Distributed Semi-Supervised Learning Algorithms for Random Vector Functional-Link Networks with Distributed Data Splitting across Samples and Features

Jin Xie[a], Sanyang Liu[a,*], Hao Dai[b], Yue Rong[c]

[a]*School of Mathematics and Statistics, Xidian University, Xi'an 710071, PR China*
[b]*School of Aerospace Science and Technology, Xidian University, Xi'an 710071, PR China*
[c]*Department of Electrical and Computer Engineering, Curtin University, Bentley, WA 6102, Australia*

Abstract

In this paper, we propose two manifold regularization (MR) based distributed semi-supervised learning (DSSL) algorithms using the random vector functional link (RVFL) network and alternating direction method of multipliers (ADMM) strategy. In DSSL problems, training data consisting of labeled and unlabeled samples are often large-scale or high-dimension and split across samples or features. These distributed data separately stored over a communication network where each node has only access to its own data and can only communicate with its neighboring nodes. In many scenarios, centralized algorithms cannot be applied to solve DSSL problems. In our previous work, we proposed a MR based DSSL algorithm, denoted as the D-LapWNN algorithm, to solve DSSL problems with distributed samples. It has been proved that the D-LapWNN algorithm, combining the wavelet neural network (WNN) with the zero-gradient-sum (ZGS) strategy, is an efficient DSSL algorithm with distributed samples or horizontally partitioned data. The drawback of the D-LapWNN algorithm is that the loss function of each node or agent over the communication network must be twice continuously differentiable. In order to extend our previous work and settle the corresponding drawback, we propose a horizontally DSSL (HDSSL) algorithm to solve DSSL problems with distributed samples. Then, we novelly propose a vertically DSSL (VDSSL) algorithm to solve DSSL problems with distributed features or vertically partitioned data. As far as we know, the VDSSL algorithm is the first work focusing on DSSL problems with distributed features. During the learning process of the proposed algorithms, nodes over the communication network only exchange coefficients rather than raw data. It means that the proposed algorithms are privacy-preserving methods. Finally, some simulations are given to show the efficiency of the proposed algorithms.

Keywords: Distributed learning (DL); Semi-supervised learning (SSL); Manifold regularization (MR); Alternating direction method of multipliers (ADMM); Random vector functional link (RVFL); Distributed features.

1. Introduction

Up to now, many supervised learning (SL) algorithms have been proposed to learn from labeled training samples. Usually, the labels of these samples are costly or difficult to get and should be artificially added. As a result, the amount of unlabeled data is much more than labeled data in the real world. To make better use of unlabeled samples, semi-supervised learning (SSL) methods have been well developed. In [1] and [2], the semi-supervised support vector machine (S3VM) and Laplacian regularization least square (LapRLS) algorithms are derived from the manifold regularization (MR) framework which is based on the assumption that similar samples have similar outputs. At the same time, the transductive support vector machine (TR-SVM) and transductive random vector functional link (TR-RVFL) algorithms, based on the transductive learning (TL) framework, are proposed in [3] and [4], respectively. Compared with MR based approaches, TL based SSL algorithms regard unknown labels as additional variables in optimization problems and perform better when labels are only required for a given test set. In addition, there are many other SSL algorithms [2, 5, 6]. These algorithms have been widely used in many practical applications such as text classification [7], scene recognition [8], financial fraud detection [9], visual location recognition [10], feature analysis for video semantic recognition [11], image-to-video adaptation for video action recognition [12], and so on. In this paper, we focus on the MR framework because its concise form and outstanding effect.

However, traditional centralized algorithms, such as [13–15], can not be used in many distributed scenarios where training data are located on distributed nodes and unable or costly to be transmitted over communication networks. Moreover, each node has only access to its own data and can only communicate with its neighboring nodes. For example, each node of a wireless sensor network (WSN) can only transfer a small amount of information limited by bandwidth [16] and medical information of each community hospital cannot be shared or centrally processed due to privacy protection [17]. For such distributed learning (DL) problems, traditional SL and SSL algorithms become useless. In order to solve these DL problems, many

*Corresponding author
Email addresses: xj6417@126.com (Jin Xie), liusanyang@126.com (Sanyang Liu), hdai@xidian.edu.cn (Hao Dai), Y.Rong@curtin.edu.au (Yue Rong)

distributed SL (DSL) and SSL (DSSL) algorithms have been proposed. In [18], the authors propose a DSL algorithm based on the random vector functional-link (RVFL) network and alternating direction method of multipliers (ADMM) strategy [19]. The authors in [20] propose a DSL algorithm based on the feed-forward neural network with random weights (FNNRW) and zero gradient sum (ZGS) strategy [21], which is denoted as the ZGS-FNNRW algorithm. In order to reduce communication times of each node over the communication network, the authors in [22] apply the event-triggered (ET) communication scheme to the ZGS-FNNRW algorithm. In [23], the authors propose two DSL algorithms based on the ZGS strategy, namely a continuous-time one and a discrete-time one. Moreover, the authors of [24] propose a diffusion adaptation framework for DL problems.

As for DSSL problems, there are some algorithms proposed. In [25], the authors propose two asynchronously distributed approaches for graph-based SSL. The first approach is based on stochastic approximation and the second one uses randomized Kaczmarz algorithm. The authors in [26] provide an error analysis for DSSL with kernel ridge regression based on a divide-and-conquer strategy. The authors in [27] propose two frameworks for distributed semi-supervised metric learning, which are based on the diffusion and ADMM strategies, respectively. The authors in [28] investigate the problem of learning a semi-supervised SVM with distributed data over a network of interconnected agents and propose a DSSL algorithm based on the in-network successive convex approximation (NEXT) framework [29]. In [30], the authors propose a MR based DSSL algorithm and denote it as the D-LapRLS algorithm. This algorithm, known as the best MR based DSSL algorithm at the time, uses the kernel method and distributed average consensus (DAC) strategy [31].. However, the D-LapRLS algorithm had the common drawback of kernel based methods namely the calculation of Euclidian distance matrix (EDM) with respect to total samples over the communication network. To solve DSSL problems and overcome the common drawback of kernel based algorithms, we proposed a novel MR based DSSL algorithm, denoted as D-LapWNN, in our previous work [32]. The D-LapWNN algorithm combines the wavelet neural network (WNN) with the ZGS strategy and works in a fully distributed fashion. It has been proved that the D-LapWNN algorithm is an efficient and privacy-preserving method. But there's a limitation to the D-LapWNN algorithm, namely the loss function of each node or agent over the communication network must be twice continuously differentiable.

In DL problems, distributed samples are often large-scale and high-dimension. However, the aforementioned DL algorithms only focus on large-scale data which are split across samples or horizontally partitioned. In other words, these algorithms only care about the amount rather than the dimension of training data. In practice, there are many applications in which data is split across features. For example, different commercial organizations possess some information of users, but these data cannot be shared due to personal privacy. However, only a few researchers have paid attentions to the distributed features, which is also called vertically partitioned data. The authors in [33] address the problem of association rule mining in vertically partitioned data where transactions are distributed across sources. In [34], the authors propose a DSL algorithm to learn an RVFL network from vertically partitioned data sources based on the ADMM strategy. As far as we know, no DSSL algorithms with vertically partioned data has been proposed yet.

In order to extend our previous work [32], we propose a horizontally DSSL (HDSSL) algorithm in this paper to solve DSSL problems with distributed samples. The HDSSL algorithm is based on RVFL network and the ADMM strategy. Owing to the ADMM strategy, the loss function of each node over the communication network does not require to be differentiable. As an extension of D-LapWNN, the HDSSL algorithm takes all advantages of the D-LapWNN algorithm and becomes more flexible. For DSSL problems with distributed features or vertically partitioned data, we propose a novel DSSL algorithm with the distributed data splitting across features. We denote this algorithm as the vertically DSSL (VDSSL) algorithm. The proposed HDSSL and VDSSL algorithms are based on RVFL networks and the ADMM strategy. During the learning process of the proposed algorithms, each node over the communication network only exchanges the updated coefficients rather than raw data. In addition, the efficiency of the proposed algorithms are proved by some illustrative simulations.

The contributions of this paper are summarized as follows.

