Journal of Computer and Communications, 2025, 13(4), 166-179
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

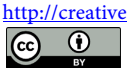
Semi-Supervised Stochastic Configuration Networks Based on Manifold Regularization Framework

Haimei Meng[1], Wu Ai[1,2*]

[1]School of Mathematics and Statistics, Guilin University of Technology, Guilin, China
[2]Guangxi Colleges and Universities Key Laboratory of Applied Statistics, Guilin, China
Email: *aiwu818@gmail.com

How to cite this paper: Meng, H.M. and Ai, W. (2025) Semi-Supervised Stochastic Configuration Networks Based on Manifold Regularization Framework. *Journal of Computer and Communications*, **13**, 166-179.
https://doi.org/10.4236/jcc.2025.134011

Received: March 5, 2025
Accepted: April 24, 2025
Published: April 27, 2025



Abstract

The stochastic configuration network (SCN) is an incremental neural network with fast convergence, efficient learning and strong generalization ability, and is widely used in fields such as medical data analysis. However, SCN is mainly used for supervised learning and its performance is limited in the case of scarce labeled data. To this end, this paper proposes semi-supervised SCN (MR-SCN) in combination with manifold regularization to make full use of unlabeled data to improve the model performance. Experimental results show that MR-SCN can still maintain high classification accuracy with a small number of labeled samples, which is better than LapRLS, SS-ELM and LapSVM, and the training time is shorter, showing good learning ability and computational efficiency.

Keywords

Semi-Supervised Learning, Manifold Regularization, Stochastic Configuration Network

1. Introduction

In many practical applications, acquiring a large amount of labeled data is usually costly, especially in fields such as medical diagnosis, autonomous driving, and natural language processing, where manual labeling is not only time-consuming and laborious, but may also require specialized knowledge. On the other hand, unlabeled data is often abundant and easily available, so how to efficiently utilize unlabeled data becomes a key issue. Supervised learning requires a large amount of labeled data but the labeling cost is high, and unsupervised learning only relies

on the intrinsic structure of the data and lacks clear guidance signals, which usually makes it difficult to achieve the effect of supervised learning. Semi-supervised learning emerges as a solution to efficiently and fully utilize cheap and easily available unlabeled data. This approach integrates the characteristics of supervised learning and unsupervised learning, leveraging the synergy between a small amount of labeled data and a large quantity of unlabeled data. It not only reduces dependence on labeled data and improve the model's generalization ability while ensuring the accuracy.

For the classification scenario of semi-supervised learning, the representative algorithms can be roughly divided into four categories, namely discriminant-based methods [1] [2], difference-based methods [3], generation-based methods [4], and graph-based methods [5]. Among them, graph-based methods are hotspots in the field of semi-supervised learning, which abstract all data into a graph and use the graph to characterize the similarities between data pairs thus revealing the distribution of the data, but their essence is still propagated by labeling [6]. Manifold regularization is a semi-supervised learning technique based on graphs, which builds a manifold structure of the data. This enables the model to leverage the distribution information from unlabeled data, thereby enhancing its learning ability.

Therefore, many researchers have combined the manifold regularization framework with different classification models to fully utilize the information of unlabeled data and improve the classification performance of the models. Zhao *et al.* proposed a semi-supervised broad learning system (SS-BLS) by combining manifold regularization with broad networks, which improves feature extraction and classification under limited labeled data by utilizing the information of the manifold structure of the data [7]. Belkin *et al.* applied manifold regularization to support vector machines (SVMs) and constructed semi-supervised SVMs to improve the model learning performance [8]. Huang *et al.* integrated the manifold regularization framework with extreme learning machines (ELMs), introducing semi-supervised and unsupervised ELMs to enhance the learning ability of ELMs with limited or unlabeled data [9]. And Li *et al.* introduced the manifold regularization framework into the multilayer extreme learning machine (ML-ELM) and proposed the LAP-ML-ELM model to enhance the adaptability of deep learning methods in semi-supervised environments [10]. These studies show that manifold regularization has broad application prospects in enhancing the learning ability of classification models, especially in semi-supervised and unsupervised learning tasks, which can effectively utilize the manifold structure of data and improve the classification accuracy and generalization ability of the model.

With the development of information technology and the improvement of data storage capacity, human society has stepped into the era of big data. Characterized by huge volume, rapid growth, and diverse types, big data has a profound impact on various fields, and at the same time brings challenges in data storage, management and analysis. In this context, machine learning (ML), as an important

branch of artificial intelligence, has attracted widespread attention because of its ability to automatically learn data patterns and make predictive decisions. Traditional analysis methods can hardly cope with the scale and complexity of big data, while deep learning (DL), as the core direction of machine learning, has made remarkable achievements in computer vision, natural language processing, financial forecasting and other fields with the help of the continuous evolution of neural networks (e.g., DBN, CNN, RNN). Advances in large-scale data and high-performance computing devices have further boosted the development of deep learning, enabling it to play a key role in the age of intelligence.

