Information Sciences 364–365 (2016) 213–221

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

SOM: Stochastic initialization versus principal components

Ayodeji A. Akinduko, Evgeny M. Mirkes, Alexander N. Gorban*

University of Leicester, Leicester, UK

ARTICLE INFO

Article history:
Received 10 April 2015
Revised 18 September 2015
Accepted 6 October 2015
Available online 27 October 2015

Keywords:
SOM
PCA
Random initiation
Benchmark test
Manifold learning

ABSTRACT

Selection of a good initial approximation is a well known problem for all iterative methods of data approximation, from *k*-means to Self-Organizing Maps (SOM) and manifold learning. The quality of the resulting data approximation depends on the initial approximation. Principal components are popular as an initial approximation for many methods of nonlinear dimensionality reduction because its convenience and exact reproducibility of the results. Nevertheless, the reports about the results of the principal component initialization are controversial.

In this work, we separate datasets into two classes: *quasilinear* and *essentially nonlinear* datasets. We demonstrate on learning of one-dimensional SOM (models of principal curves) that for the quasilinear datasets the principal component initialization of the self-organizing maps is systematically better than the random initialization, whereas for the essentially nonlinear datasets the random initialization may perform better. Performance is evaluated by the fraction of variance unexplained in numerical experiments.

1. Introduction

Principal components produce the best linear approximations of datasets ("lines and planes of closest fit to systems of points" [24]). These lines and planes are popular as initial approximations for many methods of nonlinear dimensionality reduction [13,17,19] because their convenience and exact reproducibility of the results. The quality of the resulting data approximation depends on the initial approximation but the systematic analysis of this dependence requires usually too much efforts and the reports are often controversial.

In this work, we analyze initialization of Self Organized Maps (SOM). We test and systematically compare two main approaches to initial approximation of SOM. Originally, Kohonen [18] has proposed random initiation of SOM weights but recently the principal component initialization (PCI), in which the initial map weights are chosen from the space of the first principal components, has become rather popular [5]. Nevertheless, some authors have criticized PCI [4,29] (see also discussion of PCI in recent work [30]). For example, the initialization procedure is expected to perform much better if there are more nodes in the areas where dense clusters are expected and less nodes in empty areas. In practical applications, SOM initialization is often performed in several different ways [25].

In this paper, the performance of random initialization (RI) approach is compared to that of PCI for one-dimensional SOM (models of principal curves). Performance is evaluated by the Fraction of Variance Unexplained (FVU). Datasets were classified into linear, quasilinear and nonlinear [14,15]. It was observed that RI systematically performs better for nonlinear datasets; however the performance of PCI approach remains inconclusive for quasilinear datasets.

SOM can be considered as a nonlinear generalization of the principal component analysis [32,33]. Originally developed by Kohonen [18] for visualization of distribution of metric vectors, SOM found many applications in data exploration, especially in

* Corresponding author. Tel.: +441162231433.
E-mail addresses: aaa78@le.ac.uk (A.A. Akinduko), em322@le.ac.uk (E.M. Mirkes), ag153@le.ac.uk, gorbanster@gmail.com (A.N. Gorban).

http://dx.doi.org/10.1016/j.ins.2015.10.013
0020-0255/© 2015 Elsevier Inc. All rights reserved.

data visualization, vector quantization and dimension reduction. However, like for clustering algorithms [12,26], the quality of learning of SOM is greatly influenced by the initial conditions: initial weight of the map, the neighborhood function, the learning rate, sequence of training vector and the number of iterations [18,28]. Several initialization approaches have been developed and can be broadly grouped into two classes: the random initialization and the data analysis based initialization [4]. Due to many possible initial configurations when using random approach, several attempts are usually made and the best initial configuration is adopted.

For the data analysis based approach, certain statistical data analysis and data classification methods are used to determine the initial configuration; a popular method is selecting the initial weights from the space spanned by the linear principal component. Modification to the PCA approach was done [4] and over the years other initialization methods have been proposed. An example is given by Fort et al. [11]. Careful testing is needed for comparison of different SOM initialization strategies.

In this paper, we consider the performance in terms of the quality of learning of SOM using the Random Initialization (RI) method (in which the initial weights are randomly selected from the sample data) and the Principal Component Initialization (PCI) method. The quality of learning is determined by the fraction of variance unexplained [22]. To ensure an exhaustive study, synthetic data sets distributed along various shapes of dimension two are considered in this study and the map is one-dimensional (1D). 1D SOMs are important, for example, for approximation of principal curves. The experiment was performed using the PCA, SOM and Growing SOM (GSOM) applet available online [22] and can be reproduced. The SOM learning has been done with the same neighborhood function and learning rate for both initialization approaches. Therefore, the two methods are subject to the same conditions which could influence the learning outcome of our study. To marginalize the effect of the sequence of training vectors, the applet adopts the batch learning SOM algorithm [10,11,18] described in the next Section. We also test our findings on several popular multidimensional benchmarks and on two-dimensional (2D) SOM.

For the random initialization approach, the space of initial starting weights was sampled; this is because as the size of the data set n increases, the possible choice of initial configuration for a given number of nodes k becomes enormous (n^k). The PCI was done using regular grid on the first principal component with equal variance (Mirkes, 2011). For each data set and initialization approach, the data set was trained using three or four different values of k. We use a heuristic classification of datasets in three classes, linear, quasilinear and essentially nonlinear [14,15], to organize the case study and to represent the results. We describe below the used versions of the SOM algorithms in detail in order to provide the *reproducibility* of the case study.

