Expert Systems With Applications 105 (2018) 23–33

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Robust Semi-Supervised Growing Self-Organizing Map

Ali Mehrizi, Hadi Sadoghi Yazdi*, Amir Hossein Taherinia

Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Article history:
Received 14 October 2017
Revised 22 March 2018
Accepted 23 March 2018
Available online 26 March 2018

Keywords:
Semi-supervised learning
Online learning
Dynamic self-organization network
Adaptive learning
Half quadratic

ABSTRACT

Semi-Supervised Growing Self Organizing Map (SSGSOM) is one of the best methods for online classification with partial labeled data. Many parameters can affect the performance of this method. The structure of GSOM network, activation degree and learning approach are the most important factors in SSGSOM. In this paper, a comprehensive robust mathematical formulation of the problem is proposed and then half quadratic (HQ) is used to solve it. Furthermore, an adaptive method is proposed to adjust activation degree optimally to improve the performance of SSGSOM. The results are reported on a variety of synthetic and UCI datasets and in the noisy conditions, which show superiority and robustness of the proposed method compared with the state of the art approaches.

1. Introduction

Extracting knowledge from unlabeled data, also known as unsupervised learning, improves the performance of systems. However, by increasing the volume of imported data, they require a large amount of memory and computation time and therefore become unusable or very inefficient. If a small portion of data has side information such as labels, it can increase the accuracy and speed of learning. The use of labeled data for learning together with clustering methods is called semi-supervised learning. In other words, the learning algorithms that combine unsupervised and supervised learning algorithms, are called semi-supervised learning algorithms (Zhu & Goldberg, 2009). Semi-supervised learning has been applied for many problems such as those reported in (Cong, Liu, Yuan, & Luo, 2013; Lu, Wang, Xue, & Pan, 2014; Chen, Li, Su, Cao, & Ji, 2014).

Considering the data importing method, learning algorithms are divided into two categories: online learning and batch learning. In batch learning, all the learning data are available at the beginning of the process. In contrast, in online learning, none of the data exists at the beginning and data are logged gradually. Although, online learning is adapted to the varying environment better, it lacks the accuracy of batch learning. One of the major reasons for decreasing accuracy of the online algorithms is their dependency on the sequence of data entry. In recent years, attention to online learning is growing, because by increasing volume of data, learning methods must have online capabilities and rapid responses becomes inevitable. Accordingly, many researchers have focused on eliminating defects of online learning methods to achieve better accuracy in a reasonable time. Some promising online clustering methods are online k-mean (Alpaydin, 2004), neural gas network (Martinetz, Berkovich, & Schulten, 1993), and self-organizing network (Kohonen, 1990). Among these methods, self-organization networks and neural gas networks have been more useful because they are inherently online and have properties such as topology preservation (Alahakoon, Halgamuge, & Srinivasan, 2000). Since the majority of studies on semi-supervised learning have been on offline method, in the following we first review the learning methods and then discuss studies on online semi-supervised learning.

Farid et al. (2013) presented a semi-supervised algorithm based on the ensemble learning method, which is an offline approach that determines the label of the data via voting among several classifications. They used decision trees for data classification. Then, an online algorithm was proposed by modifying the structure of the decision tree. For this purpose, in the learning phase, a random weight is allocated to every datum and the decision tree is built for them. Afterward, the allocated weights are modified based on the training data.

Leite and colleagues reported that features and main distribution of data change by time in non-stationary environments (Leite, Costa, & Gomide, 2010). In machine learning terminology, this change is called as a “concept drift”. For example, in classification, a concept can be either a class of data or the boundary of data and the drift is data boundary that changes over time. Drifts can be gradual or sudden, contracting or expanding, and definite/random or periodic change of concept. The main requirement

* Corresponding author.
E-mail addresses: mehrizi@um.ac.ir (A. Mehrizi), h-sadoghi@um.ac.ir (H. Sadoghi Yazdi), taherinia@um.ac.ir (A.H. Taherinia).

https://doi.org/10.1016/j.eswa.2018.03.046
0957-4174/

of online learning is to identify and deal with the drift. They proposed a semi-supervised method based on a granular neural network to deal with data streams. This model has five layers; the first and fifth layers of the neural network are the input and output layer, respectively. The second layer performs the data-clustering task. The third layer performs the merge operation and the fourth layer makes the decisions about data class.

Macario and de Carvalho (2012) proposed a semi-supervised FCM algorithm that adds a supervised ability to the original FCM algorithm. The main idea of this paper is that data of the same class have similar degrees of membership. Therefore, if data belongs to a certain class, differences between their cluster membership degrees are low and the data have stable degrees of membership in all clusters. They also proposed an adaptive method to prevent uncontrolled growth of a cluster.

For the first time, an online constrained clustering algorithm was presented by Halkidi, Spiliopoulou, and Pavlou (2012) in order to find clusters that are centralized and have constraints entering as a stream. The Halkidi algorithm was based on the MPCK[1]-Means clustering algorithm. The MPCK-Means algorithm was an offline algorithm and cannot be used for online data. Therefore, Halkidi considered data as a chunk that enters the algorithm over time. The MPCK-Means clustering algorithm performs offline on small chunk of data. This method can be used to reduce the computational memory.

Cong et al. (2013) introduced a self-supervised online metric learning with the low-rank constraint for scene categorization. Their work used a two-sided graph to evaluate the similarity between data and metric learning. The labeled and unlabeled data were used to update metric learning by a high score. The proposed algorithm of Kung was an online algorithm.

Li and colleagues introduced a semi-supervised learning algorithm based on the extreme learning combined with cooperative learning (Li, Zhang, Xu, Luo, & Li, 2013). The main idea of cooperative learning is using data from a learner's testing phase as training data for other learners. Looping in the learning process is the disadvantage of cooperative learning, which leads to a decrease in learning speed and propagation of the learning process errors to others. In their paper, using extreme learning was proposed to deal with these problems because it is very fast and can cover the problem of cooperative learning (Huang, Zhu, & Siew, 2004) (Huang, Li, Chen, & Siew, 2008).

Zhang and colleagues proposed a semi-supervised algorithm based on co-training (Zhang, Wen, Wang, & Jiang, 2014). This algorithm applies co-training to select the most reliable instances according to two criteria of high confidence and nearest neighbor for boosting the classifier and also exploits the most informative instances with human annotation for improving the classification performance (Zhang et al., 2014).

Dornaika and El Traboulsi (2016) proposed a Graph-Based Semi-Supervised method and its kernelized version for generic classification. Although this method is flexible, it is not online.

Beyer and Cimiano (2012) proposed a semi-supervised method based on neural gas. This paper extends the offline Growing Neural Gas classifier to an online method that predicts labels for unlabeled example and incorporates these labeled examples into the model on-the-fly (Beyer & Cimiano, 2012).

Maximo, Quiles, & Nascimento (2014) proposed a new semi-supervised growing neural gas (GNG) model, wherein instead of assigning a single scalar label value to each neuron, a vector containing the representativeness level of every class is associated with each neuron. Also, to propagate the labels among the neurons the Consensus-Based Semi-Supervised GNG employs a consensus approach (Maximo et al., 2014).

Fritzke (1994) proposed growing self-organizing map (GSOM) that is able to find a suitable network structure and size of the model automatically. Next, this researcher proposed a supervised learning method that results from the combination of the above-mentioned self-organizing network with the radial basis function (RBF) approach.

Hsu and Halgamuge (2008) improved Fritzke method and presented a semi supervised algorithm based on GSOM. In their algorithm, a two-layer model for online data classification with partial labels was provided. In the first layer of this model, GSOM is used for data clustering and when the labeled data were logged, the label of this data is used for classification. In this method, like K-mean and fuzzy C-means, there are repetitive operations with the difference, where in each step only one datum is randomly selected and entered into the GSOM.

