Kybernetika

Yinghui Wang; Peng Lin; Huashu Qin

Distributed classification learning based on nonlinear vector support machines for switching networks

Kybernetika, Vol. 53 (2017), No. 4, 595–611

Persistent URL: `http://dml.cz/dmlcz/146946`

KYBERNETIKA — VOLUME *53* (2017), NUMBER 4, PAGES 595–611

DISTRIBUTED CLASSIFICATION LEARNING BASED ON NONLINEAR VECTOR SUPPORT MACHINES FOR SWITCHING NETWORKS

Yinghui Wang, Peng Lin and Huashu Qin

In this paper, we discuss the distributed design for binary classification based on the nonlinear support vector machine in a time-varying multi-agent network when the training data sets are distributedly located and unavailable to all agents. In particular, the aim is to find a global large margin classifier and then enable each agent to classify any new input data into one of the two labels in the binary classification without sharing its all local data with other agents. We formulate the support vector machine problem into a distributed optimization problem in approximation and employ a distributed algorithm in a time-varying network to solve it. Our algorithm is a stochastic one with the high convergence rate and the low communication cost. With the jointly-connected connectivity condition, we analyze the consensus rate and the convergence rate of the given algorithm. Then some experimental results on various classification training data sets are also provided to illustrate the effectiveness of the given algorithm.

Keywords: nonlinear support vector machine, multi-agent system, distributed optimization, connectivity

Classification: 68M15, 93A14

1. INTRODUCTION

In recent years, classification problems have attracted more and more research attention in machine learning, pattern recognition, and data mining [2, 9]. The binary classification setting is of great importance to classification problems because any multi-class classification problem can be transformed into binary classification problems [23]. Support vector machine, intended to both maximize the margin and minimize the training error over labeled training sets, is one of the most famous binary classification settings. Over the past two decades, researchers have made great efforts to derive, implement and analyze the effectiveness of different SVM solutions. These efforts include the decomposition training based on dual SVMs [4, 21], the cutting-plane training based on primal SVM with violated constraints [14, 26] and the sub-gradient training based on the primal SVMs [15, 25].

Distributed SVM training becomes necessary due to the rapid growth of data and the infeasibility of a centralized system for the reason of geographical factors and limited

DOI: 10.14736/kyb-2017-4-0595

computational capabilities. The parallel design of centralized SVM training is the simplest distributed SVM training. In the parallel setting [17, 18], agents learn from small sub-data sets and send partial SVM information to the central unit. Then the central unit combines and processes the information it gets. Parallel SVM training can handle massive data sets if the central unit is available. However, if the central unit fails for some physical or computational reasons, parallel SVM training is infeasible. Therefore, the designs of the fully distributed [13, 27] SVM training without requiring any central unit are urgent to binary classification learning.

Due to the limited communication capabilities and time-varying factors of the multi-agent network, switching topologies between agents bring about great challenges in the distributed design. Many distributed optimization algorithms have been proposed with time-varying topologies [16, 20, 30]. Because the (distributed) binary classification problem can be converted to a distributed optimization problem [5, 11], some of distributed SVM training has been constructed using distributed optimization approaches. For example, [11] proposed a consensus-based distributed SVM algorithm, which adopted the idea of alternating direction method of multipliers (ADMoM) [3] and distributed decomposition training. The ADMoM method [11] , whose performance was almost the same as that of centralized SVM training, employed neighborhood information exchange. Nevertheless, the communication cost was large since the algorithm should maintain the network connectivity between the agents until it converged to centralized SVM training. Furthermore, in nonlinear SVM training, the exchanged information was very large, and [10] provided a distributed gossip algorithm based on directly rare support vectors exchange. This convenient method betrayed data privacy, where the goal was to minimize the global structural error [7] with training data sets from different agents without exchanging the training data information explicitly among them. [19] was based on an improved distributed gossip convex hull algorithm in feature space, where the computational limitations of each agent was fully considered.

This paper is motivated to study efficient distributed nonlinear SVM algorithms in a time-varying multi-agent network. To this end, a distributed stochastic sub-gradient-based SVM algorithm is proposed and analyzed. Agents in the network can only get training subsets and have neighborhood information exchange. The technical contributions of the paper include:

- **Distributed stochastic design for switching networks based on kernel approximation:** Different from the existing distributed algorithms, we propose a distributed stochastic sub-gradient nonlinear support vector machines algorithm based on kernel approximation in time-varying networks. We construct our distributed algorithm based on soft kernel SVM designs, whose corresponding feature mapping may be infinity and need dimensional reduction. The feature mapping approximation method is given to construct our algorithm in a low-dimensional feature space. Similar to [11], the distributed sub-gradient iteration should hold until it approaches to centralized SVM performances. However, our algorithm, which uses stochastic sub-gradient for each agent in iteration and is constructed in time-varying networks, is of relatively low communication cost compared with [11]. Different from [7] and [19] based on random gossip network protocol, our algorithm is based on another class of time-varying network, the jointly-connected network [16, 30].

- **Convergence and performance analysis:** We strictly give the convergence rate and consensus rate of our algorithm. Different from the existing distributed SVM algorithms [11] and [19], we adopt stochastic sub-gradient-based distributed algorithm, which can help to take advantage of the high convergence rate. Numerical simulations are given to show that our algorithm can get the same result as the centralized ones.

The rest of the paper is organized as follows. Section 2 introduces the traditional SVM model and some necessary preliminaries to the distributed nonlinear SVM problem. Then, in Section 3, we formulate a distributed nonlinear SVM learning formulation in a low-dimensional feature space and give a method to reduce the dimension of the feature space. Next, in Section 4 the distributed stochastic sub-gradient-based SVM algorithm and its performance analysis are given. Then, Section 5 details the experimental results. Finally, conclusions are presented in Section 6.

2. PRELIMINARIES AND SVM FORMULATION

In this section, we first formulate a standard nonlinear SVM problem. Then we present some preliminary knowledge related to the distributed nonlinear SVM problem.