It is demonstrated that for essentially nonlinear patterns the widely accepted presumption about advantages of PCI SOM initialization is not universal. RI (possibly with several reinitialization) often performs better than PCI.

2. Background

2.1. SOM algorithm

SOM is an artificial neural network which has a feed-forward structure with a single computational layer. Each neuron in the map is connected to all the input nodes. The classical on-line SOM algorithm can be summarised as follows:

1. Initialization: Initial weights are assigned to all the connection $w_j(0)$.
2. Competition: all nodes compete for the ownership of the input pattern. Using the Euclidean distance as criterion, the neuron with the minimum-distance wins.

$$j^* = \arg\min_{1\le j\le k} \|x(t) - w_j(t)\|,$$

where $x(t)$ is the input pattern at time t, $w_j(t)$ is jth coding vector at time t, k is the number of nodes.
3. Cooperation: the winning neuron also excites its neighboring neurons (topologically close neurons). The closeness of the ith and jth neurons is measured by the neighborhood function $\eta_{ji}(t)$: $\eta_{ii} = 1,\ \eta_{ji} \to 0$ for large $|i - j|$.
4. Learning Process (Adaptation): The winning neuron and the neighbors are adjusted with the rule given below:

$$w_i(t+1) = w_i(t) + \alpha(t)\eta_{j^*i}(x(t) - w_i(t)),$$

Thus, the weight of the winning neuron and its neighbors are adjusted towards the input patterns however the neighbors have their weights adjusted with a value less than the winning neuron. This action helps to preserve the topology of the map. A scalar factor $\alpha(t)$ (the speed of learning) defines the size of the correction; for most realizations, its value decreases with time t [18].

2.2. The batch algorithm

We use the batch algorithm of the SOM learning. This is a version of the SOM algorithm in which the whole training set is presented to the map before the weights are adjusted with the net effect over the samples [10,18,21]. The algorithm is given below (after the standard initialization).

1. Put the set of data point associated with each node equal to empty set: $C_i = \emptyset$.

2. Present an input vector x_s and find the winner neuron, which is the weight vector closest to the input data.

$$i = \arg\min_{1\le j\le k} \|x_s - w_j(t)\|, C_i \leftarrow C_i \cup \{s\}.$$

3. Repeat step 2 for all the data points in the training set.
4. Update all the weights as follows

$$w_i(t+1) = \left(\sum_{j=1}^{k} \eta_{ij}(t) \sum_{s\in C_i} x_s\right) / \sum_{j=1}^{k} \eta_{ij}(t) \tag{1}$$

where $\eta_{ij}(t)$ is the neighborhood function between the ith and jth nodes at time t, and k is the number of nodes.

2.3. SOM learning algorithm used in the case study

Before learning, all C_i are set to the empty set ($C_i = \emptyset$), and the steps counter is set to zero.

1. Associate data points with nodes (form the list of indices)

$$C_i = \left\{l : \|x_l - w_i\| \le \|x_l - w_j\| \, \forall i \ne j\right\}.$$

2. If all sets C_i evaluated at step 1 coincide with sets from the previous step of learning, then STOP.
3. Calculate the new values of coding vectors by formula (1)
4. Increment the step counter by 1.
5. If the step counter is equal to 100, then STOP.
6. Return to step 1.

The neighborhood function used for this applet has the simple B-spline form given as a B-spline:

$$\eta_{ij} = \begin{cases} 1 - \dfrac{|i-j|}{w} & \text{if } |i-j| < w \\ 0 & \text{if } |i-j| \ge w, \end{cases} \tag{2}$$

where w is the half width of the spline.

Selection of the half width regulates the ‘bending’ properties of SOM. For small w it is flexible, for large w it behaves like a rigid line (1D) or plane (2D). There exist modifications of SOM which use this analogy to bending energy directly [16]. In this work, we follow the classical ideas of Kohonen [18] and test SOM with three strategies of w selection. First of all, we use SOM with constant w. By default we take $w = 4$. In addition, we use two strategies of shrinking neighborhood range parameter w over time instances:

- Strategy 1. Start with $w = w_{\max}$. Learn until STOP. Take $w \leftarrow w - 1$. Learn until STOP. Repeat till $w = w_{\min}$. Learn until STOP.
- Strategy 2. Start with $w = w_{\max}$. Learn one epoch. Take $w \leftarrow w - 1$. Learn one epoch. Repeat till $w = w_{\min}$. Learn until STOP.

For batch learning, ‘epoch’ means just one step of the batch algorithm. In our tests for 1D SOM with n nodes we take $w_{\max} = \frac{n}{2} + 1$ and $w_{\min} = 2$.

2.4. GSOM

GSOM was developed to identify a suitable map size in the SOM and to improve the approximation of data [2]. It starts with a minimal number of nodes and grows new nodes on the boundary based on a heuristic. There are many heuristics for GSOM growing. Our version is optimized for 1D GSOM, the model of principal curves [22]. GSOM method is specified by three parameters

- Neighborhood radius (the half width). This parameter, w, is used to evaluate the neighborhood function, η_{ij} (the same as for SOM).
- Maximum number of nodes. This parameter restricts the size of the map.
- Stop when fraction of variance unexplained percent is less than a preselected threshold.

