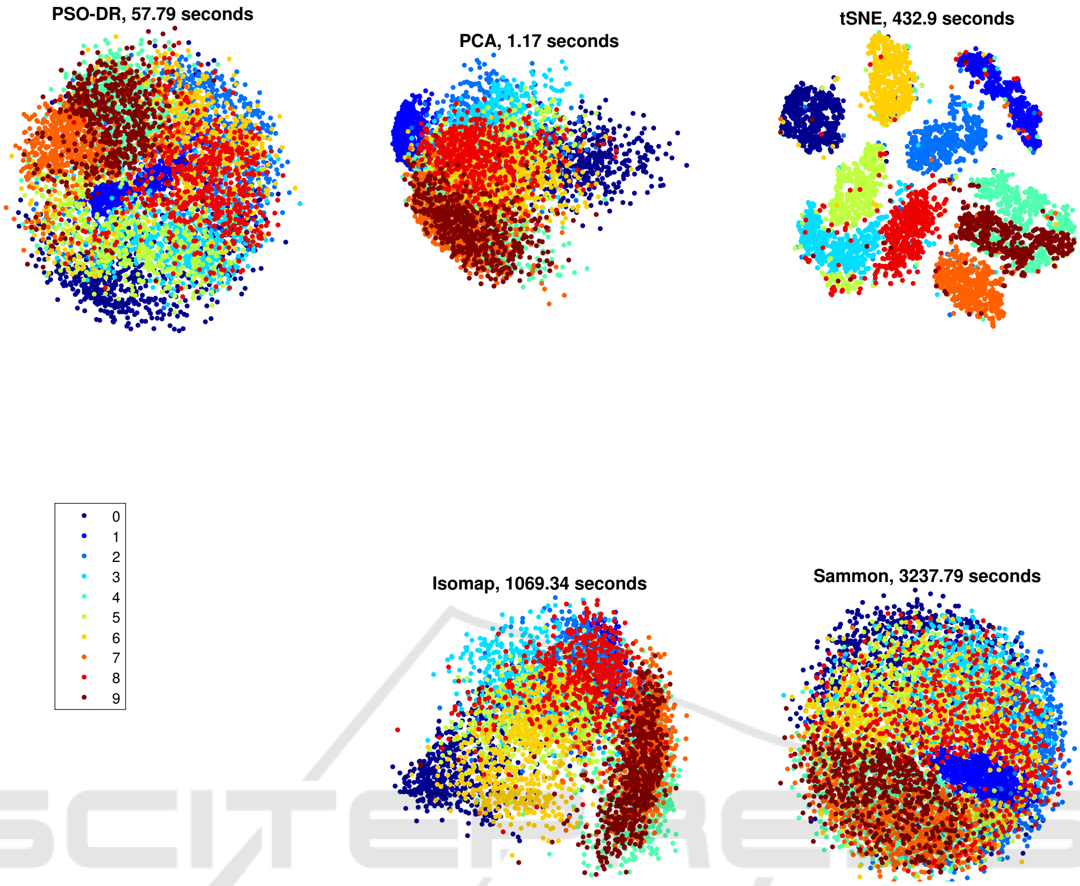

Figure 1: Visualization of the MNIST data set (6,000 digits) using the PSO-DR, PCA, tSNE, Isomap, and Sammon mapping.

forward parallelization we used all 6 cores of the CPU to estimate the low dimensional points y_i in parallel, accelerating that way the PSO-DR computation.

For the data sets MNIST, COIL-20, and FMINST, d_i and δ_i measures correspond to the Euclidean distance whereas for Swiss Roll data set δ_i is the Euclidean distance whereas d_i corresponds to the geodesic distance estimated as in the Isomap method (Tenenbaum et al., 2000).

3.3 Results

Fig. 1 shows the results of the application of PSO-DR, PCA, tSNE, Isomap, and Sammon mapping on the MNIST dataset. PSO-DR algorithm is much faster than all the rest methods except for PCA. Moreover, the visualisation quality of the proposed approach is comparable or better that the majority of the DR methods apart from tSNE which is currently the best DR method for data visualization. It is noteworthy that PSO-DR and Sammon mapping is constructing a similar ball with PSO-DR exhibiting much faster estimation. The mapping of PCA and Isomap exhibit more extensive overlap between the classes.