However, models such as CNN [11] and RNN [12] usually rely on the BP algorithm to calculate gradients layer by layer to update weights, which results in a complex training process, large number of parameters, and high computational cost. In contrast, randomized learning (RL) has shown broad prospects in the field of machine learning due to its efficient modeling capabilities. Randomized learning technology started in the 1980s and was further developed in the 1990s. Pao and Takefuji proposed the random vector function link network (RVFL) [13] [14], and Schmidt *et al.* proposed the random weight feed-forward neural network (FNNRW) [15]. The core idea is to randomly initialize the weights and biases of the hidden layer and use the least squares method to calculate the output weights, thereby simplifying the training process and improving learning efficiency. However, subsequent studies have shown that the universal approximation of RVFL and FNNRW depends on the number of hidden layer nodes and the range of random parameters, and whether the model can approximate the target function with high probability depends on whether the parameters are properly selected. To enhance the generalization ability of randomized neural networks, Wang and Li proposed the stochastic configuration network (SCN) in 2017 [16]. SCN adopts an incremental learning method and introduces a supervision mechanism to adaptively adjust the range of random parameters through inequality constraints to ensure universal approximation, reduce human intervention, and improve the learning accuracy and training efficiency of the model.

In order to improve the performance and stability of SCNs, scholars have proposed SCNs based on L1 regularization [17], SCNs based on L2 regularization [18], and SCNs based on Dropout regularization [19], which reduce the over-fitting during model increment. In addition, Wang and Li proposed a robust SCN using kernel density estimation method to calculate the penalty weights of the training samples, which improves the generalization of the learning model by reducing the negative impact of noise or outliers [20]. To solve the problem of time-consuming model training in SCNs, a bidirectional SCN algorithm [21] was introduced to categorize the addition of hidden nodes into forward learning mode and backward learning mode. In addition, block-based incremental SCNs [22] and hybrid parallel SCNs [23] were developed to improve the modeling speed and shorten the training time. Deep SCNs [24] were also developed based on SCNs, which provide faster and more extensive network generation. There are also other

excellent SCN-based models, such as deep stacked SCN [25], distributed SCN [26], etc. Although SCNs have many important variants and are widely used in areas such as hardware implementation, computer vision, medical data analysis, and fault detection and diagnosis, these models are mainly targeted at supervised learning tasks, and SCN models dedicated to semi-supervised learning have not yet emerged. In some cases, such as text categorization, information retrieval, and fault diagnosis, acquiring labeled data is both time-consuming and expensive, while a large amount of unlabeled data is simple and cheap to collect. Therefore, it is of significant research value to explore how to design a semi-supervised SCN by combining the unique supervisory mechanism of SCNs and its universal approximation properties with graph regularization frameworks, such as manifold regularization, in order to improve the learning efficiency while guaranteeing the learning accuracy.

In this paper, we will explore the combination of SCNs and semi-supervised learning, aiming to improve the model's learning ability on complex network data by constructing a semi-supervised learning framework based on SCNs. Through theoretical analysis and experimental validation, we hope to reveal the potential of SCNs in semi-supervised learning and provide new ideas and methods for future research.

2. Preliminaries

2.1. Stochastic Configuration Network (SCN)

SCN is a stochastic incremental learning model proposed in the last few years, whose network structure can grow gradually according to the training data. Specifically, it starts with a simple small neural network, randomly configures the input weights and biases through a data-dependent inequality supervision mechanism, and gradually adds new hidden layer nodes until a predefined termination condition is met, then stops and successfully generates a SCN network model. This incremental structural growth allows the SCN to adaptively scale the network to accommodate data of varying complexity.

Consider an objective function $f:\mathbb{R}^d \to \mathbb{R}^m$, then a SCN network with $L-1$ hidden nodes can be given by the following formula.

$$f_{L-1}(\boldsymbol{x}) = \sum_{l=1}^{L-1} \boldsymbol{\beta}_l g_l(\boldsymbol{x}) = \sum_{l=1}^{L-1} \boldsymbol{\beta}_l g_l(\boldsymbol{x}, \boldsymbol{w}_l, b_l), \ f_0 = 0, \tag{1}$$

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{\beta}_l = \left[\boldsymbol{\beta}_{l,1}, \boldsymbol{\beta}_{l,2}, \cdots, \boldsymbol{\beta}_{l,m}\right]^\top$ is the output weights of the l-th hidden node, $\boldsymbol{w}_l$ and b_l respectively represent the input weight and bias of the l-th hidden node, $g_l(\cdot)$ is the activation function of the l-th hidden node. In practice, the sigmoid function is frequently employed as the activation function, expressed as $g(\boldsymbol{x}, \boldsymbol{w}, b) = \dfrac{1}{1+\mathrm{e}^{-\left(\boldsymbol{w}^\top \boldsymbol{x}+b\right)}}$. The residual of the current network can be expressed as

$$e_{L-1} = f - f_{L-1} = \left[e_{L-1,1}, e_{L-1,2}, \ldots, e_{L-1,m}\right]. \tag{2}$$

If $\|e_{L-1}\|$ does not reach the preset expected residuals, the hidden layer will add new nodes. We must create a new random basis function g_L ($\boldsymbol{w}_L$ and b_L) under the supervision mechanism, and then recompute the output weights $\boldsymbol{\beta}_L$. This adjustment ensures that the model $f_L = f_{L-1} + \boldsymbol{\beta}_L g_L$ achieves a reduced residual error.