The GSOM algorithm includes learning and growing phases. The learning phase is exactly the SOM leaning algorithm. The only difference is in the number of learning steps. For SOM we use 100 batch learning steps after each learning start or restart, whereas for GSOM we select 20 batch learning steps in a learning loop.

2.5. Fraction of variance unexplained

In this study, data are approximated by broken lines (SOM and GSOM). The dimensionless least square evaluation of the error is the FVU. It is defined as the fraction: [The sum of squared distances from data to the approximating line /the sum of squared distances from data to the mean point] [22].

The distance from a point x_i to a straight line is the length of a perpendicular dropped from the point to the line p_i. This definition allows us to evaluate FVU for PCA:

$$\mathrm{FVU} = \sum_{i=1}^{n} p_i^2 / \sum_{i=1}^{n} \|x_i - \bar{x}\|^2, \tag{3}$$

where $\bar{x}$ is the mean point $\bar{x} = (1/n)\sum_{i=1}^{n} x_i$. The nominator in (3) is the sum of the squared deviation of the data points from their projections onto the line, $\sum_{i=1}^{n} p_i^2$, the denominator, $\sum_{i=1}^{n} \|x_i - \bar{x}\|^2$, is the sum of the squared deviation of the data points from their mean.

In order to define FVU for SOM, we need to solve the following problem. For the given array of coding vectors $\{y_i\}(i = 1, 2, \dots, k)$ we have to calculate the distance from each data point x to the broken line specified by a sequence of points $\{y_1, y_2, \dots, y_k\}$. For this purpose, we calculate the distance from x to each segment $[y_i, y_{i+1}]$ and find $d(x)$, the minimum of these distances.

$$\mathrm{FVU} = \sum_{i=1}^{n} d^2(x_i) / \sum_{i=1}^{n} \|x_i - \bar{x}\|^2. \tag{4}$$

In this formula, we use the squared distance $d^2(x_i)$ from the data point x_i to SOM instead of the squared distance p_i^2 from x_i to a line used in (3). We use the distance to the broken line that is less or equal than the distance to the closest node.

2.6. Initialization methods

The objective of this paper is to consider the performance of two different initialization methods for SOM using the FVU (4) as the criterion for measuring the performance or the quality of learning. The two initialization methods compared are:

- PCA initialization (PCI): The weight vectors are selected from the subspace spanned by the first n principal components. For this study, the weight vectors are chosen as a regular grid on the first principal component, with the same variance as the whole dataset. Therefore, given the number of weight vectors k, the behavior of SOM using PCA initialization, is completely deterministic and results in the only configuration. PCA initialization does not take into account the distribution of the linear projection results. It can produce several empty cells and may need a post-processing reconstitution algorithm [4]. However, since the PCA initialization is better organized, SOM computation can be made order of magnitude faster comparing to random initialization, according to Kohonen [18].
- Random Initialization (RI): k weight vectors are selected randomly, independently and equiprobably from the data points. The size of the set of possible initial configurations given a dataset of size n is n^k. Given an initial configuration, the behavior of the SOM becomes completely deterministic.

2.7. Linear, quasilinear and nonlinear models

Data sets can be modelled using linear or nonlinear manifold of lower dimension. According to [14,15] a class of quasilinear model data set was identified. In this study, data sets will be classified as linear, quasilinear or nonlinear. The non-linearity test for PCA helps to determine whether a linear model is appropriate for modelling of a data set [19].

- Linear Model. A data set is said to be linear (in dimension one, with given accuracy) if it can be approximated by a straight line with sufficient accuracy). These data can be easily approximated by the principal components without SOM. We do not study such data.
- Quasilinear Model. A dataset is called quasilinear (in dimension one) if the principal curve approximating the dataset can be univalently and orthogonally projected on the first principal component. For this study, the border cases between nonlinear and quasilinear datasets (like “S”) are also classified as quasilinear. See examples in Fig. 1a and d.
- Nonlinear Model. In this paper, we call a dataset essentially nonlinear if it does not fall into the class of quasilinear datasets. See example in Fig. 1b, c, and e.

PCI SOM and 100 RI 1D SOMs were prepared for each pattern, every strategy of w selection, and for number of nodes $n = 10, 20, 50, 75$ and 100. The typical behavior of FVU for a quasilinear pattern (‘C’ scattered, Fig. 1a, the second image in the row), and a nonlinear pattern (Horseshoe scattered Fig. 1c, the second image in the row) is presented in Table 2 for $w = 4$. The 100 samples of RI SOM are presented by the sample mean of FVU ($\overline{\text{RI FVU}}$), the fraction of samples with RI FVU $<$ PCI FVU (the column $\leq$ PCI in the table), and the sample standard deviation of RI FVU (σ(RI FVU)).

Let us characterise each pattern by two numbers: the average fraction of RI SOMs with FVU $<$ PCI FVU ($<$ PCI for short) and the average fraction of RI SOMs with FVU $=$ PCI FVU ($=$PCI for short). If we consider the choice of the pattern of a given type as a random event then we can combine the results of the tests for all individual patterns in two tables, Table 3 for different morphologies of patterns and Table 4 for quasilinear and nonlinear patterns.

