
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Doostmohammadian, Mohammadreza; Aghasi, Alireza; Charalambous, Themistoklis; Khan, Usman A.

Distributed support vector machines over dynamic balanced directed networks

Published in:
IEEE Control Systems Letters

DOI:
[10.1109/LCSYS.2021.3086388](https://doi.org/10.1109/LCSYS.2021.3086388)

Published: 01/01/2022

Document Version
Peer reviewed version

Please cite the original version:
Doostmohammadian, M., Aghasi, A., Charalambous, T., & Khan, U. A. (2022). Distributed support vector machines over dynamic balanced directed networks. *IEEE Control Systems Letters*, 6, 758-763. [9446550]. <https://doi.org/10.1109/LCSYS.2021.3086388>

© 2021 IEEE. This is the author's version of an article that has been published by IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Distributed support vector machines over dynamic balanced directed networks

Mohammadreza Doostmohammadian, *Member, IEEE*, Alireza Aghasi, *Member, IEEE*, Themistoklis Charalambous, *Senior Member, IEEE*, and Usman A. Khan, *Senior Member, IEEE*

Abstract—In this paper, we consider the binary classification problem via distributed Support Vector Machines (SVMs), where the idea is to train a network of agents, with limited share of data, to cooperatively learn the SVM classifier for the global database. Agents only share processed information regarding the classifier parameters and the gradient of the local loss functions instead of their raw data. In contrast to the existing work, we propose a continuous-time algorithm that incorporates network topology changes in discrete jumps. This hybrid nature allows us to remove chattering that arises because of the discretization of the underlying CT process. We show that the proposed algorithm converges to the SVM classifier over time-varying weight balanced directed graphs by using arguments from the matrix perturbation theory.

Index Terms—Support Vector Machines, distributed optimization, matrix perturbation theory.

I. INTRODUCTION

MACHINE-learning has been an area of significant research in recent signal processing and control literature [1]–[4]. Among the supervised-learning methods, Support Vector Machines (SVMs) find several applications ranging from image/video processing to bioinformatics. Motivated by the recent applications in robotic networks and the Internet of Things, we are interested in developing distributed solutions for SVM classification. The basic idea is to process the raw data at each node to train a local classifier and then fuse these classifiers among the neighboring nodes. D-SVM (distributed SVM) finds applications where a subset of the data is acquired by different nodes/servers/agents possibly at different geographic locations, privacy is of concern, and communication to a fusion center is infeasible.

In binary classification, SVM defines the maximum-margin hyperplane (the classifier) determined by the closest data samples (Support Vectors). The preliminary work on D-SVM (referred as Distributed Parallel SVM [5] and Parallel SVM [6]) is focused on local computation/sharing of the support vectors [5]–[9]. These local support vectors are updated either via a fusion center [6]–[8], over a Hamiltonian

multi-agent cycle [5], or via a distributed method based on alternating direction method of multipliers (ADMM) [9]. A major drawback is that these approaches require sharing raw data over the network, raising data privacy and information security issues. More recently, consensus-based distributed optimization methods are proposed in [10]–[26], where instead of raw data, agents share *processed* information, which in case of leakage to unauthorized parties reveals little information about the original data. Among these, the solution in [16] requires distributed computation of the Hessian inverse, while [17], [18] consider a penalty term on consensus constraint violation with certain optimality gap [27]. In contrast, Lagrangian and ADMM-based methods proposed in [22], [23], [25] can achieve null constraint violation by combining the benefits of dual decomposition and augmented Lagrangian for constrained optimization. Such methods converge linearly to primal/dual optimal solutions for *strongly convex* loss functions [22]. Prediction-correction algorithm is proposed in [25] based on prediction of optimal conditions in time and correction on gradient descent or (damped) Newton method. Particular application in online kernel-based nonlinear regression learning is considered in [26], using a penalized stochastic gradient descent with low-dimensional subspace projection. A sub-gradient push-sum strategy over digraphs is proposed in [19], and its regret-based extension over dynamic networks is proposed in [21]. The perturbed push-sum descent with linear convergence rate over digraphs is given in [24]. Similarly, [23] proposes a linearly convergent solution over time-varying networks based on small-gain analysis. A double time-scale algorithm is proposed by [20] with finite number of communications per gradient-update iteration. Other methods include finite/fixed-time algorithms [11]–[15] that are prone to steady-state *chattering* due to non-Lipschitz dynamics.

In this paper, a D-SVM method is proposed that overcomes the challenges of semi-centralized (fusion center based) solutions and the chattering phenomena. Moreover, in contrast to Refs. [10]–[26], where either continuous-time (CT) or discrete-time (DT) protocols are considered, we propose a hybrid algorithm to address the topology switching of the multi-agent network in DT incorporated in a CT gradient-descent update [28]. Our hybrid approach enables more flexibility in considering mixed-dynamics [28], [29], which allows solving D-SVM via CT protocols over general *dynamic* digraphs in DT domain, in contrast to DT dynamics where the sampling times may be constrained [20]–[26], [30]. To analyze the proposed hybrid model, we use *matrix perturbation theory* [31] to char-

The work of UAK was supported in part by the U.S. National Science Foundation under awards CMMI-1903972 and CBET-1935555.