Consider the standard semi-supervised binary classification problem. The training set for agent i consists of n_i labeled samples $\{(x_{ij}, y_{ij})\}_{j=1}^{n_i}$ with $x_{ij} \in \mathbf{R}^p$, $y_{ij} \in \{-1, 1\}$. Given a feature mapping $\phi : \mathbf{R}^p \to \mathbf{R}^{2d}$, a convex loss function $C(\cdot) : \mathbf{R} \to \mathbf{R} \cup \{\infty\}$, coefficients $\lambda > 0$ and the local variables w_i, b_i, the primal nonlinear support vector machines problem [5] can be stated as follows:

$$\min_{w_i \in \mathbf{R}^{2d}, b_i \in \mathbf{R}} \frac{\lambda}{2} ||w_i||^2 + \frac{1}{n_i} \sum_{j=1}^{n_i} C(y_{ij}, h_i(\phi(x_{ij})), c_{ij}), \tag{1}$$

where $h_i(x) = w_i^\top x + b_i$ is the maximum-margin linear discriminant function; c_{ij} are the slack variables which contribute to the nonlinear separable training data sets. We choose hinge loss function as the loss function, given by

$$C(y, h(x), z) = \max\{0, 1 - yh(x) - z\}^c, \tag{2}$$

where c is a positive integer. Different from [6, 24], we choose $c = 1$ for the second term in (1) for accuracy.

Here, the training data set is not available on the central unit, but each agent i in the M-agent network has access to a stream of n_i labeled samples with $\sum_{i=1}^M n_i = N$.

As usual, the network with M agents is modeled by a graph $\mathcal{G}(t) = (\mathcal{M}, \mathcal{E}(t))$, $t = 0, 1, 2, \ldots$ with vertex set $\mathcal{M} := \{1, \ldots, M\}$ representing the agent/node set, and edges $\mathcal{E}(t)$ describing (time-varying) communication links among the agents at time t. The one-hop neighborhood of agent i at time t is defined by $\mathcal{N}_i(t) = \{q | (q, i) \in \mathcal{E}(t)\}$. Define the weights $a_i^q(t)$ of the adjacency matrix $A(t) \in \mathbf{R}^{M \times M}$ matching $\mathcal{G}(t)$, i.e. $a_i^q(t) > 0$ if $(q, i) \in \mathcal{E}(t)$ or $q = i$; $a_i^q(t) = 0$ otherwise. Moreover, we make the following assumptions on the network model:

Assumption 2.1. The graph $\mathcal{G}(t) = (\mathcal{M}, \mathcal{E}(t))$ and the adjacency matrix $A(t)$ satisfy:

(a) $A(t)$ is doubly stochastic, i. e. $A(t)\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^\top \mathbf{A(t)} = \mathbf{1}^\top$.

(b) For all $i \in \mathcal{M}$, $a_i^q(t) \geqslant v$ if $(q,i) \in \mathcal{E}(t)$ or $q = i$, where v is a positive scalar.

(c) The graph $\cup_{m=t}^{t+B-1} \mathcal{G}(m) = (\mathcal{M}, \mathcal{E}(t) \cup \mathcal{E}(t+1) \cup \cdots \cup \mathcal{E}(t+B-1))$ is strongly connected for all $t \geqslant 0$ and some positive integer B.

Assumption 2.1(c) provides a widely used connectivity condition [20] for the distributed optimization. In addition, Assumption 2.1(c) guarantees that each agent i can collect information from its one-hop neighbors at least once during each period of B, though the network topology is time-varying.

The following concepts are about convex function. A differentiable function f is convex over a convex set Ω, whose sub-gradient at a point x is denoted by $\nabla f(x)$, if

$$f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle, \ \ \forall x, y \in \Omega. \tag{3}$$

Moreover, the differentiable, convex function f is σ-strongly convex over the convex set Ω if

$$f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{\sigma}{2}||x - y||^2, \ \ \forall x, y \in \Omega. \tag{4}$$

3. DISTRIBUTED NONLINEAR SVM LEARNING

In this section, we first formulate a distributed nonlinear SVM problem in a low-dimensional feature space in Section 3.1. Then, a feature mapping approximation of ϕ_i for finding the low-dimensional feature space is introduced in Section 3.2.

3.1. Distributed nonlinear SVM in a low-dimensional feature space

In distributed binary classification learning, the goal is to find a global $h(x)$, and then enable each agent to classify any new input vector $x \in \mathbf{R}^p$ to one of $\{-1, 1\}$ without communicating its samples to other agents $q \neq i$. In a network consisting of M agents, the proposed distributed reformulation of (1) becomes:

$$\begin{gathered}\min_{w_i \in \mathbf{R}^{2d}, b_i \in \mathbf{R}} \frac{\lambda}{2} \sum_{i=1}^{M} ||w_i||^2 + \sum_{i=1}^{M} \frac{1}{n_i} \sum_{j=1}^{n_i} C(y_{ij}, h_i(\phi(x_{ij})), c_{ij}) \\ s.\,t.\ w_i = w_q,\ b_i = b_q,\ i \in \mathcal{M},\ q \in \mathcal{M}. \end{gathered}\tag{5}$$

Problem (5) can be solved in the distributed system because each agent i can optimize (1) and also meet $w_i = w_q$, $b_i = b_q$, by communicating with its one-hop neighbors. Moreover, Assumption 2.1 ensures the network-wide consensus.

Define, for notational brevity:

$$\begin{cases} \xi_i & = [w_i^\top, b_i]^\top, \\ \phi^{'}(X_{ij}) & = [\phi^\top(x_{ij}), 1]^\top \end{cases} \tag{6}$$

It follows from (6) that $w_i = (I_{2d+1} - \Pi_{2d+1})\xi_i$, where Π_{2d+1} is a $(2d+1) \times (2d+1)$ matrix with zeros except for $[\Pi_{2d+1}]_{(2d+1)(2d+1)} = 1$. Therefore, problem (5) can be rewritten as

$$\min_{\xi_i \in \mathbf{R}^{2d+1}} \frac{\lambda}{2} \sum_{i=1}^{M} \xi_i^\top (I_{2d+1} - \Pi_{2d+1})\xi_i + \sum_{i=1}^{M} \frac{1}{n_i} \sum_{j=1}^{n_i} C(y_{ij}, \xi_i^\top \phi^{'}(x_{ij}), c_{ij})$$
$$s.\ t.\ \xi_i = \xi_q,\ i \in \mathcal{M},\ q \in \mathcal{M}. \tag{7}$$

Substitute ξ_i in (7) according to the Karush–Kuhn–Tucker condition [22]:

$$\xi_i = \sum_{j=1}^{N} \alpha_{ij} \phi^{'}(x_{ij}), \tag{8}$$

and set $\alpha_i = [\alpha_{i1}, \alpha_{i2}, \alpha_{in_i}]^\top$ and $\phi_i^{'} = [\phi^{'}(x_{i1}), \phi^{'}(x_{i2}), \ldots, \phi^{'}(x_{in_i})]$. Then we transform (7) to the following form:

$$\min_{\alpha_i \in \mathbf{R}^{n_i}} \frac{\lambda}{2} \sum_{i=1}^{M} \alpha_i^\top (\phi_i^{'})^\top (I_{2d+1} - \Pi_{2d+1}) \phi_i^{'} \alpha_i + \sum_{i=1}^{M} \frac{1}{n_i} \sum_{j=1}^{n_i} C(y_{ij}, \Phi_i^j \alpha_i, c_{ij})$$
$$s.\ t.\ \alpha_i^\top \phi_i = \alpha_q^\top \phi_q,\ i \in \mathcal{M},\ q \in \mathcal{M}, \tag{9}$$

where $\Phi_i \in \mathbf{R}^{n_i \times n_i}$ is defined by

$$\Phi_i^{jt} := \phi^{'}(x_{ij})^\top \phi^{'}(x_{it}), j, t = 1, 2 \ldots, n_i, \tag{10}$$

and Φ_i^j stands for the jth row of matrix Φ_i.