In practice, let the input matrix be $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ and the targets matrix be $\boldsymbol{T} \in \mathbb{R}^{N\times m}$. Denote $h_L(\boldsymbol{X})$ as the output of the L-th hidden node for $\boldsymbol{X}$

$$h_L(\boldsymbol{X}) = \left[g_L(\boldsymbol{x}_1), \cdots, g_L(\boldsymbol{x}_N)\right]^\top, \tag{3}$$

where $g_L(\boldsymbol{x}_n) = g(\boldsymbol{x}_n, \boldsymbol{w}_L, b_L), n = 1, 2, \cdots, N$. The output matrix of the entire hidden layer can be written as

$$\boldsymbol{H}_L(\boldsymbol{X}) = \left[h_1(\boldsymbol{X}), \cdots, h_L(\boldsymbol{X})\right]. \tag{4}$$

The residual matrix is represented as $e_{L-1}(\boldsymbol{X}) \in \mathbb{R}^{N\times m}$, where $e_{L-1,q}(\boldsymbol{X}) = \left[e_{L-1,q}(\boldsymbol{x}_1), \cdots, e_{L-1,q}(\boldsymbol{x}_N)\right]^\top \in \mathbb{R}^N$, $q = 1, 2, \cdots, m$. According to the global approximation theorem proposed by Wang [16], a new set of random basis functions $g_L(\boldsymbol{w}_L, b_L)$ is generated when the residuals $\|e_{L-1}\|$ do not reach the pre-set target values . If the new vector g_L satisfies the inequality

$$\xi_{L,q}(\boldsymbol{X}) = \frac{\left(e_{L-1,q}^\top(\boldsymbol{X}) \cdot h_L(\boldsymbol{X})\right)^2}{h_L^\top(\boldsymbol{X}) \cdot h_L(\boldsymbol{X})} - (1 - r - \mu_L) e_{L-1,q}^\top(\boldsymbol{X}) e_{L-1,q}(\boldsymbol{X}) \geq 0, \tag{5}$$

then the new input weights $\boldsymbol{w}_L$ and bias b_L can enter the candidate node pool, and the size of the candidate pool is denoted by $T_{\max}$, where $\xi_L(\boldsymbol{X})$ is defined as

$$\xi_L(\boldsymbol{X}) = \sum_{q=1}^{m} \xi_{L,q}(\boldsymbol{X}). \tag{6}$$

The node with the largest $\xi_L(\boldsymbol{X})$ in the candidate pool becomes the added node. Then, the output weights can be obtained by the following optimization problem

$$\left[\boldsymbol{\beta}_1^*, \boldsymbol{\beta}_2^*, \cdots, \boldsymbol{\beta}_L^*\right]^\top = \arg\min_{\beta} \left\| f - \sum_{l=1}^{L} \boldsymbol{\beta}_l g_l \right\|. \tag{7}$$

Then, we have $\lim_{L\to+\infty} \left\| f - f_L^* \right\| = 0$, where $f_L^* = \sum_{l=1}^{L} \boldsymbol{\beta}_l^* g_l$, $\boldsymbol{\beta}_l^* = \left[\boldsymbol{\beta}_{l,1}^*, \boldsymbol{\beta}_{l,2}^*, \cdots, \boldsymbol{\beta}_{l,m}^*\right]^\top$.

2.2. Semi-Supervised Learning

Manifold regularization is a semi-supervised learning approach based on graphs, which creates a manifold structure for the data. This allows the model to leverage the distributional information from unlabeled data, thereby enhancing its learning capacity. The core idea of this framework is that high-dimensional data is usually distributed on a low-dimensional manifold, so the model can be constrained by manifold regularization to make it change smoothly on the data

manifold, thereby improving the generalization ability of classification and regression tasks. The manifold regularization framework is usually based on semi-supervised learning, combining information from labeled and unlabeled data, and mainly includes the following assumptions:

Assumption 1 (Smoothness Assumption [8]) *If two input samples* $\boldsymbol{x}_1, \boldsymbol{x}_2$ *are close to each other, then the corresponding conditional probabilities* $\mathcal{P}(\boldsymbol{y} \mid \boldsymbol{x}_1), \mathcal{P}(\boldsymbol{y} \mid \boldsymbol{x}_2)$ *should be similar.*

Assumption 2 (Cluster Assumption [8]) *The decision boundary should be positioned within a low-density region of the input space* X.

Assumption 3 (Manifold Assumption [8]) *The marginal distribution* $\mathcal{P}_X$ *is defined over a low-dimensional manifold* $\mathcal{M}$ *embedded in* X.

Based on the above assumptions, the manifold regularization framework introduces an additional manifold regularization term into the traditional supervised learning loss function to maintain the smoothness of the model over the manifold structure.

To impose the assumptions on the data, the manifold regularization framework minimizes the following loss function

$$\|f\|_M^2 = \frac{1}{2}\sum_i^N \sum_j^N W_{ij}\left(\mathcal{P}(\boldsymbol{y} \mid \boldsymbol{x}_i) - \mathcal{P}(\boldsymbol{y} \mid \boldsymbol{x}_j)\right)^2, \tag{8}$$

where W_{ij} is the pairwise similarity between two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and $W = \left[W_{i,j}\right]_{N\times N}$, N is the total number of samples, including l labeled samples and u unlabeled samples. Regarding the computation of W, according to the work in [27], W can be defined by the Gaussian kernel function, *i.e.*,

$$W_{i,j} = \mathrm{e}^{-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}}, \tag{9}$$

where $\sigma > 0$ is the regulation parameter.

It is well known that it is practically difficult to calculate the conditional probabilities $\mathcal{P}(\boldsymbol{y} \mid \boldsymbol{x}_i)$ and $\mathcal{P}(\boldsymbol{y} \mid \boldsymbol{x}_j)$. Therefore, for the convenience of calculation, the predicted output of the model can be used to replace the conditional probabilities and obtain the following approximate expression

$$\|f\|_M^2 = \frac{1}{2}\sum_i^N \sum_j^N W_{ij}\left(\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_j\right)^2, \tag{10}$$

where $\hat{\boldsymbol{y}}_i$ and $\hat{\boldsymbol{y}}_j$ are the predictions for samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively.