MD/TC are with the School of Electrical Engineering at Aalto University, Espoo, Finland (mohammadreza.doostmohammadian@aalto.fi, themistoklis.charalambous@aalto.fi). MD is also with the Faculty of Mechanical Engineering at Semnan University, Semnan, Iran (doost@semnan.ac.ir). AA is with Robinson College of Business at Georgia State University, GA, USA (aaghasi@gsu.edu). UAK is with the Electrical and Computer Engineering Department at Tufts University, MA, USA (khan@ece.tufts.edu).

acterize the eigen-spectrum of the proposed dynamics, which enables convergence analysis in the hybrid DT-CT setup. Due to Lipschitz-continuity of the proposed CT approach, it's DT approximation is free of the aforementioned *chattering* inherent to the non-Lipschitz dynamics [11]–[15]. Further, the proposed solution is free of penalty-based approximation inaccuracies in [17], [18], [27].

We now describe the rest of the paper. Section II recaps some preliminaries on algebraic graph theory, while Section III formulates the D-SVM problem. Section IV states our CT gradient descent method to address D-SVM, whereas the convergence analysis over dynamic WB-digraphs is available in Section V. Section VI provides an illustrative example, and finally, Section VII concludes the paper.

II. PRELIMINARIES ON ALGEBRAIC GRAPH THEORY

We represent the multi-agent network by a strongly-connected directed graph (SC digraph) \mathcal{G} . Assuming a positive weight w_{ij} for every link (from node j to node i) and zero otherwise, the irreducible weighted adjacency matrix of \mathcal{G} is $W = \{w_{ij}\}$, and the Laplacian matrix $\bar{W} = \{\bar{w}_{ij}\}$ with $\bar{w}_{ij} = w_{ij}$ for $i \neq j$ and $\bar{w}_{ij} = -\sum_{j=1}^n w_{ij}$ for $i = j$. The SC property of the graph is directly related to the rank of its Laplacian matrix as given in the next lemma.

Lemma 1: [32] The given Laplacian \bar{W} for a SC digraph has eigenvalues whose real-parts are non-positive with one isolated eigenvalue at zero.

Next, we define a WB-digraph as an SC digraph with equal weight-sum of incoming and outgoing links at every node i , i.e., $\sum_{j=1}^n w_{ji} = \sum_{j=1}^n w_{ij}$, implying the following lemma.

Lemma 2: [32] For the Laplacian \bar{W} of a WB-digraph, the vectors $\mathbf{1}_n^\top$ and $\mathbf{1}_n$ are respectively the left and right eigenvector associated with the zero eigenvalue, i.e., $\mathbf{1}_n^\top \bar{W} = \mathbf{0}_n$ and $\bar{W} \mathbf{1}_n = \mathbf{0}_n$, where $\mathbf{1}_n$ and $\mathbf{0}_n$ are the column vectors of 1's and 0's of size n , respectively.

In the rest of the paper, $\|A\|_\infty$ denotes the infinity norm of a matrix, i.e., $\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$.

III. PROBLEM STATEMENT

Consider binary classification of N points $\mathbf{x}_i \in \mathbb{R}^{m-1}$, $i = 1, \dots, N$, each belonging to one of two classes labeled by $l_i \in \{-1, 1\}$. Using the entire training set, the SVM problem is to find a hyperplane $\omega^\top \mathbf{x} - \nu = 0$, for $\mathbf{x} \in \mathbb{R}^{m-1}$, based on the maximum margin linear classification to partition the data into two classes. Subsequently, a new test data point $\hat{\mathbf{x}}$ belongs to the class labeled as $g(\hat{\mathbf{x}}) = \text{sgn}(\omega^\top \hat{\mathbf{x}} - \nu)$. In the linearly non-separable case, the data points are first projected into a high-dimensional space \mathcal{F} via a nonlinear mapping $\phi(\cdot)$ associated with a *kernel function* $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$. By proper mapping $\phi(\cdot)$, a linear optimal hyperplane can be found in \mathcal{F} such that $g(\hat{\mathbf{x}}) = \text{sgn}(\omega^\top \phi(\hat{\mathbf{x}}) - \nu)$ determines the class of $\hat{\mathbf{x}}$. The SVM problem is to find the optimal ω and ν by minimizing the following convex loss [33]:

$$\min_{\omega, \nu} \quad \omega^\top \omega + C \sum_{j=1}^N \max\{1 - l_j(\omega^\top \phi(\mathbf{x}_j) - \nu), 0\}^p \quad (1)$$

where $p = \{1, 2, \dots\}$ defines smoothness and the positive constant C determines the margin size. We adopt the standard convention of modifying the hinge loss in SVM, that is not differentiable, with a twice differentiable cost function [12]. Therefore, $\max\{z, 0\}^p$ for $p = 1$ in (1) is replaced by $L(z, \mu) = \frac{1}{\mu} \log(1 + \exp(\mu z))$. It can be shown that the maximum gap between the two functions inversely scales with μ , i.e., $L(z, \mu) - \max\{z, 0\} \leq \frac{1}{\mu}$, and the two can become arbitrarily close by selecting μ sufficiently large [34].