Proposition 3.1. $\alpha_i \in \mathbf{R}^{n_i}$ is the solution of (9). Then, with w_i defined by (8), $(w_i, b_i) \in \mathbf{R}^{2d} \times \mathbf{R}$ is a solution of (5).

Suppose the original feature mapping is $\phi^o : \mathbf{R}^p \to \mathbf{H}$, where $\mathbf{H}$ is a Hilbert space. There exists a kernel $k^o : \mathbf{R}^p \times \mathbf{R}^p \to \mathbf{R}$ satisfying the conditions of Mercer Theorem [2], and the Hilbert space $\mathbf{H}$ is induced by the given kernel k^o. Suppose there exists a low-dimensional feature approximation $\phi : \mathbf{R}^p \to \mathbf{R}^{2d}$ of the original feature mapping ϕ_i^o for agent i, where

$$k_i^o(x_{ij}, x_{it}) \approx \phi(x_{ij})^\top \phi(x_{it}), j, t = 1, 2 \ldots, n_i. \tag{11}$$

Construct a matrix $Q_i \in \mathbf{R}^{n_i \times 2d}$ and $K_i \in \mathbf{R}^{n_i \times (2d+1)}$ for agent i's training data set by defining the jth rows of Q_i and K_i, respectively, as

$$Q_i^j = \phi(x_{ij})^\top, K_i^j = \phi^{'}(x_{ij})^\top,\ \ j = 1, 2, \ldots n_i, \tag{12}$$

Therefore,

$$\Phi_i := K_i K_i^\top \approx \Phi_i^o = [\phi^{'}(x_{ij})^\top \phi^{'}(x_{it})]_{j,t=1,2,\ldots,n_i}. \tag{13}$$

Φ_i is a positive semidefinite rank-$(2d+1)$ kernel approximation to Φ_i^o. Substitute $\Phi_i = K_i K_i^\top$ in (9) and we get:

$$\min_{\alpha_i \in \mathbf{R}^{n_i}} \frac{\lambda}{2} \sum_{i=1}^{M} \alpha_i^\top K_i (I_{2d+1} - \Pi_{2d+1}) K_i^\top \alpha_i + \sum_{i=1}^{M} \frac{1}{n_i} \sum_{j=1}^{n_i} C(y_{ij}, K_i^j K_i^\top \alpha_i, c_{ij})$$
$$s.\ t.\ K_i^\top \alpha_i = K_q^\top \alpha_q,\ i \in \mathcal{M},\ q \in \mathcal{M}. \tag{14}$$

The variable substitution

$$\gamma_i = K_i^\top \alpha_i \tag{15}$$

leads to

$$\min_{\gamma_i \in \mathbf{R}^{2d+1}} \frac{\lambda}{2} \sum_{i=1}^{M} \gamma_i^\top (I_{2d+1} - \Pi_{2d+1}) \gamma_i + \sum_{i=1}^{M} \frac{1}{n_i} \sum_{j=1}^{n_i} C(y_{ij}, K_i^j \gamma_i, c_{ij})$$
$$s.\ t.\ \gamma_i = \gamma_q,\ i \in \mathcal{M},\ q \in \mathcal{M}. \tag{16}$$

According to (6) and (12), $K_i^{j\top} = \phi(x_{ij})' \in \mathbf{R}^{2d+1}$ holds. The Problem (16) can be regarded as a distributed nonlinear SVM optimization with ϕ_i or $\phi_i' \in \mathbf{R}^{2d+1}, i = 1, 2, \ldots, M$, whose solution can be obtained by algorithms discussed in Section 4. The feature mapping approximation technique to find ϕ_i' is given in Section 3.2.

The following result can be easily obtained.

Proposition 3.2. $\gamma_i \in \mathbf{R}^{2d+1}$ is the solution of (9). Then, with α_i defined in (15) and w_i defined in (8), $(w_i, b_i) \in \mathbf{R}^{2d} \times \mathbf{R}$ is a solution of (5).

3.2. Feature Mapping Approximation of ϕ_i

In this section, a method to find matrix Q_i satisfying (13) is discussed, which needs to construct approximate mappings ϕ based on random projections.

Use feature mapping method to define Q_i. Find a mapping $\phi : \mathbf{R}^p \to \mathbf{R}^{2d}$ that satisfies

$$\langle \phi^o(x_{ij}), \phi^o(x_{it}) \rangle = \mathbb{E}[\langle \phi(x_{is}), \phi(x_{it}) \rangle]. \tag{17}$$

The feature mapping ϕ_i for agent i can be explicitly approximated by random projection method in [1],

$$\begin{cases} \phi(x) = \sqrt{\frac{1}{2d}}[\cos(v_1^\top x), \cos(v_2^\top x), \ldots, \cos(v_d^\top x), \sin(v_1^\top x), \sin(v_2^\top x), \ldots, \sin(v_d^\top x)]^\top \\ \phi'(x) = \sqrt{\frac{1}{2d}}[\cos(v_1^\top x), \cos(v_2^\top x), \ldots, \cos(v_d^\top x), \sin(v_1^\top x), \sin(v_2^\top x), \ldots, \sin(v_d^\top x), \sqrt{2d}]^\top \end{cases} \tag{18}$$

where $v_1, v_2 \ldots, v_d \in \mathbf{R}^p$ are i.i.d. sampled from a distribution whose density function is $p(v)$. The kernels we use can determine the density function $p(v)$. For the Gaussian kernel for agent i,

$$k_i^o(x_{is}, x_{it}) = \exp\left(-\sigma ||x_{is} - x_{it}||_2^2\right), \tag{19}$$

we get

$$p(v) = \frac{1}{(4\pi\sigma)^{d/2}} \exp(-\frac{||v||_2^2}{4\sigma}), \tag{20}$$

where $p(v)$ is from the Fourier transformation of k_i^o.