- The HDSSL algorithm is proposed to extend the D-LapWNN algorithm. It focus on solving DSSL problems with large-scale dataset splitting across samples or horizontally partitioned data.
- The VDSSL algorithm is novelly proposed for solving DSSL problems with vertically partitioned data. In these problems, training data is high dimension and distributed across features over the communication network.
- The proposed algorithms are based on the RVFL network and the MR framework rather than kernel method, which avoid them to calculate the global EDM with respect to total samples.
- The proposed algorithms work in fully distributed fashions, where each node over the communication network only has access to its own data and communicates the updated coefficients rather than raw data. It means that the proposed algorithms are privacy preserving methods.
- Owing to the ADMM strategy, the proposed algorithms can be improved more easily by changing the loss function. Compared with the ZGS strategy used in the D-LapWNN algorithm, the ADMM is more flexible.

The rest of this paper is organized as follows. Firstly, some preliminaries are introduced in Section 2. In Section 3, we formulate the DSSL problems with the distributed training data splitting across samples and features. Then, we propose two novel algorithms to solve the corresponding DSSL problems. Section 4 provides some illustrative simulations to show the efficiency and advantages of the proposed algorithms. Then, some conclusions are drawn in Section 5.

2. Preliminaries

In this section, we will introduce some basic knowledge of this paper in 4 parts. Part 1 shows some definitions and conclusions of RVFL networks. In Part 2, some conceptions and conclusions of SSL are introduced. Then, the ADMM and DAC strategy are described in Part 3 and 4, respectively.

Notation: In this paper, we give the following notation: $\mathbf{R}$ denotes the set of real numbers, $\mathbf{x} \in \mathbf{R}^n$ stands for a $n \times 1$ real vectors, $\mathbf{A} \in \mathbf{R}^{n \times n}$ represents an $n \times n$ real matrix, $\mathbf{A}^T$ is the transpose of $\mathbf{A}$, $\|\cdot\|$ denotes the Euclidean norm, $\mathtt{Tr}(\mathbf{A})$ stands for the trace of $\mathbf{A}$ and $\|\mathbf{x}\|_{\mathbf{A}}^2$ denotes $\mathbf{x}^T\mathbf{A}\mathbf{x}$.

2.1. Random Vector Functional-Link

Given the training dataset $\mathcal{S} = \{(\mathbf{x_i}, y_i)\}_{i=1}^N$. For each sample $\mathbf{x_i} \in \mathbf{R}^d$, its output or label is known as $y_i \in \mathbf{R}$, $i = 1, ..., N$. The SLFNN is applied to approximate the mapping of the training dataset $\mathcal{S}$.

According to [35], an functional-link network (FLN) is a special kind of SLFNN. An FLN can be regarded as a weighted sum of L non-linear transformations of the training sample, which can be modeled as

$$f(\mathbf{x}) = \sum_{l=1}^{L} w_l h(\mathbf{x}; \mathbf{\Theta}_l) \triangleq \mathbf{w}^T \mathbf{h}(\mathbf{x}) \tag{1}$$

where the lth transformation $h(\mathbf{x}; \mathbf{\Theta}_l)$ is the base function h parametrized by a parameter set $\mathbf{\Theta}_l$, w_l is the weight coefficient of the lth hidden node, $\mathbf{h}(\mathbf{x}) = [h(\mathbf{x}; \mathbf{\Theta}_1), h(\mathbf{x}; \mathbf{\Theta}_2), ..., h(\mathbf{x}; \mathbf{\Theta}_L)]^T$ and $\mathbf{w} = [w_1, w_2, ..., w_L]^T$.

As for the RVFL network, each parameter set $\mathbf{\Theta}_i$ of the ith hidden neuron is randomly chosen from a probability distribution and then fixed [36, 37]. Thus, the task of RVFL network is to minimize the loss of the learning errors, which is formulated as

$$\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathbf{R}^L} \frac{1}{2}\left\|\mathbf{y} - \mathbf{H}\mathbf{w}\right\|^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2, \tag{2}$$

where λ is a positive parameter and $\mathbf{H} = [\mathbf{h}(\mathbf{x_1}), \mathbf{h}(\mathbf{x_2}), ..., \mathbf{h}(\mathbf{x_N})]^T$ is an $N \times L$ hidden matrix.

In this paper, we use the sigmoid function as the base function. Given the parameters $\mathbf{a}$ and b, the expression of the sigmoid function is given as

$$h(\mathbf{x}; \mathbf{a}, b) = \frac{1}{1 + e^{-(\mathbf{a}^T\mathbf{x} + b)}}, \tag{3}$$

It is obvious to get the solution of the optimization problem (2) as the following expression

$$\mathbf{w}^* = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}_L\right)^{-1}\mathbf{H}^T\mathbf{y}. \tag{4}$$

When $N < L$, the solution of the optimization problem (2) can be rewritten as

$$\mathbf{w}^* = \mathbf{H}^T\left(\mathbf{H}\mathbf{H}^T + \lambda\mathbf{I}_N\right)^{-1}\mathbf{y}. \tag{5}$$

2.2. Semi-Supervised Learning

As extensions of traditional SL techniques, SSL approaches succeed in making use of the unlabeled data. In the SSL problem, the dataset $\mathcal{S} = \mathcal{S}^{\text{l}} \bigcup \mathcal{S}^{\text{u}}$ contains two subsets, namely a labeled dataset $\mathcal{S}^{\text{l}} = \{(x_i^{\text{l}}, y_i^{\text{l}})\}_{i=1}^{N^{\text{l}}}$ and an unlabeled dataset $\mathcal{S}^{\text{u}} = \{x_j^{\text{u}}\}_{j=1}^{N^{\text{u}}}$. Obviously, $N^{\text{l}} + N^{\text{u}} = N$. Thus, the task of SSL is to learn mappings from labeled and unlabeled training samples.

In this part, we will introduce a general MR based SSL framework derived from [38] and [4]. In the MR framework, samples in $\mathcal{S}$ are assumed to satisfy the following two assumptions.

Assumption 1 ([2]). *Both the labeled data $\mathbf{x_l} \in \mathcal{S}^{\text{l}}$ and the unlabeled data $\mathbf{x_u} \in \mathcal{S}^{\text{u}}$ are drawn from the same marginal distribution $\mathcal{P}_X$.*

Assumption 2 ([2]). *If two input samples $\mathbf{x_1}$, $\mathbf{x_2} \in \mathcal{S}$ in a high-density region are close, then so should be the corresponding conditional probabilities $\mathcal{P}(y|\mathbf{x_1}), \mathcal{P}(y|\mathbf{x_2})$.*

Under Assumptions 1 and 2, unlabeled samples carry information owing to the labeled samples which are similar to them. The main idea of the MR framework is to add a manifold regularization item, which is described as

$$r_{MR} = \frac{1}{2}\sum\nolimits_{i,j=1}^{N} W_{ij}\left\|\mathcal{P}(y|\mathbf{x_i}) - \mathcal{P}(y|\mathbf{x_j})\right\|^2, \tag{6}$$

where $\mathbf{W} = [W_{ij}]_{N \times N}$ is a weight matrix to measure similarities among each pair of samples in $\mathcal{S}$. According to [4], the definition of $\mathbf{W}$ can be given as

$$\mathbf{W}_{ij} = e^{-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{2\sigma^2}}, \tag{7}$$

where $\sigma > 0$ is a tunable parameter.

In fact, it is difficult to calculate the conditional probability $\mathcal{P}(y|\mathbf{x_i})$ and $\mathcal{P}(y|\mathbf{x_j})$. Therefore, the manifold regularization item r_{MR} is generally rewritten as

$$r_{MR} = \frac{1}{2}\sum\nolimits_{i,j}^{N} W_{ij}\left\|\hat{y}_i - \hat{y}_j\right\|^2, \tag{8}$$

where $\hat{y}_i = \mathbf{w}^T\mathbf{h}(\mathbf{x_i})$ and $\hat{y}_j = \mathbf{w}^T\mathbf{h}(\mathbf{x_j})$ are the predictions of the outputs of the samples $\mathbf{x_i}$ and $\mathbf{x_j}$, respectively.

Defining $\hat{\mathbf{y}} = [\hat{y}_1, ..., \hat{y}_N]^T$ and $\mathbf{L} = \mathbf{D} - \mathbf{W}$, r_{MR} can be rewritten in a matrix form as

$$r_{MR} = \mathtt{Tr}(\hat{\mathbf{y}}^T\mathbf{L}\hat{\mathbf{y}}), \tag{9}$$

where $\mathbf{L}$ is known as the Laplacian matrix and $\mathbf{D}$ is a diagonal matrix with the ith element $\mathbf{D}[ii] = \sum_j W_{ij}$.