The tables clearly demonstrate that for nonlinear patterns RI perfoms significantly better than for quasilinear ones. The learning strategies with graduate decrease of w in time, from $1 + n/2$ to 2, produce RI SOMs closer to PCI SOM than the strategy with constant $w = 4$. For these strategies, there exist non-negligible number of cases where the results of RI SOM learning coincide with PCI SOM (see the columns ‘=PCI’ in Tables 3 and 4).

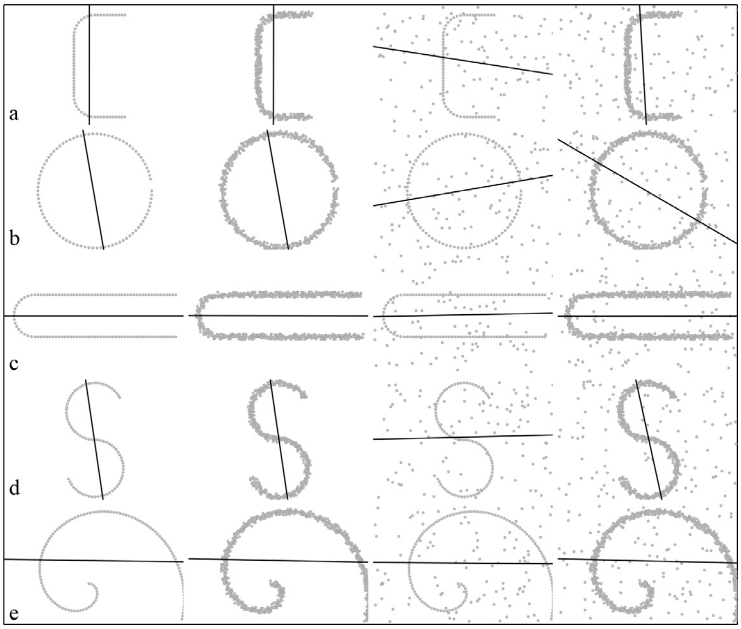

Fig. 1. (a) Quasilinear data set; (b, c, e) nonlinear data set; (d) a border case between nonlinear and quasilinear dataset. The first principal component approximations are shown (black line). The left column contains clear patterns, the second column from the left contains scattered patterns, the second column from the right contains the clear patterns with added noise, and the right column contains the scattered patterns with added noise.

Table 1
Classification of data sets used in the case study and presented in Fig. 1.

Etalon	Clear	Scattering	Noise	Noise & scattering
C	Quasilinear	Quasilinear	Nonlinear	Quasilinear
Circle	Nonlinear	Nonlinear	Nonlinear	Nonlinear
Horseshoe	Nonlinear	Nonlinear	Nonlinear	Nonlinear
S	Quasilinear	Quasilinear	Nonlinear	Quasilinear
Spiral	Nonlinear	Nonlinear	Nonlinear	Nonlinear

Table 2
The results of testing for two patterns of different types. For each row, 100 trials of RI are used, $\overline{\text{RI FVU}}$ in each row is the average value of these 100 trials, and σ(RI FVU) is the empirical standard deviation for these 100 trials.

Pattern	Nodes	PCI FVU	$\overline{\text{RI FVU}}$	$\leq$ PCI (%)	σ(RI FVU)
'C' scattered	10	0.0651	0.0660	46	0.0083
	20	0.0119	0.0184	48	0.0078
	50	0.0023	0.0034	7	0.0008
	75	0.0017	0.0020	21	0.0004
	100	0.0015	0.0016	49	0.0002
Horseshoe scattered	10	0.1730	0.1717	72	0.0048
	20	0.0828	0.0507	100	0.0141
	50	0.0183	0.0062	100	0.0020
	75	0.0065	0.0026	100	0.0008
	100	0.0042	0.0017	100	0.0003

Let us compare the RI SOM to data approximation by GSOM (instead of PCI SOM). For the spiral patterns the histograms are presented in Fig. 2 and GSOM performs better than PCI SOM. The statistics for all patterns is presented in Tables 5 and 6 (for the neighborhood function with fixed half width $w = 4$). We can see that for the nonlinear patterns the relative performance of RI SOM with respect to GSOM is better than for the quasilinear ones.

Let us estimate the number of RI SOMs which we can learn (with fixed $w = 4$) to obtain the FVU less than that of PCI SOM with probability 90%. For the quasilinear patterns we estimate the probability of obtaining RI SOM with FVU worse than for PCI SOM as 0.7111 (see the first row in Table 4). Probability of obtaining seven RI SOMs with FVU not less than for PCI SOM is $0.7111^7 \approx 0.0919 < 0.1$. Therefore, it is sufficient to try seven RI SOMs to obtain FVU less than for PCI SOM with probability $\approx$ 90%. For the nonlinear patterns the situation is even better: if we estimate the probability $\mathbf{P}(\text{RI FVU} > \text{PCI FVU})$ as 0.442 (see the second row

Table 3
The results of testing RI versus PCI for patterns of different morphology with 95% confidence intervals. For each morphology, we take all the corresponding data sets from Fig. 1. In each row, for each data set and the number of nodes $n = 10, 20, 50, 75$ and 100, 100 trials of RI SOM are used (500 RI SOM in total for each data set). Statistical data for individual patterns are presented in [1].