According to [15], we can assume that sampling of each $v_{ij} \in \mathbf{R}^p$ for agent i takes $O(p)$ time. The feature mapping approximation method requires $O(pd)$ operations for each training point. However, for a fixed d, the feature mapping approximation method gives lower prediction accuracy than the previous one. Next, we describe how to predict a new data point without recovering the α_i in (9), when we are given the solution γ in (16). By the definitions (8), (13) and (15), we obtain

$$h_i(\phi(x)) = w_i^T \phi_i(x) + b_i = \phi_i^{'}(x)^T \sum_{j=1}^{N} \alpha_{ij} \phi_i^{'}(x_{ij}) = \phi_i^{'}(x)^T K_i^T \alpha_i = \phi_i^{'}(x)^T \gamma_i. \tag{21}$$

Therefore, the time complexity is $O(nd)$.

4. DISTRIBUTED STOCHASTIC SUB-GRADIENT-BASED SVM ALGORITHM

In order to solve the nonlinear SVM formulation (16), we discuss a distributed stochastic sub-gradient-based SVM Algorithm. In general, we transfer problem (16) into the following optimization problem:

$$\begin{aligned} &\min_{\gamma\in\Omega} \; F(\gamma) = \sum_{i=1}^{M} F_i(\gamma) = \sum_{i=1}^{M} (f_i(\gamma) + \frac{1}{n_i} \sum_{j=1}^{n_i} g_{ij}(\gamma)) \\ &\text{where} \quad f_i(\gamma) = \frac{\lambda}{2} \gamma^\top (I_{2d+1} - \Pi_{2d+1}) \gamma, \; g_{ij}(\gamma) = C(y_{ij}, K_i^j \gamma_i, c_{ij}) \end{aligned} \tag{22}$$

where $\Omega = \{||\gamma||_2 \leqslant R\}$ is a bounded closed convex set in $\mathbf{R}^{2d+1}$.

Remark 4.1. [25] requires F_i to be strongly convex in all w_i and b_i, thus modifies F_i to be strongly convex in all γ_i. We also follow this assumption in this paper. Therefore, we have that for all $\gamma \in \Omega$, where $\Omega = \{||\gamma||_2 \leqslant R\}$ is a bounded closed convex set, $||\nabla f_i(\gamma)|| \leqslant M_f$ and $||\nabla g_{ij}(\gamma)|| \leqslant M_g$. Still, for function $g_{ij}(\gamma)$, we assume that there exists a $\iota > 0$ such that $\min_{g_{ij}(\gamma)=0} ||\nabla g_{ij}(x)|| \geqslant \iota$.

Note that each F_i is available to agent i only. Define $\nabla F_{ij}(\gamma)$ as the sub-gradient of $f_i(\gamma)+g_{ij}(\lambda)$. The distributed stochastic sub-gradient-based SVM algorithm (Algorithm 1) is proposed to solve problem (22). The property of F_i and Lemma 4.2 help us to construct Algorithm 1 and analyze its convergence performance.

Lemma 4.2. (Yuan et al. [30]) Under Assumption 2.1, denote $A(t:s)$ as the transition matrix with $A(t:s) = A(t) \cdots A(s)$ and $A(t:t) = A(t)$. For all i, q and all $t \geqslant s$, we have

$$|[A(t:s)]_{iq} - \frac{1}{N}| \leqslant \varsigma^{-2} \eta^{t-s+1}.$$

where $\varsigma = 1 - \frac{v}{4M^2}$, and $\eta = \varsigma^{1/B}$.

Algorithm 1 Distributed Stochastic Sub-gradient-based SVM Algorithm

```
Input: $X_i = \{(x_{i1}, y_{i1}), \ldots, (x_{iN}, y_{in_i})\}$,$\Phi_i^0$,positive integer $T$, Bounded convex set $\Omega$,
Set $\gamma_i(0) = \mathbf{0}$,$\mu(t) = \frac{1}{t}$.
for $t = 0, 1, \ldots T$ do
        for $i = 1, 2, \ldots, M$ do
            Choose $\kappa_i(t) \in \{1, 2 \ldots n_i\}$ at random.
            $K_i^j = \phi_i^{'}(x_{ij})$ in (18).
            $z_i(t) = \gamma_i(t) - \mu(t)\nabla F_{i\kappa_i(t)}(\gamma_i(t))$.
            $\bar{\gamma}_i(t+1) = \sum_{q=1}^{M} a_i^q(t) z_q(t)$.
            $\gamma_i(t+1) = P_\Omega(\bar{\gamma}_i(t+1))$.
        end for
end for
Output: $\gamma_i(T)$
```

Denote

$$\rho(t) = \frac{1}{M}\sum_{i=1}^{M}\gamma_i(t). \tag{23}$$

and $\tilde{\gamma}_i(T) = P_\Omega(\hat{\gamma}_i(T))$ where $\hat{\gamma}_i(T) = \frac{1}{T}\sum_{t=1}^{\top}\gamma_i(t)$. We need to analyze the convergence performance of $F_i(\cdot)$ at $\tilde{\gamma}_i(T)$ for every $i \in M$ and the convergence rate of Algorithm 1. Theorem 4.3 shows that the consensus rate among the agents in the time-varying network is at an expected rate $O(\ln T/T)$.

Theorem 4.3. Let the step-size sequence be $\mu(t) = \dfrac{1}{\sigma t}$, $t = 1, 2, \ldots, T$. With Assumption 2.1 and Remark 4.1, for all $i \in \mathcal{M}$ and any iteration $T \geqslant 3$, we have

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{M}||\gamma_i(T) - \rho(t)|| \leqslant C_1\frac{\ln T}{T}$$

where $C_1 = \frac{2M}{\sigma}(\frac{3M}{\varsigma^2(1-\eta)} + 4)(M_f + M_g)$.