Shen, Yu, Kamiya, and Hasegawa (2010) and colleagues offered a three-layer architecture that represents the topological structure of training data (with SOM), learned node labeling, and classifying construction.

In our previous work, we proposed an online constraint semi-supervised learning based on GSOM (Allahyar, Yazdi, & Harati, 2015), based on a two-layer model that came from the development of the HSU model (Hsu & Halgamuge, 2008).

Self-Organizing Maps (SOM) is a suitable learning method for data clustering (Kohonen, 1990). This method is based on artificial neural networks (ANNs) and its structure is formed online. Accordingly, the network is suitable for online learning. SOM was first proposed in 1990 by Kohonen (1990). Dimensions of SOM are a parameter that strongly influences the final result of clustering (Alahakoon et al., 2000). In the Kohonen's model, the user sets this parameter manually. To resolve this problem, ALkahon presented Growing Self-organizing Map (GSOM) (Alahakoon et al., 2000). In ALkahon models, expansion dimensions are performed automatically depending on type and structure of data (Alahakoon et al., 2000).

Hsu and Halgamuge (2008) presented Semi-supervised Growing Self-Organizing Map (SSGSOM) by adding a supervised layer to GSOM. Hsu model is an online semi-supervised model that can perform clustering and classification of data very well (Hsu & Halgamuge, 2008). Their model uses labeled data to control the Expansion nodes in the SOM. This model employs an innovative method to determine the effective parameters in the objective function (Hsu & Halgamuge, 2008). In the present paper, we redefine the objective function and then discuss the determination of the optimum effective parameters.

Table 1 summarizes the significant semi-supervised methods. In this table, type of algorithm, advantages, and unresolved problems of each algorithm are presented.

The main contributions of the proposed method are:

- Redefining robust cost function of online Semi-Supervised GSOM by half quadratic,
- An adaptive online semi-supervised GSOM algorithm is proposed.

The remainder of this paper is organized as follows: In Section 2, the primary concepts for the proposed algorithm are explained. In Section 3, the proposed algorithm is presented. In Section 4, the results of the performed experiments on different data are shown. In the final section, the conclusion and guidelines for future works are expressed.

[1] Metric pairwise constrained K-means.

Table 1
Summary of Semi-supervised Algorithms.

Proposed algorithm	Advantage	Unsolved problems
Farid et al.	√ Semi Supervised learning	❖ Slow execution speed because of creating the decision tree
Leite et al.	√ Fuzzy	❖ Slow execution speed
	√ Resistant data types	❖ Not improve clustering by using the feedback of labeled data
Macario et al.	√ Fuzzy clustering	❖ Offline method
	√ Adaptive distance	
Shen et al.	√ online	❖ Not adaptive
	√ Based on SOM	❖ Not robust
Halkidi et al.	√ Data binding	❖ Receiving data in chunk format
Cong et al.	√ Metric learning	❖ Receiving in chunk format
		❖ Slow execution speed
Li et al.	√ Cooperative extreme learning	❖ Not suitable for online data
		❖ Slow execution speed because of cooperative learning
Zhang et al.	√ Co-training	❖ Offline
		❖ Slow (because used active learning to determine appropriate data for labeling)
Dornaika et al.	√ Flexible method	❖ Offline
	√ Graph-based	❖ Adjust many parameters
Beyer et al.	√ Based on neural gas	❖ Not adaptive
	√ Online	❖ Not robust
Maximo et al.	√ Based on neural gas	❖ Not adaptive
	√ Online	❖ Not robust
Allahyar et al.	√ Online	❖ Not adaptive
	√ Based on GSOM	❖ Not robust
Hsu et al.	√ online	❖ Determine the parameters of the GSOM innovatively
	√ Based on GSOM	
	√ Improve clustering by using feedback of labeled data	❖ Not robust
Proposed method of the paper	√ Online	
	√ Based on GSOM	
	√ Adaptive	
	√ Robust	

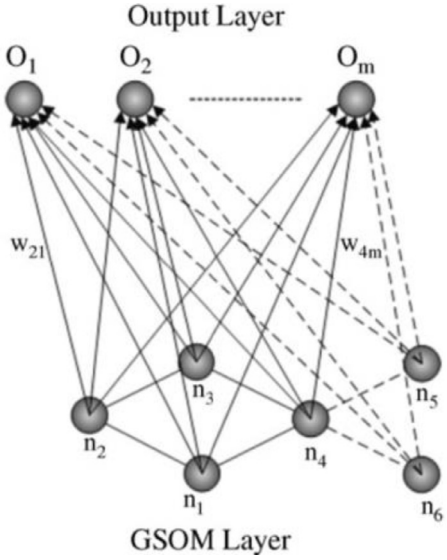

Fig. 1. Semi-supervised GSOM Structure (Hsu & Halgamuge, 2008).

2. Preliminary concept

In this section, the preliminary concepts of semi-supervised GSOM are expressed. GSOM is a method that is able to show the structure of data (Fritzke, 1994). Hsu and colleagues added the semi-supervised ability by adding a supervised layer on top of this network (Hsu & Halgamuge, 2008). Fig. 1 shows the structure of this model.

In the presented semi-supervised model, the number of nodes in the upper layer is equal to the number of classes (M) and the number of nodes in the bottom layer represents the number of clusters (C). There is a full connection of weights between the two layers that is indicated as $w^{c,m}$. Its purpose is to train the weights in such a way that by entering data into the cluster layer, the nearest node is activated and then the suitable class is determined for it. Based on the weights between the two layers, training of weights between the two layers is done by the labeled data and according to Eq. (1) for each data sample (Hsu & Halgamuge, 2008).

$$\Delta w_k^{c,m} = \beta \times \left(\zeta_k^m - PS_k^m\right) \times o_k^c \tag{1}$$

In Eq. (1), ζ_k^m is actual label for datum k that is in class m and the PS_k^m is the estimated value of the semi-supervised GSOM for datum k that is in class m; where ζ_k^m and PS_k^m are defined respectively by Eqs. (2) and (3) (Hsu & Halgamuge, 2008).

$$\zeta_k^m = \begin{cases} 0 & l_k \neq \hat{m}_k \text{ or missing label} \\ 1 & l_k = \hat{m}_k \end{cases} \tag{2}$$

$$PS_k^m = \sum_{c=1}^{C} w_k^{c,m} \times o_k^c \quad m = \{1, 2, \ldots, M\} \tag{3}$$

In Eqs. (1) and (3), o_k^c the function determines the similarity of the entered data with cluster nodes and is known as the activation degree. The value of activation degree for each c node is calculated by using a Gaussian function that is given in Eq. (4) (Hsu & Halgamuge, 2008).

$$o_k^c = exp\left(-\frac{\|x_k - v_c\|^2}{\left(\delta_k^c\right)^2}\right) \quad c \in \{1, \ldots, C\} \tag{4}$$

where δ_k^c is average distance of all neighbor's nodes of c for kth sample and is calculated by (5) and $|Ne(c)|$ in (5) represents number of neighbor's node c (Hsu & Halgamuge, 2008).

$$\delta_k^c = \frac{\sum_{k \in Ne(c)} \|v_k - v_c\|}{|Ne(c)|} \tag{5}$$

More information about how growing GSOM in (Hsu & Halgamuge, 2008).

3. Proposed method

Since the number of labeled data may be few, optimal operation of semi-supervised algorithms is very important. Although the method proposed by Hsu is semi-supervised and online, it faces the following problems:

- The objective function is optimized only based on the weight of classifier layer, but in fact, the objective function also depends on the parameters of the shape and location of the nodes in the clustering layer and the degree of activation.
- Growing of GSOM is not optimized, because the activation function plays an important role in amount of similarity of data for each node, but Hsu algorithm adjusts this parameter by the average distance to neighbor nodes. This innovative solution is a local solution and this method is not optimal by changing data.
- It suffers the problem of instability versus noisy data.