Strategy	Pattern	< PCI (%)	=PCI (%)
$w = 4$	Clear	32.59 (17.51–47.68)	2.41 (0.07–4.75)
	Scattered	40.08 (26.13–54.03)	4.48 (0.41–8.55)
	Noised	55.24 (41.50–68.98)	0.28 (0.00–0.61)
	Scattered and noised	62.36 (49.14–75.58)	2.24 (0.00–4.92)
Strategy 1	Clear	2.95 (0.00–6.36)	73.50 (61.06–85.94)
	Scattered	10.56 (2.44–18.68)	79.36 (69.11–89.61)
	Noised	24.96 (15.00–34.92)	48.48 (37.63–59.33)
	Scattered and noised	26.20 (14.10–38.30)	61.00 (47.57–74.43)
Strategy 2	Clear	41.14 (26.47–55.80)	25.18 (11.61–38.75)
	Scattered	35.16 (23.63–46.69)	26.36 (17.73–34.99)
	Noised	60.20 (48.44–71.96)	2.68 (0.41–4.95)
	Scattered and noised	61.04 (51.48–70.60)	6.24 (2.97–9.51)

Table 4
The results of testing RI versus PCI for quasilinear and nonlinear patterns with 95% confidence intervals. For each type of patterns we take all the corresponding data sets from Fig. 1 (their classification is presented in Table 1). In each row, for each data set and the number of nodes $n = 10, 20, 50, 75$ and 100, 100 trials of RI SOM are used (500 RI SOM in total for each data set).

Strategy	Pattern model	< PCI (%)	=PCI (%)
$w = 4$	Quasilinear	28.89 (18.56–39.23)	3.46 (0–7.18)
	Nonlinear	55.80 (46.97–64.62)	1.90 (0.67–3.13)
Strategy 1	Quasilinear	4.64 (0.23–9.06)	89.26 (82.90–95.62)
	Nonlinear	21.42 (15.04–27.80)	55.48 (48.22–62.74)
Strategy 2	Quasilinear	33.96 (24.11–43.82)	23.14 (13.50–32.78)
	Nonlinear	56.00 (48.61–63.39)	11.42 (6.77–16.07)

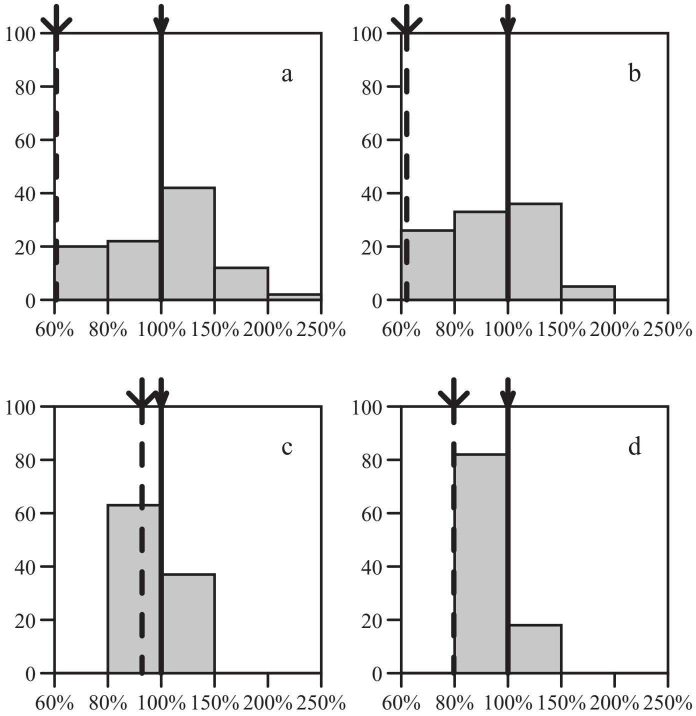

Fig. 2. A typical example of distribution of RI SOM FVU in percent of PCI FVU (with fixed $w = 4$). Vertical solid line with thin arrow above corresponds to PCI SOM FVU. Vertical dashed line with wide arrow above corresponds to GSOM FVU. All four histograms illustrate the distribution of RI SOM FVU with 20 SOM nodes for the spiral pattern: (a) clear spiral, (b) scattered spiral, (c) noised spiral, and (d) scattered and noised spiral. More statistical data for individual patterns are presented in [1].

Table 5
The results of testing RI versus GSOM for patterns of different morphology with 95% confidence intervals

Pattern	$<$ GSOM (%)	=GSOM (%)
Clear	24.23 (6.91–41.54)	5.36 (1.19–9.54)
Scattered	9.40 (0.95–17.85)	7.48 (1.56–13.40)
Noised	72.56 (60.57–84.55)	1.36 (0–2.81)
Scattered and noised	62.00 (47.85–76.15)	2.52 (0–5.08)

Table 6
The results of testing RI versus GSOM for quasilinear and nonlinear patterns with 95% confidence intervals.

Pattern model	$<$ GSOM (%)	=GSOM (%)
Quasilinear	24.96 (11.60–38.33)	8.04 (2.19–13.89)
Nonlinear	49.75 (39.77–59.74)	2.57 (1.21–3.92)

in Table 4) then for three RI SOMs the probability to find at least one FVU better than for PCI SOM is greater than 90%. All these numbers are valid for our choice of patterns and their smearing (Fig. 1).