Proof. $\rho(t)$ evolves as follows:

$$\rho(t) = \frac{1}{M}\sum_{i=1}^{M} z_i(t-1) + \frac{1}{M}\sum_{i=1}^{M} p_i(t), \tag{24}$$

where $p_i(t) = P_\Omega(\bar{\gamma}_i(t)) - \bar{\gamma}_i(t)$. Summing up equation (24) over $s = 1, 2, \ldots, t$, we obtain

$$\rho(t) = \rho(1) - \sum_{s=1}^{t-1}\mu(s)\frac{1}{M}\sum_{i=1}^{M}\nabla F_{i\kappa_i(s)}(\gamma_i(s)) + \sum_{s=1}^{t-1}\frac{1}{M}\sum_{i=1}^{M} p_i(s+1). \tag{25}$$

Analogously, $\gamma_i(t)$ evolves as follows:

$$\begin{aligned}\gamma_i(t) = \sum_{q=1}^{M}[A(t-1:1)]_{iq}\gamma_q(1) - \sum_{s=1}^{t-1}\mu(s)\sum_{q=1}^{M}[A(t-1:s)]_{iq}\nabla F_{q\kappa_q(s)}(\gamma_q(s))\\ + \sum_{s=1}^{t-2}\sum_{q=1}^{M}[A(t-1:s+1)]_{iq}p_q(s+1) + p_i(t).\end{aligned} \tag{26}$$

Combining (25) and (26), we get

$$\begin{aligned}||\gamma_i(t)-\rho(t)|| \leqslant{} & ||\sum_{q=1}^{M}[A(t-1:1)]_{iq}\gamma_q(1)-\rho(1)|| + ||p_i(t)||\\ & + \sum_{s=1}^{t-1}\mu(s)\sum_{q=1}^{M}|[A(t-1:s)]_{iq}-\frac{1}{M}|||\nabla F_{q\kappa_q}(\gamma_q(s))||\\ & + \sum_{s=1}^{t-2}\sum_{q=1}^{M}|[A(t-1:s+1)]_{iq}-\frac{1}{M}|||p_q(s+1)|| + ||\frac{1}{M}\sum_{i=1}^{M}p_i(t)||.\end{aligned} \tag{27}$$

For all $q \in \mathcal{M}$, $\gamma_q(1)=0$ and $\rho(1)=0$ holds. Therefore,

$$||\sum_{q=1}^{M}[A(t-1:1)]_{iq}\gamma_q(1)-\rho(1)|| = 0. \tag{28}$$

The second term $||p_i(t)||$ is bounded as follows:

$$\begin{aligned}||p_i(t)|| \leqslant{} & ||P_\Omega[\bar{\gamma}_i(t)] - \sum_{q=1}^{M}a_i^q(t-1)\gamma_q(t-1)||\\ & + \mu(t-1)\sum_{q=1}^{M}a_i^q(t-1)||\nabla F_{q\kappa_q(t-1)}(\gamma_q(t-1))||\\ \leqslant{} & 2\mu(t-1)\sum_{q=1}^{M}a_i^q(t-1)||.\end{aligned} \tag{29}$$

According to Remark 4.1, the following inequality holds:

$$||\nabla F_{i\kappa_i}(\gamma_i(t))|| \leqslant ||\nabla f_{i\kappa_i}(\gamma_i(t))|| + \lambda||\nabla g_{i\kappa_i}(\gamma_i(t))|| \leqslant M_f + M_g. \tag{30}$$

Hence,

$$||p_i(t)|| \leqslant 2\mu(t-1)\sum_{q=1}^{M}a_i^q(t-1)||\nabla F_{q\kappa_q}(\gamma_q(t-1))|| \leqslant 2(M_f+M_g)\mu(t-1). \tag{31}$$

Therefore, according to (27) and Lemma 4.2, we have

$$||\gamma_i(t) - \rho(t)|| \leqslant M(M_f + M_g)\varsigma^{-2} \sum_{s=1}^{t-1} \mu(s)\eta^{t-s} + 2M(M_f + \lambda M_g)\varsigma^{-2} \sum_{s=1}^{t-2} \mu(s)\eta^{t-s-1} + 4(M_f + M_g)\mu(t-1). \tag{32}$$

Since $\varsigma < 1$,

$$\sum_{t=1}^{T} ||\gamma_i(t) - \rho(t)|| \leqslant 3M(M_f + M_g)\varsigma^{-2} \sum_{t=2}^{T} \sum_{s=1}^{t-2} \mu(s)\eta^{t-s-1} + 4(M_f + M_g) \sum_{t=2}^{T} \mu(t-1). \tag{33}$$

Note that $\sum_{t=2}^{T} \sum_{s=1}^{t-2} \mu(s)\eta^{t-s-1} \leqslant \frac{1}{\varsigma^2(1-\eta)} \sum_{t=1}^{T-1} \mu(t)$, and, for all $T \geqslant 3$, $\sum_{t=1}^{T} \mu(t) \leqslant \frac{2}{\sigma} \ln T$. Thus, the conclusion follows. □

The following lemma analyzes the convergence performance of the Distributed Stochastic Sub-gradient-based SVM Algorithm (Algorithm 1).

Lemma 4.4. With Assumptions 2.1 and Remark 4.1, for all $i \in \mathcal{M}$ and any iteration $t \geqslant 1$, we have

$$F(\gamma_q(t)) - F(\gamma) \leqslant \frac{1}{2\mu(t)}((1 - \sigma\mu(k)) \sum_{i=1}^{M} r_i(t) - \sum_{i=1}^{M} r_i(t+1)) + \frac{M}{2}(M_f + M_g)^2\mu(t) + (M_f + M_g) \sum_{i=1}^{M} ||\gamma_i(t) - \gamma_q(t)||,$$

where $r_i(t) = ||\gamma_i(t) - \gamma||^2$, for all $\gamma \in \Omega$ and $t \geqslant 1$.

Proof. Clearly, the following inequality holds

$$r_i(t+1) \leqslant \sum_{q=1}^{M} a_i^q(t)||z_q(t+1) - \gamma||^2. \tag{34}$$

Therefore,

$$||z_q(t+1) - \gamma||^2 = r_q(t) + \mu^2(t)||\nabla F_{q\kappa_q(t)}(\gamma_q(t))||^2 + 2\mu(t)\langle \gamma - \gamma_q(t), \nabla F_{q\kappa_q(t)}(\gamma_q(t))\rangle. \tag{35}$$

Substitute (35) into (34). Summing over all $i \in \mathcal{M}$ gives

$$\sum_{i=1}^{M} r_i(t+1) \leqslant \sum_{q=1}^{M} (r_q(t) + \mu^2(t)||\nabla F_{q\kappa_q(t)}(\gamma_q(t))||^2 + 2\mu(t)\langle \gamma - \gamma_q(t), \nabla F_{q\kappa_q(t)}(\gamma_q(t))\rangle). \tag{36}$$