Fig. 2 presents the respective resutls for COIL-20 data set (labels 1-20 refer to each one of the 20 objects). Again PSO-DR is faster than tSNE and Sammon mapping. Nevertheless, the time differences are not of the same magnitude and, in addition, PSO-DR is now slightly slower than Isomap. This is due to the fact that COIL-20 has much fewer high dimensional points. Thus, PSO-DR's superiority in terms of fast computation is mostly revealed in bigger data sets. Moreover, the similarity of PSO-DR and Sammon mapping is again observed. Nevertheless, Sammon mapping is almost 5 times slower than the PSO-DR approach.

Fig. 3 shows the results of the same DR methods on FMNIST data set. Similarly, PSO-DR is significantly faster from all the other approaches, except for PCA, with better quality of visualization from PCA and Isomap, similar quality with Sammon mapping, and comparable visualization quality with the tSNE approach.

In Fig. 4 the mappings of the Swiss Roll data set using the PSO-DR, PCA, Laplacian Eigenmap,

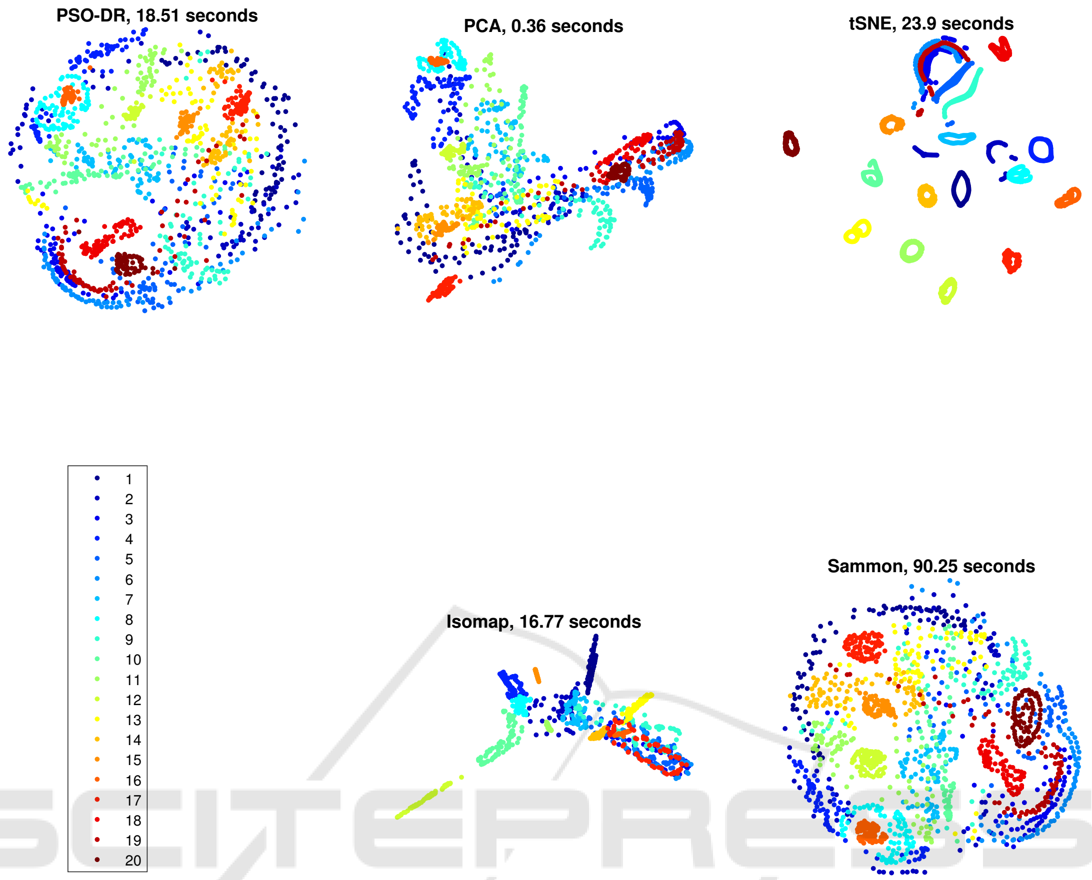

Figure 2: Visualization of the COIL-20 data set using the PSO-DR, PCA, tSNE, Isomap, and Sammon mapping (labels 1-20 refer to each one of the 20 objects).