By defining the total predicted output $\hat{Y} = \left[\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \cdots, \hat{\boldsymbol{y}}_N\right]$ and the diagonal elements to be $D_{ii} = \sum_{j=1}^N W_{ij}$ of the diagonal matrix D, we can simplify (10) to a matrix form

$$\|f\|_M^2 = Tr\left(\hat{Y}^\top \mathcal{L} \hat{Y}\right), \tag{11}$$

where $\mathcal{L} \in \mathbb{R}^{N\times N}$ is the Laplacian matrix and $\mathcal{L} = D - W$. Furthermore, in order to facilitate calculation, Belkin *et al.* [8] recommended using the normalized Laplacian matrix $\tilde{\mathcal{L}} = D^{-\frac{1}{2}} \mathcal{L} D^{-\frac{1}{2}}$ instead of $\mathcal{L}$.

3. Semi-Supervised SCNs Based on Manifold Regularization Framework

In the practical applications of semi-supervised learning, a common situation is that labeled data is scarce while unlabeled data is abundant. This situation of uneven data distribution is one of the challenges that semi-supervised learning methods need to focus on. To fully leverage the substantial amount of unlabeled data and improve the classification accuracy in the presence of scarce labeled samples, we combine the SCN with the manifold regularization framework and propose a semi-supervised SCN algorithm called the MR-SCN algorithm. In semi-supervised learning, the dataset $\mathcal{S}$ is set to contain N^l labeled samples and N^u unlabeled samples, denoted as $\mathcal{S}^l = \left\{\left(\boldsymbol{x}_n^l, \boldsymbol{t}_n^l\right)\right\}_{n=1}^{N^l}$, $\mathcal{S}^u = \left\{\boldsymbol{x}_n^u\right\}_{n=1}^{N^u}$, where $N = N^l + N^u$ is the total number of samples. The traditional SCN is a supervised learning, which only needs labeled data for model training. For the labeled data set $\mathcal{S}^l = \left\{\left(\boldsymbol{x}_n^l, \boldsymbol{t}_n^l\right)\right\}_{n=1}^{N^l}$, we can get the optimization objective function of the standard SCN as

$$\begin{aligned} \min \quad & \sum_{n=1}^{N^l} \left\|\boldsymbol{\xi}_n\right\|^2 + \frac{C}{2}\|\boldsymbol{\beta}\|^2 \\ \text{s.t.} \quad & \boldsymbol{h}\left(\boldsymbol{x}_n^l\right)\boldsymbol{\beta} = \boldsymbol{t}_n^{l\top} - \boldsymbol{\xi}_n^\top, n = 1, 2, \cdots, N^l. \end{aligned} \tag{12}$$

where the first term is the loss function, $\boldsymbol{\xi}_n$ is the training error vector of the output neuron corresponding to the training sample $\boldsymbol{x}_n^l$, $\boldsymbol{h}\left(\boldsymbol{x}_n^l\right) = \left[g_1\left(\boldsymbol{x}_n^l\right), \cdots, g_L\left(\boldsymbol{x}_n^l\right)\right] = \left[g\left(\boldsymbol{x}_n^l, \boldsymbol{w}_1, b_1\right), \cdots, g\left(\boldsymbol{x}_n^l, \boldsymbol{w}_L, b_L\right)\right]$ is the output of the hidden node relative to the input sample $\boldsymbol{x}_n^l$; the second term is the L_2 regularization term, and $C > 0$ is the regularization parameter.

By incorporating the loss function of the manifold regularization framework into the conventional supervised SCN formula (12), we can derive a semi-supervised SCN algorithm.

$$\begin{aligned} \min \quad & \sum_{n=1}^{N^l} \left\|\boldsymbol{\xi}_n\right\|^2 + \frac{C}{2}\|\boldsymbol{\beta}\|^2 + \frac{\eta}{2} Tr\left(\boldsymbol{F}^\top \mathcal{L} \boldsymbol{F}\right) \\ \text{s.t.} \quad & \boldsymbol{h}\left(\boldsymbol{x}_n^l\right)\boldsymbol{\beta} = \boldsymbol{t}_n^{l\top} - \boldsymbol{\xi}_n^\top, \ n = 1, 2, \cdots, N^l, \\ & \boldsymbol{f}_n = \boldsymbol{h}\left(\boldsymbol{x}_n\right)\boldsymbol{\beta}, \ n = 1, 2, \cdots, N, \end{aligned} \tag{13}$$

where $\mathcal{L} \in \mathbb{R}^{N \times N}$ is the graph Laplacian matrix constructed by labeled samples and unlabeled samples, $\boldsymbol{F} \in \mathbb{R}^{N \times m}$ is the final output matrix of the network, whose n-th row represents the output $\boldsymbol{f}\left(\boldsymbol{x}_n\right)$ of the n-th sample , and $\eta > 0$ is the trade-off parameter of the manifold regularization term.