According to Remark 4.1, the term $||\nabla F_{q\kappa_q(t)}(\gamma_q(t))||^2$ satisfies the following inequality:

$$||\nabla F_{q\kappa_q(t)}(\gamma_q(t))||^2 \leqslant (M_f + M_g)^2. \tag{37}$$

Combining (36) and (37) yields

$$\sum_{i=1}^{M}\langle\gamma_i(t) - \gamma, \nabla F_{i\kappa_i(t)}(\gamma_i(t))\rangle \leqslant \frac{1}{2\mu(t)}(\sum_{i=1}^{M} r_i(t) - \sum_{i=1}^{M} r_i(t+1)) + \frac{M}{2}(M_f + M_g)^2. \tag{38}$$

Due to $F_i(\gamma) \leqslant F_i(\gamma_i(t)) + \langle\gamma - \gamma_i(t), \nabla F_{i\kappa_i(t)}(\gamma_i(t))\rangle + \frac{\sigma}{2}||\gamma_i(t) - \gamma||^2)$, the following inequality holds:

$$\begin{aligned}&\sum_{i=1}^{M}(F_i(\gamma_i(t)) - F_i(\gamma) + \frac{\sigma}{2}||\gamma_i(t) - \gamma||^2)\\ &\geqslant F(\gamma_q(t)) - F(\gamma) + \frac{\sigma}{2}\sum_{i=1}^{M} r_i(t) - (M_f + M_g)\sum_{i=1}^{M}||\gamma_i(t) - \gamma_q(t)||,\end{aligned} \tag{39}$$

Combining (39) with (38) yields the conclusion. □

Theorem 4.5 shows that the convergence of the Distributed Stochastic Sub-gradient-based SVM Algorithm (Algorithm 1) is also at an expected rate $O(\ln T/T)$.

Theorem 4.5. Let $\mu(t) = \dfrac{1}{\sigma t}$, $t = 1, 2, \ldots, T$. Denote $\xi^* = \arg\min_{\xi\in\Omega} f(\xi)$ and set $\kappa = \dfrac{\ln T}{T}$. With Assumption 2.1 and Remark 4.1, for all $q \in \mathcal{M}$ and any iteration $T \geqslant 3$, we have

$$F(\tilde{\gamma}_q(T)) - F(\gamma^*) \leqslant C_2\frac{\ln T}{T},$$

where $C_2 = \frac{M}{\sigma}(\frac{12M}{\varsigma^2(1-\eta)} + 17)(M_f + M_g)^2$.

Proof. Summing the inequality in Lemma 4.4 over $t = 1, \ldots, T$, we have

$$\sum_{t=1}^{T} F(\gamma_q(t)) - F(\gamma) \leqslant I_T + II_T + III_T, \tag{40}$$

where

$$\begin{cases} I_T &= \sum_{t=1}^{T}\frac{1}{2\mu(t)}((1 - \sigma\mu(k))\sum_{i=1}^{M} r_i(t) - \sum_{i=1}^{M} r_i(t+1)),\\ II_T &= \frac{M}{2}(M_f + M_g)^2\sum_{t=1}^{T}\mu(t),\\ III_T &= (M_f + M_g)\sum_{t=1}^{T}\sum_{i=1}^{M}||\gamma_i(t) - \gamma_q(t)||.\end{cases} \tag{41}$$

Since $\mu(t) = \frac{1}{\sigma t}$, the terms I_T, II_T and III_T are bounded:

$$\begin{cases} I_T & = -\frac{1}{2\mu(T)} \sum_{i=1}^{M} r_i(T+1) \leqslant 0, \\ II_T & = \frac{M}{2}(M_f + \lambda M_g)^2 \sum_{t=1}^{T} \frac{1}{\sigma t} \leqslant \frac{M}{\sigma}(M_f + M_g)^2 \ln T, \\ III_T & \leqslant 2(M_f + \lambda M_g) C_1 \ln T \leqslant \frac{4M}{\sigma}(\frac{3M}{\varsigma^2(1-\eta)} + 4)(M_f + M_g)^2 \ln T. \end{cases} \tag{42}$$

Combining (42) with (40) yields

$$[F(\hat{\gamma}_q(T))) - F(\gamma^*)] \leqslant \frac{M}{\sigma}(\frac{12M}{\varsigma^2(1-\eta)} + 17)(M_f + M_g)^2 \frac{\ln T}{T}, \tag{43}$$

We complete the proof by $F(\hat{\gamma}_q(T)) \leqslant \sum_{t=1}^{\top} F(\gamma_q(t))$. □

Remark 4.6. If we choose $\mu(t) = \frac{1}{\bar{\sigma} t}$ with $\bar{\sigma} \leqslant \sigma$, the convergence rate of the Distributed Stochastic Sub-gradient-based SVM Algorithm (Algorithm 1) is also at an expected rate $O(\frac{\ln T}{T})$. In fact, although we consider a class of deterministic time-varying multi-agent networks, the analysis idea can be simplified to some stochastic time-varying networks, for example, random sleeping networks (referring to [29]).

5. SIMULATION

In this section, two simulations are given to evaluate the performance of the distributed stochastic sub-gradient-based SVM algorithm on open data sets from UCI data set. Simulation 5.1 is on Iris data set to illustrate the feature mapping performance of Algorithm 1 on a two-agent network, the test accuracy rate is given for different choices of d and iterations T. Simulation 5.2 is on DSPS data set, which reveals that Algorithm 1 can get the same test accuracy rate as the centralized algorithms.

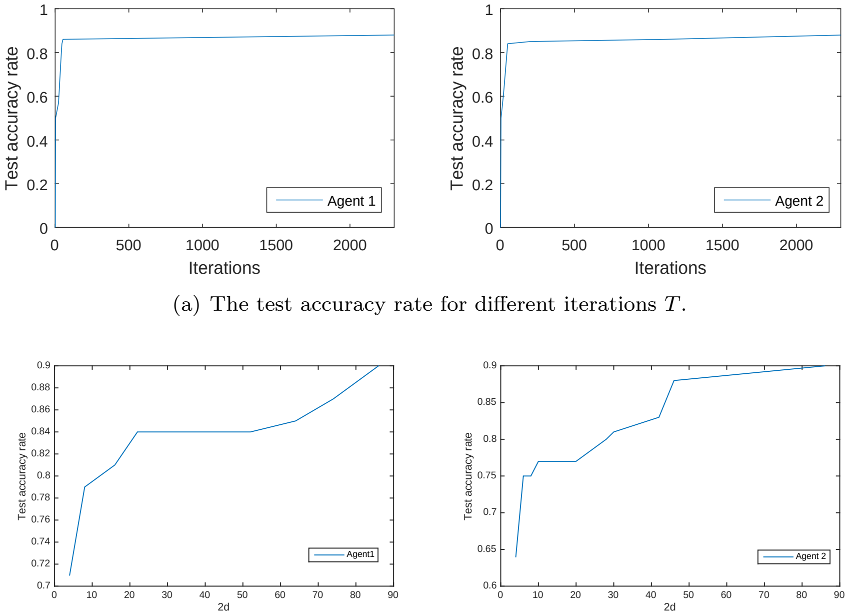
Simulation 5.1. We choose three different species of a well-known plant, Iris, (that is, Iris-setosa, Iris-versicolor and Iris-virginica) for the classification. Each species data set for the classification consists of 50 four-dimensional entities. Consider a two-agent network; each agent can get all these 150 data points and the labels Y_i.