Accordingly, to overcome the mentioned problems, the objective function was redefined and resolved in the present work.

3.1. Definition of the problem

According to Section 2, in the semi-supervised GSOM error for each data k is defined as:

$$e_k = \sum_{m=1}^{M} \left(l_k^m - ps_k^m\right) \tag{6}$$

And cost function based on the expectation of error function is defined as:

$$g(e_k) = \sum_{k=1}^{N} \exp\left(-\eta e_k^2\right) \tag{7}$$

$$J = \max_{w,\ \delta,\ v} \sum_{k=1}^{N} \exp\left(-\eta e_k^2\right) \tag{8}$$

The half-quadratic technique Rockafellar, 1970; He, Hu, Zheng, & Guo, 2010; Zhang et al., 2014) is often used to solve the nonlinear optimization problem (He, Hu, Zheng, & Kong, 2011). Therefore, we derived an algorithm to solve ((8) based on the half quadratic. Based on the theory of convex conjugated functions, we can solve (8) as following proposition.

Proposition 1. *There exists a convex conjugated function ϕ of $g(e)$ such that*

$$g(e) = \sup_{p} \left(\eta e^2 p - \varphi(p)\right) \tag{9}$$

where convex function $\phi(p)$ is defined as $\phi(p) = -p\log(-p) + p$. Therefore, J is rewritten as follows:

$$J = \max_{w,\delta,v} \sum_{k=1}^{N} \left(\eta e_k^2 p_k - \varphi(p_k)\right) \tag{10}$$

According to Proposition 1, we notice that when $[w,\ \delta,\ v]$ is fixed, the following equation holds:

$$\max_{w,\delta,\ v} J(w,\delta,\ v) = \max_{w,\delta,\ v,p} \hat{J}(w,\delta,\ v,\ p) \tag{11}$$

Then, the alternative solution is used:

$$step\ 1: \max_{p} \hat{J}(w,\delta,\ v,\ p) \ \Rightarrow p = \ -\exp\left(-\eta e^2\right).$$

$$step\ 2: \ \max_{w,\delta,\ v} \hat{J}(w,\delta,\ v,\ p) \ \Rightarrow \max_{w,\ \delta,v} J = \max_{w,\delta,\ v} \sum_{k=1}^{N} \eta e_k^2 p_k.$$

By defining $q_k = -p_k = \exp(-\eta e_k^2)$:

$$\min_{w,\ \delta,v} J = \min_{w,\delta,\ v} \sum_{k=1}^{N} \eta e_k^2 q_k \tag{12}$$

Let assume η is constant and expand (12) as:

$$\overset{Eq.(8)}{\Rightarrow} J = \sum_{k=1}^{n} \left(\left(\sum_{m=1}^{M} \left(l_k^m - ps_m^k\right)\right)^2 \times q_k\right)$$

$$\overset{Eq.3}{\Rightarrow} J = \sum_{k=1}^{n} \left(\left(\sum_{m=1}^{M} \left(l_k^m - \left(\sum_{c=1}^{C} w_k^{c,m} \times o_k^c\right)\right)\right)^2 \times q_k\right)$$

$$= \sum_{k=1}^{n} \left(\left(\sum_{m=1}^{M} \sum_{c=1}^{C} \left(l_k^m - w_k^{c,m} \times o_k^c\right)\right)^2 \times q_k\right)$$

$$\overset{Eq.4}{\Rightarrow} J = \sum_{k=1}^{n} \left(\left(\sum_{m=1}^{M} \sum_{c=1}^{C} \left(l_k^m - w_k^{c,m} \times exp\left(-\frac{\|x_k - v_c\|^2}{\sigma_c^2}\right)\right)\right)^2 \times q_k\right) \tag{13}$$

Therefore:

$$\min_{w,\delta,v,p} J(w,\sigma,v) = \min_{w,\delta,v,p} \sum_{k=1}^{n} \left(\left(\sum_{m=1}^{M} \sum_{c=1}^{C} \left(l_k^m - w_k^{c,m} \times exp\left(-\frac{\|x_k - v_c\|^2}{\sigma_c^2}\right)\right)\right)^2 \times q_k\right) \tag{14}$$

Based on (14), the effective parameters include weight classifier layer (w), the shape and location of the nodes in the clustering layer (v) and the width of activation degree function (δ). It is so clear that the parameter δ is important because it affects w directly. Moreover, the calculation of the optimal value of parameter v in the objective function practically is impossible and will be very time consuming. Therefore, if the optimal value of parameter δ can be determined, other parameters will be optimal. In conclusion, it can be derived:

$$J(w,\delta,v) \cong J(w,\delta) \tag{15}$$

Next, we use the gradient search method in the following form:

$$\begin{pmatrix} w \\ d \end{pmatrix}_{k+1} = \begin{pmatrix} w \\ d \end{pmatrix}_{k} - \mu \hat{\nabla} \begin{pmatrix} w \\ d \end{pmatrix}_{k} \tag{16}$$

Now, Eq. (16) is simplified in two steps: one step is an adjustment of clustering layer and the other is the adjustment of classification layer. For clustering layer, it is:

$$\delta_{k+1} = \delta_k - \mu_\delta \hat{\nabla}_{\delta_k} \tag{17}$$

And in classification layer,

$$w_{k+1} = w_k - \mu_w \hat{\nabla}_{w_k} \tag{18}$$

By derivation of each variable, the minimal cost function is achieved by the corresponding variable (μ_δ and μ_w).

3.2. Parameters adaption

To obtain the adaptive value of δ, we use a least mean algorithm for Eq. (17) as:

$$\hat{\nabla}_{\delta_k} = \frac{\partial J_k}{\partial_{\delta_k}} = 2e_k \frac{\partial e_k}{\partial_{\delta_k}} \tag{19}$$

The optimal value of δ is calculated through a derivation of the objective function (14) based on δ as follows:

$$\frac{\partial e_k}{\partial \delta_k} = \frac{\partial \left(\sum_{k=1}^{n} \sum_{m=1}^{M} \sum_{c=1}^{C} \left(l_k^m - w_k^{c,m} exp\left(-\frac{\|x_k - v_c\|^2}{\sigma_c^2}\right)\right) \times q_k\right)}{\partial \delta_k}$$

$$= 0 - \frac{\partial\left(\sum_{t=1}^{n}\left(\sum_{m=1}^{M}\sum_{c=1}^{C}\left(w_k^{c,m} exp\left(-\frac{\|x_t-v_c\|^2}{\sigma_c^2}\right)\right)\right)\times q_k\right)}{\partial\delta_k} \quad (20)$$

If c is replaced with Best Matching Node (BMN of winning node) then (20) is rewritten as follows,

$$= -\frac{\partial}{\partial\delta_k}\left(\sum_{k=1}^{n}\left(\sum_{m=1}^{M}\left(w_k^{BMN,m}\times\exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\right)\times q_k\right)\right)$$
$$= -\frac{\sum_{k=1}^{n}\sum_{m=1}^{M}\left(2\times w_k^{BMN,m}\times exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\times\left(\|x_k-v_{BMN}\|^2\right)\right)\times q_k}{\delta_{BMN}^3} \quad (21)$$

So, according to Eq. (19), δ is updated as follows:

$$\delta_{k+1} = \delta_k + 4\mu_\delta \times \frac{\sum_{k=1}^{n}\sum_{m=1}^{M}\left(I_k^m - PS_m\right)\times\left(w_k^{BMN,m}\times\exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\times\left(\|x_k-v_{BMN}\|^2\right)\right)\times q_k}{\delta_{BMN}^3}$$
$$\nabla\delta = 4\mu_\delta \times \frac{\sum_{k=1}^{n}\sum_{m=1}^{M}\left(w_k^{BMN,m}\times\left(I_k^m - PS_m\right)\times\exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\times\left(\|x_k-v_{BMN}\|^2\right)\right)q_k}{\delta_{BMN}^3} \quad (22)$$

The coefficient μ_δ affects the convergence speed of δ. For the sake of simplicity, the learning rate can be assumed as a constant and positive value that is reduced by time and exceeding the boundary confidence.