According to [2], we can replace $\mathbf{L}$ with a normalized matrix, which is defined as $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

By modifying the formulation of RVFL (2), the formulation of SSL-RVFL is defined as follows

$$\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathbf{R}^L} \frac{1}{2}\left\|\mathbf{H}\mathbf{w} - \tilde{\mathbf{y}}\right\|_{\mathbf{C}}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{\eta}{2}\mathtt{Tr}\left(\hat{\mathbf{y}}^T\tilde{\mathbf{L}}\hat{\mathbf{y}}\right), \tag{10}$$

where λ and η are the positive parameters, $\tilde{\mathbf{y}}$ is the augmented output of the training data with the rows corresponding to the labeled samples equal to $\mathbf{y}$ and the rest equals to 0, $\mathbf{C} = [C_{ij}]$ is an $N \times N$ diagonal binary matrix with the element $C_{ii} = 1$ when the ith training sample is labeled and the rest equals to 0.

By calculating the gradient of (10) and set it to 0, we can easily get the corresponding solution as follows

$$\mathbf{w}^* = \left(\mathbf{H}^T\mathbf{C}\mathbf{H} + \eta\mathbf{H}^T\tilde{\mathbf{L}}\mathbf{H} + \lambda\mathbf{I}_L\right)^{-1}\mathbf{H}^T\mathbf{C}\tilde{\mathbf{y}}. \tag{11}$$

According to [4], this algorithm can be denoted as the Lap-RVFL algorithm. When $N < L$, the solution (11) can be equivalently rewritten as

$$\mathbf{w}^* = \mathbf{H}^T\left(\mathbf{C}\mathbf{H}\mathbf{H}^T + \eta\tilde{\mathbf{L}}\mathbf{H}\mathbf{H}^T + \lambda\mathbf{I}_N\right)^{-1}\mathbf{C}\tilde{\mathbf{y}}. \tag{12}$$

2.3. Alternating Direction Method of Multipliers

The ADMM strategy proposed in [19] is an optimization procedure, which breaks the optimization problem into smaller pieces which are easier to handle. It focuses on solving optimization problems with variables $\mathbf{x} \in \mathbf{R}^{D_1}$ and $\mathbf{z} \in \mathbf{R}^{D_2}$, shown as

$$\begin{cases} \min & f(\mathbf{x}) + g(\mathbf{z}), \\ s.t. & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} + \mathbf{c} = \mathbf{0}, \end{cases} \tag{13}$$

where f and g are convex functions, $\mathbf{A} \in \mathbf{R}^{p\times D_1}$, $\mathbf{B} \in \mathbf{R}^{p\times D_2}$, and $\mathbf{c} \in \mathbf{R}^p$.

To solve the optimization problem (13), we rewrite the expression as the following augmented Lagrangian form

$$\mathcal{L}_\rho = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{r}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} + \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{s} + \mathbf{B}\mathbf{z} + \mathbf{c}\|^2, \tag{14}$$

where $\rho > 0$ is a tunable parameter and $\mathbf{r} \in \mathbf{R}^p$ is the Lagrange multipliers vector.

In order to solve the problem (13), the iteration procedure of the ADMM strategy is given by

$$\begin{cases} \mathbf{x}(k+1) = \arg\min\limits_{\mathbf{x}}\{\mathcal{L}_\rho(\mathbf{x}(k), \mathbf{z}(k), \mathbf{r}(k))\}, \\ \mathbf{z}(k+1) = \arg\min\limits_{\mathbf{z}}\{\mathcal{L}_\rho(\mathbf{x}(k+1), \mathbf{z}(k), \mathbf{r}(k))\}, \\ \mathbf{r}(k+1) = \mathbf{r}(k) + \rho(\mathbf{A}\mathbf{x}(k+1) + \mathbf{B}\mathbf{z}(k+1) + \mathbf{c}). \end{cases} \tag{15}$$

2.4. Distributed Average Consensus

The DAC strategy is designed to compute the global average of the local vectors over the communication network [31]. The kth iteration of node i is shown as

$$\mathbf{x}_i(k) = \sum\nolimits_{j=1}^{V} \mathbf{D}_{ij}\mathbf{x}_j(k-1). \tag{16}$$

where $\mathbf{D}$ the connectivity matrix.

If the communication network is connected and $\mathbf{D}$ is appropriately chosen, the iterative sequence with expression (16) converges to the global average.

According to [31], the connectivity matrix $\mathbf{D}$ can be defined as follows

$$\mathbf{D}_{ij} = \begin{cases} \frac{1}{d+1}, & j \in \mathcal{N}_i, \\ 1 - \frac{d_j}{d+1}, & i = j, \\ 0, & otherwise, \end{cases} \tag{17}$$

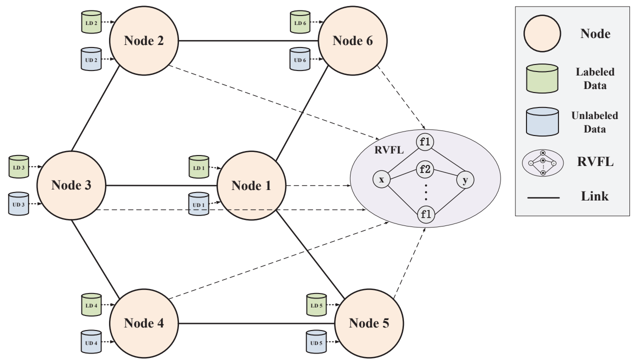


Figure 1: An illustration of DSSL problems. Training data consisting of labeled and unlabeled samples are separately distributed on each node over the communication network. Moreover, each node is individually assigned an RVFL network to learn mapping from training data. Thus, the task of DSSL algorithms are to learn the global coefficients of the RVFL network.

where $\mathcal{N}_i$ stands for the index set of node i's neighboring nodes, d_j denotes the degree of node j in the graph $\mathcal{G}$ corresponding to the communication network and d represents the maximum degree of all nodes.

3. Distributed Semi-Supervised Learning Algorithms

In DSSL problems, training data consisting of labeled and unlabeled samples are often extremely large-scale or high-dimension and separately stored over the communication network. For an communication network including V nodes, it can be modeled as a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$, in which $\mathcal{V} = \{v_1, v_2, ..., v_V\}$ is a finite nonempty node set.

In this section, we aim to solve DSSL problems illustrated in Fig.1. In this section, two DSSL frameworks, based on the ADMM strategy and RVFL networks, are proposed to solve DSSL problems with distributed data splitting across samples and features in Subsection 3.1 and 3.2, separately.

3.1. Horizontally Distributed Semi-Supervised Learning

In this part, we firstly consider traditional DSSL problems with distributed data splitting across samples. It means that a part of training samples are stored on each node over the communication network. Then, we propose the corresponding framework to solve DSSL problems with horizontally split data.

3.1.1. Problem Formulation

As shown in Fig.1, the goal of horizontally DSSL (HDSSL) is to learn a global mapping using the RVFL network and horizontally partioned data on each node over the communication network.

In HDSSL problems, training samples are horizontally distributed on V nodes over the communication network. The dataset used in the HDSSL problem can be described as

$$\mathcal{S} = \bigcup\nolimits_{i\in\mathcal{V}} \mathcal{S}_i^{=},$$

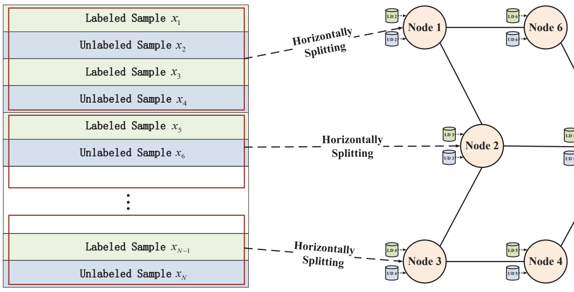


Figure 2: An illustration of distributed samples or horizontally partitioned data. Training samples are horizontally split across samples and separately distributed on each node over the communication network.

where $\mathcal{S}_i^{=} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{N_i}$ denotes the subset stored on node i, N_i is the number of training samples in node i, $\sum_{i\in\mathcal{V}} N_i = N$ and N denotes the total number of training samples over the communication network. An illustration of the training data of HDSSL problems is shown in Fig.2.