Name	$\phi'(x_{ij})$	λ	c_{ij}	$h_i(x)$
setosa vs versicolor	$\phi(x_{ij})$ in (18)	0.005	0.01	$h_{i_1}(x) = w_{i_1}^\top x + b_{i_1}$
versicolor vs virginica	$\phi(x_{ij}) = [x_{ij}, 1]^\top$	7×10^{-7}	0.001	$h_{i_2}(x) = w_{i_2}^\top x + b_{i_2}$
setosa vs virginica	$\phi(x_{ij}) = [x_{ij}, 1]^\top$	7×10^{-7}	0.001	$h_{i_3}(x) = w_{i_3}^\top x + b_{i_3}$

Tab. 1. Iris data set and training parameters.

Transform the Iris multivariate classification into three SVM problems using one-versus-one method [28]. Table 1 illustrates the three Iris SVM tasks and the different training parameters we choose. Denote $y_{i_k j}$ as the output of h_{i_k}. Take the super majority voting principle to get the final classifier as follows:

$$H_i(x) = \begin{cases} y_{i_k j}, & \text{if } h_{i_k}(x) > 0.5 \sum_{k=1}^{3} h_{i_k}(x); \\ \text{reject}, & \text{otherwise}. \end{cases} \tag{44}$$



(a) The test accuracy rate for different iterations T.

(b) The test accuracy rate for different dimension d.

Fig. 1. The test accuracy rate for Iris data set.

Figure 1(a) shows that the testing performance keeps improved as iterations T increases for a given dimension $d = 50$. Figure 1(b) shows that the testing performance improves as dimension d increases for $T = 3000$. In practice, we can give a large dimension d for our algorithm to solve the SVM classification problem.

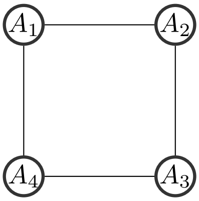


Fig. 2. Topology of the 4-agent network.

Simulation 5.2. We choose USPS(binary codes training sets) as the training sets. Consider a four-agent network. The topology of the network is connected as shown in Figure 2. Agent $1, 2, 3$ all have access to 1823 training points and agent 4 has access to 1822 training points. We need to classify these 7291 training points into 10 different groups through one-versus-rest method. Use USPS(binary codes validation set) (There are total 2007 samples in the validation set.) to test the test accuracy performance of Algorithm 1 for these 4 agents.

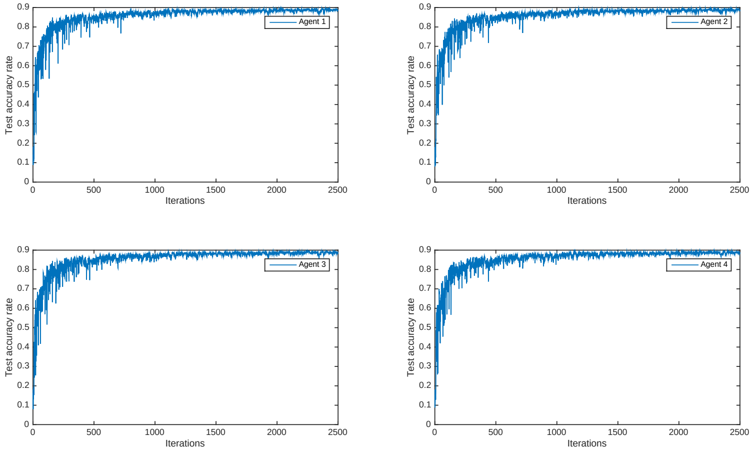


Fig. 3. The test accuracy rate for different iterations T.

Figure 3 shows the test accuracy rate for different iterations T, with $c_{ij} = 0$, $\phi(x_{ij}) = [x_{ij}, 1]^\top$ and $\lambda = 7 \times 10^{-7}$. The testing performance keeps improved as iterations T increases. Figure 4 shows the test accuracy rate for different λ, with $c_{ij} = 0$, $\phi(x_{ij}) = [x_{ij}, 1]^\top$ and $T = 2500$. The testing performance keeps improved as λ decreases. The test accuracy rates of our algorithm are 0.8864, 0.8834, 0.8839 and 0.8839 which are the same as centralized stochastic SVM algorithm.

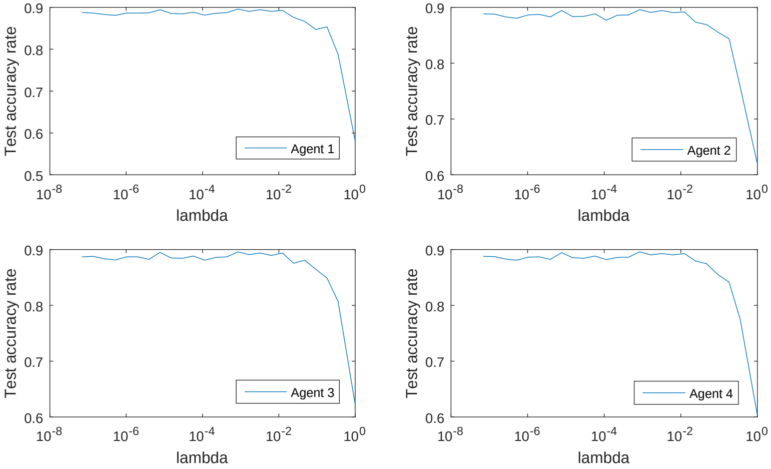


Fig. 4. The test accuracy rate for different λ.