Let

$$\boldsymbol{H}^l = \begin{bmatrix} \boldsymbol{h}\left(x_1^l\right) \\ \boldsymbol{h}\left(x_2^l\right) \\ \vdots \\ \boldsymbol{h}\left(x_{N^l}^l\right) \end{bmatrix}, \ \boldsymbol{T}^l = \begin{bmatrix} \boldsymbol{t}_1^{l\top} \\ \boldsymbol{t}_2^{l\top} \\ \vdots \\ \boldsymbol{t}_{N^l}^{l\top} \end{bmatrix}, \ \boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}\left(x_1\right) \\ \boldsymbol{h}\left(x_2\right) \\ \vdots \\ \boldsymbol{h}\left(x_N\right) \end{bmatrix}. \tag{14}$$

Substituting the constraints in formula (13) into the objective function, we can obtain the equivalent unconstrained optimization objective function

$$\min \ \frac{1}{2}\left\|\boldsymbol{H}^{l}\boldsymbol{\beta}-\boldsymbol{T}^{l}\right\|^{2}+\frac{C}{2}\|\boldsymbol{\beta}\|^{2}+\frac{\eta}{2}\boldsymbol{\beta}^{\top}\boldsymbol{H}^{\top}\mathcal{L}\boldsymbol{H}\boldsymbol{\beta}. \tag{15}$$

According to (15), we can easily derive the optimal coefficient vectors of MR-SCN by gradient operation as follows

$$\boldsymbol{\beta}^{*}=\left(\boldsymbol{H}^{l\top}\boldsymbol{H}^{l}+C\boldsymbol{I}+\eta\boldsymbol{H}^{\top}\mathcal{L}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{l\top}\boldsymbol{T}^{l\top}. \tag{16}$$

4. Numerical Experiments

In this section, we conduct experiments using four publicly available benchmark datasets to evaluate the performance of the proposed MR-SCN. The 2Moons dataset is a classical artificially generated binary categorization dataset, which consists of two clusters of points in the shape of two interleaved half moons, usually each category contains the same number of data points. The G50C dataset is a typical binary classification dataset, where each class is generated from a 50-dimensional multivariate Gaussian distribution. The COIL20 dataset is an image recognition task that consists of 1,440 images of 20 different objects taken from different angles, each with a size of 32 × 32 gray pixels. The Image Segmentation dataset is derived from the UCI Machine Learning Repository, which contains multiple image samples, each sample consists of multiple features, including information such as color, texture and location, labeled into different classes, and the dataset contains multiple classes such as buildings, trees and people and is suitable for evaluating the performance of various image processing and machine learning algorithms. All input variables are normalized before conducting the experiments. All simulation experiments conducted in this study were performed using MATLAB R2022b.

In the experiment, in order to meet the needs of semi-supervised learning, we divide the dataset into three parts: labeled dataset L, unlabeled dataset U and test dataset T, and the specific features are shown in Table 1.

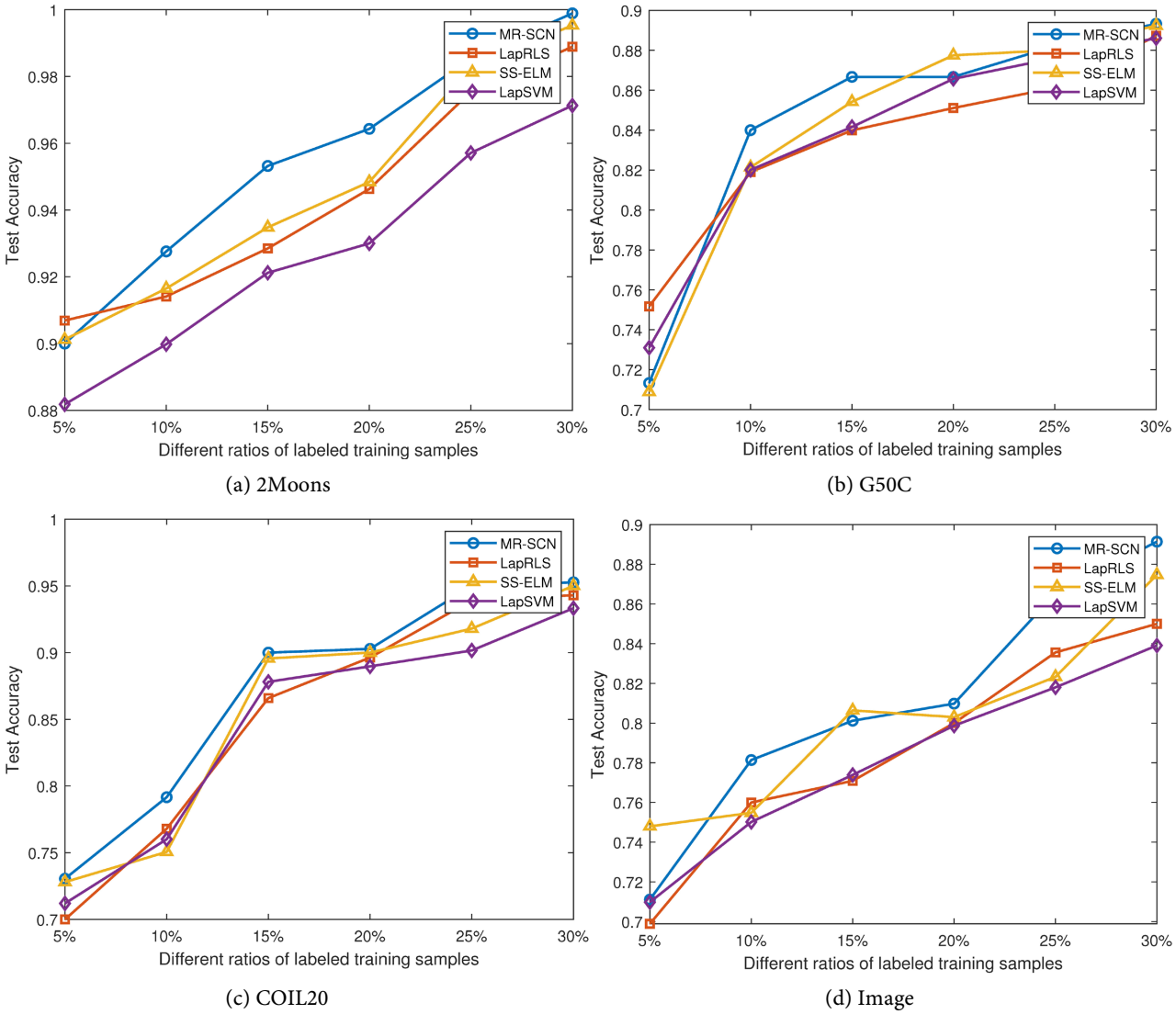
Table 1. Details of the datasets.

