International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **10(A)**, October **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **6959–6974**

AN IMPROVED LVQ ALGORITHM WITH DATA-STRUCTURE PRESERVING VISUALIZATION

Sarwar Tapan and Dianhui Wang

Department of Computer Science and Computer Engineering
La Trobe University
Melbourne, Victoria 3086, Australia
dh.wang@latrobe.edu.au

Received July 2011; revised November 2011

Abstract. *Data-structure preserved visualization of high-dimensional data reveals the dataset borders and the spread and overlapping tendency of the class borders in a more informative manner than the usual data-topology preserved mapping produced by Self-Organizing Maps (SOMs). Hence, an extension of SOM called Probabilistic Regularized SOM (PRSOM) is proposed for the data-structure preservation in the visualization; however, PRSOM is less suitable for the classification task due to its regularized positioning of the prototypes. In many practical applications, a good classification rate and data-structure informative visualization of high-dimensional data are simultaneously required from an employed method. However, it is difficult to find a method in the current literature that can perform these two tasks effectively. This paper proposes a variant of the Learning Vector Quantization (LVQ) algorithm as Data-Structure Preserving LVQ (LVQ_{dsp}) by combining the classical LVQ1 algorithm with a proposed visualization mechanism to support these two tasks on high-dimensional datasets. Simulations on several benchmark datasets demonstrated LVQ_{dsp}'s promising capability of producing data-structure preserving visualizations in addition to offering excellent classification rates.*

Keywords: Data classification, Data-structure preserving visualization, Self-organizing maps, Learning vector quantization

1. **Introduction.** Data-structure preserving visualization and the classification of high-dimensional and complex data have been two active research interests over the past two decades. However, it is difficult to find a method in the current literature that can perform excellent classification on a labeled dataset, in addition to producing a data-structure preserving visualization of the dataset simultaneously.

Data-structure preserving visualization can be defined as a $\Re^n \rightarrow \Re^l$ ($n > 3$, $l \leq 3$) mapping where high-dimensional data from the $\Re^n$ space are projected on a low-dimensional ($\Re^l$) space in such a way that the global relationship among the data samples in $\Re^n$ space in terms of similarity quantification, often given by the Euclidean distance, is effectively represented in the $\Re^l$ space, so that the inherent relationship among the data samples in the $\Re^n$ space can be *visualized*. Such visualization reveals the borders of the dataset, and the spread and the overlapping tendency of the class borders that are insufficiently revealed by the usual data topology preserved mapping produced by Self-Organizing Maps (SOM) [1] as shown in [2, 3].

Being more informative, data-structure preserved mapping inspired the development of the two extensions of SOM called Visualization induced SOM (ViSOM) [2] and Probabilistic Regularized SOM (PRSOM) [3] which are able to produce excellent data-structure

preserving visualizations of high-dimensional data [2, 3]. Both PRSOM and ViSOM integrate a Multidimensional Scaling (MDS) [4] component with the SOM algorithm for data-structure preservation. PRSOM uses a probabilistic approach motivated by the Soft Topographic Vector Quantization (STVQ) algorithm [5], and is developed based on the foundation introduced by the ViSOM [2] algorithm. PRSOM regularizes the spread and positioning of the prototype vectors in the input space according to the inter-node distances in the 2D output grid, so that the inter-prototype distances in the input space resemble the inter-node distances in the output grid in a trained map. Hence, PRSOM is able to preserve the data-topology and the data-structure information in the mapping [3]. PRSOM has been proven more effective than Curvilinear Component Analysis (CCA) [6], Sammon's Mapping (SM) [7] and particularly ViSOM [2] in terms of data-structure preserving visualization of high-dimensional data [3].

However, due to the constrained (regularized) positioning of the prototype vectors, PRSOM performs poor data quantization by the prototypes that eventually makes PRSOM less suitable for the classification task. By using a larger map size with an empirical parameter tuning, PRSOM can improve its data quantization performance; however, the required computation then becomes impractical. Other classical methods of high-dimensional data visualization, e.g., MDS [4] and SM [7] can produce data-structure informative visualizations; however, they are not meant for vector quantization, learning and classification tasks. On the other hand, numerous methods with Artificial Neural Networks (ANN) and non-ANN architecture have been widely applied to the classification task. However, it is difficult to find a classification-focused method that can simultaneously support data-structure preserving visualization in the current literature.

Therefore, this paper proposes a practical method that can offer an excellent classification rate and data-structure preserving visualization of high-dimensional data. We integrated the LVQ1 [8], i.e., the basic version of the Learning-Vector-Quantization (LVQ) family, and a proposed visualization mechanism to propose the Data-Structure Preserving LVQ (LVQ_{dsp}) as a new variant of LVQ. A cost function is associated by combining the vector-quantization error and the data-structure preservation error in the mapping. Simulations on several benchmark datasets are then conducted which show that the LVQ_{dsp} algorithm has a recognizable capability of producing data-structure preserving visualization of high-dimensional data in addition to offering an excellent classification rate.

The remainder of this paper is organized as follows. Section 2 briefly describes the SOM and PRSOM algorithms since they are used in the evaluation of LVQ_{dsp}'s performance. Section 3 describes the proposed LVQ_{dsp} algorithm. Section 4 describes two quantification criteria for the evaluation of data-structure preservation. Section 5 then reports the simulations on the benchmark datasets. A discussion of the advantages, usability and deficiency of the LVQ_{dsp} algorithm is then given in Section 6. Section 7 concludes this work.

2. SOM and PRSOM Algorithms.

2.1. Self-organizing maps (SOM).

SOM [1] usually preserves data topology in a 2D regular grid. Each node i in the grid is initially associated with a prototype vector $w_i = [w_{i1}, \ldots, w_{in}] \in \Re^n$. In the incremental SOM training, each data sample $x(t) \in \Re^n$ in discrete time step t is presented to the network, then a winner node c is selected based on the closest similarity as,

$$c = \arg\min_i \|x(t) - w_i(t)\|, \tag{1}$$

where $\| * \|$ is the Euclidean distance between two vectors. Then, the winner node with a set of neighboring nodes N_c is updated as,

$$\begin{aligned} w_i(t+1) &= w_i(t) + \alpha(t) h_{ic}(t)[x(t) - w_i(t)], \quad \forall i \in N_c, \\ w_i(t+1) &= w_i(t), \quad \forall i \notin N_c, \end{aligned} \tag{2}$$

where $h_{ic}(t)$ is the neighborhood function that can be defined using a *shrinking* neighborhood range $\sigma(t)$ as,

$$h_{ic}(t) = \exp\left\{-\frac{\|r_i - r_c\|^2}{2\sigma(t)^2}\right\}, \tag{3}$$

where $\|r_i - r_c\|$ is the Euclidean distance between a node i and the winner node c in the output grid. Data-topology preservation is a very useful and discriminative feature of SOM that has been employed in various application fields, e.g., [9-11].

2.2. **Probabilistic regularized SOM (PRSOM).** PRSOM [3] modifies the learning rules of the ViSOM [2] algorithm using a probabilistic approach for soft assignments. ViSOM introduced an extension of SOM by decomposing the force $J_{ix}(t) = [x(t) - w_i(t)]$ representing the force between any prototype w_i and $x(t)$ into two parts as, $[x(t) - w_i(t)] = [x(t) - w_c(t)] + [w_c(t) - w_i(t)] = J_{cx}(t) + J_{ic}(t) = J_{ix}(t)$, where $J_{cx}(t)$ is the force from the winner prototype $w_c(t)$ to $x(t)$ and $J_{ic}(t)$ is the lateral force from $w_i(t)$ to $w_c(t)$. This enables the data-structure preservation by mesh-like spreading of the prototypes [3].