In *distributed* SVM (D-SVM), the data points are distributed over a network of n agents and each agent i possesses a local dataset with N_i data points denoted by $\mathbf{x}_j^i, j = 1, \dots, N_i$. Since each agent has access to partial data, the locally found values ω_i and ν_i , obtained by solving (1) over the local dataset $\mathbf{x}_j^i, j = 1, \dots, N_i$, may differ for each agent i . The idea behind D-SVM is thus to develop a distributed mechanism to learn the global classifier parameters by making sure that no agent reveals its local data to any other agent. The corresponding distributed optimization problem is given by:

$$\min_{\omega_1, \nu_1, \dots, \omega_n, \nu_n} \quad \sum_{i=1}^n f_i(\omega_i, \nu_i) \quad (2)$$

$$\text{subject to } \omega_1 = \dots = \omega_n, \quad \nu_1 = \dots = \nu_n, \quad (3)$$

where each local cost $f_i: \mathbb{R}^m \rightarrow \mathbb{R}$ is approximated as (with $z = 1 - l_j(\omega_i^\top \phi(\mathbf{x}_j^i) - \nu_i)$ and large enough $\mu > 0$)

$$f_i(\omega_i, \nu_i) = \omega_i^\top \omega_i + C \sum_{j=1}^{N_i} \frac{1}{\mu} \log(1 + \exp(\mu z)). \quad (4)$$

Let $\mathbf{x}_i = [\omega_i^\top; \nu_i] \in \mathbb{R}^m$ and let $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_n] \in \mathbb{R}^{mn}$ be the global state with the symbol ‘;’ denoting the column concatenation. Then, Problem (2) takes the following form:

$$\min_{\mathbf{x} \in \mathbb{R}^{mn}} F(\mathbf{x}), \quad F(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x}_i) \quad (5)$$

subject to $\mathbf{x}_1 = \mathbf{x}_2 = \dots = \mathbf{x}_n$.

We next provide the following lemma on the cost functions.

Lemma 3: [12] Each local cost f_i is twice differentiable and strictly convex, i.e., the $m \times m$ Hessian matrix $\nabla^2 f_i(\mathbf{x}_i)$ is positive definite, for all non-zero $\mathbf{x}_i \in \mathbb{R}^m$.

Clearly, any solution $\mathbf{x}_i^*, i = 1, \dots, n$, of (5) must satisfy $\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^*) = \mathbf{0}_m$, such that $\mathbf{x}_1^* = \dots = \mathbf{x}_n^* = \bar{\mathbf{x}}^*$, for some $\bar{\mathbf{x}}^* \in \mathbb{R}^m$. In other words, the optimality condition $\nabla F(\mathbf{x}^*) = \mathbf{0}_{mn}$ must hold for some $\mathbf{x}^* \in \mathbb{R}^{mn}$ such that $\mathbf{x}^* = \mathbf{1}_n \otimes \bar{\mathbf{x}}^*$, where $\nabla F: \mathbb{R}^{mn} \rightarrow \mathbb{R}^{mn}$ is the gradient of $F: \mathbb{R}^{mn} \rightarrow \mathbb{R}$.

IV. PROPOSED ALGORITHM: DYNAMICS AND AUXILIARY RESULTS

To solve problem (5), we consider the following continuous-time linear dynamics:

$$\dot{\mathbf{x}}_i = - \sum_{j=1}^n w_{ij}^q (\mathbf{x}_i - \mathbf{x}_j) - \alpha \mathbf{y}_i, \quad (6)$$

where $\mathbf{x}_i(t) \in \mathbb{R}^m$ represents the state of agent i at time $t \geq 0$, $\dot{\mathbf{x}}_i = \frac{d\mathbf{x}_i}{dt}$, $W_q = \{w_{ij}^q\}$ is the weighted adjacency associated

with \mathcal{G}_q (q is the switching index), and $\alpha > 0$ is the step-size. We note that instead of the standard descend direction $\nabla f_i(\mathbf{x}_i)$, the \mathbf{x}_i -update descends towards an auxiliary variable $\mathbf{y}_i(t) \in \mathbb{R}^m$, which tracks the sum of local gradients, asymptotically, and is updated via the following dynamics:

$$\dot{\mathbf{y}}_i = -\sum_{j=1}^n a_{ij}^q (\mathbf{y}_i - \mathbf{y}_j) + \frac{d}{dt} \nabla f_i(\mathbf{x}_i), \quad (7)$$

where $\dot{\mathbf{y}}_i = \frac{d\mathbf{y}_i}{dt}$ and $A_q = \{a_{ij}^q\}$ is the weighted adjacency matrix with the same zero/non-zero structure as the matrix W . Let $\mathbf{y} = [\mathbf{y}_1; \mathbf{y}_2; \dots; \mathbf{y}_n] \in \mathbb{R}^{mn}$ and note that $\frac{d}{dt} \nabla f_i(\mathbf{x}_i) = \nabla^2 f_i(\mathbf{x}_i) \dot{\mathbf{x}}_i$. We emphasize that the proposed algorithm, (6) and (7), is in continuous-time, however, the structure of the underlying graph \mathcal{G}_q may change in a discrete fashion. This makes the proposed dynamics *hybrid* where the states, \mathbf{x} and \mathbf{y} , evolve in CT collated with DT switching signal q . We make the following assumption on W_q and A_q .