Isomap, and LLE methods are presented. In this implementation of the PSO-DR method instead of using the Euclidean distance as a measure for d_i, the geodesic distance as estimated also in Isomap approach is used. Thus, it is revealed that PSO-DR results in almost identical representation with Isomap whereas the rest of the state of the art approaches have poorer performance. The PSO-DR approach, though, is slightly slower than Isomap. Nevertheless, this computation time corresponds to only 6 cores of parallel computation.

It should be stressed out that if Euclidean distance is used for the d_i measure for the case of the Swiss Roll data set, the performance of the PSO-DR approach is poorer. Nevertheless, the versatility of the proposed approach makes it possible to easily adjust it to the needs of the data set under consideration and select the dissimilarity measures of the input and output spaces accordingly.

4 DISCUSSION AND FUTURE WORK

The experiments presented here demonstrate that PSO-DR is a simple, fast and versatile algorithm for DR for data visualization where multiple choices for distance measures both for d_i and δ_i are possible in a simple and straightforward way.

PSO-DR exhibits comparative or better visualization quality with the majority of the state of the art approaches that it was compared to. In essence, apart from tSNE, PSO-DR outperforms the rest of the approaches in terms of visualization quality. Nevertheless, tSNE is much slower than PSO-DR especially for large data sets and it performs poorly on certain datasets like the Swiss Roll dataset. Moreover, for tSNE, as for many other non-linear DR approaches, out-of-sample extension is not straightforward (Van Der Maaten et al., 2009). On the contrary, the out-of-sample extension in PSO-DR is inherent in its functionality, as any new input can be mapped directly