Assuming that K is the number of nodes in the map, χ is a normalization constant, and h_{jk} is a neighborhood function given in Equation (3), the probabilistic assignment of $x(t)$ to j^{th} node can be expressed as,

$$P_j(x(t)) = \frac{1}{\chi} \times \left(\left\| \sum_{k=1}^{K} h_{jk} \left[x(t) - w_k(t) \right] \right\|^2 \right)^{-1}, \tag{4}$$

where $P_j(x(t))$ achieves the highest probability assignment if w_j is the best matching prototype to $x(t)$. Then, the neighborhood-affected probabilistic assignment denoted as $p_j(x(t))$ for the j^{th} node can be given as $p_j(x(t)) = \sum_{i=1}^{K} h_{ij} P_i(x(t))$, where $\sum_{i=1}^{K} h_{ij} = 1$.

Having the learning rate denoted as $\alpha(t)$, and a normalized probabilistic assignment $p'_j(x(t))$ representing $p'_j(x(t)) = p_j(x(t)) / \max_k \{p_k(x(t)\}$, PRSOM's prototype updating can be written as,

$$w_j(t+1) = w_j(t) + \alpha(t) p'_j(x(t)) \sum_{i=1}^{K} p_i(x(t)) \left[J_{ix}(t) + J_{ji}(t) \Omega(t) \right], \tag{5}$$

where $J_{ix}(t) = [x(t) - w_i(t)]$, $J_{ji}(t) = [w_i(t) - w_j(t)]$, and $\Omega(t) = \gamma \dfrac{\left(d_{ij}^2(t) - \lambda \delta_{ij}^2\right)}{\left(\lambda \delta_{ij}^2 + I_{ij}\right)}$ is a MDS [4] component integrated for data-structure preservation, I is an identity matrix to avoid the denominator term becoming zero [3], d_{ij} and δ_{ij} are the distances between the prototype of node i and j in the input space and their corresponding grid positions, respectively. λ and γ are the resolution and the regularization parameters, respectively.

Setting appropriate γ and λ are challenging and require empirical assignments, due to their data-dependent nature. A smaller λ may cause the prototypes to be placed too densely to cover the data spread that will produce an inappropriate data representation by the prototypes. On the other hand, a large λ may cause the prototypes to be positioned beyond the data regions which increases the chances of dead nodes.

3. Data-Structure Preserving LVQ (LVQ_{dsp}).

3.1. **Data-structure preservation.** For a given dataset $X = \{x_1, x_2, \ldots, x_M\}\ \forall x \in \Re^n$, suppose that the corresponding counterparts in the low-dimensional (e.g., 2D) space are known as $Y = \{y_1, y_2, \ldots, y_M\}\ \forall y \in \Re^l$. Since the inherent data-structure can be represented by the global relationship among the data samples given by a similarity measure, such relationship in the $\Re^n$ and $\Re^l$ spaces can be expressed by the following two relationship matrices as,

$$DX^n = \begin{bmatrix} R(x_1,x_1) & \cdots & R(x_1,x_M) \\ \vdots & \ddots & \vdots \\ R(x_M,x_1) & \cdots & R(x_M,x_M) \end{bmatrix}, \quad DY^l = \begin{bmatrix} R(y_1,y_1) & \cdots & R(y_1,y_M) \\ \vdots & \ddots & \vdots \\ R(y_M,y_1) & \cdots & R(y_M,y_M) \end{bmatrix},$$

where $R(x_i, x_j) = \dfrac{\|x_i - x_j\|}{\mu^n}$ and $R(y_i, y_j) = \dfrac{\|y_i - y_j\|}{\mu^l}\ \forall i, j \in M$, where μ^n and μ^l represent the maximum distance between all points in the $\Re^n$ and $\Re^l$ space, respectively. Then, the data-structure of X can be called preserved in the $\Re^n \to \Re^l$ mapping if: $\|DX^n - DY^l\| \approx 0$. A data-structure revealing mapping using $\forall x \in X$ and $\forall y \in Y$ is not practical for the learning algorithms since data quantization is required by the learning algorithms for knowledge representation and generalization.

If data quantization is performed by K number of nodes ($K < M$), then a set of prototypes, i.e., $W = \{w_1, w_2, \ldots, w_K\}\ \forall w \in \Re^n$, are positioned in such a way that W approximates X sufficiently through training. Each w_i can have a 2D coordinate vector $v_i = [v_{i1}, v_{i2}]$, so that the set of coordinate vectors, i.e., $V = \{v_1, v_2, \ldots, v_K\}$, can be positioned in the output layer for visualization. Then, the data-structure of X can be revealed by V through W, provided that the following two conditions are sufficiently satisfied.

1. First, sufficient preservation of the inter-point relative distances, i.e., $\|DW^n - DV^l\| \approx 0$, where DW^n and DV^l are the inter-point relationship matrices for W and V, respectively in a trained network. DW^n and DV^l can be produced similarly as shown for DX^n and DY^l above.
2. Second, sufficient quantization of X by W, i.e., $\frac{1}{M}\sum_{i=1}^{M} \|x_i - w_p\| \approx 0$, where $p = \arg\min_j \{\|x_i - w_j\|\}$.

These two conditions define the cost function of the LVQ_{dsp} algorithm.

3.2. **LVQ1.** LVQ1 is the prime representative of the large LVQ family [8]. Let us assume that $\forall x \in \Re^n$ are derived from a finite set of classes with overlapping distributions. Initially, several prototypes $w \in \Re^n$ are assigned to each class of data samples. For data sample $x_p(t)$ in a discrete time step t, the winner node $w_c(t)$ is selected as,

$$\|x_p(t) - w_c(t)\| = \arg\min_i \|x_p(t) - w_i(t)\|. \tag{6}$$

Then, only the winner node is updated using a monotonically decreasing learning rate $\alpha(t)$ $(0 < \alpha(t) < 1)$ as,

$$\begin{aligned} &w_c(t+1) = w_c(t) + \alpha(t)[x_p(t) - w_c(t)], \text{ if } x_p(t) \text{ and } w_c(t) \text{ belong to the same class, and} \\ &w_c(t+1) = w_c(t) - \alpha(t)[x_p(t) - w_c(t)], \text{ if } x_p(t) \text{ and } w_c(t) \text{ belong to different classes.} \end{aligned} \tag{7}$$

The LVQ1 algorithm is routinely fast, and is able to produce a classification rate at least as good as other supervised classifiers [8].

3.3. **Data-structure preserving visualization.** First, a low-dimensional output layer needs to be created to enable the proposed mechanism to produce visualization. The output layer can be considered as a 2D space that holds the visualization. Each node i is then associated with a display vector v_i, which is randomly positioned in the output layer. Let μ_{out} (user defined) specify the *expected* maximum Euclidean distance between the display vectors in a trained map. Let there be another scaling constant μ_{in} to represent the maximum Euclidean distance between the data samples in the input space.

The idea is, when the incremental LVQ1 selects a winner for a given data sample, then the winner's display vector $v_c(t)$ triggers every other node's display vector $v_i(t)$ to adjust the relative distance in respect to it. The adjustment is conducted by using an adaptation factor $f_{ic}(t)$ for each node i in time step t as,

$$f_{ic}(t) = \frac{\|v_i(t) - v_c(t)\|}{\mu_{out}} - \frac{\|w_i(t) - w_c(t)\|}{\mu_{in}}, \quad i \neq c, \tag{8}$$

where $f_{ic}(t) > 0$ forces $v_i(t)$ of node i to move towards $v_c(t)$ of winner c, while $v_i(t) < 0$ forces $v_i(t)$ to move in the opposite direction. Then, the display vector $v_i(t)$ of every other node i is updated in respect to the display vector $v_c(t)$ of the winner node c as,

$$v_i(t+1) = v_i(t) + \xi_{ic}(t) f_{ic}(t)[v_c(t) - v_i(t)]\eta(t), \tag{9}$$

where $\eta(t)$ is a monotonically decreasing adaptation parameter $0 < \eta(t) < 1$, and $\xi_{ic}(t)$ is an adaptation smoothing operator based on the relative similarity between prototype pair $\{w_i(t), w_c(t)\}$ to allow soft adaptation of $v_i(t)$ with respect to $v_c(t)$. Being motivated by Equation (3), $\xi_{ic}(t)$ can be written as,

$$\xi_{ic}(t) = \exp\left\{-\frac{\|w_i(t) - w_c(t)\|^2}{\varphi \mu_{in}^2}\right\}, \tag{10}$$

where φ can have empirical assignments, e.g., $\varphi = 2$ can be a workable assignment (as used in the simulations).