To obtain adaptive w, likewise the δ, we have:

$$\hat{\nabla}_{w_t}\left(e_k^2\right) = \frac{\partial J_k}{\partial w_k} = 2e_k\frac{\partial e_k}{\partial w_k} \quad (23)$$

where

$$\frac{\partial e_k}{\partial w_k} = \frac{\partial\left(\sum_{k=1}^{n}\sum_{m=1}^{M}\sum_{c=1}^{C}\left(I_k^m - w_k^{c,m} exp\left(-\frac{\|x_k-v_c\|^2}{\sigma_c^2}\right)\right)\times q_k\right)}{\partial w_k}$$
$$= 0 - \frac{\partial\left(\sum_{k=1}^{n}\sum_{m=1}^{M}\sum_{c=1}^{C}\left(w_k^{c,m}\times exp\left(-\frac{\|x_k-v_c\|^2}{\sigma_c^2}\right)\right)\times q_k\right)}{\partial w_k} \quad (24)$$

If c is replaced with Best Matching Node (BMN of winning node) then (24) is rewritten as follows:

$$= -\frac{\partial\left(\left(\sum_{k=1}^{n}\left(\sum_{m=1}^{M}\left(w_k^{BMN,m}\times\exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\right)\times q_k\right)\right)\right)}{\partial w_k}$$
$$= -\sum_{k=1}^{n}\left(\left(\exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\right)\times q_k\right) \quad (25)$$

Therefore, w is updated as follows:

$$w_{k+1} = w_k + 2\mu_w\times\sum_{k=1}^{n}\sum_{m=1}^{M}\left(I_k^m - ps_k^m\right)\times\exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\times q_k$$
$$\nabla w = 2\mu_w\times\sum_{k=1}^{n}\sum_{m=1}^{M}\left(I_k^m - ps_k^m\right)\times\exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\times q_k \quad (26)$$

3.3. Parameters adaption through back propagation method

In addition to Section 3.1, the back propagation method can also be used for error estimation. In the back propagation method, the error is defined as Eq. (27).

$$e_k = \sum_{m=1}^{M}\left(I_k^m - ps_k^m\right)\times w_k^m \quad (27)$$

Hence, (21) id rewritten as:

$$\frac{\partial e_k}{\partial\delta_k} = \frac{\partial\left(\sum_{k=1}^{n}\sum_{m=1}^{M}\sum_{c=1}^{C}\left(I_k^m - w_k^{c,m}\times exp\left(-\frac{\|x_k-v_c\|^2}{\sigma_c^2}\right)\right)\times w_k^{c,m}\times q_k\right)}{\partial\delta_k}$$

After simplification:

$$\frac{\partial e_k}{\partial\delta_k} = -\sum_{k=1}^{n}\sum_{m=1}^{M}\left(2\times\left(w_k^{BMN,m}\right)^2\times exp\left(-\frac{\|x_k-v_c\|^2}{\sigma_{BMN}^2}\right)\times\left(\frac{\|x_k-v_c\|^2}{\sigma_{BMN}^3}\right)\right)\times q_k\right)$$
$$\Rightarrow\nabla\delta = 4\mu_\delta \times \frac{\sum_{k=1}^{n}\sum_{m=1}^{M}\left(\left(w_k^{BMN,m}\right)^2\times\left(I_k^m - ps_k^m\right)\times\exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\times\left(\|x_k-v_{BMN}\|^2\right)\right)q_k}{\delta_{BMN}^3} \quad (28)$$

Also, for w, it is as follows:

$$\frac{\partial e_k}{\partial w_k} = \frac{\partial\left(\sum_{k=1}^{n}\sum_{m=1}^{M}\sum_{c=1}^{C}\left(I_k^m - PS_m\right)\times w_m^k\times q_k\right)}{\partial w_k} \quad (29)$$

After simplification, we have:

$$\frac{\partial e_k}{\partial w_k} = -2\times\sum_{k=1}^{n}\sum_{m=1}^{M}\sum_{c=1}^{C} w_k^{BMN,m} \times \exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\times q_k$$
$$\Rightarrow\nabla w = 4\mu_w\times\sum_{k=1}^{n}\sum_{m=1}^{M}\left(I_k^m - ps_k^m\right)\times\left(w_k^{BMN,m}\right)^2 \times \exp\left(-\frac{\|x_k-v_{BMN}\|^2}{\sigma_{BMN}^2}\right)\times q_k \quad (30)$$

3.4. η tuning

In this section, we discuss adaption of the η parameter. We define a constraint for η parameter over the minimization problem as follows:

$$\min_{\eta_i}\sum_{i=1}^{N}\eta_i^2 e_i^2 q_i,\ \sum\eta_i = p,\ \eta a_i\in[0,1] \quad (31)$$

Producing the Lagrange equation by adding the penalty term, we have:

$$l = \sum_{i=1}^{N}\eta_i^2 e_i^2 q_i - \lambda\left(\sum\eta_i - p\right) \quad (32)$$

Then, we obtain η_i as follows:

$$\frac{\partial l}{\partial\eta_i} = 0\ \Rightarrow\eta_i = \frac{\lambda}{2e_i^2 q_i}\ \overset{\sum\eta_i = p}{\Rightarrow}\ \eta_i = \frac{p}{\sum_{j=1}^{n}\frac{e_j^2 q_j}{e_i^2 q_i}} \quad (33)$$

To deal with division by zero problems, we add ξ parameter to the denominator of the equation as follows:

$$\eta_i = \frac{p}{\sum_{j=1}^{n}\frac{e_i^2 q_i}{e_j^2 q_j} + \xi} \quad (34)$$

3.5. Structure of the proposed method

As shown in Fig. 2, the proposed algorithm receives labeled and unlabeled data continuously. Depending on the type of the data, two events may happen:

- Unlabeled data are received: GSOM performs clustering and updates weight of nodes in the clustering layer. In this study, we assumed that most of data sizes are unlabeled.
- Data with known labeled is received: First, the winner node in the clustering layer is identified and then the label of data is estimated by the system based on this node. The system error is calculated from the difference between actual Label and estimated Label system. This error, which updates weight between layers of clustering and classification, is applied to determine the width degree of activation of each node.

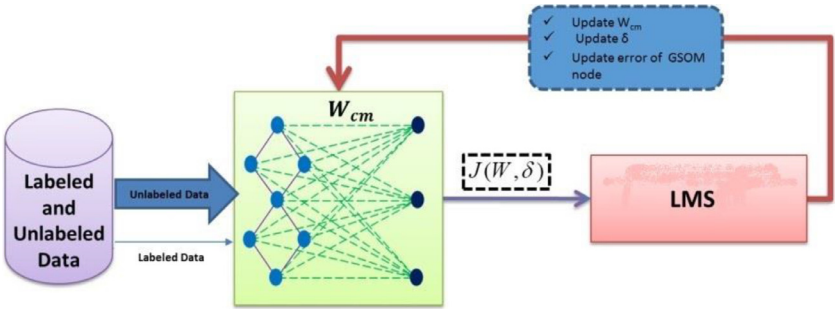

Fig. 2. The proposed model for creation adaptive Semi-supervised GSOM.