Of course, our conclusion is limited by the choice of the patterns morphology (smooth curves smeared by two procedures, scattering with small radius and uniformly distributed nose) and by the choice of the smooth curves (‘C’, spiral, etc, see Fig. 1). This limitation is unavoidable in any benchmark testing. Moreover, the low dimension of patterns (2D) and SOMs (1D) may be also considered as the limitation of the analysis. Therefore, in the next section we check the validity of our observations on several popular multidimensional benchmarks.

3. Comparison of PCI and RI for multidimensional datasets

It is interesting to learn the comparative results n-dimensional ($n > 3$) and complex datasets. We tested PCI and RI on several well known benchmarks available in UC Irvine Machine Learning Repository [20]:

- The famous Fisher’s Iris dataset (4 features and 150 samples) [8];
- Wine [9] (the short UCI version, 13 features and 177 samples);
- Forest fires (13 features and 515 samples) [6];
- Abalone (8 features and 4176 samples) [23].

For 1D SOM the results are presented in Table 7. The number of nodes varies from 10 to 100. The neighborhood function η_{ij} is one dimensional B-spline (2) with the half width $w = 4$. The results for 2D SOM with the square grid of nodes are presented in Table 8. The neighborhood function η_{ij} for 2D SOM has the same form (2) with $w = 4$, where $|i - j|$ stands for the Euclidean distance between the nodes on the grid. 100 randomly initiated SOMs are used for each row of the tables.

For 1D SOM, the sample mean of FVU for RI ($\overline{\text{RI FVU}}$) for sufficiently large number of nodes becomes smaller than PCI FVU. For 2D SOM this effect is also observed with the only exclusion for Abalone dataset ($\overline{\text{RI FVU}}$ becomes close to PCI FVU but not

Table 7
The results of RI and PCI testing for multidimensional data sets (1D SOM).

Database	Nodes	PCI FVU	$\overline{\text{RI FVU}}$	$\leq$ PCI (%)	σ(RI FVU)
Iris	10	0.075	0.077	49	0.0030
	20	0.045	0.048	11	0.0029
	50	0.027	0.025	96	0.0013
	100	0.022	0.015	100	0.0009
Wine	10	0.497	0.505	32	0.0100
	20	0.419	0.426	10	0.0064
	50	0.313	0.308	79	0.0061
	100	0.238	0.222	100	0.0059
Forest fires	10	0.702	0.699	69	0.0070
	20	0.601	0.603	34	0.0050
	50	0.475	0.472	62	0.0097
	100	0.358	0.344	87	0.0122
Abalone	10	0.256	0.257	98	0.0016
	20	0.162	0.147	98	0.0538
	50	0.094	0.086	98	0.0025
	100	0.070	0.066	100	0.0011

Table 8
The results of RI and PCI testing for multidimensional data sets (2D SOM).

Database	Nodes	PCI FVU	$\overline{\text{RI FVU}}$	$\leq$ PCI (%)	σ(RI FVU)
Iris	5 × 5	0.066	0.071	15	0.0037
	10 × 10	0.027	0.029	25	0.0014
	15 × 15	0.016	0.016	57	0.0010
	20 × 20	0.011	0.010	79	0.0007
Wine	5 × 5	0.474	0.473	67	0.0044
	10 × 10	0.339	0.340	47	0.0066
	15 × 15	0.253	0.249	74	0.0052
	20 × 20	0.191	0.188	68	0.0054
Forest fires	5 × 5	0.663	0.661	78	0.0047
	10 × 10	0.514	0.514	53	0.0059
	15 × 15	0.410	0.402	81	0.0099
	20 × 20	0.331	0.309	100	0.0105
Abalone	5 × 5	0.258	0.212	93	0.0187
	10 × 10	0.105	0.097	88	0.0063
	15 × 15	0.069	0.070	38	0.0022
	20 × 20	0.057	0.058	25	0.0014

smaller than this value). The sample standard deviation of RI FVU (σ(RI FVU)) presented in the tables allows us to estimate the confidence intervals for RI FVU and $\overline{\text{RI FVU}}$.

The fraction of samples with $\text{RI FVU} < \text{PCI FVU}$ (the column $\leq$ PCI in the tables) is large enough to claim that the random initiation with selection of the best SOM a posteriori may be more efficient than the principal component initiation.

4. Discussion

The simple systematical case study demonstrates that the widely accepted presumption about advantages of PCI SOM initialization is not universal. The frequency of RI SOM with FVU that is less than FVU for PCI SOM is 55% for the nonlinear patterns selected as benchmarks for our study. This means that four random initializations are sufficient to obtain the FVU which is less or equal to the PCI SOM FVU with probability 95% in these cases. For the quasilinear patterns the situation is different and the performance of PCI SOM is better. Nevertheless, it is sufficient for the selected quasilinear benchmarks to try RI SOM seven times to obtain FVU less than for PCI SOM with probability 90%.

The proposed classification of datasets into two classes, quasilinear and nonlinear, is important for understanding of dynamics of manifold learning and for selection of the initial approximation. The linear configurations may be considered as a limit case of the quasilinear ones. We defined quasilinear (in dimension one) dataset using the principal curve and studied one-dimensional SOMs. In applications, SOMs of higher dimensions (two or even three) are used much more often. Therefore, the next step should be the development of the concept of quasilinear datasets for higher dimensions of approximants.