Dataset	L	U	T	Attributes	Class
2Moons	15	285	100	2	2
G50C	50	314	186	50	2
COIL20	40	1000	400	1024	20
Image	50	1450	810	19	7

Here, we first compare and analyze the proposed MR-SCN algorithm with the traditional supervised learning algorithm SCN, and the experimental results are shown in Table 2. From Table 2, it can be seen that the proposed semi-supervised algorithm MR-SCN is able to achieve comparable classification results with the supervised learning algorithm SCN on most datasets in terms of classification

Table 2. Performance of the two algorithms on different datasets.

Dataset	Method	Training RMSE	Test RMSE	Training Time(s)
2Moons	SCN	0.0099	0.0127	0.345
	MR-SCN	0.0091	0.1088	20.0072
G50C	SCN	0.0071	0.0406	2.4566
	MR-SCN	0.0094	0.0811	54.6992
COIL20	SCN	0.0109	0.1835	30.3567
	MR-SCN	0.0095	0.1063	78.0631
Image	SCN	0.2334	0.4399	15.9883
	MR-SCN	0.0093	0.1825	57.1824



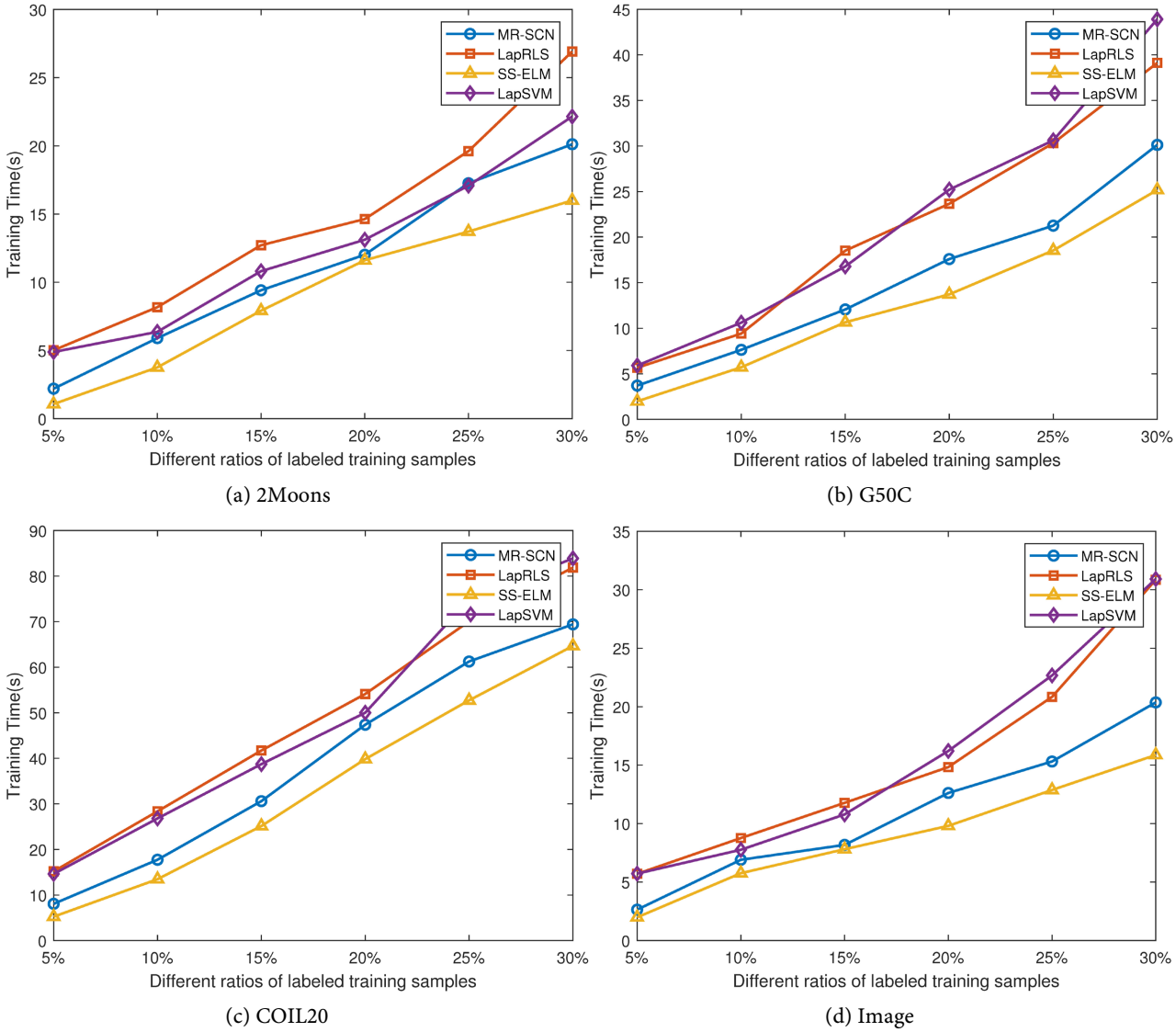
(a) 2Moons

(b) G50C

(c) COIL20

(d) Image

Figure 1. Test accuracy for different numbers of labeled training samples.