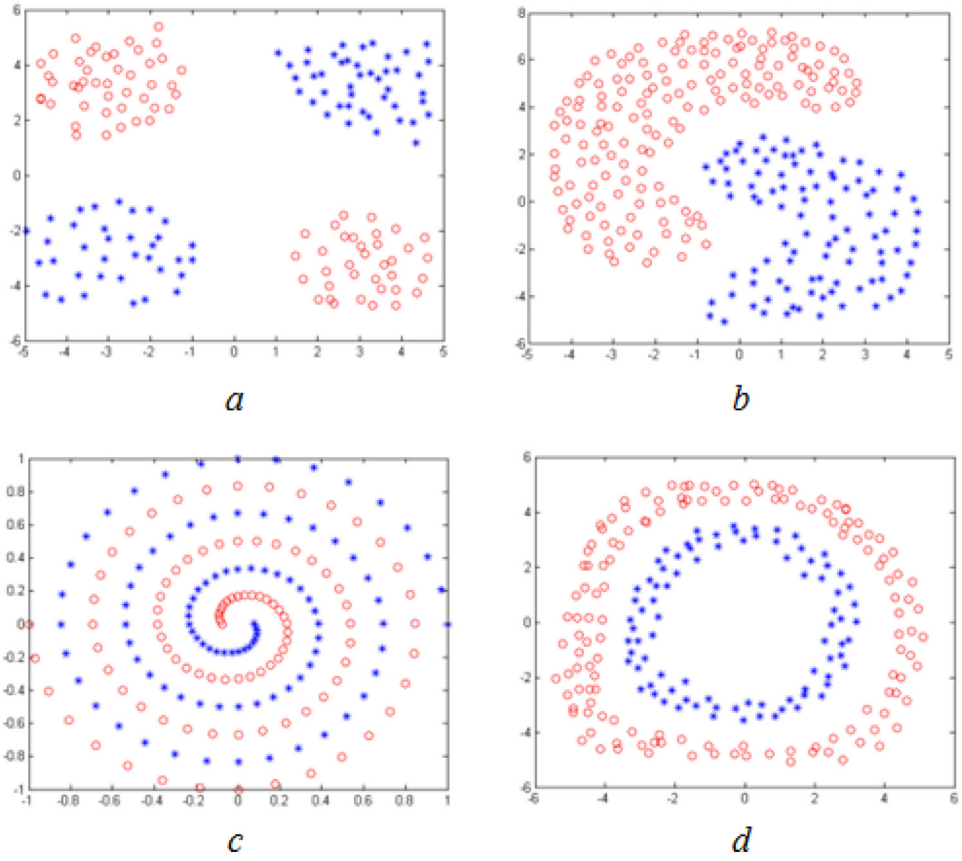

Fig. 3. Synthetic evaluated datasets: (a) four cluster (b) two moon (c) spiral (d) two ring.

4. Performance evaluation and tests

In this section, the efficiency of the proposed algorithm is evaluated. The datasets used for the evaluation include a synthetic and UCI repository,[2] which are detailed in the next sections.

Three online algorithms were selected among the state of the art methods mentioned in Section 1. Table 2 shows title and abbreviation for the two proposed method and state of the art methods.

The measures considered for assessing the proposed method and those from the literature include Accuracy and F-measure. The Accuracy measure calculates based on (35). In (35), TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively.

$$Acc_q = \frac{TP}{TP + TN + FP + FN} \tag{35}$$

[2] http://archive.ics.uci.edu/ml/

Table 2
Abbreviation and title of proposed and state of the art methods.

Abbreviated name	Title of method
ASSGSOM	Adaptive Semi Supervised GSOM
BASSGSOM	Back propagation Adaptive Semi Supervised GSOM
SSGSOM	Hsu's Semi-supervised GSOM{Hsu, 2008 #20}
CS2GS	Constrained Semi-Supervised GSOM{Allahyar, 2015 #2}
HSS	Halkidi's Semi Stream algorithm{Halkidi, 2012 #18}

Also, Precision and Recall measures are defined as follows:

$$PR = \frac{TP}{TP + FP} \tag{36}$$

$$RE = \frac{TP}{TP + FN} \tag{37}$$

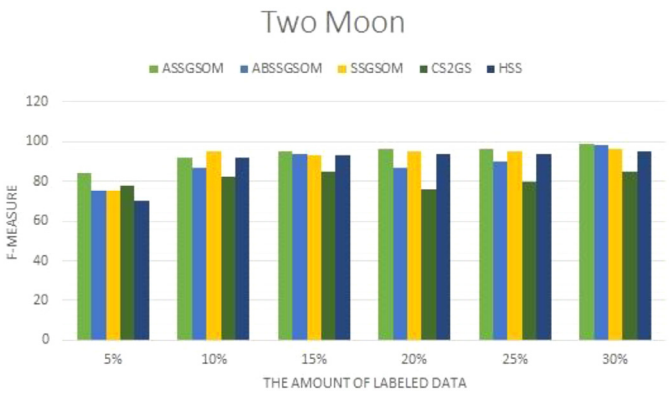

Fig. 4. Result on Two Moon dataset.

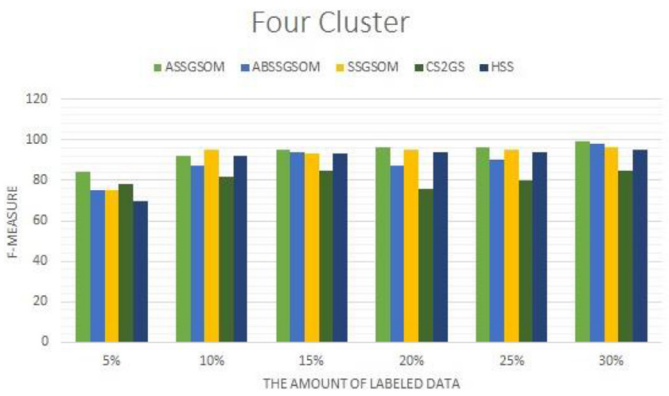

Fig. 5. Result on Four Cluster dataset.

And then *F*-measure, which is based on precision and recall, is defined as follows:

$$FM = 2 \times \frac{PR_q \times RE_q}{PR_q + RE_q} \tag{38}$$

For all algorithms that are based on GSOM (i.e., SSGSOM, CS2GS, and both proposed methods), the scale factor (SF) is considered as a constant value and based on growing threshold of GSOM is calculated as (39). Parameter d in (39) represents data dimension.

$$GT = -d * \log(SF) \tag{39}$$

Furthermore, the regularization parameter for GSOM layer and classification layer is very influential on the results. So, in all of the experiments, constant values of 0.4 and 0.6 were used for the regularization parameter for GSOM and Classification Layer, respectively. Besides, the initial value of w (weight of classification layer), v (weight of GSOM node) and other settings related to GSOM, are selected randomly and are the same for all algorithms. The order of data entry, which affects the performance of the algorithms is the same for all of them.

The test scenario is that at first each data set is divided into the train and test subsets considering a 10-folds cross-validation. Then, 5% of training data is labeled randomly and all the methods are trained using them. Next, the performance of each algorithm is evaluated using test data. This process is replicated 5 times and the average of results is reported. In the following, the number of labeled training data is increased by 5% and the test scenario is repeated. This procedure will continue until the number of labeled data is 30%.

4.1. The evaluation results on synthetic datasets

In this section, the performances of the proposed methods and state of the art methods are shown on synthetic datasets. The structures of the synthetic datasets are shown in Fig. 3. All these data sets have two classes. Also, the number of data in two moons, four clusters, two ring and spiral sets are 258, 160, 237, and 3000, respectively.