It is possible to generalize this definition to dimension $k > 1$ using injectivity of projection of the k-dimensional principal manifold onto the space of first k principal components. Nevertheless, it may be desirable to consider the quasilinearity of the data distribution without such a complex intermediate concept as "principal manifold". Indeed, SOM is often considered as an approximation of the principal manifold [32,33] and it is reasonable to avoid usage of the principal manifolds of the definition of quasilinearity which will be used for selection of the initial approximation in manifold learning. Let us operate with the probability distributions directly.

Consider a probability distribution in the dataspace with probability density $\mathbf{p}(x)$. Assume that there is a gap between k first eigenvalues of the correlation matrix and the rest of its spectrum. Then the projector Π_k of the dataspace onto the space of first k principal components is defined unambiguously. This projector is orthogonal with respect to the standard inner product in the space of the normalized data. We call the distribution $\mathbf{p}(x)$ *quasilinear in dimension k* if the conditional distribution

$$\mathbf{p}(x|\Pi_k(x)=y)$$

is for each y either log-concave or zero.

The requirement of log-concavity is motivated by the properties of such distributions: convolution of log-concave distributions and their marginal distributions are also log-concave [7]. Therefore, this class of distributions is much more convenient than the naïve unimodal distributions [3]. Most of the commonly used parametric distributions are log-concave and log-concave distributions necessarily have subexponential tails. Non-parametric maximum likelihood estimations for log-concave distributions are developed even in multidimensional case [31].

Finally, let us formulate a hypothesis: if the probability distribution is quasilinear in dimension k then the PCI will perform better than RI, at least for sufficiently large data sets. This hypothesis is supported by our tests.

5. Conclusion

Selection of an initial approximation is crucial for all methods of manifold learning [13]. It seems very natural to start from the best linear approximation, that is, from the principal components. This initial approximation guarantees reproducibility of the results and the produced nonlinear approximation can be considered as an improvement of the linear one. Nevertheless, our tests demonstrate that the randomization of the initial approximation can help a lot in manifold learning for the nonlinear datasets. Of course, there may be many heuristical rules for the further improvement of the initiation, for example, to respect the cluster structure. The optimal choice of the initial approximation depends on the geometry of the dataset. For the essentially nonlinear datasets randomized initial approximation performs better (for the benchmarks used in our work). For the quasilinear datasets use of principal components as the initial approximation may be recommended.

We tested a simple algorithm of random initialization based on random selection of data points. Advanced algorithms of randomized initialization should take into account the data structure at various scales. Random initialization can be also included into heuristic strategies of global optimization like genetic algorithms and evolution strategies. The idea of use of these heuristics in SOM training was proposed long ago [27]. Nevertheless, the detailed analysis and testing of these approaches is needed for practical applications and requires future research.

References

[1] A.A. Akinduko, E.M. Mirkes, 2012, Initialization of self-organizing maps: principal components versus random initialization. A case study, arXiv:1210.5873 [stat.ML].

[2] D. Alahakoon, S.K. Halgamuge, B. Srinivasan, Dynamic self-organizing maps with controlled growth for knowledge discovery, IEEE Trans. Neural Netw. 11 (3) (2000) 601–614.

[3] M.Y. An, Log-concave probability distributions: Theory and statistical testing, Duke University Dept of Economics Working Paper 95-03, 1997. Available at SSRN: http://ssrn.com/abstract=1933 or http://dx.doi.org/10.2139/ssrn.1933.

[4] M. Attik, L. Bougrain, F. Alexandre, Self-organizing map initialization, in: W. Duch, J. Kacprzyk, E. Oja, S. Zadrozny (Eds.), Artificial Neural Networks: Biological Inspirations, vol. 3696, Springer, Berlin Heidelberg, 2005.LNCS, pp. 357–362

[5] A. Ciampi, Y. Lechevallier, Clustering large, multi-level data sets: an approach based on Kohonen self-organizing maps, in: D.A. Zighed, J. Komorowski, J. Zytkow (Eds.), PKDD 2000, LNCS (LNAI), vol. 1910, 2000, pp. 353–358.

[6] P. Cortez, A. Morais, A data mining approach to predict forest fires using meteorological data, in: J. Neves, M.F. Santos, J. Machado (Eds.), Proceedings of the 13th EPIA 2007 New Trends in Artificial Intelligence, APPIA, 2007, pp. 512–523.

[7] S. Dharmadhikari, K. Joag-dev, Unimodality, Convexity, and Applications, Academic Press, 1988.

[8] R.A. Fisher, The use of multiple measurements in taxonomic problems, Annual Eugenics 7 (Part II) (1936) 179–188.[Reprinted in R.A. Fisher, Contributions to Mathematical Statistics, Wiley, NY, 1950.]

[9] M. Forina, C. Armanino, M. Castino, M. Ubigli, Multivariate data analysis as a discriminating method of the origin of wines, Vitis 25 (3) (1986) 189–201.

[10] J.-C. Fort, M. Cottrell, P. Letrémy, Stochastic on-line algorithm versus batch algorithm for quantization and self-organizing maps, in: Proceedings of the 2001 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing 11., 2001, pp. 43–52.

[11] J.-C. Fort, P. Letrémy, M. Cottrell, Advantages and drawbacks of the batch Kohonen algorithm, in: M. Verleysen (Ed.), ESANN'2002 Proceedings, European Symposium on Artificial Neural Networks, Bruges (Belgium), 2002, pp. 223–230.