The task of HDSSL is to compute the global coefficients of the RVFL network of each node over the communication network by minimize the global cost function. Thus, the HDSSL problem can formulated as the following expression,

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbf{R}^L} \frac{1}{2}\sum_{i=1}^{V} \left\|\mathbf{H}_i\mathbf{w} - \tilde{\mathbf{y}}_i\right\|_{\mathbf{C}_i}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{\eta}{2}\sum_{i=1}^{V} \mathtt{Tr}\left(\mathbf{w}^T\mathbf{H}_i^T\tilde{\mathbf{L}}_i\mathbf{H}_i\mathbf{w}\right), \tag{18}$$

where the definitions of $\tilde{\mathbf{y}}_i$, $\mathbf{H}_i$, $\mathbf{C}_i$, and $\tilde{\mathbf{L}}_i$ are similar to the centralized SSL problems.

3.1.2. Horizontally DSSL Algorithm

Since training samples are separately stored over the communication network and cannot be centrally processed, the optimization problem (18) cannot be solved using traditional SSL algorithms. Thus, we rewrite the problem (18) as the following form,

$$\begin{cases} \min & \frac{1}{2}\sum_{i=1}^{V}\left\|\hat{\mathbf{y}}_i - \tilde{\mathbf{y}}_i\right\|_{\mathbf{C}_i}^2 + \frac{\lambda}{2}\|\mathbf{z}\|^2 + \frac{\eta}{2}\sum_{i=1}^{V}\mathtt{Tr}\left(\hat{\mathbf{y}}_i^T\tilde{\mathbf{L}}_i\hat{\mathbf{y}}_i\right), \\ s.t. & \mathbf{w}_i - \mathbf{z} = 0, i = 1, ..., V, \end{cases} \tag{19}$$

where $\hat{\mathbf{y}}_i = \mathbf{H}_i\mathbf{w}_i$.

Obviously, the equivalent problem (19) is a specific instance of the optimization problem (13) which can be solved by using the ADMM strategy. Moreover, the corresponding augmented Lagrangian form of the problem (19) is given by

$$\mathcal{L}_\rho = \frac{1}{2}\sum_{i=1}^{V}\left\|\hat{\mathbf{y}}_i - \tilde{\mathbf{y}}_i\right\|_{\mathbf{C}_i}^2 + \frac{\lambda}{2}\|\mathbf{z}\|^2 + \frac{\eta}{2}\sum_{i=1}^{V}\mathtt{Tr}\left(\hat{\mathbf{y}}_i^T\tilde{\mathbf{L}}_i\hat{\mathbf{y}}_i\right) + \sum_{i=1}^{V}\mathbf{r}^T(\mathbf{w}_i - \mathbf{z}) + \frac{\rho}{2}\sum_{i=1}^{V}\|\mathbf{w}_i - \mathbf{z}\|^2, \tag{20}$$

where $\rho > 0$ is a tunable parameter and $\mathbf{r} \in \mathbf{R}^p$ is the Lagrange multipliers vector.

Algorithm 1 HDSSL

Choose the parameters λ, η, ρ, the accuracy parameter ϵ for early termination and the iteration number K. Initialize $\mathbf{z}(0)$ and $\mathbf{r}_i(0)$ to $\mathbf{0}$,
for $i \in \mathcal{V}$ **do**
  Calculate $\mathbf{H}_i$ and $\tilde{\mathbf{L}}_i$,
  $\mathbf{P}_i \leftarrow \mathbf{H}_i^T\mathbf{C}_i\mathbf{H}_i + \eta\mathbf{H}_i^T\tilde{\mathbf{L}}_i\mathbf{H}_i + \lambda\mathbf{I}_L$,
end
for $k \leftarrow 0$ **to** $K-1$ **do**
  for $i \in \mathcal{V}$ **do**
    $\mathbf{w}_i(k+1) \leftarrow \mathbf{P}_i^{-1}(\mathbf{H}_i^T\mathbf{C}_i\tilde{\mathbf{y}}_i - \mathbf{r}_i(k) + \rho\mathbf{z}(k))$,
  end
  $\bar{\mathbf{w}} \leftarrow \frac{1}{V}\sum_{i=1}^{V}\mathbf{w}_i(k+1)$ using the DAC strategy,
  $\bar{\mathbf{r}} \leftarrow \frac{1}{V}\sum_{i=1}^{V}\mathbf{r}_i(k)$ using the DAC strategy.
  $\mathbf{z}(k+1) \leftarrow \frac{V}{\lambda+V\rho}(\rho\bar{\mathbf{w}} + \bar{\mathbf{r}})$,
  for $i \in \mathcal{V}$ **do**
    $\mathbf{r}_i(k+1) \leftarrow \mathbf{r}_i(k) + \rho(\mathbf{w}_i(k+1) - \mathbf{z}(k+1))$,
  end
  if $\|\mathbf{w}(k) - \mathbf{z}(k)\| < \epsilon$ **then**
    break,
  end
end
return $\mathbf{z}(k)$.

According to the iterative format of (15), the problem of HDSSL is rewritten as follows

$$\begin{cases} \mathbf{w}_i(k+1) = \arg\min_{\mathbf{w}_i}\left\{\frac{1}{2}\left\|\hat{\mathbf{y}}_i - \tilde{\mathbf{y}}_i\right\|_{\mathbf{C}_i}^2 + \frac{\eta}{2}\mathtt{Tr}\left(\hat{\mathbf{y}}_i^T\tilde{\mathbf{L}}_i\hat{\mathbf{y}}_i\right) \right. \\ \qquad \left. + \mathbf{r}^T(\mathbf{w}_i - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{w}_i - \mathbf{z}\|^2\right\}, \\ \mathbf{z}(k+1) = \arg\min_{\mathbf{z}}\left\{\frac{\lambda}{2}\|\mathbf{z}\|^2 + \sum_{i=1}^{V}\mathbf{r}^T(\mathbf{w}_i(k+1) - \mathbf{z}) \right. \\ \qquad \left. + \frac{\rho}{2}\sum_{i=1}^{V}\|\mathbf{w}_i(k+1) - \mathbf{z}\|^2\right\}, \\ \mathbf{r}_i(k+1) = \mathbf{r}_i(k) + \rho(\mathbf{w}_i(k+1) + \mathbf{z}(k+1)). \end{cases} \tag{21}$$

Thus, the HDSSL algorithm is designed as the following expression.

$$\begin{cases} \mathbf{w}_i(k+1) = \mathbf{P}_i^{-1}(\mathbf{H}_i^T\mathbf{C}_i\tilde{\mathbf{y}}_i - \mathbf{r}_i(k) + \rho\mathbf{z}(k)), \\ \mathbf{z}(k+1) = \frac{V}{\lambda+V\rho}(\rho\bar{\mathbf{w}} + \bar{\mathbf{r}}), \\ \mathbf{r}_i(k+1) = \mathbf{r}_i(k) + \rho(\mathbf{w}_i(k+1) - \mathbf{z}(k+1)), \end{cases} \tag{22}$$

where $\mathbf{P}_i = \mathbf{H}_i^T\mathbf{C}_i\mathbf{H}_i + \eta\mathbf{H}_i^T\tilde{\mathbf{L}}_i\mathbf{H}_i + \lambda\mathbf{I}_L$, the initial value of $\mathbf{r}_i$ and $\mathbf{z}$ can be chosen as $\mathbf{0}$, $\bar{\mathbf{w}} = \frac{1}{V}\sum_{i=1}^{V}\mathbf{w}_i(k+1)$ and $\bar{\mathbf{r}} = \frac{1}{V}\sum_{i=1}^{V}\mathbf{r}_i(k)$ are globally average values over the communication network, which can be calculated using the DAC strategy.

Remark 1. *The ZGS strategy used in the D-LapWNN algorithm is proposed to solve unconstrained, separable and convex optimization problems, which can be described as* $\mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x}) = \arg\min_{\mathbf{x}}\sum_{i=1}^{V} f_i(\mathbf{x})$, *where* f_i *is a twice continuously differentiable and strongly convex function. These problems can be solved by using the ADMM strategy, either. Moreover, the ADMM strategy works even when* f_i *is not differentiable. For example, the ADMM strategy can solve the problem*

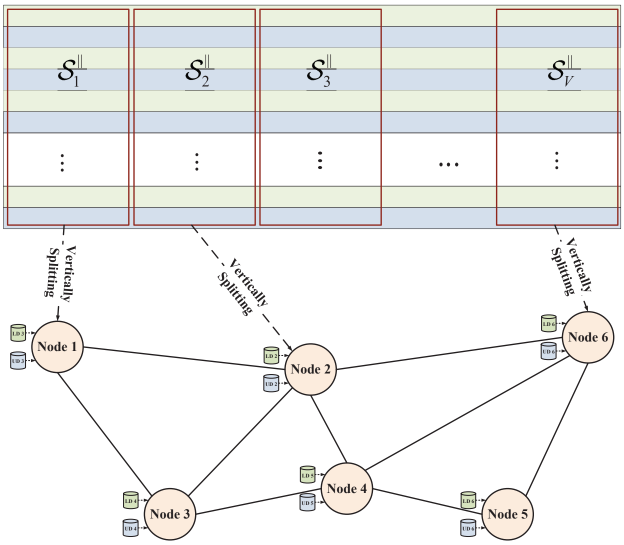


Figure 3: An illustration of distributed features or vertically partitioned data. Training data consisting of labeled and unlabeled samples are split across the features. Then, each part of features is stored on the corresponding node over the communication network.