Figs. 4–7 present the obtained results using the synthetic datasets based on F-measure.

As shown in Figs. 4–7, ASSGSOM method is superior compared to other methods. The closest results to this method are those obtained by SSGSOM method. Moreover, a statistical *t*-test was established to determine whether the difference between these methods is statistically significant. Table 3, presents the *t*-test comparison between ASSGSOM and SSGSOM. In this test, 15% of the data is labeled and a number of iteration is 35.

In Table 3, if the value of *t*-test is 1, it means that the method with a higher average is statistically superior. For example, in the

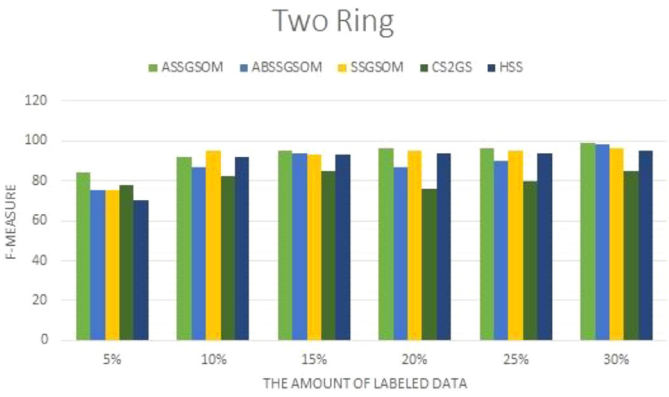

Fig. 6. Result on Two Ring dataset.

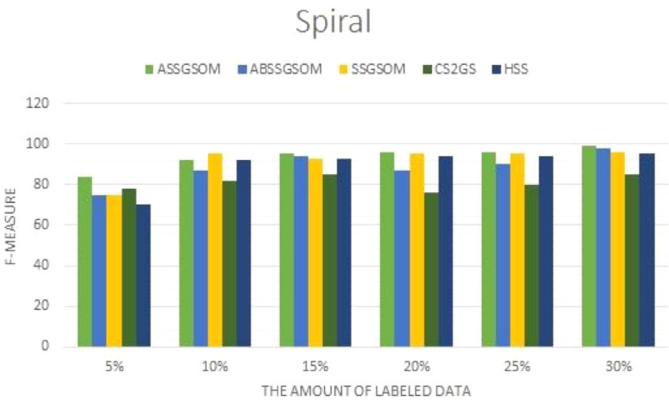

Fig. 7. Result on Spiral dataset.

Two Ring dataset, the value of Accuracy *t*-test column for ASSGSOM is 1. It means that Accuracy result of ASSGSOM is statistically superior to the SSGSOM because the average of the accuracy of the ASSGSOM algorithm (98%) from SSGSOM (93%) is higher, then the ASSGSOM algorithm is superior.

4.2. The evaluation results on UCI datasets

In this section, the performance of the proposed methods on UCI datasets is evaluated. Several datasets from UCI repository (Blake & Merz, 1998) were selected to evaluate the proposed method. Table 4 shows the details of each dataset.

In Figs. 8–13, results on a part of UCI datasets are shown and detailed results are reported in Table 5.

Results in Figs. 8–13 indicate the relative superiority of ASSGSOM compared to state of art methods. As can be seen, when the

Table 3
The *t*-test results between ASSGSOM and SSGSOM.

Dataset	method	Accuracy (%)	*F*-Measure (%)	Accuracy *t*-test	*F*-Measure *t*-test
Two Moon	ASSGSOM	96	95	1	1
	SSGSOM	88	87		
Four Cluster	ASSGSOM	94	94	1	1
	SSGSOM	86	85		
Two Ring	ASSGSOM	98	98	1	1
	SSGSOM	93	92		
Spiral	ASSGSOM	88	87	1	1
	SSGSOM	86	85		

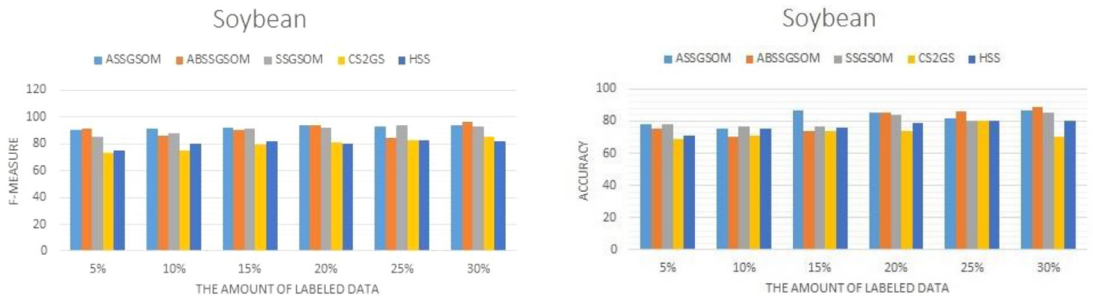

Fig. 8. Result on Spiral dataset.

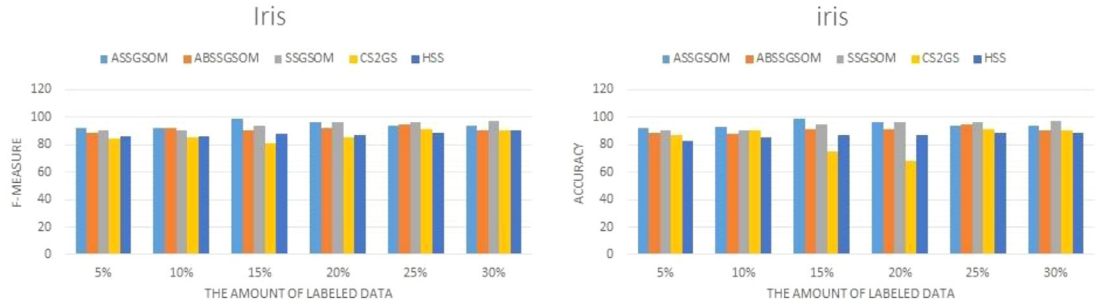

Fig. 9. Result on Iris dataset.

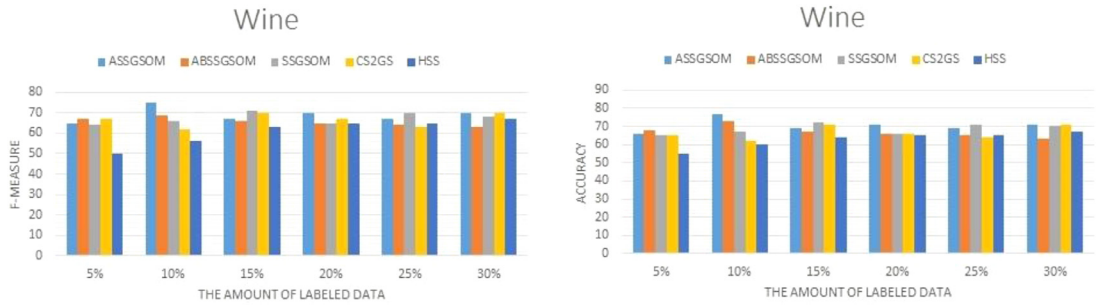

Fig. 10. Result on Wine dataset.