Assumption 1: The weights $W_q = \{w_{ij}^q\}$ and $A_q = \{a_{ij}^q\}$ ($w_{ij}^q, a_{ij}^q \geq 0$) are associated with strongly-connected WB-digraphs. Further, $\sum_{j=1}^n w_{ij}^q < 1$ and $\sum_{j=1}^n a_{ij}^q < 1$. Following Assumption 1, we obtain from (6) and (7):

$$\sum_{i=1}^n \dot{\mathbf{y}}_i = \sum_{i=1}^n \frac{d}{dt} \nabla f_i(\mathbf{x}_i), \quad (8)$$

$$\sum_{i=1}^n \dot{\mathbf{x}}_i = -\alpha \sum_{i=1}^n \mathbf{y}_i. \quad (9)$$

Integrating (8) with respect to t and initializing the auxiliary variable $\mathbf{y}(0) = \mathbf{0}_{nm}$, we have

$$\sum_{i=1}^n \dot{\mathbf{x}}_i = -\alpha \sum_{i=1}^n \mathbf{y}_i = -\alpha \sum_{i=1}^n \nabla f_i(\mathbf{x}_i), \quad (10)$$

which shows that the time-derivative of the sum of states \mathbf{x}_i 's is towards sum gradient. Therefore, the equilibrium ($\dot{\mathbf{x}}_i = \mathbf{0}_m$) of the dynamics (6)-(7) is \mathbf{x}^* satisfying $(\mathbf{1}_n^\top \otimes I_m) \nabla F(\mathbf{x}^*) = \mathbf{0}_m$ (I_m as the identity matrix of size m), which is the optimal state of problem (5) [10].

Lemma 4: Initializing from any $\mathbf{x}(0) \neq \mathbf{1}_n \otimes \mathbf{x}_0$, for some non-zero $\mathbf{x}_0 \in \mathbb{R}^m$, and $\mathbf{y}(0) = \mathbf{0}_{nm}$, the state $[\mathbf{x}^*; \mathbf{0}_{nm}]$ with $(\mathbf{1}_n^\top \otimes I_m) \nabla F(\mathbf{x}^*) = \mathbf{0}_m$ is an invariant equilibrium point of the dynamics (6)-(7).

Proof: From (10), the following uniquely holds at $\mathbf{x} = \mathbf{x}^* = \mathbf{1}_n \otimes \bar{\mathbf{x}}^*$,

$$\sum_{i=1}^n \dot{\mathbf{x}}_i = -\alpha (\mathbf{1}_n^\top \otimes I_m) \nabla F(\mathbf{x}^*) = \mathbf{0}_m.$$

Further, from (6) we have $\dot{\mathbf{x}}_i = \mathbf{0}_m$ and from (7),

$$\dot{\mathbf{y}}_i = \frac{d}{dt} \nabla f_i(\bar{\mathbf{x}}^*) = \nabla^2 f_i(\bar{\mathbf{x}}^*) \dot{\mathbf{x}}_i = \mathbf{0}_m,$$

which shows that $[\mathbf{x}^*; \mathbf{0}_{nm}]$ is an invariant equilibrium point of the dynamics (6)-(7). ■

Lemma 4 only shows that $[\mathbf{x}^*; \mathbf{0}_{nm}]$, with \mathbf{x}^* as the optimal point of (5), is the equilibrium of the networked dynamics (6)-(7). The first term in Eq. (6) drives the agents to reach consensus on \mathbf{x}_i 's, while the second term along with Eq. (7) implements the gradient correction [35], [36]. The pseudo-code of the proposed D-SVM is given in Algorithm 1.

Algorithm 1: The proposed D-SVM algorithm.

- 1 **Given:** data $\chi_j \in \mathbb{R}^{m-1}$, $j = 1, \dots, N$, costs $f_i(\mathbf{x}_i)$ with $\mathbf{x}_i = [\omega_i^\top; \nu_i] \in \mathbb{R}^m$ as SVM parameters, agents $i = 1, \dots, n$, WB-digraphs \mathcal{G}_q , weights W_q, A_q , switching signal q , running-time T_{end}
 - 2 **Initialization:** $\mathbf{y}_i(0) = \mathbf{0}_m$, $\mathbf{x}_i(0)$ is set randomly
 - 3 **for** $t < T_{end}$ **do**
 - 4 Every agent i finds $\nabla f_i(\mathbf{x}_i)$;
 - 5 Every agent i shares \mathbf{x}_i and \mathbf{y}_i over \mathcal{G}_q ;
 - 6 Every agent i updates \mathbf{x}_i and \mathbf{y}_i via Eq. (6)-(7);
 - 7 **Return:** \mathbf{x}^* as optimal SVM parameters ω_i^*, ν_i^* ;
-

V. PROOF OF CONVERGENCE

In this section, we show that dynamics (6)-(7) converge to the equilibrium state described in Lemma 4. Define the nm -by- nm Hessian matrix $H := \text{blockdiag}[\nabla^2 f_i(\mathbf{x}_i)]$. The dynamics (6)-(7) can be written in a compact form as

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{y}} \end{pmatrix} = M(t, \alpha, q) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \quad (11)$$

$$M(t, \alpha, q) = \begin{pmatrix} \bar{W}_q \otimes I_m & -\alpha I_{mn} \\ H(\bar{W}_q \otimes I_m) & \bar{A}_q \otimes I_m - \alpha H \end{pmatrix}. \quad (12)$$