3.4. **Cost function.** LVQ_{dsp}'s cost function is composed of two terms, i.e., an error component for the vector quantization and the data-structure preservation. LVQ_{dsp} performs supervised vector quantization using the LVQ1 learning rules. Hence, LVQ_{dsp}'s quantization error function denoted as F_{mse} can be given by the mean square error (mse) computation as,

$$F_{mse}(t) = \frac{1}{M} \sum_{p=1}^{M} \|x_p(t) - w_c(t)\|. \tag{11}$$

In LVQ_{dsp}, the data-structure preservation is defined as an approximation of the relative distance among the prototypes in the input space by their corresponding display vectors in the output layer. The following F_{dsp} term gives the *average* relative-distance (data-structure) preservation error using the adaptation factor given in Equation (8) as,

$$F_{dsp}(t) = \frac{1}{K(K-1)} \sum_{i \neq j}^{K} \left| \frac{\|v_i(t) - v_j(t)\|}{\mu_{out}} - \frac{\|w_i(t) - w_j(t)\|}{\mu_{in}} \right|, \tag{12}$$

where K, μ_{in} and μ_{out} are defined earlier. Then, the total cost function of LVQ_{dsp} can be written as,

$$F(t) = F_{mse}(t) + F_{dsp}(t). \tag{13}$$

Figure 1 shows an empirical demonstration on the above cost function during random runs on five benchmark datasets. Dataset details and LVQ_{dsp} settings are given in Section 5. It can be seen that LVQ_{dsp}'s relative-distance preservation given in Equations (8) and

(9) can minimize the cost component F_{dsp} through training, while the quantization error term F_{mse} is minimized by LVQ1's established learning rules. Hence, LVQ_{dsp}'s overall cost F is minimized. The algorithm is defined next.

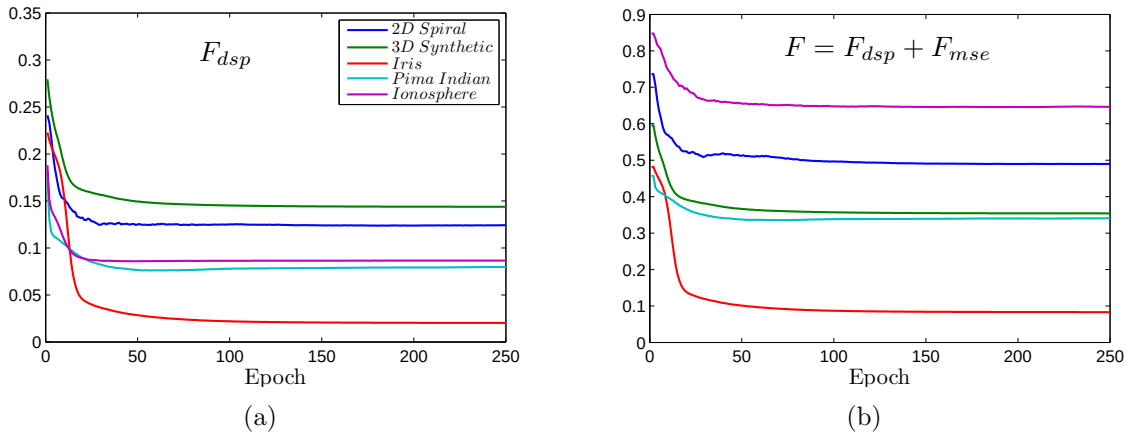

FIGURE 1. (a) The decreasing of the relative-distance (data-structure) preservation error F_{dsp} through training; and (b) the decreasing of the overall cost $F = F_{mse} + F_{dsp}$ during random runs on five datasets

3.5. ***LVQ*$_{dsp}$ algorithm.** Assume that $\forall x \in \Re^n$ are derived from a finite set of classes with overlapping distributions, and several prototypes $w \in \Re^n$, preferably selected from data samples, are assigned to each class. A 2D(3D) output layer is initialized by the user defined μ_{out} value. Each node i has a display vector v_i randomly positioned in the output layer. The maximum distance between the data samples in the input space is computed and assigned to μ_{in}. Then, the LVQ_{dsp} algorithm with incremental training can be defined as follows.

1. *Select the winner node w_c for data sample $x_p(t)$ using Equation (6) in time step t.*
2. *Apply LVQ1 training for w_c using Equation (7).*
3. *Compute the adaptation factor $f_i(t)$ for each node i using Equation (8).*
4. *Update the display vector $v_i(t)$ of each node i using Equation (9).*
5. *Repeat Step 1 to Step 4 until the stopping condition is satisfied.*

4. **Evaluation of Data-Structure Preservation.** Several quantification criteria, e.g., Topographic Function [12] and Topographic Product [13] have been proposed to quantify the neighborhood ordering in a data topology preserved mapping. However, these criteria are not suitable to quantify the data-structure preserved mapping since the evaluation of the data-structure preservation requires quantification of the preservation of the inter-prototype relative-distance by their corresponding counterparts in the output layer. Two possible criteria are therefore proposed.

4.1. **Relative-distance approximation error (RDAE).** First, LVQ_{dsp}'s normalization constants $\mu_{in,out}$ are generalized as $\beta_{in,out}$ for quality evaluation of other methods. For a given trained network, let the maximum distance between the prototype vectors in the input space be denoted as β_{in}, and the maximum distance between the nodes' positioning in the output layer be denoted as β_{out}. Then, the Relative-Distance Approximation Error

($RDAE$) can be given as:

$$RDAE = \frac{1}{K(K-1)} \sum_{i \neq j}^{K} \left| \frac{d_{i,j}^n}{\beta_{out}} - \frac{d_{i,j}^l}{\beta_{in}} \right|, \tag{14}$$

where $d_{i,j}^n$ and $d_{i,j}^l$ are the distances between node i and j in the input and output layers respectively, and K is the number of nodes in the network. $RDAE$ gives the average relative-distance approximation error occurring in the mapping.

4.2. **Relative-distance-based neighborhood dissimilarity (RDND).** First, we need to randomly select L number of nodes to be the neighborhood centers such that $L < K$ from a trained network having K number of nodes. Due to the global association, each prototype w_j belongs to the neighborhood of each neighborhood center $C_i = w_i$ to some extent that could be quantified by their in-between relative-distance. Let Q_i^n represent the neighborhood measure of each neighborhood center $C_i = w_i$ as:

$$Q_i^n = \sum_{j=1}^{K} \left[1 - \frac{d_{i,j}^n}{\beta_{in}} \right], \tag{15}$$

where $d_{i,j}^n$ is the distance between the prototype w_j of any node j and the prototype of neighborhood center C_i in the input space, and β_{in} is the maximum distance between the prototypes. Then, the global neighborhood distribution among L neighborhood centers in the input space can be written as: $Q^n = [Q_1^n, Q_2^n, Q_3^n, \ldots, Q_L^n]$ such that $\sum_{q=1}^{L} Q_q^n = 1$. Similarly, Q^l can be the neighborhood measure in output layer, where v_i represents the neighborhood center C_i in the output layer and β_{out} is defined in Section 4.1.