(a) 2Moons

(b) G50C

(c) COIL20

(d) Image

Figure 2. Training time for different numbers of labeled training samples.

performance. This shows that MR-SCN can effectively use unlabeled data for training, so that it can still maintain high classification accuracy with fewer labeled samples. However, in terms of training time, MR-SCN is much higher than SCN. The main reason is that MR-SCN needs to calculate the Laplacian matrix, which involves the construction of graph structure and feature extraction, adding additional computational overhead.

In order to investigate the effect of different number of labeled samples on the performance of the algorithm, we selected six different labeled sample ratios in the experiments: 5%, 10%, 15%, 20%, 25% and 30%. We compare and analyze the experimental results using three algorithms, SS-ELM, LapRLS and LapSVM. In this experiment, the hidden layer activation function of both MR-SCN and SS-ELM algorithms adopts the Sigmoid function, and the regularization parameters of all four algorithms are from the range of $\left[10^{-5},10^{-4},\cdots,10^{4},10^{5}\right]$.

Figure 1 shows the trend of the test accuracy of each algorithm with different proportions of labeled training samples. It can be observed that all four methods, MR-SCN, LapRLS, SS-ELM and LapSVM, show good classification performance on all datasets, and the overall test accuracy tends to increase with the increase of the proportion of labeled training samples, implying that more labeled data helps the model to more accurately learn the features of the data distributions and improve the generalization ability. Furthermore, when the proportion of labeled training samples remains constant, the MR-SCN method consistently achieves higher test accuracy compared to the three other methods. This suggests that MR-SCN demonstrates superior classification performance in semi-supervised learning settings. Particularly when the proportion of labeled samples is large, the MR-SCN method achieves test accuracies nearing 100% on the 2Moons and COIL20 datasets, demonstrating its strong ability to learn and generalize effectively. In terms of training time, as can be seen from Figure 2, the training time of all methods increases as the proportion of labeled training samples increases, which is due to the fact that more labeled data increases the amount of computation. With the same proportion of labeled samples, LapRLS and LapSVM take the longest training time, while MR-SCN and SS-ELM have relatively short and similar training times. This indicates that MR-SCN also has high computational efficiency while ensuring the classification performance.

5. Conclusions

In this paper, we introduce MR-SCN, a semi-supervised learning algorithm built upon the manifold regularization framework and SCN. This method can effectively utilize a small set of labeled data along with a large amount of unlabeled data for semi-supervised classification. It not only improves the computational speed and generalization ability, but also overcomes the limitation of the SCN's dependence on the labeled data, further expanding the scope of SCN's application. To evaluate the performance of MR-SCN, we performed experiments using four different datasets: 2Moons, G50C, COIL20 and Image. The results show that compared with the supervised learning algorithm SCN, MR-SCN maintains high classification accuracy with fewer labeled samples, but its training time increases due to the computation of Laplacian matrix. In addition, MR-SCN outperforms LapRLS, SS-ELM and LapSVM in terms of classification accuracy across different labeled training sample ratios. It also requires less training time, demonstrating excellent learning capability and computational efficiency. Overall, MR-SCN can effectively balance the classification performance and computational cost in semi-supervised learning tasks, and has high application value.

However, the MR-SCN algorithm also has certain limitations. The four datasets used in this experiment, 2Moons, G50C, COIL20 and Image, gradually increase in size. Correspondingly, the training time of MR-SCN on these datasets also gradually increases. This is because the input weights and biases of SCN are set based on the data-driven supervision mechanism, and the calculation of the manifold

regularization term requires considering the similarity between all samples. For large-scale datasets, this results in higher computational costs and longer training time.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No. 62166013), the Natural Science Foundation of Guangxi (No. 2022GXNSFAA035499) and the Foundation of Guilin University of Technology (No. GLUTQD2007029).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Tavernier, J., Simm, J., Meerbergen, K., Wegner, J.K., Ceulemans, H. and Moreau, Y. (2019) Fast Semi-Supervised Discriminant Analysis for Binary Classification of Large Data Sets. *Pattern Recognition*, **91**, 86-99. https://doi.org/10.1016/j.patcog.2019.02.015

[2] Wang, S., Lu, J., Gu, X., Du, H. and Yang, J. (2016) Semi-Supervised Linear Discriminant Analysis for Dimension Reduction and Classification. *Pattern Recognition*, **57**, 179-189. https://doi.org/10.1016/j.patcog.2016.02.019

[3] Peng, J., Estrada, G., Pedersoli, M. and Desrosiers, C. (2020) Deep Co-Training for Semi-Supervised Image Segmentation. *Pattern Recognition*, **107**, Article 107269. https://doi.org/10.1016/j.patcog.2020.107269

[4] Zhang, B., Bai, B. and Su, J. (2007) Semi-Supervised Text Classification Based on Self-Training EM Algorithm. *Journal-National University of Defense Technology*, **29**, 65-69.