(18) *when the L_2 norm used in the regular item is substituted to the L_1 norm in some scenarios, but the ZGS strategy become useless. Thus, the proposed HDSSL algorithm is more flexible than the D-LapWNN algorithm.*

The pseudo code of the HDSSL algorithm is shown in Algorithm 1.

3.2. Vertically Distributed Semi-Supervised Learning

In this part, we consider another kind of DSSL problems where training samples are high-dimension and vertically split across features. It means that partial features of the total training data are separately stored on each node over the communication. Based on the ADMM strategy and RVFL networks, we propose another framework to solve vertically DSSL (VDSSL) problems with vertically split data.

3.2.1. Problem Formulation

Similar to the definition of the distributed data used in HDSS-L problems, dataset in VDSSL problems can be described as

$$\mathcal{S} = \bigcup_{i\in\mathcal{V}} \mathcal{S}_i^{\|},$$

where $\mathcal{S}_i^{\|} = \{(\mathbf{x}_j^{d_i}, y_j)\}_{j=1}^N$, $\mathbf{x}_j = [\mathbf{x}_j^{d_1}|\mathbf{x}_j^{d_2}|...|\mathbf{x}_j^{d_V}]$, $d = \sum_i^V d_i$ and d denotes the dimension of each training sample. An illustration of the training data of VDSSL problems is shown in Fig.3.

Similarly, the VDSSL problem can formulated as the following expression

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbf{R}^V} \frac{1}{2}\Big\| \sum_{i=1}^V \mathbf{H}_i\mathbf{w}_i - \tilde{\mathbf{y}}\Big\|_{\mathbf{C}}^2 + \frac{\lambda}{2}\sum_{i=1}^V \|\mathbf{w}_i\|^2 + \frac{\eta}{2}\sum_{i=1}^V \mathtt{Tr}\Big(\mathbf{w}_i^T\mathbf{H}_i^T\tilde{\mathbf{L}}_i\mathbf{H}_i\mathbf{w}_i\Big). \tag{23}$$

Algorithm 2 VDSSL

Choose the parameters $\lambda, \eta, \rho, \epsilon, K$, and initialize $\mathbf{u}(0)$, $\bar{\mathbf{z}}(0)$, $\mathbf{w}(0)$, $\bar{\mathbf{y}}(0)$ to $\mathbf{0}$,
for $i \in \mathcal{V}$ **do**
  Calculate $\mathbf{H}_i$ and $\tilde{\mathbf{L}}_i$,
  $\mathbf{Q}_i \leftarrow \mathbf{H}_i^T\mathbf{H}_i + \frac{\eta}{\rho}\mathbf{H}_i^T\tilde{\mathbf{L}}_i\mathbf{H}_i + \frac{\lambda}{\rho}\mathbf{I}_L$,
end
for $k \leftarrow 0$ **to** $K-1$ **do**
  for $i \in \mathcal{V}$ **do**
    $\mathbf{w}_i(k+1) \leftarrow \mathbf{Q}_i^{-1}\mathbf{H}_i^T\Big(\mathbf{H}_i\mathbf{w}_i(k) + \bar{\mathbf{z}}(k) - \bar{\mathbf{y}}(k) - \mathbf{u}(k)\Big)$,
  end
  $\bar{\mathbf{y}}(k+1) \leftarrow \frac{1}{V}\sum_{i=1}^V \mathbf{H}_i\mathbf{w}_i(k+1)$ using the DAC strategy,
  $\bar{\mathbf{z}}(k+1) \leftarrow (V\mathbf{C} + \rho\mathbf{I}_N)^{-1}(\mathbf{C}\tilde{\mathbf{y}} + \rho\bar{\mathbf{y}}(k+1) + \rho\mathbf{u}(k))$,
  $\mathbf{u}(k+1) \leftarrow \mathbf{u}(k) + \bar{\mathbf{y}}(k+1) - \bar{\mathbf{z}}(k+1)$,
  if $\|\bar{\mathbf{z}}(k+1) - \bar{\mathbf{y}}(k+1)\| < \epsilon$ **then**
    break,
  end
end
return $\mathbf{w}(k+1)$.

3.2.2. Vertically DSSL Algorithm

Since training samples are separately stored over the communication network and cannot be centrally processed, the optimization problem (23) cannot be solved using traditional SSL algorithms. Thus, we rewrite the problem (23) as the following form,

$$\begin{cases} \min & \frac{1}{2}\Big\| \sum\limits_{i=1}^V \mathbf{z}_i - \tilde{\mathbf{y}}\Big\|_{\mathbf{C}}^2 + \frac{\lambda}{2}\sum\limits_{i=1}^V \|\mathbf{w}_i\|^2 + \frac{\eta}{2}\sum\limits_{i=1}^V \mathtt{Tr}\Big(\hat{\mathbf{y}}_i^T\tilde{\mathbf{L}}_i\hat{\mathbf{y}}_i\Big), \\ s.t. & \hat{\mathbf{y}}_i - \mathbf{z}_i = 0, i = 1, ..., V, \end{cases} \tag{24}$$

where $\hat{\mathbf{y}}_i = \mathbf{H}_i\mathbf{w}_i$.

Similarly, the corresponding augmented Lagrangian form of the problem (24) is

$$\mathcal{L}_\rho = \frac{1}{2}\Big\| \sum_{i=1}^V \mathbf{z}_i - \tilde{\mathbf{y}}\Big\|_{\mathbf{C}}^2 + \frac{\lambda}{2}\sum_{i=1}^V \|\mathbf{w}_i\|^2 + \frac{\eta}{2}\sum_{i=1}^V \mathtt{Tr}\Big(\hat{\mathbf{y}}_i^T\tilde{\mathbf{L}}_i\hat{\mathbf{y}}_i\Big) + \sum_{i=1}^V \mathbf{r}^T(\hat{\mathbf{y}}_i - \mathbf{z}_i) + \frac{\rho}{2}\sum_{i=1}^V \|\hat{\mathbf{y}}_i - \mathbf{z}_i\|^2. \tag{25}$$

By substituting $\mathbf{u} = \frac{1}{\rho}\mathbf{r}$, we have

$$\sum_{i=1}^V \mathbf{r}^T(\hat{\mathbf{y}}_i - \mathbf{z}_i) + \frac{\rho}{2}\sum_{i=1}^V \|\hat{\mathbf{y}}_i - \mathbf{z}_i\|^2 = \frac{\rho}{2}\sum_{i=1}^V \|\hat{\mathbf{y}}_i - \mathbf{z}_i + \mathbf{u}\|^2 - \frac{\rho}{2}\sum_{i=1}^V \|\mathbf{u}\|^2 \tag{26}$$

According to [19], the iterative expressions of this special ADMM problem can be rewritten as follows

$$\begin{cases} \mathbf{w}_i(k+1) = \arg\min\limits_{\mathbf{w}_i} \Big\{\frac{\lambda}{2}\|\mathbf{w}_i\|^2 + \frac{\eta}{2}\mathtt{Tr}\Big(\hat{\mathbf{y}}_i^T\tilde{\mathbf{L}}_i\hat{\mathbf{y}}_i\Big) \\ \qquad + \frac{\rho}{2}\Big\|\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_i(k) - \bar{\mathbf{z}}(k) + \bar{\mathbf{y}}(k) + \mathbf{u}(k)\Big\|^2\Big\}, \\ \bar{\mathbf{z}}(k+1) = \arg\min\limits_{\bar{\mathbf{z}}} \Big\{\frac{V\rho}{2}\Big\|\bar{\mathbf{z}} - \bar{\mathbf{y}}(k+1) - \mathbf{u}(k)\Big\|^2 \\ \qquad + \frac{1}{2}\Big\|V\bar{\mathbf{z}} - \tilde{\mathbf{y}}\Big\|_{\mathbf{C}}^2\Big\}, \\ \mathbf{u}(k+1) = \mathbf{u}(k) + \bar{\mathbf{y}}(k+1) - \bar{\mathbf{z}}(k+1), \end{cases} \tag{27}$$