A good data-structure preservation in the $\Re^n \rightarrow \Re^l$ mapping should have minimal dissimilarity between the relative-distance-based neighborhood representation. Hence, for a large number of trials, taking randomly selected L number of neighborhood centers in each trial, the RDND quantification can be written as:

$$RDND = E\{ \| Q^n - Q^l \| \}, \tag{16}$$

where $E\{\}$ is the mathematical expectation. This criteria statistically quantifies the data-structure preservation of a network by quantifying the relative-distance-based neighborhood dissimilarity in the mapping, provided that a large number of prototypes are used to sufficiently approximate the data.

5. Performance Evaluation.

5.1. **Visual observation.** A 2D Twin Spiral dataset [14], a simulated 3D Synthetic dataset [3], Iris Flower dataset [15], Wisconsin Breast Cancer (WBC) dataset [3], Ionosphere dataset [15], and Pima Indian Diabetes dataset [15] are used in the simulations. LVQ_{dsp}'s data-structure preservation performance is compared with the PRSOM algorithm; since, PRSOM [3] is the most recent and a superior algorithm over ViSOM and SM for data-structure preserved visualization. SOM's data-topology preserved visualizations are then included as references.

Figure 2 to Figure 7 with informative captions show the visualizations of the benchmark datasets produced by LVQ_{dsp}, SOM and PRSOM. The parameters of PRSOM are set according to [3]. It can be seen in the figures that LVQ_{dsp} is able to reveal the borders of datasets, and the spread and the overlapping tendency of class borders of the datasets, which are found promisingly similar to that of PRSOM [3] and better than the SOM's visualizations.

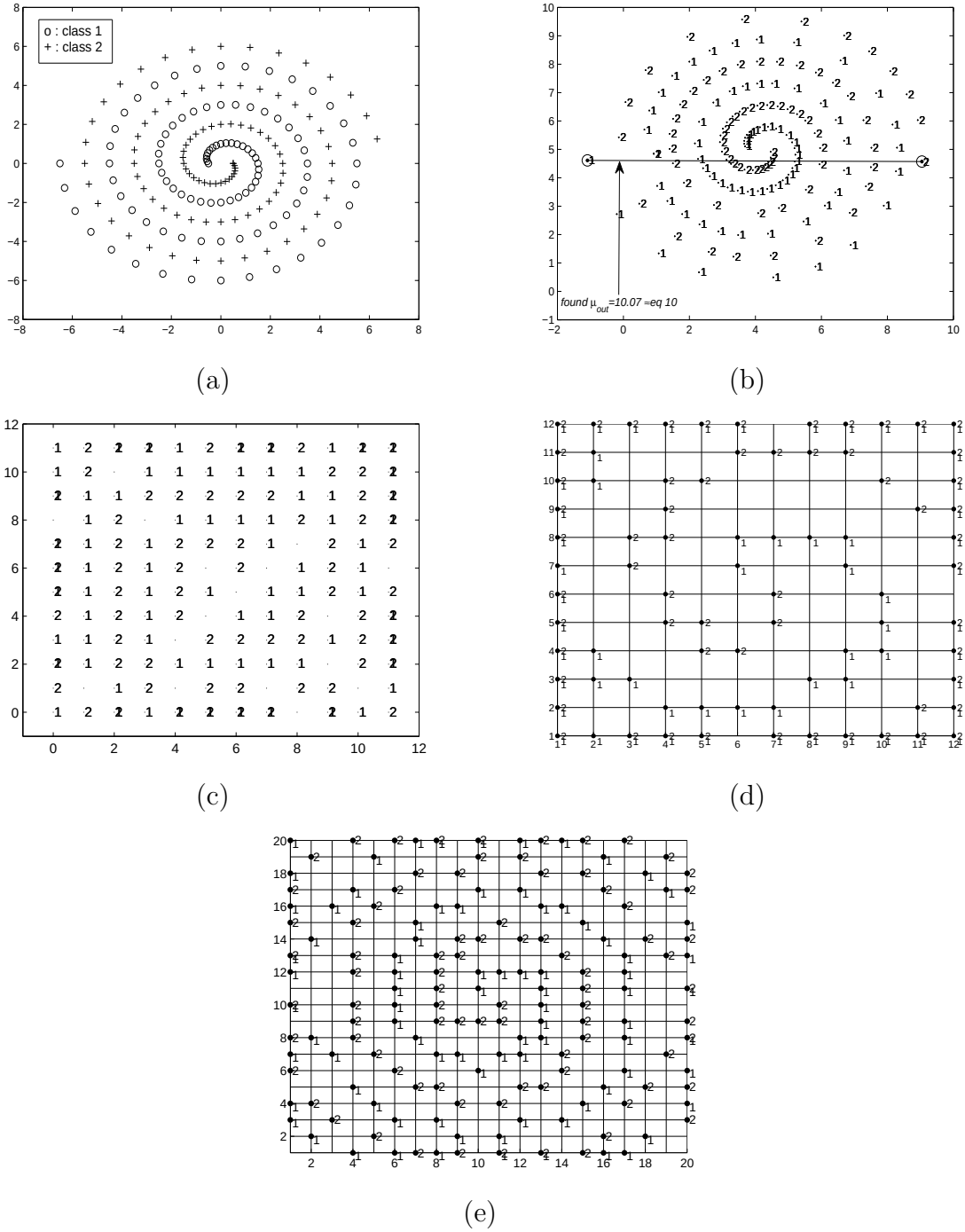

FIGURE 2. Twin spiral dataset (2D, 2-class, 194-samples), $\Re^2 \rightarrow \Re^2$ mapping. (a) View of the 2D twin spiral dataset, (b) visualization by LVQ_{dsp} (12×12 nodes, $\mu_{out} = 10$, 500 iterations), (c) visualization by SOM (12×12 nodes, 1000 iterations), (d) visualization by PRSOM (12×12 nodes, $\gamma = 1.5$, $\lambda = 0.1$, 1000 iterations), (e) visualization by PRSOM (20×20 nodes, $\gamma = 1.5$, $\lambda = 0.1$, 1000 iterations).

In LVQ_{dsp}'s visualization, a straight line is drawn to indicate the maximum distance between the nodes in the output layer. Note that after the training, obtained μ_{out} value approximates the user defined (as specified in the figure captions) *expected* maximum distance between the nodes in the output layer.