The networked dynamics (11)-(12) represent a *hybrid dynamical system* because: (i) the matrix H varies in CT; and (ii) the structure of Laplacian matrices \bar{W}_q and \bar{A}_q may change in DT in case of *dynamic network topology*, which is motivated by robotic networks and dynamic resource availability at the agents. In this direction, \bar{W}_q and \bar{A}_q follow a switching signal q (and a *jump map*) fulfilling all the proper assumptions for stability¹; see also [37] for related work on regularity conditions on the weight matrices. In this hybrid setup, towards convergence analysis, (i) we evaluate the stability properties of the matrix M at every time-instant using the matrix perturbation theory [31]. Specifically, we show that, under Assumptions 1, the algebraic multiplicity of zero eigenvalues of M is m and the rest of eigenvalues have negative real parts. Recall that our methodology *only mandates strict convexity* (Lemma 3) in contrast to *strong convexity* condition in [20]–[26]; (ii) then, using a Lyapunov analysis, we show that the rate of convergence (decrease in Lyapunov function) depends

¹The proposed model (11) represents a “differential equation whose right-hand side is chosen from a family of functions based on a switching signal” [28]. Define the hybrid state $\zeta = ((\mathbf{x}; \mathbf{y}), q, \tau)$ with τ as the timer state and $q : t \in \mathbb{R}_{\geq 0} \rightarrow Q = \{1, 2, \dots, \bar{q}\}$ as the index of the *network topology* \mathcal{G}_q (and the Laplacians \bar{W}_q, \bar{A}_q) over a bounded time-interval. Then, the flow map is $\mathcal{F} : (\dot{\mathbf{x}}; \dot{\mathbf{y}}) = M(t, \alpha, q)(\mathbf{x}; \mathbf{y}), \dot{q} = 0, \dot{\tau} \in [0, \frac{1}{\tau_D}]$ with the flow (domain) set $\zeta \in \mathcal{C} = \mathbb{R}^{2mn} \times Q \times [0, 1]$. Then, the change in the hybrid state at each jump (known as the *jump map*) is $\mathcal{J} : (\mathbf{x}; \mathbf{y})^+ = (\mathbf{x}; \mathbf{y}), q^+ \in Q, \tau^+ = 0$ over the jump domain set $\zeta \in \mathcal{D} = \mathbb{R}^n \times Q \times \{1\}$, implying that the hybrid system jumps to a new mode $q \in Q$ whenever $\zeta \in \mathcal{D}$ with the time-interval length depending on the timer rate $\dot{\tau}$ for each mode q . For example, for minimum length time-interval τ_D , the rate is $\dot{\tau} = \frac{1}{\tau_D}$ implying that $\tau = 1$ (the jump happens) at the time τ_D . Clearly, at the jump, q switches to a new mode, τ starts over, while the state $(\mathbf{x}; \mathbf{y})$ is continuous and unchanged. Such a jump map is categorized as a piece-wise constant mapping with *finite* number of discontinuities (jumps) in each time interval and satisfies the so-called “Basic Assumption” for stability [28].

on the largest non-zero eigenvalue of M ; and, (iii) following the continuity of the Lyapunov function at the jump points, we generalize the convergence to the entire (hybrid) time horizon [28]. In the rest of this paper for notation simplicity, we drop the dependence of M on (t, α, q) and dependence of \bar{A}_q, \bar{W}_q on q , unless where needed, despite the fact that they are a function of mapping q , time t , and stepsize α .

Lemma 5: [38], [39] Let an n -by- n matrix $P(\alpha)$ depend smoothly on a real parameter $\alpha \geq 0$. Assume $P(0)$ has $l < n$ equal eigenvalues, denoted by $\lambda_1 = \dots = \lambda_l$, associated with right and left eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_l$ and $\mathbf{u}_1, \dots, \mathbf{u}_l$, which are linearly independent. Let $\lambda_i(\alpha)$ denote the eigenvalues of $P(\alpha)$, as a function of α , corresponding to $\lambda_i, i \in \{1, \dots, l\}$, and $P' = \frac{dP(\alpha)}{d\alpha}|_{\alpha=0}$. Then, $\frac{d\lambda_i}{d\alpha}|_{\alpha=0}$ is the i -th eigenvalue of the following l -by- l matrix,

$$\begin{pmatrix} \mathbf{u}_1^\top P' \mathbf{v}_1 & \dots & \mathbf{u}_1^\top P' \mathbf{v}_l \\ & \ddots & \\ \mathbf{u}_l^\top P' \mathbf{v}_1 & \dots & \mathbf{u}_l^\top P' \mathbf{v}_l \end{pmatrix}.$$

Theorem 1: Let Assumption 1 hold. For sufficiently small α , all eigenvalues of M have non-positive real-parts, $\forall t, q$, and algebraic multiplicity of zero eigenvalue is m .

Proof: Let $M = M_0 + \alpha M_1$ with

$$\begin{aligned} M_0 &= \begin{pmatrix} \bar{W} \otimes I_m & \mathbf{0}_{mn \times mn} \\ H(\bar{W} \otimes I_m) & \bar{A} \otimes I_m \end{pmatrix}, \\ M_1 &= \begin{pmatrix} \mathbf{0}_{mn \times mn} & -I_{mn} \\ \mathbf{0}_{mn \times mn} & -H \end{pmatrix}, \end{aligned}$$

where $\mathbf{0}_{mn \times mn}$ is the zero matrix of size mn . Since matrix M_0 is block (lower) triangular we have,

$$\sigma(M_0) = \sigma(\bar{W} \otimes I_m) \cup \sigma(\bar{A} \otimes I_m), \quad (13)$$