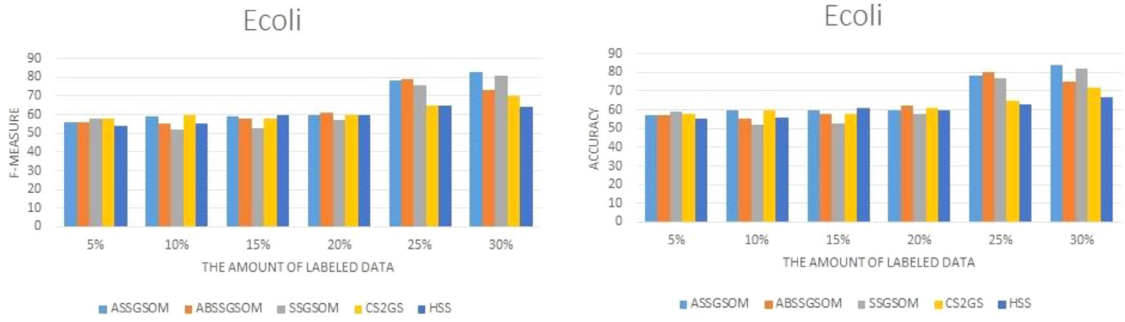

Fig. 11. Result on E.coli dataset.

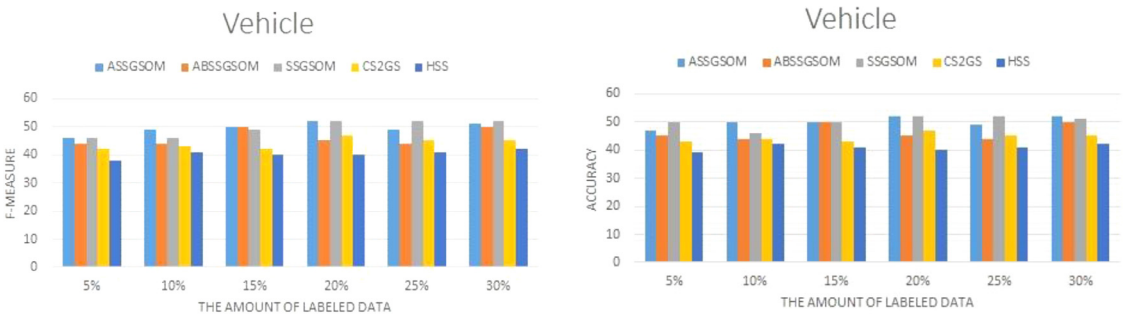

Fig. 12. Result on Vehicle dataset.

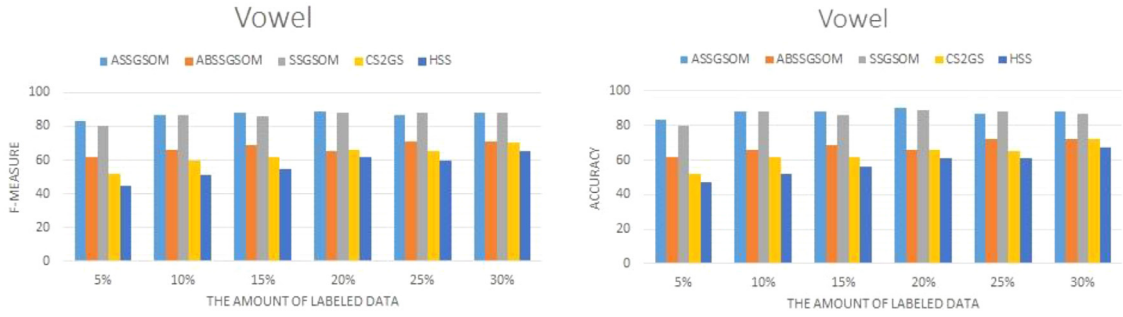

Fig. 13. Result on Vowel dataset.

Table 4
Detail of used UCI repository.

Dataset name	Number of class	Number of features	Number of data
Soybean	4	35	47
Iris	3	4	150
Wine	3	13	178
Sonar	2	60	208
E.coli	6	6	332
Ionosphere	2	34	351
WDBC	2	30	569
Scale	3	4	625
Vehicle	4	18	846
Vowel	11	10	990

number of labeled data is increased the proposed method shows less fluctuate and it becomes more stable.

Results in different UCI dataset are summarized in Table 5. Evaluation is done in the condition that the number of labeled data is 30% of the total data in the dataset.

The results in Table 5 show that when the number of Attributes is high (such as Soybeans and Sonar), ABSSGSOM is better. Except for the results on Iris dataset, in the other datasets ASSGSOM and SSGSOM methods are superior compared to other methods. The results of these two methods are very close to each other but ASSGSOM is superior totally.

Table 5
The results of the proposed methods and state of art methods on UCI datasets.

Dataset/Method	Measure	ASSGSOM (%)	BASSGSOM	SSGSOM (%)	CS2GS (%)	HSS (%)
Soybean	*F*-Measure	94	96%	93	85	82
	Accuracy	87	89%	85	70	80
Iris	*F*-Measure	94	90%	97	90	90
	Accuracy	94	90	97	90	89
Wine	*F*-Measure	70	63%	68	70	67
	Accuracy	71	63%	70	71	67
Sonar	*F*-Measure	60	67%	57	57	51
	Accuracy	61	67%	58	58	51
E.coli	*F*-Measure	83	73%	81	70	64
	Accuracy	84	75%	82	72	67
Ionosphere	*F*-Measure	90	85%	88	86	80
	Accuracy	91	87%	89	87	81
WDBC	*F*-Measure	90	87%	88	87	80
	Accuracy	90	87%	88	87	80
Scale	*F*-Measure	75	70%	71	68	55
	Accuracy	84	76%	84	69	56
Vehicle	*F*-Measure	51	50%	52	45	42
	Accuracy	52	50%	51	45	42
Vowel	*F*-Measure	88	71%	88	64	60
	Accuracy	88	72%	87	65	61

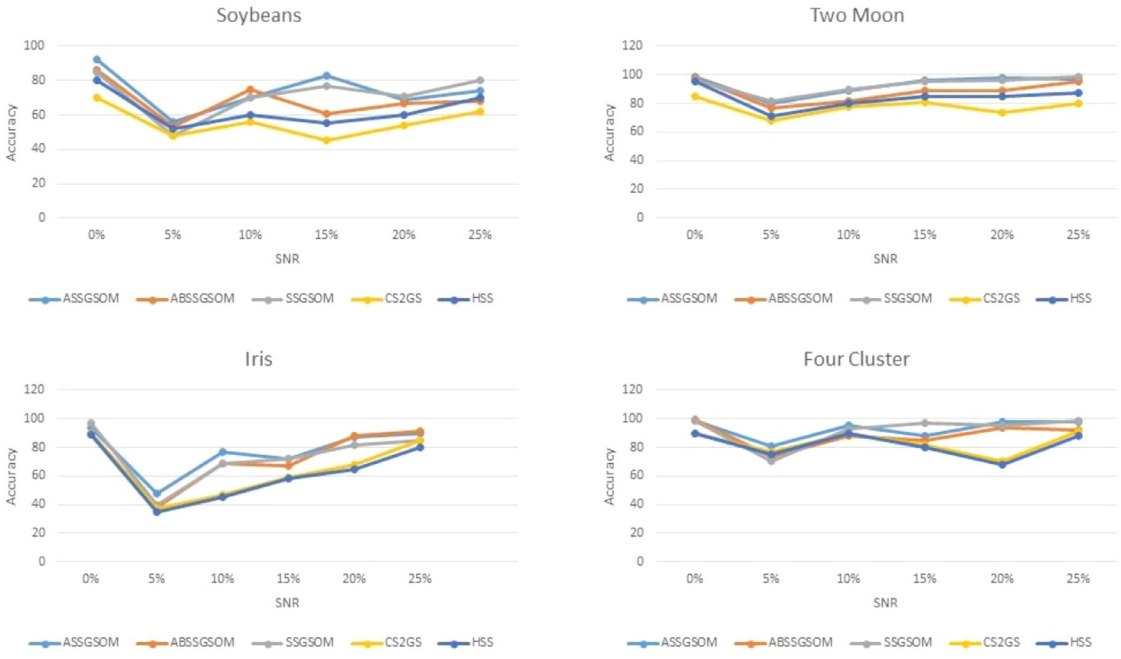

Fig. 14. Effect of the noise on the four datasets selected.