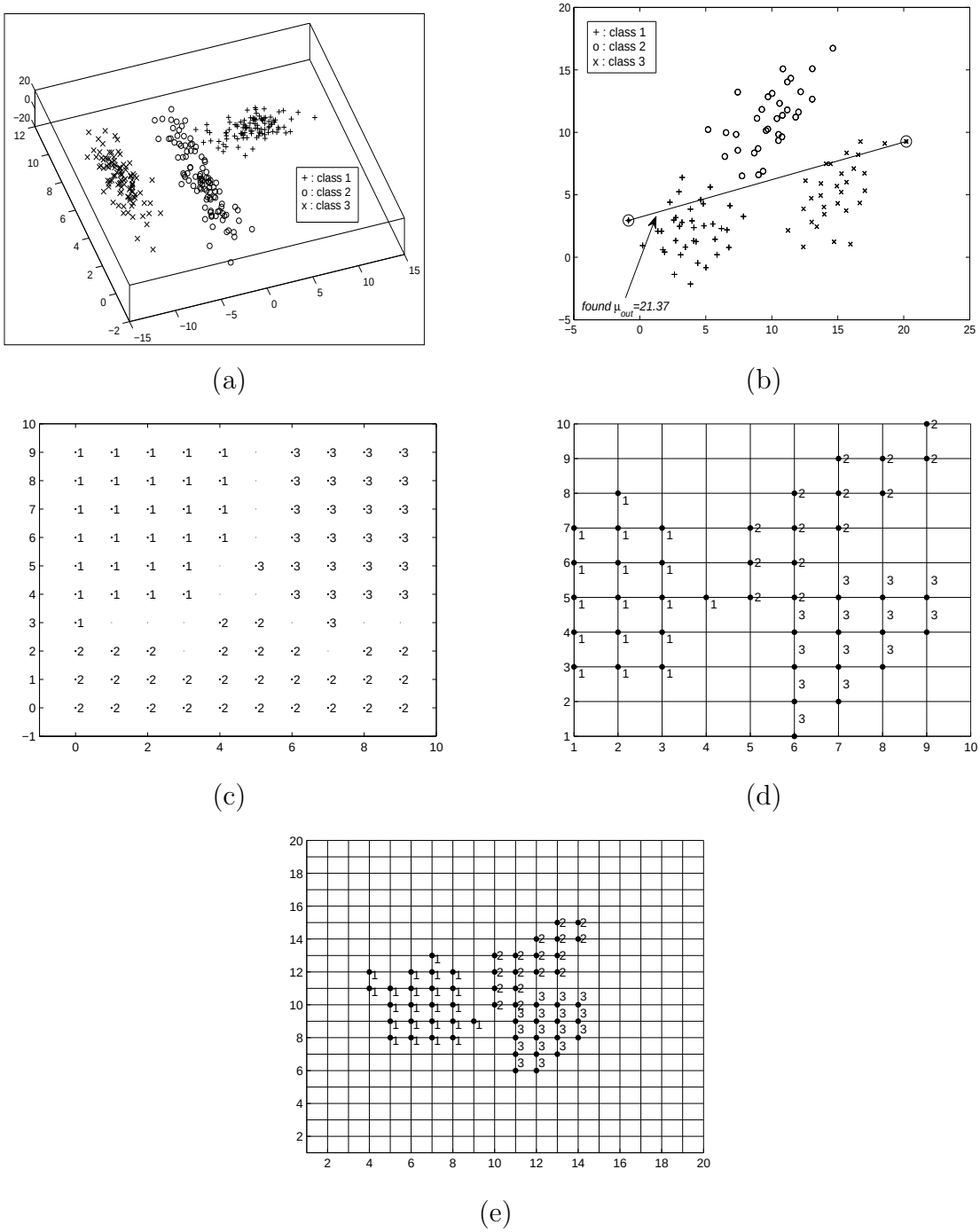

FIGURE 3. 3D Synthetic dataset (3D, 3-class, 300-samples), $\Re^3 \to \Re^2$ mapping. (a) View of 3D synthetic dataset, (b) visualization by LVQ_{dsp} (10×10 nodes, assigned $\mu_{out} = 20$, 500 iterations), (c) visualization by SOM (10×10 nodes, 1000 iterations), (d) visualization by PRSOM (10×10 nodes, $\gamma = 1.5$, $\lambda = 0.01$, 1000 iterations), (e) visualization by PRSOM (20×20 nodes, $\gamma = 1.5$, $\lambda = 0.01$, 1000 iterations).

5.2. **Quantitative evaluation of data-structure preservation.** Quantitative evaluation of the data-structure preservation performance is necessary for more precise evaluation. Table 1 presents a 10-run average of the RDAE and RDND scores produced by LVQ_{dsp}, PRSOM and SOM on the datasets. For comparison reasons, all networks are

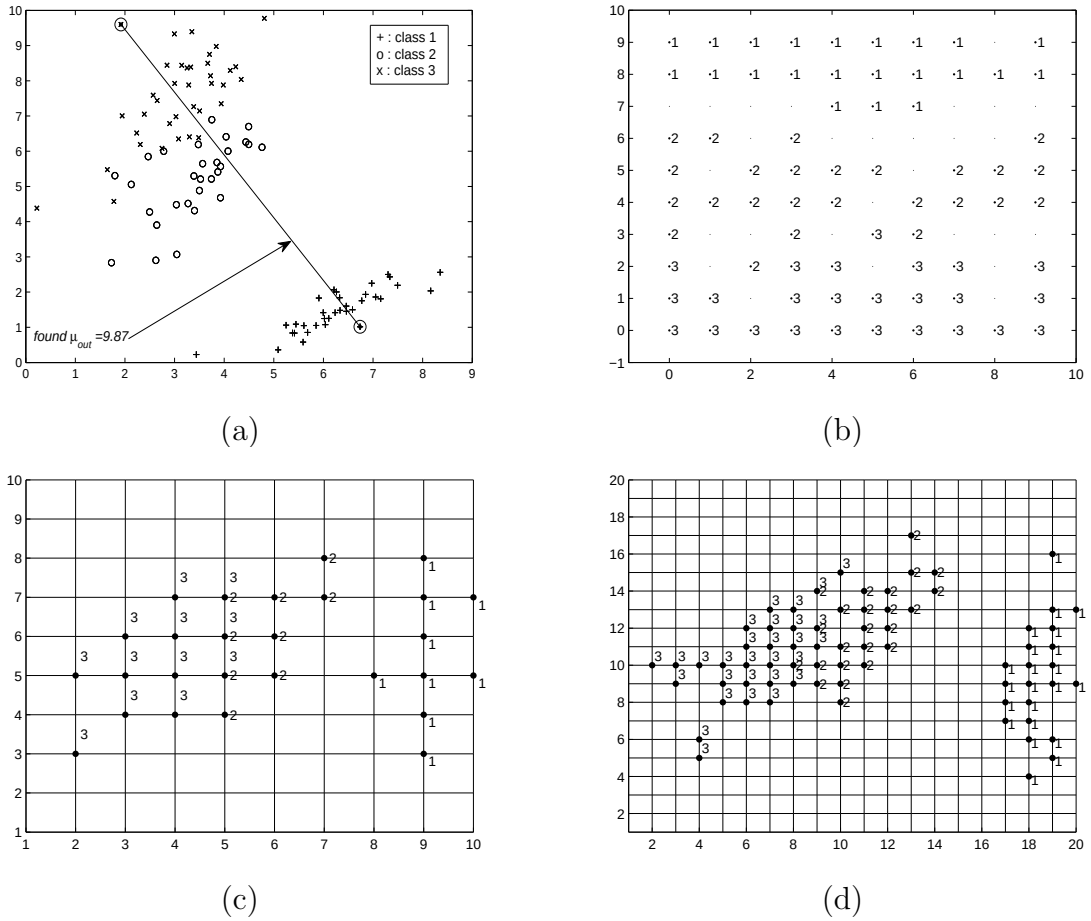

FIGURE 4. Iris flower dataset (4D, 3-class, 150-samples), $\Re^4 \rightarrow \Re^2$ mapping. (a) Visualization by LVQ_{dsp} (10×10 nodes, assigned $\mu_{out} = 10$, 500 iterations), (b) visualization by SOM (10×10 nodes, 1,000 iterations), (c) visualization by PRSOM (10×10 nodes, $\gamma = 1.5$, $\lambda = 0.03$, 1,000 iterations), (d) visualization by PRSOM (20×20 nodes, $\gamma = 1.5$, $\lambda = 0.06$, 1,000 iterations).

given same size. Then, the performance scores are normalized, i.e., $best = 0, worst = 1$, for the three methods considered on each dataset to improve readability. Table 1 shows that both RADE and RDND's promising scores recognize LVQ_{dsp}'s data-structure preservation ability in respect to that of PRSOM, and better than that of SOM.