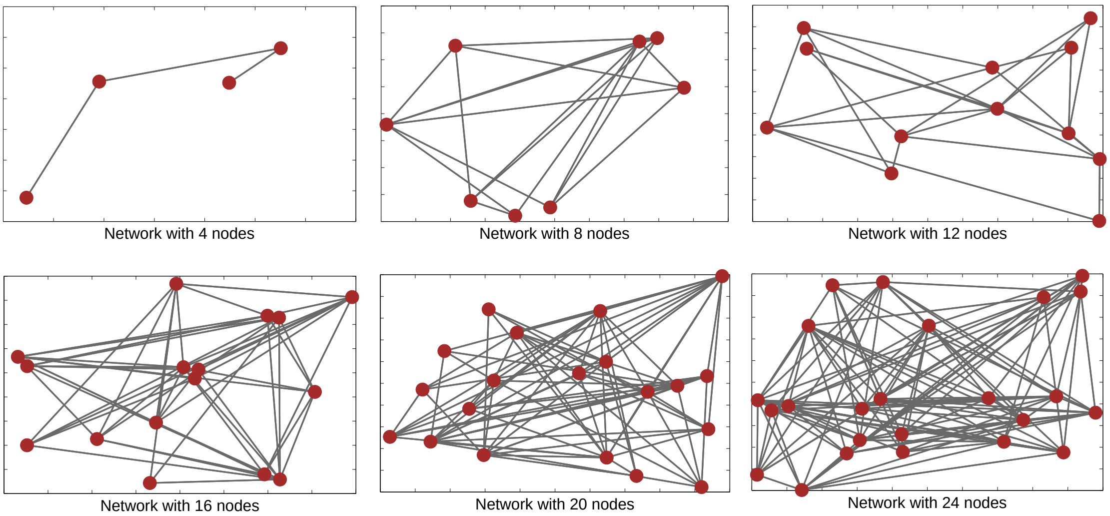


Figure 4: *Randomly generated communication networks with different number of nodes.*

Table 1: Description of the datasets used in the D-LapWNN and HDSSL algorithms. These data will be horizontally partitioned or splitting across samples.

Dataset	Instance	Labeled	Features	Task	Source
SinC	4000	400	1	Regression	Artificial
2-Moon	800	32	2	Classification	Artificial [1]
Concrete	1027	410	8	Regression	UCI Repository
WDBC	569	56	30	Classification	UCI Repository

Table 2: Parameters used in the simulations. The parameter γ is used in the D-LapWNN algorithm. And the parameter ρ is used in the HDSSL algorithm.

Dataset	λ	η	γ	ρ	L
SinC	10^{-2}	10^{-4}	10^{-3}	10^{-3}	50
2-Moon	10^{-3}	10^{-5}	10^{-3}	10^{-3}	50
Concrete	10^{-4}	10^{-5}	10^{-4}	10^{-4}	50
WDBC	10^{-3}	10^{-4}	10^{-3}	10^{-3}	50

where $\hat{\mathbf{y}}_i = \mathbf{H}_i\mathbf{w}_i$, $\hat{\mathbf{y}}_i(k) = \mathbf{H}_i\mathbf{w}_i(k)$, $\bar{\mathbf{z}}(0) = \frac{1}{V}\sum_{i=1}^{V}\mathbf{z}_i(0)$ and $\bar{\mathbf{y}}(k) = \frac{1}{V}\sum_{i=1}^{V}\hat{\mathbf{y}}_i(k)$.

Thus, the corresponding VDSSL algorithm is designed as the follows

$$\begin{cases}\mathbf{w}_i(k+1) = \mathbf{Q}_i^{-1}\mathbf{H}_i^T\Big(\hat{\mathbf{y}}_i(k) + \bar{\mathbf{z}}(k) - \bar{\mathbf{y}}(k) - \mathbf{u}(k)\Big),\\ \bar{\mathbf{z}}(k+1) = \Big(V\mathbf{C} + \rho\mathbf{I}_N\Big)^{-1}\Big(\mathbf{C}\tilde{\mathbf{y}} + \rho\bar{\mathbf{y}}(k+1) + \rho\mathbf{u}(k)\Big),\\ \mathbf{u}(k+1) = \mathbf{u}(k) + \bar{\mathbf{y}}(k+1) - \bar{\mathbf{z}}(k+1),\end{cases} \tag{28}$$

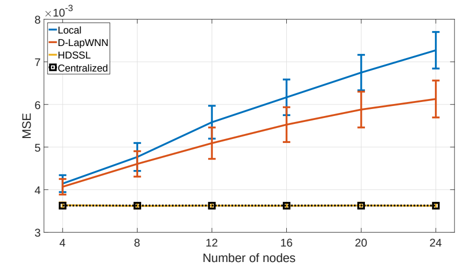
where $\mathbf{Q}_i = \mathbf{H}_i^T\mathbf{H}_i + \frac{\eta}{\rho}\mathbf{H}_i^T\tilde{\mathbf{L}}_i\mathbf{H}_i + \frac{\lambda}{\rho}\mathbf{I}_L$, the initial value of $\mathbf{w}_i$, $\mathbf{u}$ and $\bar{\mathbf{z}}$ can be chosen as $\mathbf{0}$.

Remark 2. *The VDSSL algorithm is novelly proposed to solve DSSL problems with vertically partitioned data or distributed features. Data in these problems are often high-dimension. In some scenarios, distributed features are separately stored over communication networks and can not be collected due to privacy protection or environment limitation. According to the VDSSL algorithm described in* (28)*, nodes over the communication network exchange updated coefficients rather than raw data during the learning process. It means that the VDSSL algorithm is a privacy-preserving method. In the consensus step, the* $\bar{\mathbf{y}}(k) = \frac{1}{V}\sum_{i=1}^{V}\mathbf{H}_i\mathbf{w}_i(k)$ *can be calculated by using the DAC strategy.*

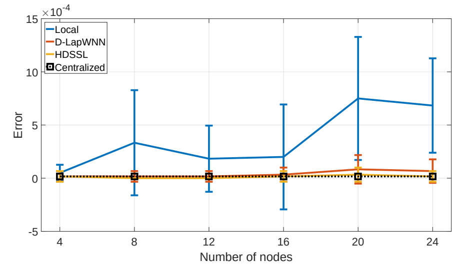
The pseudo code of the VDSSL algorithm is shown in Algorithm 2.

4. Simulations

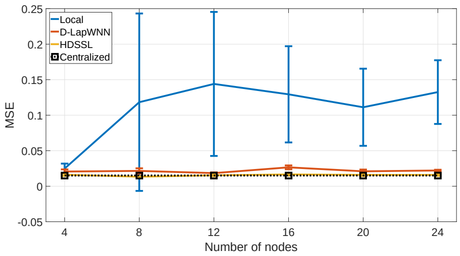
In order to verify the efficiency of the proposed algorithms, some simulations are shown in this section. In Part 1, we compare the proposed HDSSL algorithm with the D-LapWNN



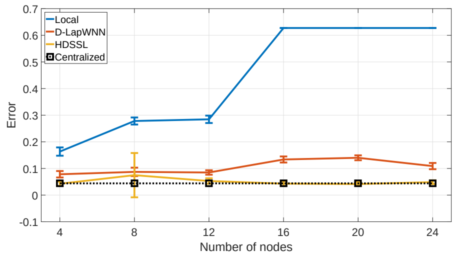
(a) SinC Dataset.



(b) 2-Moon Dataset.

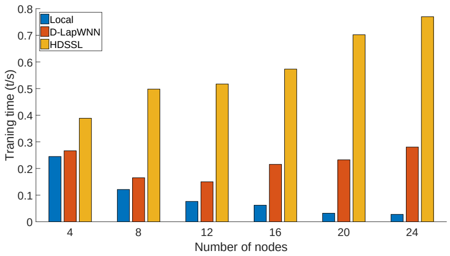


(c) Concrete Dataset.

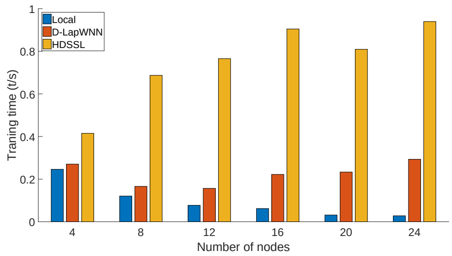


(d) WDBC Dataset.