6. CONCLUSION

In this paper, we discussed the support vector machines formulation for binary classification and a distributed SVM algorithm, for the cases with decentralized training data unavailable to all agents. We proposed a distributed stochastic sub-gradient-based SVM problem, and solved the binary classification problem by making agents achieve global consensus only through one-hop neighborhood communication with time-varying topologies. Moreover, we analyzed the convergence rate and consensus rate of the given algorithm. Finally, We did simulations using various real classification training sets, and compared its performance to a centralized SVM training to demonstrate the effectiveness our algorithm.

ACKNOWLEDGEMENT

This work is supported by the National Key Research and Development Program of China (2016YFB0901902), National Natural Science Foundation of China (61573344, 61333001).

(Received April 13, 2017)

REFERENCES

[1] R. Ali and B. Recht: Random features for large-scale kernel machines. In: Advances in Neural Information Processing System, MIT Press, Massachusetts 2008, pp. 1177–1184.

[2] S. Bernhard, and A. J. Smola: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Massachusetts 2002.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning *3* (2011), 1–122. DOI:10.1561/2200000016

[4] C. C. Chang and C. J. Lin: LIBSVM: a library for support vector machines. JACM Trans. Intell. Systems Technol. *2* (2011), 1–27. DOI:10.1145/1961189.1961199

[5] O. Chapelle: Training a support vector machine in the primal. Neural Computation *19* (2007), 1155–1178. DOI:10.1162/neco.2007.19.5.1155

[6] O. Chapelle and A Zien: Semi-supervised classification by low density separation. In: Proc. International Conference on Artificial Intelligence and Statistics, Barbados 2005.

[7] C. Cortes and V. Vapnik: Support-vector networks. Machine Learning *20* (1995), 273–297. DOI:10.1007/bf00994018

[8] P. Drineas and M. W. Mahoney: On the Nyström method for approximating a Gram matrix for improved kernel-based learning. J. Machine Learning Research *6* (2005), 2153–2175.

[9] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy: Advances in Knowledge Discovery and Data Mining. AAAI Press, Menlo Park 1996.

[10] K. Flouri, B. Beferull-Lozano, and P. Tsakalides: Distributed consensus algorithms for SVM training in wireless sensor networks. In: 16th European Signal Processing Conference, Lausanne 2008. DOI:10.1109/icdsp.2009.5201180

[11] P A. Forero, A. Cano, and G. B. Giannakis: Consensus-based distributed support vector machines. J. Machine Learning Research *11* (2010), 1663–1707.

[12] V. Franc and S. Sonnenburg: Optimized cutting plane algorithm for support vector machines. In: Proc. 25th International Conference on Machine Learning, Helsinki 2008. DOI:10.1145/1390156.1390197

[13] J. Hu: On robust consensus of multi-agent systems with communication delays. Kybernetika *45* (2009), 768–784.

[14] T. Joachims, T. Finley, and C. J. Yu: Cutting-plane training of structural SVMs. Machine Learning *77* (2009), 27–59. DOI:10.1007/s10994-009-5108-8

[15] S. Lee and S. J. Wright : Approximate Stochastic Sub-gradient Estimation Training for Support Vector Machines. In: Mathematical Methodologies in Pattern Recognition and Machine Learning, Springer, New York 2011, pp. 67–82. DOI:10.1007/978-1-4614-5076-4_5

[16] Y. Lou , Y. Hong, and S. Wang: Distributed continuous-time approximate projection protocols for shortest distance optimization problems. Automatica *69* (2016), 289–297. DOI:10.1016/j.automatica.2016.02.019

[17] Y. Lu, V. Roychowdhury, and L. Vandenberghe: Distributed parallel support vector machines in strongly connected networks. IEEE Trans. Neural Networks *19* (2008), 1167–1178. DOI:10.1109/tnn.2007.2000061

[18] W. Kim, J. Park, J. Yoo, H. J. Kim, and C. G. Park: Target localization using ensemble support vector regression in wireless sensor networks. IEEE Trans. Cybernetics *43* (2013), 1189–1198. DOI:10.1109/tsmcb.2012.2226151

[19] W, Kim, M. S. Stanković, K. H. Johansson, and H.J. Kim:˙A distributed support vector machine learning over wireless sensor networks. IEEE Trans. Neural Cybernetics *45* (2015), 2599–2611.

[20] A. Nedic and O. Asuman: Distributed sub-gradient methods for multi-agent optimization. IEEE Trans. Automatic Control *54* (2009), 48–61. DOI:10.1109/tac.2008.2009515

[21] J. C. Platt: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods, MIT Press, Massachusetts 1999, pp. 185–208.

[22] B. T. Polyak : Introduction to Optimization. Springer, New York 1987.

[23] R. Rifkin and A. Klautau: In defense of one-vs-all classification. J. Machine Learning Research *5* (2004), 101-141.

[24] S. Scardapane, R. Fierimonte, P. D. Lorenzo, M. Panella, and A. Uncini: Distributed semi-supervised support vector machines. Neural Networks *80* (2016), 43–52. DOI:10.1016/j.neunet.2016.04.007

[25] S. Shalev-Shwartz, Y. Singer, and N. Srebro: Pegasos: Primal estimated sub-gradient solver for svm. In: Proc. 24th International Conference on Machine Learning, Oregon 2007. DOI:10.1145/1273496.1273598

[26] S. Sra, S. Nowozin, and S. J. Wright: Optimization for Machine Learning. MIT Press, Massachusetts 2012.

[27] X. Wang and Y. Chen: Quantized distributed output regulation of multi-agent systems. Kybernetika *52* (2016), 427–440. DOI:10.14736/kyb-2016-3-0427

[28] J. Weston and C. Watkins: Support vector machines for multi-class pattern recognition. ESANN *99* (1999), 219–224.

[29] P. Yi and Y. Hong: Stochastic sub-gradient algorithm for distributed optimization with random sleep scheme. Control Theory Technol. *13* (2015), 333–347. DOI:10.1007/s11768-015-5100-8

[30] D. Yuan, D. W. C. Ho and Y. Hong: On Convergence rate of distributed stochastic gradient algorithm for convex optimization with inequality constraints. SIAM J. Control Optim. *54* (2016), 2872–2892. DOI:10.1137/15m1048896

Yinghui Wang, School of Mathematical Sciences,University of Chinese Academy of Sciences, Beijing. P. R. China; Key Laboratory of Systems and Control, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing. P. R. China.
e-mail: wangyinghuisdu@163.com

Peng Lin, School of Mathematical Sciences,University of Chinese Academy of Sciences, Beijing. P. R. China; Key Laboratory of Systems and Control, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing. P. R. China.
e-mail: penglin@amss.ac.cn

Huashu Qin, Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing. P. R. China.
e-mail: qin@iss.ac.cn