Effective data-structure representation through the prototypes requires: i) a sufficient number of prototypes in the data region; and ii) a sufficient approximation of the data by the prototypes. Hence, in order to observe how these methods perform in terms of data quantization, a 10-run average of the MSE produced by the networks on each dataset are presented in Table 1 which shows that LVQ_{dsp} performs the best, and PRSOM performs the worst data representation for a given network size on each dataset. SOM's quantization performance is found to be reasonably better than PRSOM. PRSOM performs poor data representation due to its regularized positioning of the prototypes, which eventually makes it ineffective in the classification task.

5.3. **Classification performance comparison.** Being a supervised method, LVQ_{dsp} can be anticipated to produce a better classification rate than SOM and PRSOM. However, due to the context of this paper, it is necessary to quantitatively compare LVQ_{dsp}'s

TABLE 1. A 10-run average of relative-distance preservation and quantization performance

			Relative-distance preservation		Vector quantization
Dataset	Network	Size	$RDAE$	$RDND$	MSE
Twin Spiral	PRSOM	12×12	0.2018	1	1
(2D)	LVQ_{dsp}	12×12	0	0	0
(194 samples)	SOM	12×12	1	0.4206	0.3024
3D Synthetic	PRSOM	10×10	0	0	1
(3D)	LVQ_{dsp}	10×10	0.0842	0.0928	0
(300 samples)	SOM	10×10	1	1	0.0949
Iris Flower	PRSOM	10×10	0	0	1
(4D)	LVQ_{dsp}	10×10	0.0067	0.0726	0
(150 samples)	SOM	10×10	1	1	0.2070
Pima Indian	PRSOM	15×15	0	0	1
(8D)	LVQ_{dsp}	15×15	0.2911	0.2254	0
(768 samples)	SOM	15×15	1	1	0.0270
WBC	PRSOM	15×15	0	0	1
(9D)	LVQ_{dsp}	15×15	0.1227	0.0180	0
(683 samples)	SOM	15×15	1	1	0.0175
Ionosphere	PRSOM	15×15	0	0	1
(34D)	LVQ_{dsp}	15×15	0.3442	0.1058	0
(351 samples)	SOM	15×15	1	1	0.0938

$RDAE$ = Relative-Distance Approximation Error.
$RDND$ = Relative-Distance-based Neighborhood Dissimilarity.
MSE = Mean Square Error.
Scores are normalized ($best = 0$, $worst = 1$) for the 3 networks on each dataset.

TABLE 2. Classification performance comparison

Dataset	A 20-run average accuracy (%)							
	Training data (memorization)				Test data (generalization)			
	LVQ_{dsp}	SOM+LVQ	SOM	PRSOM	LVQ_{dsp}	SOM+LVQ	SOM	PRSOM
Twin Spiral	**99.66**	97.66	93.66	86.51	**74.25**	68.33	66.20	63.25
Pima Indian	**86.36**	81.18	79.15	75.23	**73.24**	69.53	66.88	65.88
Ionosphere	**97.67**	95.89	94.36	91.05	**90.01**	87.32	83.21	82.42
Iris Flower	**99.66**	98.85	98.33	93.45	**96.66**	93.33	89.98	88.25
WBC	**99.25**	98.16	97.28	96.15	**95.93**	94.34	91.35	89.36

classification ability against the visualization methods considered, i.e., SOM (unsupervised) and PRSOM (unsupervised and regularized). LVQ's supervised fine tuning can be employed on a SOM's trained map to improve its classification ability, if the class labels are known. Let us denote this coupling as SOM+LVQ and include it in this comparison. In SOM+LVQ learning, 500 epoch of fine tuning with small constant learning rate $\alpha = 0.001$ is applied on SOM's trained map.

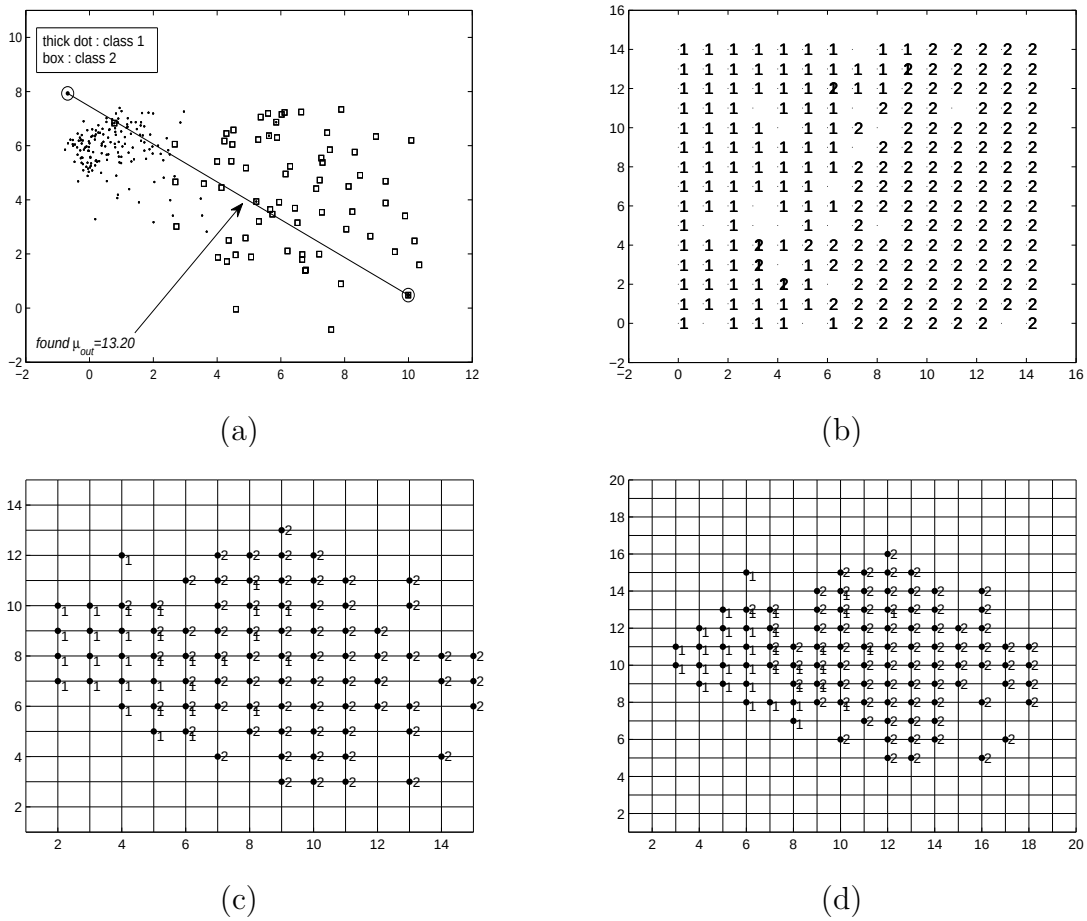

FIGURE 5. Wisconsin breast cancer (WBC) dataset (9D, 2-class, 683-samples), $\Re^9 \to \Re^2$ mapping. (a) Visualization by LVQ_{dsp} (15×15 nodes, assigned $\mu_{out} = 15$, 500 iterations), (b) visualization by SOM (15×15 nodes, 1000 iterations), (c) visualization by PRSOM (15×15 nodes, 1000 iterations, $\gamma = 1.5$, $\lambda = 0.03$), (d) visualization by PRSOM (20×20 nodes, 1,000 iterations, $\gamma = 1.5$, $\lambda = 0.03$).

A 20-run average of the classification rates produced by these methods on the datasets are presented in Table 2. During each run, 80% of the data samples are randomly selected for training and the remaining 20% are used for testing. After training, nodes are labeled according to the majority winning class of the data samples falling in the activation regions, and by carefully removing the 'tie' situations. Then, $1 - NN$ based classification is conducted for the testing data (generalization) and the training data (memorization). Table 2 clearly indicates the superiority of the LVQ_{dsp} algorithm in terms of the classification performance over SOM and PRSOM algorithms. Note that, SOM+LVQ can improve SOM's classification ability to partially satisfy the context of this paper by producing a good classification rate and data topology preserving visualization.