[12] A.P. Ghosh, R. Maitra, A.D. Peterson, Systematic evaluation of different methods for initializing the k-means clustering algorithm, IEEE Transactions on Knowledge and Data Engineering (2010) 522–537.

[13] A.N. Gorban, B. Kégl, D.C. Wunsch, A.Y. Zinovyev (Eds.), Principal Manifolds for Data Visualization and Dimension Reduction, Springer, Berlin – Heidelberg, 2008. vol. 58, LNCSE.

[14] A.N. Gorban, A.A. Rossiev, Neural network iterative method of principal curves for data with gaps, J. Comput. Syst. Sci. Int. 38 (5) (1999) 825–830.

[15] A.N. Gorban, A.A. Rossiev, D.C. Wunsch II, Neural network modeling of data with gaps: method of principal curves, Carleman's formula, and other, in: USA-NIS Neurocomputing opportunities workshop, Washington DC, 1999. arXiv:cond-mat/0305508

[16] A.N. Gorban, A. Zinovyev, Principal manifolds and graphs in practice: from molecular biology to dynamical systems, Int. J. Neural Syst. 20 (3) (2010) 219–232.

[17] K. Kiviluoto, E. Oja, S-map: A network with a simple self-organization algorithm for generative topographic mappings, in: M.I. Jordan, M.J. Kearns, S.A. Solla (Eds.), Advances in Neural Information Processing Systems, Vol. 10, MIT Press, Cambridge, MA, 1998.pp. 549–555

[18] T. Kohonen, Self-Organization and Associative Memory, Springer, Berlin, 1984.

[19] U. Kruger, J. Zhang, L. Xie, Development and apllications of nonlinear principal component analysis - a review, in: A.N. Gorban, B. Kégl, D.C. Wunsch, A.Y. Zinovyev (Eds.), Principal Manifolds for Data Visualization and Dimension Reduction, vol. 58, Springer, Berlin Heidelberg, 2008, pp. 1–44. LNCSE.

[20] M. Lichman, 2013, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science.

[21] H. Matsushita, Y. Nishio, Batch-Learning self-organizing map with false-neighbor degree between neurons, in: IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. IEEE World Congress on Computational Intelligence, 2008, pp. 2259–2266.

[22] E.M. Mirkes, Principal Component Analysis and Self-Organizing Maps: applet. University of Leicester, 2011. http://www.math.le.ac.uk/people/ag153/homepage/PCA_SOM/PCA_SOM.html.

[23] W.J. Nash, T.L. Sellers, S.R. Talbot, A.J. Cawthorn, W.B. Ford, The population biology of abalone (Haliotis species) in Tasmania. 1. Blacklip Abalone (H. rubra) from the North Coast and Islands of Bass Strait, Sea Fish. Div. Tech. Rep. 48 (1994).

[24] K. Pearson, On lines and planes of closest fit to systems of points in space, Lond., Edinburgh, Dublin Philos. Mag. J. Sci. 2 (11) (1901) 559–572.

[25] M. Pellicer-Chenoll, X. Garcia-Massó, J. Morales, P. Serra-Añó, M. Solana-Tramunt, L.-M. González, J.-L. Toca-Herrera, Physical activity, physical fitness and academic achievement in adolescents: a self-organizing maps approach, Health Educ. Res. (2015), doi:10.1093/her/cyv016.

[26] J.M. Pena, J.A. Lozano, P. Larranaga, An Empirical comparison of four initialization methods for the k-means algorithm, Pattern Recogn. Lett. 20 (1999) 1027–1040.

[27] D. Polani, On the optimization of self-organizing maps by genetic algorithms, in: E. Oja, S. Kaski (Eds.), Kohonen Maps, Elsevier, 1999. pp. 157–169.

[28] M.-C. Su, T.-K. Liu, H.-T. Chang, Improving the self-organizing feature map algorithm using an efficient initialization scheme, Tamkang J. Sci. Eng. 5 (1) (2002) 35–48.

[29] T. Vatanen, I.T. Nieminen, T. Honkela, T. Raiko, K. Lagus, Controlling self-organization and handling missing values in SOM and GTM, in: P.A. Estévez, J.C. Príncipe, P. Zegers (Eds.), Advances in Self-Organizing Maps, Advances in Intelligent Systems and Computing, 198, 2013, pp. 55–64.

[30] T. Vatanen, M. Osmala, T. Raiko, K. Lagus, M. Sysi-Aho, M. Orešič, T. Honkela, H. Lähdesmäki, Self-organization and missing values in SOM and GTM, Neurocomputing 147 (5) (2015) 60–70.

[31] G. Walther, Inference and modeling with log-concave distributions, Stat. Sci. 24 (3) (2009) 319–327.

[32] H. Yin, et al., The Self-organizing maps: background, theories, extensions and applications, in: J. Fulcher, et al. (Eds.), Computational Intelligence: A Compendium: Studies in Computational Intelligence, Springer, Berlin Heidelberg, 2008.pp. 715–762

[33] H. Yin, Learning nonlinear principal manifolds by self-organising maps, in: A.N. Gorban, B. Kégl, D.C. Wunsch, A.Y. Zinovyev (Eds.), Principal Manifolds for Data Visualization and Dimension Reduction, vol. 58, Springer, Berlin Heidelberg, 2008, pp. 69–96. LNCSE.