where $\sigma(\cdot)$ represents the eigenspectrum of the matrix. From Lemma 1, both matrices \bar{W} and \bar{A} have $n-1$ eigenvalues in the LHP (left-half plane) and one isolated eigenvalue at zero. Therefore, matrix M_0 has m sets of eigenvalues associated with m dimensions of vector states \mathbf{x}_i i.e.,

$$\text{Re}\{\lambda_{2n,j}\} \leq \dots \leq \text{Re}\{\lambda_{3,j}\} < \lambda_{2,j} = \lambda_{1,j} = 0,$$

where $j = \{1, \dots, m\}$. Using Lemma 5, we analyze the spectrum of M by considering it as the perturbed version of M_0 via the term αM_1 . We check the variation of the zero eigenvalues $\lambda_{1,j}$ and $\lambda_{2,j}$ by adding the (small) perturbation αM_1 . Denote these perturbed eigenvalues by $\lambda_{1,j}(\alpha)$ and $\lambda_{2,j}(\alpha)$. To apply Lemma 5, define the right eigenvectors corresponding to $\lambda_{1,j}$ and $\lambda_{2,j}$ as,

$$V = [V_1 \ V_2] = \begin{pmatrix} \mathbf{1}_n & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{1}_n \end{pmatrix} \otimes I_m, \quad (14)$$

Similarly, the left eigenvectors are V^\top . These eigenvectors are defined using Lemma 2 and satisfy $V^\top V = I_{2mn}$. Recall that, $\frac{dM(\alpha)}{d\alpha}|_{\alpha=0} = M_1$ and following Lemma 5,

$$V^\top M_1 V = \begin{pmatrix} \mathbf{0}_{m \times m} & \mathbf{0}_{m \times m} \\ -nI_m & -(\mathbf{1}_n \otimes I_m)^\top H(\mathbf{1}_n \otimes I_m) \end{pmatrix}. \quad (15)$$

Following the definition of the Hessian matrix H ,

$$-(\mathbf{1}_n \otimes I_m)^\top H(\mathbf{1}_n \otimes I_m) = -\sum_{i=1}^n \nabla^2 f_i(\mathbf{x}_i) \prec 0, \quad (16)$$

where the last inequality follows the strict convexity of the loss function (see Lemma 3). Recall that from Lemma 5 the derivatives $\frac{d\lambda_{1,j}}{d\alpha}|_{\alpha=0}$ and $\frac{d\lambda_{2,j}}{d\alpha}|_{\alpha=0}$ depend on the eigenvalues of (15), which clearly form a lower triangular matrix with m zero eigenvalues and m negative eigenvalues (following (16)). Therefore, $\frac{d\lambda_{1,j}}{d\alpha}|_{\alpha=0} = 0$ and $\frac{d\lambda_{2,j}}{d\alpha}|_{\alpha=0} < 0$, which implies that considering αM_1 as a perturbation, the m zero eigenvalues $\lambda_{2,j}(\alpha)$ of M move toward the LHP while $\lambda_{1,j}(\alpha)$'s remain zero. We recall that the eigenvalues are a continuous functions of the matrix elements [31], and therefore, for sufficiently small α we have,

$$\begin{aligned} \text{Re}\{\lambda_{2n,j}(\alpha)\} &\leq \dots \leq \text{Re}\{\lambda_{3,j}(\alpha)\} \\ &\leq \lambda_{2,j}(\alpha) < \lambda_{1,j}(\alpha) = 0, \end{aligned} \quad (17)$$

which completes the proof. \blacksquare

Theorem 1, similar to [10]–[15], only requires *strict convexity* of the loss function, as compared to strong convexity in [20]–[26]. Moreover, the matrix perturbation method allows eigen-spectrum analysis of the time-varying matrix M , including possible discrete jumps in the hybrid mode. From Theorem 1, for sufficiently small α , the matrix M has m zero eigenvalues, while all other eigenvalues remain in the LHP. In order to determine upper-bound on α ensuring the results of Theorem 1, some relevant concepts regarding the eigen-spectrum $\sigma(M_0)$ and $\sigma(M)$ are provided next. Define the optimal matching distance [40] as $d(\sigma(M), \sigma(M_0)) = \min_{\pi} \max_{1 \leq i \leq 2nm} (\lambda_i - \lambda_{\pi(i)}(\alpha))$, where $\pi(i)$ represents the i th permutation over all possible permutations $\{1, \dots, 2nm\}$. It can be verified that, $d(\sigma(M), \sigma(M_0))$ is the smallest-radius circle centered at $\lambda_{1,j}, \dots, \lambda_{2n,j}$, which includes all the eigenvalues of M denoted by $\lambda_{1,j}(\alpha), \dots, \lambda_{2n,j}(\alpha)$. Loosely speaking, $d(\sigma(M), \sigma(M_0))$ represents the farthest distance between the eigenvalues of M and M_0 . From Theorem 1, the first $2m$ eigenvalues of the perturbed matrix M are $\lambda_{1,j}(\alpha) = 0$ and $\lambda_{2,j}(\alpha) < 0$. To show that all the other $(2n-2)m$ eigenvalues $\lambda_{3,j}(\alpha), \dots, \lambda_{2n,j}(\alpha)$ remain in the LHP, it is sufficient that $d(\sigma(M), \sigma(M_0)) < \underline{\lambda}$ with $\underline{\lambda} = \min_{1 \leq j \leq m} |\text{Re}\{\lambda_{3,j}\}|$. This guarantees that the distance between the $(2n-2)m$ eigenvalues of M_0 and M is less than $\underline{\lambda}$ and therefore all the $(2n-2)m$ eigenvalues of M remain in the LHP. In this direction, the following lemma provides a useful bound on $d(\sigma(M), \sigma(M_0))$ and subsequently bound α .