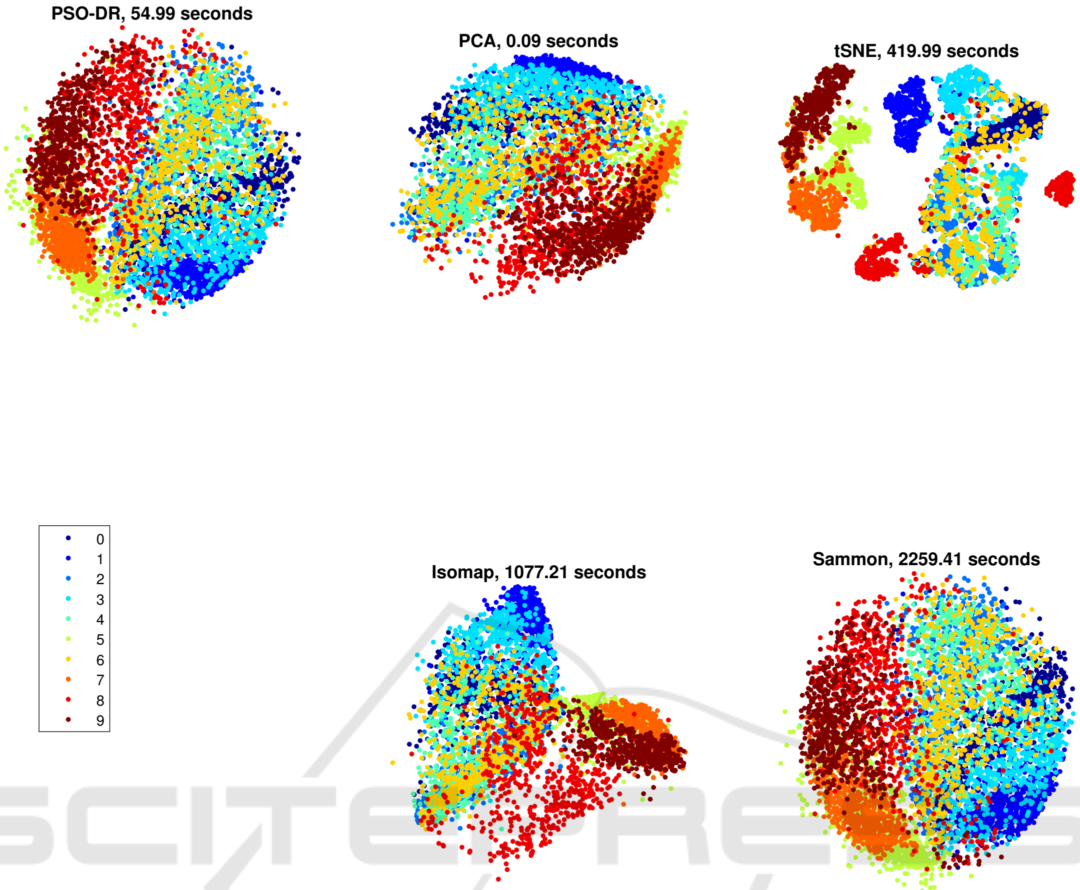

Figure 3: Visualization of the FMNIST data set (6,000 fashion items) using the PSO-DR, PCA, tSNE, Isomap, and Sammon mapping. (labeling was used for convenience, Labels 0 to 9 correspond to the 10 different fashion items).

by comparing it with a fixed set of reference beacon-points.

In addition, the independence of the mapping of each high dimensional input from other inputs can favor the straightforward parallel implementation of the algorithm. Thus, its computation time depends on and scales in inverse proportion with the number of parallel computation units (cores). Furthermore, the design of the PSO-DR algorithm makes it a good choice for ever increasing data sets, even with live streamed data points as it uses a specified set of beacons to compare the new data points and map them in the low dimensional space. Nevertheless, in this work, we did not investigated the functionality of the PSO-DR method in such cases and we will consider evaluating it in future research.

In essence, the only part of the PSO-DR algorithm where the mapping of an input depends on the previous inputs is the definition of the beacon set Y_b. As soon as this step is completed, PSO-DR mapping can be performed independently with regard to the different inputs. The alternative approach where all mappings depend on the previous inputs, thus, no beacons are defined but all x_i in X are mapped depending on the distance from all the previous inputs would result in slower computation without better visualization quality. Fig. 5 shows the low dimensional mapping of MNIST data set based on such an approach. The result is comparable with the one shown in Fig. 1 whereas the time needed is multiple times larger that the one needed with the beacons approach.

5 CONCLUSIONS

In this paper a new approach for DR for data visualization is presented. The PSO-DR algorithm is based on the PSO optimization for mapping high dimensional inputs to low dimensional spaces with fast and versatile way with good data visualization quality. Despite the promising performance of the proposed approach future research will focus on testing