6. **Discussion.**

6.1. **Advantages and deficiencies.** The main advantage and unique feature of the LVQ_{dsp} is that it can support an excellent classification rate and a data-structure preserved visualization simultaneously on a labeled dataset. On the other hand, PRSOM supports an excellent data-structure preserved visualization, however it fails to support a

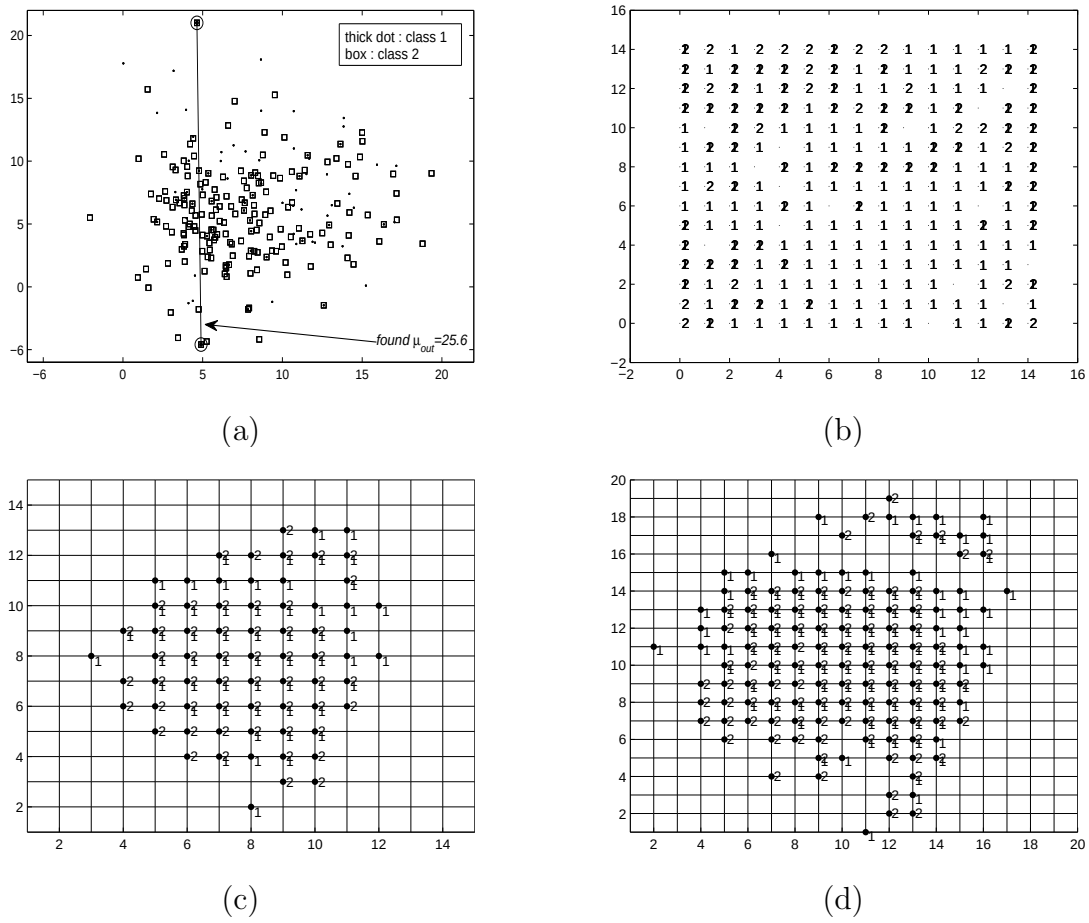

FIGURE 6. Pima Indian diabetes dataset (8D, 2-class, 768-samples), $\Re^8 \rightarrow \Re^2$ mapping. (a) Visualization by LVQ_{dsp} (15×15 nodes, assigned $\mu_{out} = 25$, 500 iterations), (b) visualization by SOM (15×15 nodes, 1000 iterations), (c) visualization by PRSOM (15×15 nodes, $\gamma = 1.5$, $\lambda = 0.02$, 1000 iterations), (d) visualization by PRSOM (20×20 nodes, $\gamma = 1.5$, $\lambda = 0.007$, 1000 iterations).

good classification rate. Other data-structure preserving methods, e.g., MDS [4] and SM [7] are not classically meant for a data classification task, and SOM/(SOM+LVQ) can offer a workable classification rate, however cannot reveal data-structure information in the visualization. Therefore, in a practical situation when an excellent classification rate is primarily required in addition to a data-structure preserved visualization on a labeled dataset, LVQ_{dsp} can be a reasonable choice.

LVQ_{dsp} can efficiently visualize a workable data-structure of high-dimensional data using a comparatively smaller map size and quicker training than PRSOM as observed in the simulations. Additionally, PRSOM tends to produce a large number of dead nodes due to the regular positioning of the prototypes, while LVQ_{dsp} produces almost 0% dead nodes with careful initialization. In addition to that, LVQ_{dsp} is free of any sensitive parameters, which makes it more efficient and employable.

LVQ_{dsp} improves the $LVQ1$ algorithm to be able to produce data-structure preserving visualization. However, it imposes the requirement of a comparatively larger network size than the $LVQ1$ algorithm requirement as a necessary condition for effective data-structure visualization, while this map size increment has little or no impact on the classification