Lemma 6: [40] For $M = M_0 + \alpha M_1$, we have $\forall t, q$,

$$d(\sigma(M), \sigma(M_0)) \leq 4(\|M_0\|_{\infty} + \|M\|_{\infty})^{1-\frac{1}{nm}} \|\alpha M_1\|_{\infty}^{\frac{1}{nm}}.$$

Lemma 7: Define $\gamma = \max_{1 \leq i \leq nm} \sum_{j=1}^{nm} |H_{ij}|$ and $\underline{\lambda} = \min_{1 \leq j \leq m} |\text{Re}\{\lambda_{3,j}\}|$. Then, the real-part of the eigenvalues, $\text{Re}\{\lambda_{3,j}(\alpha)\}, \dots, \text{Re}\{\lambda_{2n,j}(\alpha)\}$, is negative, if $0 < \alpha < \bar{\alpha}$ where for $\gamma < 1$,

$$\bar{\alpha} = \underset{\alpha > 0}{\text{argmin}} |4(\max\{4+4\gamma+\alpha\gamma, 4+2\gamma+\alpha\})^{1-\frac{1}{nm}} \alpha^{\frac{1}{nm}} - \underline{\lambda}|, \quad (18)$$

and for $\gamma \geq 1$,

$$\bar{\alpha} = \underset{\alpha > 0}{\text{argmin}} |4(4+4\gamma+\alpha\gamma)^{1-\frac{1}{nm}} (\alpha\gamma)^{\frac{1}{nm}} - \underline{\lambda}|. \quad (19)$$

Proof: From Assumption 1 and Lemmas 2 and 3, $\|M_0\|_\infty \leq 2(1+\gamma)$. This is because, from Assumption 1 and Lemma 3, the row sum of the absolute values of matrix \bar{W} and $H(\bar{W} \otimes I_m)$ are at most 2 and 2γ , respectively. Thus,

$$\|M\|_\infty \leq \max\{2 + \gamma(2 + \alpha), 2 + \alpha\},$$

$$\|\alpha M_1\|_\infty \leq \max\{\alpha\gamma, \alpha\}.$$

Then, for $\gamma < 1$,

$$4(2(1 + \gamma) + \max\{2 + \gamma(2 + \alpha), 2 + \alpha\})^{1 - \frac{1}{nm}} \alpha^{\frac{1}{nm}} < \underline{\lambda},$$

and for $\gamma \geq 1$,

$$4(4 + \gamma(4 + \alpha))^{1 - \frac{1}{nm}} (\alpha\gamma)^{\frac{1}{nm}} < \underline{\lambda}.$$

Since the left-hand-side of the above inequalities are monotonically increasing for $\alpha > 0$, the largest α satisfying the above inequalities is given by (18)-(19). ■

Despite the conservative upper-bound in Lemma 7, the eigenvalue condition in Theorem 1 may be valid for possible less-conservative choice of $\alpha > \bar{\alpha}$. For a proper α , matrix M has m zero eigenvalues associated with the eigenvectors V_1 in (14), and the null space of the time-varying matrix M , $\mathcal{N}(M) = \text{span}\{[1_n; \mathbf{0}_n] \otimes I_m\}$, is independent of time.

Theorem 2: Let the conditions in Lemma 4, Lemma 7, and Theorem 1 hold. The proposed dynamics (6)-(7) converges to $[\mathbf{x}^*; \mathbf{0}_{nm}]$ with \mathbf{x}^* as the optimal solution of problem (5).

Proof: Consider the positive-definite Lyapunov function $V(\delta) = \frac{1}{2} \delta^\top \delta = \frac{1}{2} \|\delta\|_2^2$ with $\delta = [\mathbf{x}; \mathbf{y}] - [\mathbf{x}^*; \mathbf{0}_{mn}] \in \mathbb{R}^{2mn}$. Since, from Lemma 4, $[\mathbf{x}^*; \mathbf{0}_{nm}]$ is an invariant state of the dynamics (11)-(12), we have $\dot{\delta} = [\dot{\mathbf{x}}; \dot{\mathbf{y}}] - [\dot{\mathbf{x}}^*; \mathbf{0}_{mn}] = M([\mathbf{x}; \mathbf{y}] - [\mathbf{x}^*; \mathbf{0}_{mn}]) = M\delta$, where $M[\mathbf{x}; \mathbf{0}_{mn}] = \mathbf{0}_{2mn}$. Then, the time-derivative of the proposed Lyapunov function is $\dot{V} = \delta^\top \dot{\delta} = \delta^\top M\delta$. Following Theorem 1, recall that $\lambda_{1,j}(\alpha) = 0$, while the remaining eigenvalues have negative real parts, i.e., $\text{Re}\{\lambda_{i,j}(\alpha)\} < 0$, for $2 \leq i \leq 2n, 1 \leq j \leq m$. It is known that [32],

$$\delta^\top M\delta \leq \max_{1 \leq j \leq m} \text{Re}\{\lambda_{2,j}(\alpha)\} \delta^\top \delta. \quad (20)$$