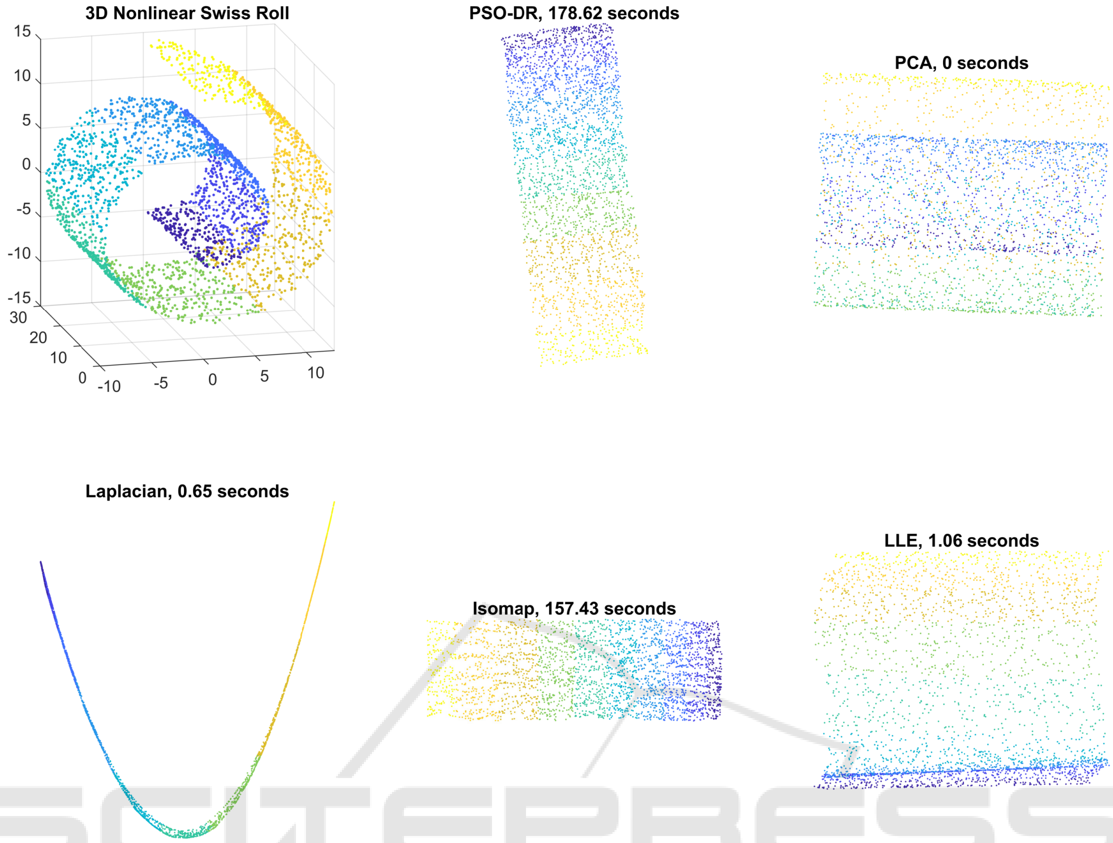

Figure 4: Visualization of the nonlinear Swiss Roll data set in 3D space and by using the PSO-DR, PCA, Laplacian Eigenmaps, Isomap, and LLE.

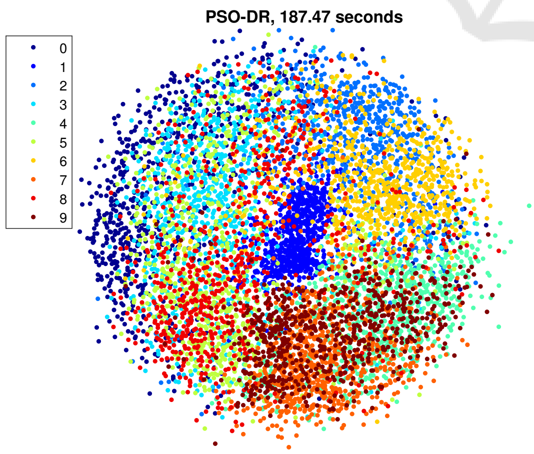

Figure 5: Visualization of the MNIST data set using the PSO-DR without beacon-points.

with more data sets and especially with live streamed input points.

ACKNOWLEDGEMENTS

This work was supported by the European Union's Horizon 2020 Research and Innovation Program through the project SUITCEYES under Grant 780814.

REFERENCES

Belkin, M. and Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591.

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

Eberhart, R. and Kennedy, J. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pages 1942–1948. Citeseer.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping.

In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Koziel, S. and Yang, X.-S. (2011). *Computational optimization, methods and algorithms*, volume 356. Springer.

Nene, S. A., Nayar, S. K., Murase, H., et al. (1996). Columbia object image library (coil-20).

Pedersen, M. E. H. (2010). Good parameters for particle swarm optimization. *Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001*, pages 1551–3203.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, 100(5):401–409.

Shi, Y. and Eberhart, R. C. (1998). Parameter selection in particle swarm optimization. In *International conference on evolutionary programming*, pages 591–600. Springer.

Shi, Y. and Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1945–1950. IEEE.

Shi, Y. et al. (2001). Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, volume 1, pages 81–86. IEEE.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419.

Van Der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative review. *J Mach Learn Res*, 10(66-71):13.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.