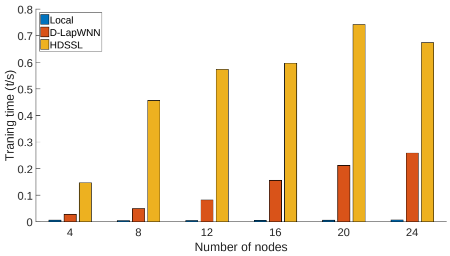
Figure 5: Training error of the aforementioned algorithms using different datasets listed in Table 1 over the communication networks shown in 4.



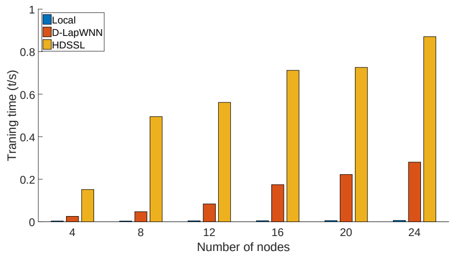
(a) SinC Dataset.



(b) 2-Moon Dataset.



(c) Concrete Dataset.



(d) WDBC Dataset.

Figure 6: Average training time of the HDSSL, local Lap-RVFL and centralized RVFL algorithms using distributed samples over the communication networks shown in 4.

algorithm and centralized Lap-RVFL network with total data. We choose the Gaussian wavelet function as the mother wavelet function in the D-LapWNN algorithm. Then, some simulations of the centralized Lap-RVFL and VDSSL algorithms with distributed features are provided in Part 2.

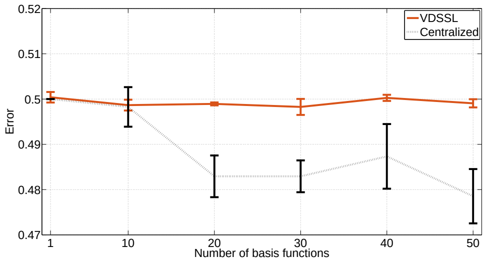
For simplicity, samples or features of each dataset are averagely allocated over the communication network. To indicate the accuracy of the algorithms, the mean squared error (MSE) is introduced in regression tasks. For classification tasks, we use the percentage of misclassification.

4.1. Horizontally DSSL Algorithm

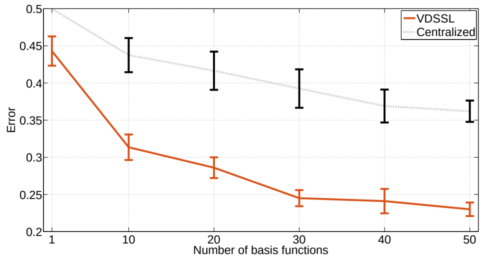
In this part, some simulations are given to verify the efficiency of the proposed HDSSL algorithm. The HDSSL algorithm

Table 3: Description of the datasets used in the VDSSL algorithm. These data will be vertically partitioned or split across features over the communication network.

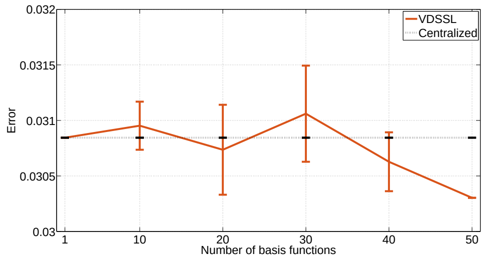
Dataset	Instance	Labeled	Features	Task	Source
g241c	1500	150	241	Classification	Artificial [2]
BCI	400	40	117	Classification	Artificial [2]
Ozone	1848	180	72	Classification	UCI Repository
WDBC	569	56	30	Classification	UCI Repository



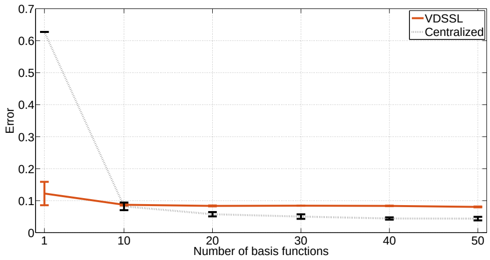
(a) g241c



(b) BCI

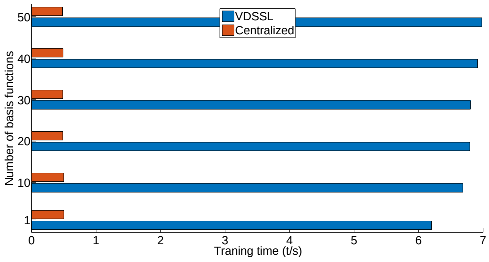


(c) Ozone

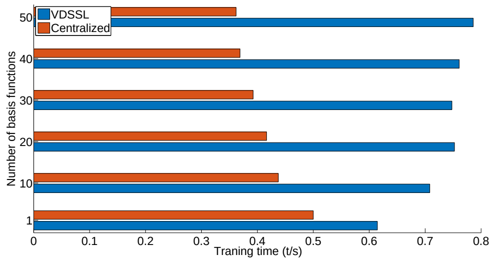


(d) WDBC

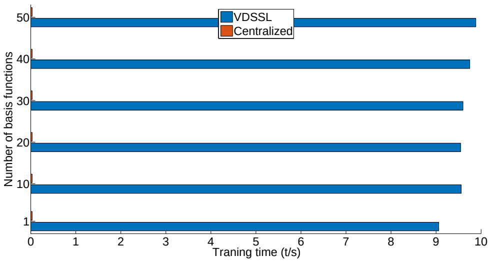
Figure 7: The influence of the number of basis functions on the performance of the VDSSL algorithm using different dataset.



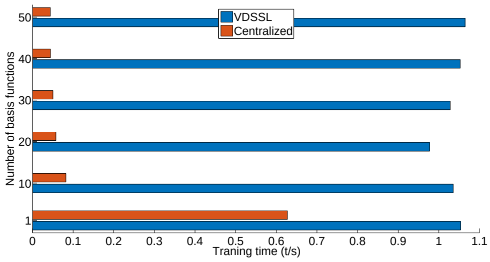
(a) g241c



(b) BCI



(c) Ozone



(d) WDBC

Figure 8: The influence of the number of basis functions on the training time of the VDSSL algorithm using different dataset.

is used for regression and classification using the datasets described in Table 1. These data are horizontally partitioned and averagely distributed over the communication networks with different number of nodes, which are shown in Fig.4. To make the results more convincing, we compare the proposed HDSSL algorithm with the previously proposed D-LapWNN, local Lap-RVFL, and centralized Lap-RVFL algorithms.

The parameters used in this part are listed in Table 2. In order to show the influence of varying the number of nodes in the communication networks, we record the simulation results of the D-LapRLS and proposed algorithms in different communication network with different number of nodes shown in Fig.4. We vary the number of nodes from 4 to 24 and repeat each simulation for 10 times. Simulation results are shown in Fig.5 and Fig.6.

Simulation results show that the proposed HDSSL algorithm is efficient enough in solving DSSL problems with regression and classification tasks. Compared with the D-LapWNN algorithm, the HDSSL algorithm achieves the higher accuracy but costs a few more time. This is caused by the iterative processes owing to the DAC strategy. Moreover, the accuracy of the HDSSL does not decrease with the increase of the number of nodes in the communication network.

4.2. *Case 2: Vertically DSSL Algorithm*

In this part, the proposed VDSSL algorithm is applied to classify the data listed in Table 3. These data are vertically partitioned or split across features and averagely distributed over the communication network with 8 nodes shown in Fig.4. In addition, we compare the proposed VDSSL algorithm with the centralized Lap-RVFL algorithm with the total features. In order to analyze the influence of the number of basis functions used in the RVFL network on the performance of the VDSSL algorithm, we apply the VDSSL algorithm to the “BCI” dataset and vary the number of basis functions from 1 to 51. Then, we repeat each simulation for 10 times and show the results in Fig.7 and 8. For simplicity, all data sets use the same parameters, where $\lambda = 10^{-2}$, $\eta = 10^{-4}$, $\rho = 10^{-3}$ and $L = 50$.

The results show that the proposed VDSSL is efficient in solving DSSL problems with vertically partitioned data. For the “Ozone” and “BCI” datasets, the VDSSL algorithm performs better than the centralized Lap-RVFL algorithm when the number of basis functions is increased. As for the “g241c” dataset, neither the VDSSL algorithm nor the centralized Lap-RVFL algorithm performs very well. It means that the performance of the proposed VDSSL algorithm depends on the SSL algorithm used in the VDSSL algorithm. Moreover, we applied the VDSSL algorithm on the “WDBC” dataset to prove the efficiency of the proposed VDSSL algorithm on the dataset with less features. But the results show that the accuracy of the VDSSL algorithm does not change when the number of basis functions is changed.