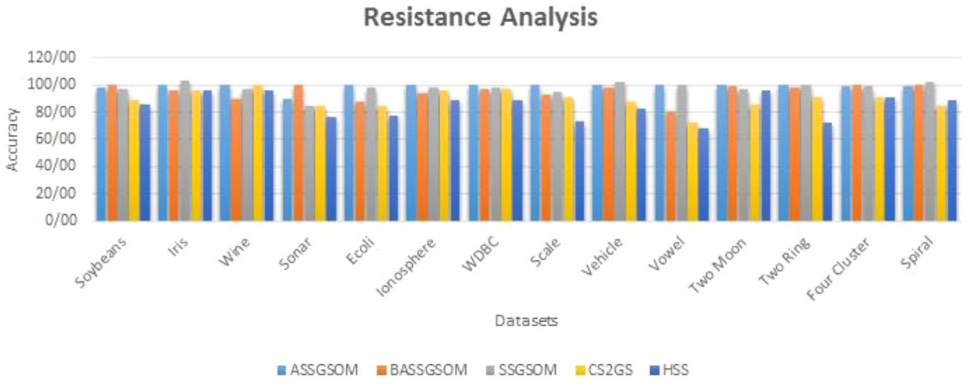

Fig. 15. Resistance analysis of the accuracy of the all evaluated datasets.

4.3. The effect of noise

In this section, noise effect on the proposed methods is evaluated. Fig. 14 presents the results for the datasets contaminated with noise. Eq. (40) is used for calculating the SNR.[3] In this formula, D is the domain of signal.

$$SNR_{dB} = 20log_{10}\left(\frac{D_{signal}}{D_{noise}}\right) \tag{40}$$

4.4. Resistance analysis

Resistance analysis shows the stability of the algorithm in different condition (Geng, Zhan, & Zhou, 2005). Eq. (41) presents how to calculate resistance analysis. Fig. 15 exhibits resistance analysis for the discussed dataset.

$$RA_q = \frac{Acc_q}{\max Acc_j} \; \forall j \in \{1, 2, \ldots, Q\} \tag{41}$$

[3] Signal to noise ratio.

5. Conclusion and future works

In this paper, a comprehensive robust mathematical formulation of the semi-supervised GSOM was proposed and then half quadratic (HQ) was used to solve it. Furthermore, an adaptive method was proposed to adjust activation degree optimally for achieving a better performance of semi-supervised GSOM. Evaluation in noisy conditions and on different datasets suggests that the proposed method has superiority and stability compared to the state of the art methods.

In the future, we add a regular term to cost function to avoid over-fitting the complex dataset. Also, we plan to develop a robust active semi-supervised SSGSOM. Using this active learning, we can determine which data are important for increasing the accuracy. Therefore, a higher degree of accuracy can be achieved with a smaller number of labeled data.

References

Alahakoon, D., Halgamuge, S. K., & Srinivasan, B. (2000). Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Transactions on Neural Networks, 11*(3), 601–614.

Allahyar, A., Yazdi, H. S., & Harati, A. (2015). Constrained semi-supervised growing self-organizing map. *Neurocomputing, 147*, 456–471.

Alpaydin, E. (2004). *Introduction to machine learning*. MIT Press.

Beyer, O., & Cimiano, P. (2012). Online semi-supervised growing neural gas. *International Journal of Neural Systems, 22*(05), 1250023.

Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases, Irvine, CA: Department of Information and Computer Science, University of California, 55. [http://www.ics.uci.edu/~mlearn/MLRepository.html].

Chen, S., Li, S., Su, S., Cao, D., & Ji, R. (2014). Online semi-supervised compressive coding for robust visual tracking. *Journal of Visual Communication and Image Representation, 25*(5), 793–804.

Cong, Y., Liu, J., Yuan, J., & Luo, J. (2013). Self-supervised online metric learning with low rank constraint for scene categorization. *IEEE Transactions on Image Processing, 22*(8), 3179–3191.

Dornaika, F., & El Traboulsi, Y. (2016). Learning flexible graph-based semi-supervised embedding. *IEEE Transactions on Cybernetics, 46*(1), 206–218.

Farid, D. M., Zhang, L., Hossain, M. A., Rahman, C. M., Strachan, R., Sexton, G., et al. (2013). An adaptive ensemble classifier for mining concept drifting data streams. *Expert Systems with Applications, 40*(15), 5895–5906.

Fritzke, B. (1994). Growing cell structures—a self-organizing network for unsupervised and supervised learning. *Neural Networks, 7*(9), 1441–1460.

Geng, X., Zhan, D.-C., & Zhou, Z.-H. (2005). Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 35*(6), 1098–1107.

Halkidi, M., Spiliopoulou, M., & Pavlou, A. (2012). *A semi-supervised incremental clustering algorithm for streaming data advances in knowledge discovery and data mining* (pp. 578–590). Springer.

He, R., Hu, B. G., Zheng, W. S., & Guo, Y. (2010). Two-Stage Sparse Representation for Robust Recognition on Large-Scale Database. In *AAAI: Vol. 10* (p. 1).

He, R., Hu, B.-G., Zheng, W.-S., & Kong, X.-W. (2011). Robust principal component analysis based on maximum correntropy criterion. *IEEE Transactions on Image Processing, 20*(6), 1485–1494.

Hsu, A., & Halgamuge, S. K. (2008). Class structure visualization with semi-supervised growing self-organizing maps. *Neurocomputing, 71*(16), 3124–3130.

Huang, G.-B., Li, M.-B., Chen, L., & Siew, C.-K. (2008). Incremental extreme learning machine with fully complex hidden nodes. *Neurocomputing, 71*(4), 576–583.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine: A new learning scheme of feedforward neural networks. In *Proceedings of the IEEE international joint conference on neural networks*.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE, 78*(9), 1464–1480.

Leite, D., Costa, P., & Gomide, F. (2010). Evolving granular neural network for semi–supervised data stream classification. In *Proceedings of the international joint conference on neural networks (IJCNN)*.

Li, K., Zhang, J., Xu, H., Luo, S., & Li, H. (2013). A semi-supervised extreme learning machine method based on co-training⋆. *Journal of Computational Information Systems, 9*(1), 207–214.

Lu, K., Wang, Q., Xue, J., & Pan, W. (2014). 3D model retrieval and classification by semi-supervised learning with content-based similarity. *Information Sciences, 281*, 703–713.

Macario, V., & de Carvalho, F. d. A. (2012). An adaptive semi-supervised fuzzy clustering algorithm based on objective function optimization. In *Proceedings of the IEEE international conference on fuzzy systems (FUZZ-IEEE)*.

Martinetz, T. M., Berkovich, S. G., & Schulten, K. J. (1993). Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks, 4*(4), 558–569.

Maximo, V. R., Quiles, M. G., & Nascimento, M. C. (2014). A consensus-based semi–supervised growing neural gas. In *Proceedings of the international joint conference on neural networks (IJCNN)*.

Rockafellar, R. (1970). *Convex analysis*. Princeton, NJ: Princeton University Press *Google Scholar*.

Shen, F., Yu, H., Kamiya, Y., & Hasegawa, O. (2010). An online incremental semi–supervised learning method. *Journal of Advanced Computational Intelligence and Intelligent Informatics, 14*(6), 593–605.

Zhang, Y., Wen, J., Wang, X., & Jiang, Z. (2014). Semi-supervised learning combining co-training with active learning. *Expert Systems with Applications, 41*(5), 2372–2378.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 3*(1), 1–130.