[5] Song, Z., Yang, X., Xu, Z. and King, I. (2023) Graph-Based Semi-Supervised Learning: A Comprehensive Review. *IEEE Transactions on Neural Networks and Learning Systems*, **34**, 8174-8194. https://doi.org/10.1109/tnnls.2022.3155478

[6] Chong, Y., Ding, Y., Yan, Q. and Pan, S. (2020) Graph-Based Semi-Supervised Learning: A Review. *Neurocomputing*, **408**, 216-230. https://doi.org/10.1016/j.neucom.2019.12.130

[7] Zhao, H., Zheng, J., Deng, W. and Song, Y. (2020) Semi-Supervised Broad Learning System Based on Manifold Regularization and Broad Network. *IEEE Transactions on Circuits and Systems I: Regular Papers*, **67**, 983-994. https://doi.org/10.1109/tcsi.2019.2959886

[8] Belkin, M., Niyogi, P. and Sindhwani, V. (2006) Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research*, **7**, 2399-2434.

[9] Huang, G., Song, S., Gupta, J.N.D. and Cheng Wu, (2014) Semi-Supervised and Unsupervised Extreme Learning Machines. *IEEE Transactions on Cybernetics*, **44**, 2405-2417. https://doi.org/10.1109/tcyb.2014.2307349

[10] Li, K., Zhang, J., Xu, H., Luo, S. and Li, H. (2013) A Semi-Supervised Extreme Learning Machine Method Based on Co-Training. *Journal of Computer Information Systems*, **9**, 207-214.

[11] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017) ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, **60**, 84-90. https://doi.org/10.1145/3065386

[12] Medsker, L.R. and Jain, L. (2001) Recurrent Neural Networks: Design and Applications. CRC Press.

[13] Pao, Y.-H. and Takefuji, Y. (1992) Functional-Link Net Computing: Theory, System Architecture, and Functionalities. *Computer*, **25**, 76-79. https://doi.org/10.1109/2.144401

[14] Pao, Y., Park, G. and Sobajic, D.J. (1994) Learning and Generalization Characteristics of the Random Vector Functional-Link Net. *Neurocomputing*, **6**, 163-180. https://doi.org/10.1016/0925-2312(94)90053-1

[15] Schmidt, W.F., Kraaijveld, M.A. and Duin, R.P.W. (1992) Feedforward Neural Networks with Random Weights. Proceedings., 11*th IAPR International Conference on Pattern Recognition. Vol. II. Conference B: Pattern Recognition Methodology and Systems*, The Hague, 30 August-3 September 1992, 1-4. https://doi.org/10.1109/icpr.1992.201708

[16] Wang, D. and Li, M. (2017) Stochastic Configuration Networks: Fundamentals and Algorithms. *IEEE Transactions on Cybernetics*, **47**, 3466-3479. https://doi.org/10.1109/tcyb.2017.2734043

[17] Wang, Q., Dai, W., Lu, Q., Fu, X. and Ma, X. (2022) A Sparse Learning Method for SCN Soft Measurement Model. *Control and Decision*, **37**, 3171-3182.

[18] Zhao, L., Zou, S., Guo, S. and Huang, M. (2020) Ball Mill Load Condition Recognition Model Based on Regularized Stochastic Configuration Networks. *Control Engineering of China*, **27**, 1-7.

[19] Li, W., Zeng, Z., Qu, H. and Sun, C. (2019) A Novel Fiber Intrusion Signal Recognition Method for OFPS Based on SCN with Dropout. *Journal of Lightwave Technology*, **37**, 5221-5230. https://doi.org/10.1109/jlt.2019.2930624

[20] Wang, D. and Li, M. (2017) Robust Stochastic Configuration Networks with Kernel Density Estimation for Uncertain Data Regression. *Information Sciences*, **412**, 210-222. https://doi.org/10.1016/j.ins.2017.05.047

[21] Cao, W., Xie, Z., Li, J., Xu, Z., Ming, Z. and Wang, X. (2021) Bidirectional Stochastic Configuration Network for Regression Problems. *Neural Networks*, **140**, 237-246. https://doi.org/10.1016/j.neunet.2021.03.016

[22] Dai, W., Li, D., Zhou, P. and Chai, T. (2019) Stochastic Configuration Networks with Block Increments for Data Modeling in Process Industries. *Information Sciences*, **484**, 367-386. https://doi.org/10.1016/j.ins.2019.01.062

[23] Dai, W., Zhou, X., Li, D., Zhu, S. and Wang, X. (2022) Hybrid Parallel Stochastic Configuration Networks for Industrial Data Analytics. *IEEE Transactions on Industrial Informatics*, **18**, 2331-2341. https://doi.org/10.1109/tii.2021.3096840

[24] Wang, D. and Li, M. (2018) Deep Stochastic Configuration Networks with Universal Approximation Property. 2018 *International Joint Conference on Neural Networks* (*IJCNN*), Rio de Janeiro, 8-13 July 2018, 1-8. https://doi.org/10.1109/ijcnn.2018.8489695

[25] Pratama, M. and Wang, D. (2019) Deep Stacked Stochastic Configuration Networks for Lifelong Learning of Non-Stationary Data Streams. *Information Sciences*, **495**, 150-174. https://doi.org/10.1016/j.ins.2019.04.055

[26] Ai, W. and Wang, D. (2020) Distributed Stochastic Configuration Networks with Cooperative Learning Paradigm. *Information Sciences*, **540**, 1-16.

https://doi.org/10.1016/j.ins.2020.05.112

[27] Fierimonte, R., Scardapane, S., Uncini, A. and Panella, M. (2017) Fully Decentralized Semi-Supervised Learning via Privacy-Preserving Matrix Completion. *IEEE Transactions on Neural Networks and Learning Systems*, **28**, 2699-2711. https://doi.org/10.1109/tnnls.2016.2597444