Besides, it can be seen that the VDSSL and centralized Lap-RVFL algorithms performs better when the number of basis functions increase. However, the change of the number of basis functions does not affect the training time.

5. Conclusions

In this paper, we extend the previously proposed D-LapWNN algorithm and propose the HDSSL algorithms to solve DSSL problems with horizontally partitioned data. Then, we novelly propose the VDSSL algorithm to sovle DSSL problems with vertically partitioned data. These two algorithms are based on the RVFL network and the ADMM strategy. Owing to the ADMM strategy, the proposed algorithms only exchange coefficients during learning processes, which means they are privacy-preserving methods.

The simulation results show that the proposed algorithms are efficient enough in DSSL problems with large-scale or high-dimension data, which is distributed over the communication network. The HDSSL and VDSSL algorithms are fully distributed algorithms. Compared with the previously proposed D-LapWNN algorithm with distributed samples, the HDSSL algorithm achieves the higher accuracy. As for distributed features, the VDSSL is efficient and gets the similar results to the centralized Lap-RVFL network using total data over the communication network.

In the end, it is worth providing some future works. the proposed HDSSL and VDSSL algorithms can be developed into the case of directed and the time-varying communication networks, which are more practical in applications. Besides, it worth to extend the proposed algorithms to finite-time algorithms in order to meet engineering requirements.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61503292, 61877046, 61673014 and 61703117), the Natural Science Foundation of Shaanxi Province (Grant Nos. 2017JM1001 and 2018JM6079), the Natural Science Foundation of Guangxi (No. 2017GXNSFBA198113), the Fundamental Research Funds for the Central Universities and the Innovation Fund of Xidian University.

References

[1] M. Belkin, P. Niyogi, V. Sindhwani, Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples, JMLR.org, 2006.

[2] O. Chapelle, B. Schölkopf, A. Zien, Semi-Supervised Learning, MIT Press, 2006.

[3] J. Wang, X. Shen, On transductive support vector machines, Prediction & Discovery 27 (6) (2007) 1463–1462.

[4] S. Scardapane, D. Comminiello, M. Scarpiniti, A. Uncini, A semi-supervised random vector functional-link network based on the transductive framework, Information Sciences 364 (C) (2015) 156–166.

[5] T. Miyato, S.-i. Maeda, S. Ishii, M. Koyama, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, IEEE transactions on pattern analysis and machine intelligence.

[6] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, Y.-L. He, Fuzziness based semi-supervised learning approach for intrusion detection system, Information Sciences 378 (2017) 484–497.

[7] C. L. Liu, W. H. Hsaio, C. H. Lee, T. H. Chang, T. H. Kuo, Semi-supervised text classification with universum learning., IEEE Transactions on Cybernetics 46 (2) (2016) 462–473.

[8] X. Lu, X. Li, L. Mou, Semi-supervised multitask learning for scene recognition, IEEE Transactions on Cybernetics 45 (9) (2017) 1967–1976.

[9] J. M. Liu, J. Tian, Z. X. Cai, Y. Zhou, R. H. Luo, R. R. Wang, A hybrid semi-supervised approach for financial fraud detection, in: International Conference on Machine Learning and Cybernetics, 2017, pp. 217–222.

[10] P. Liu, P. Yang, C. Wang, K. Huang, T. Tan, A semi-supervised method for surveillance-based visual location recognition, IEEE Transactions on Cybernetics PP (99) (2016) 1–14.

[11] M. Luo, X. Chang, L. Nie, Y. Yang, A. G. Hauptmann, Q. Zheng, An adaptive semisupervised feature analysis for video semantic recognition, IEEE Transactions on Cybernetics 48 (2) (2017) 648–660.

[12] J. Zhang, Y. Han, J. Tang, Q. Hu, J. Jiang, Semi-supervised image-to-video adaptation for video action recognition., IEEE Transactions on Cybernetics 47 (4) (2016) 960–973.

[13] Y. Grandvalet, Y. Bengio, Semi-supervised learning by entropy minimization, in: Advances in neural information processing systems, 2005, pp. 529–536.

[14] P. Y. Simard, D. Steinkraus, J. C. Platt, et al., Best practices for convolutional neural networks applied to visual document analysis., in: Icdar, Vol. 3, 2003.

[15] G. Hinton, D. Van Camp, Keeping neural networks simple by minimizing the description length of the weights, in: in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory, Citeseer, 1993.

[16] J. B. Predd, S. R. Kulkarni, P. H. Vincent, Distributed learning in wireless sensor networks, Signal Processing Magazine IEEE 23 (4) (2005) 56–69.

[17] F. Yan, S. Sundaram, S. V. N. Vishwanathan, Y. Qi, Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties, IEEE Transactions on Knowledge & Data Engineering 25 (11) (2013) 2483–2493.

[18] S. Scardapane, D. Wang, M. Panella, A. Uncini, Distributed learning for random vector functional-link networks, Information Sciences 301 (2015) 271–284.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations & Trends in Machine Learning 3 (1) (2011) 1–122.

[20] W. Ai, W. Chen, J. Xie, A zero-gradient-sum algorithm for distributed cooperative learning using a feedforward neural network with random weights, Information Sciences 373 (2016) 404–418.

[21] J. Lu, C. Y. Tang, Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case, IEEE Transactions on Automatic Control 57 (9) (2012) 2348–2354.

[22] W. Ai, W. Chen, J. Xie, Distributed learning for feedforward neural networks with random weights using an event-triggered communication scheme, Neurocomputing 224 (2017) 184–194.

[23] J. Xie, W. Chen, H. Dai, Distributed cooperative learning algorithms using wavelet neural network, Neural Computing & Applicationsdoi:https://doi.org/10.1007/s00521-017-3134-1.

[24] A. H. Sayed, Adaptive networks, Proceedings of the IEEE 102 (4) (2014) 460–497.

[25] K. Avrachenkov, V. S. Borkar, K. Saboo, Distributed and Asynchronous Methods for Semi-supervised Learning, Springer International Publishing, 2016.

[26] X. Chang, S. B. Lin, D. X. Zhou, Distributed semi-supervised learning with kernel ridge regression, Journal of Machine Learning Research 18 (2017) 1–22.

[27] P. Shen, X. Du, C. Li, Distributed semi-supervised metric learning, IEEE Access 4 (99) (2017) 8558–8571.

[28] S. Scardapane, R. Fierimonte, L. P. Di, M. Panella, A. Uncini, Distributed semi-supervised support vector machines., Neural Networks 80 (2016) 43–52.

[29] P. D. Lorenzo, G. Scutari, Next: In-network nonconvex optimization, IEEE Transactions on Signal & Information Processing Over Networks 2 (2) (2016) 120–136.

[30] R. Fierimonte, S. Scardapane, A. Uncini, M. Panella, Fully decentralized semi-supervised learning via privacy-preserving matrix completion., IEEE Transactions on Neural Networks & Learning Systems 28 (11) (2017) 2699–2711.

[31] R. Olfati-Saber, J. A. Fax, R. M. Murray, Consensus and cooperation in networked multi-agent systems, Proceedings of the IEEE 95 (1) (2007) 215–233.

[32] J. Xie, S. Liu, H. Dai, A distributed semi-supervised learning algorithm based on manifold regularization using wavelet neural network, Neural Networksdoi:https://doi.org/10.1016/j.neunet.2018.10.014.

[33] J. Vaidya, Privacy preserving association rule mining in vertically partitioned data, Journal of Computer Applications 39 (13) (2006) 29–35.

[34] S. Scardapane, M. Panella, D. Comminiello, A. Uncini, Learning from distributed data sources using random vector functional-link networks, Procedia Computer Science 53 (1) (2015) 468–477.

[35] Y. H. Pao, G. H. Park, D. J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (2) (1994) 163–180.

[36] M. Alhamdoosh, D. Wang, Fast decorrelated neural network ensembles with random weights, Information Sciences 264 (6) (2014) 104–117.

[37] B. Igelnik, Y. H. Pao, Stochastic choice of basis functions in adaptive function approximation and the functional-link net, IEEE Transactions on Neural Network 6 (6) (1995) 1320–1329.

[38] G. Huang, S. Song, J. N. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Transactions on Cybernetics 44 (12) (2017) 2405–2417.