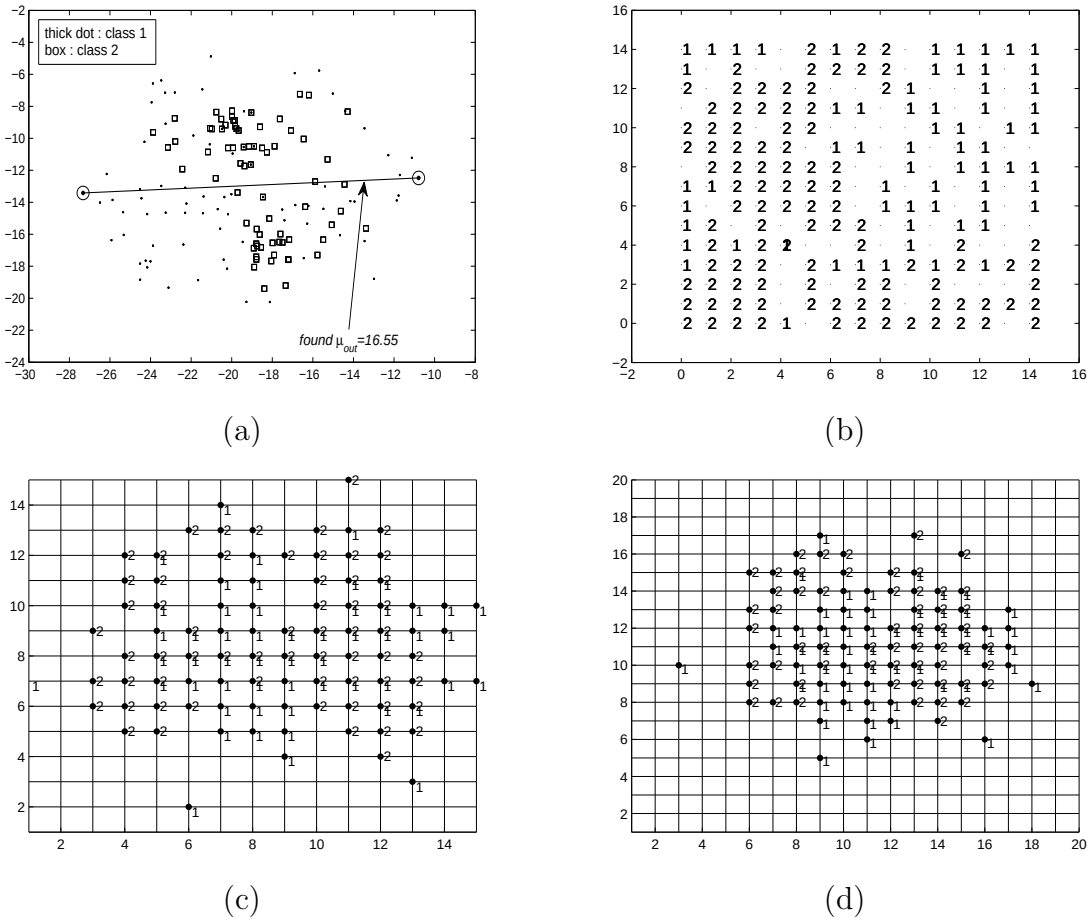

FIGURE 7. Ionosphere dataset (34D, 2-class, 351-samples), $\Re^{34} \rightarrow \Re^2$ mapping. (a) Visualization by LVQ_{dsp} (15×15 nodes, assigned $\mu_{out} = 15$, 500 iterations), (b) visualization by SOM (15×15 nodes, 1000 iterations), (d) visualization by PRSOM (15×15 nodes, $\gamma = 0.5$, $\lambda = 0.07$, 1000 iterations), (e) visualization by PRSOM (20×20 nodes, $\gamma = .5$, $\lambda = 0.07$, 1000 iterations).

result. Using a large number of nodes needs careful consideration as an excessively larger network size can cause overfitting to the network and reduce the generalization ability in classification task.

Additionally, the cost function associated with LVQ_{dsp} is rather heuristic driven that can be mathematically improved. LVQ_{dsp} is currently able to produce a workable data-structure preservation, however improvement seems necessary and will be beneficial. A soft probabilistic approach can be investigated in this regard.

6.2. **Applicability.** Being able to reveal additional information to the usual data-topology preserved visualization, data-structure preserving visualization has great usefulness in applications of the same kind that utilize data-topology preserved visualization, which has been successfully applied in numerous scientific and engineering applications.

Additionally, LVQ_{dsp}'s visualization is more advantageous and realistic in finding nearest neighbors of a node than the grid-based visualizations of SOM and PRSOM algorithms, since the grid-based representation fails to assign *ranking* among the immediate-neighbors of a node in the grid. In addition to this, the nearest-neighbors of a node in a SOM grid

do not resemble their actual similarity in the high-dimensional space as effectively as a data-structure preserving visualization.

Data-structure informative visualization can be a supportive feature to the prototype-based-classifiers (e.g., LVQ) to increase the transparency and reliability of the classification result to the user, instead of being a *black-box* classifier. The unseen data sample can be mapped to the best matching node in the visualization to reveal the characteristics of the data sample with respect to global data distribution, which is often useful to take decisive actions on the data and the classification result.

6.3. **Similarity to MDS.** Traditional MDS [4] preserves the inter point distances in the $\Re^n \rightarrow \Re^l$ mapping. The objective function, often called stress, can be given as [3], $H = \sum_{i<j} \left(\frac{d^l_{i,j} - d^n_{i,j}}{d^n_{i,j}} \right)^2$, where $d^n_{i,j}$ and $d^l_{i,j}$ are the distances between the data pair i, j in the $\Re^n$ and $\Re^l$ spaces, respectively. This MDS stress has an intuitively similar interpretation of LVQ_{dsp}'s relative distance approximation error (F_{dsp}) given in Equation (12). Hence, LVQ_{dsp}'s visualization technique has a similar objective as the MDS methods, however it has a simpler mechanism and it could be integrated with any incremental learning algorithm of LVQ type as well as *unsupervised* vector quantization (VQ) type. Classical MDS methods (e.g., [4, 7]) are not meant for prototype-based data representation that can be used for a classification task, hence they are excluded from the comparison.

LVQ_{dsp} is briefly reported in our previous work [16]. Recently, a variant of LVQ called Generalized Matrix LVQ (GMLVQ) [17] has been proposed to produce low-dimensional projections of high-dimensional datasets after performing dimension reduction using a matrix transformation-based concept, however its visualization does not address the data-structure preservation.

7. **Conclusions.** In this paper a variant of LVQ, named Data-Structure Preserving LVQ (LVQ_{dsp}), is proposed by combining the LVQ1 algorithm and a proposed visualization mechanism. LVQ_{dsp} is able to produce data-structure preserving visualization of a high-dimensional dataset to reveal some useful information about the data distribution, besides producing an excellent classification rate on a labeled dataset. The proposed LVQ_{dsp} addresses the practical scope where these two tasks need to be performed by a single method, while the existing methods are unable to satisfactorily support these two tasks simultaneously.

REFERENCES

[1] T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, vol.43, no.1, pp.59-69, 1981.

[2] H. Yin, ViSOM – A novel method for multivariate data projection and structure visualization, *IEEE Trans. on Neural Net.*, vol.13, no.1, pp.237-243, 2002.

[3] S. Wu and T. M. Chow, PRSOM: A new visualization method by hybridizing multidimensional scaling and self-organizing map, *IEEE Trans. on Neural Net.*, vol.16, no.6, pp.1362-1380, 2005.

[4] R. N. Shepard and J. D. Carroll, Parametric representation of nonlinear data-structures, in *Int. Symp. Multivariate Anal.*, P. R. Krishnaiah (ed.), New York, Academic, 1965.

[5] T. Graepel, M. Burger and K. Obermayer, Phase transitions in stochastic self-organizing maps, *Physical Review E*, vol.56, no.4, pp.3876-3890, 1997.

[6] P. Demartines and J. Herault, Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets, *IEEE Trans. on Neural Net.*, vol.8, no.1, pp.148-154, 1997.

[7] J. J. W. Sammon, A nonlinear mapping for data-structure analysis, *IEEE Trans. on Computers*, vol.C-18, no.5, pp.401-409, 1969.

[8] T. Kohonen, *Self-Organization and Associative Memory*, 2nd Edition, Springer-Verlag, 1984.

[9] T. Samatsu, K. Tachikawa and Y. Shi, Visualization for fuzzy retrieval using self-organizing maps, *ICIC Express Letters*, vol.3, no.4(B), pp.1345-1350, 2009.
[10] G. Noriega, Self-organizing maps as a model of brain mechanisms potentially linked to autism, *IEEE Trans. on Neural Systems and Rehabilitation Eng.*, vol.15, no.2, pp.217-226, 2007.
[11] Y. Liu, X. Wang and C. Wu, ConSOM: A conceptional self-organizing map model for text clustering, *Neurocomputing*, vol.71, no.4-6, pp.857-862, 2008.
[12] T. Villmann, R. Der, M. Herrmann and T. M. Martinetz, Topology preservation in self-organizing feature maps: Exact definition and measurement, *IEEE Trans. on Neural Net.*, vol.8, no.2, pp.256-266, 1997.
[13] H.-U. Bauer and K. Pawelzik, Quantifying the neighborhood preservation of self-organizing feature maps, *IEEE Trans. on Neural Networks*, vol.3, no.4, pp.570-579, 1992.
[14] G. Fung and O. L. Mangasarian, Proximal support vector machine classifiers, *Int. Conf. on Knowledge Discovery and Data Mining KDD'01*, pp.77-86, 2001.
[15] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, University of California, Irvine, 2010.
[16] S. Tapan and C. S. Teh, Hybridization of learning vector quantization (LVQ) and adaptive coordinates (AC) for data classification and visualization, *ICIAS 2007*, pp.505-510, 2007.
[17] K. Bunte, B. Hammer, A. Wismüller and M. Biehl, Adaptive local dissimilarity measures for discriminative dimension reduction of labeled data, *Neurocomputing*, vol.73, no.7-9, pp.1074-1092, 2010.