Since M varies in time, $\max_{1 \leq j \leq m} \text{Re}\{\lambda_{2,j}(\alpha)\}$ also changes in time. However, from Theorem 1, it is always negative, implying that $\dot{V} < 0$ for $\delta \neq \mathbf{0}_{2mn}$, while V remains continuous at the jump (switching) points. We thus have $\dot{V} = 0 \Leftrightarrow \delta = \mathbf{0}_{2mn}$ and, from LaSalle's invariance principle, convergence to the invariant set $\{\delta = \mathbf{0}_{2mn}\}$ follows [28]. ■

From (20), the convergence rate of the dynamics (11)-(12) depends on $\text{Re}\{\lambda_{2,j}(\alpha)\}$ and the parameter α . Therefore, to improve the convergence rate, α needs not to be very small.

VI. SIMULATION: NONLINEAR SVM EXAMPLE

We consider the example given in [41] with $N = 6000$ uniformly distributed sample data points in Fig. 1 (Left), represented in two classes: blue *'s and red o's. Clearly, these points $\chi_i = [\chi_i(1); \chi_i(2)]$ are not linearly separable in \mathbb{R}^2 . The nonlinear mapping $\phi(\chi_i) = [\chi_i(1)^2; \chi_i(2)^2; \sqrt{2}\chi_i(1)\chi_i(2)]$, proposed by [41], properly maps the data to \mathbb{R}^3 such that the projected points are linearly separable (see Fig. 1 (Right)) with the kernel function

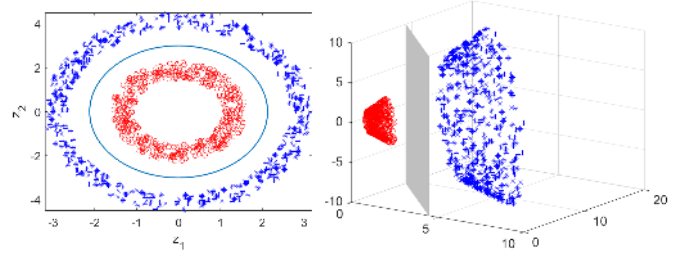


Fig. 1. (Left) Training data and the optimal nonlinear classifier (the ellipse) in 2D. (Right) The same points mapped into 3D space via a nonlinear mapping. Linear SVM optimally classifies the data points via the gray hyperplane which represents the ellipse in the left figure by inverse mapping.

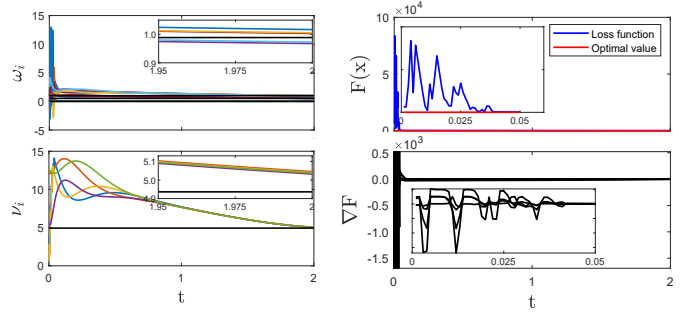


Fig. 2. The time-evolution of the SVM classifier parameters ω_i and ν_i (at all 5 agents) under dynamics (11)-(12) along with overall loss function $F(\mathbf{x})$ and sum of the gradients $\sum_{i=1}^5 \nabla f_i(\mathbf{x}_i)$. The optimal values based on the centralized SVM are also shown for comparison.

$K(\chi_i, \chi_j) = (\phi(\chi_i)^\top \phi(\chi_j))^2$. We evaluate the proposed dynamics (11)-(12) (with $\alpha = 10$) for D-SVM over a network \mathcal{G}_q of $n = 5$ agents considered as the union of a cycle and a 2-hop digraph (as in [2]) with weight-balanced links. Using the loss function (2)-(4) with $\mu = 3$ and $C = 1.5$, every agent finds the optimal hyperplane parameters $\mathbf{x}_i = [\omega_i^\top; \nu_i]$ ($\omega_i \in \mathbb{R}^3$) and shares \mathbf{x}_i along with the auxiliary variable \mathbf{y}_i over \mathcal{G}_q . Using MATLAB's `randperm`, the node's permutation is randomly changed every 0.05 sec to simulate a dynamic network with switching signal $q: t \rightarrow Q = \{1, 2, \dots, N!\}$ and timer rate $\dot{\tau} = \frac{1}{0.05} = 20$. The time-evolution of $\mathbf{x}_i = [\omega_i^\top; \nu_i] \in \mathbb{R}^4$, loss function $F(\mathbf{x})$, and sum of the gradients $\sum_{i=1}^5 \nabla f_i(\mathbf{x}_i) \in \mathbb{R}^4$ are shown in Fig. 2. The agents reach consensus on the optimal value $\bar{\mathbf{x}}^* = [\bar{\omega}(1), \bar{\omega}(2), \bar{\omega}(3), \bar{\nu}]^\top$, which represents the separating ellipse $\bar{\omega}(1)z_1^2 + \bar{\omega}(2)z_2^2 - \bar{\nu} = 0$ (z_1 and z_2 as the Cartesian coordinates in \mathbb{R}^2). For comparison, similar D-SVM solutions under finite-time [13] (with $\beta_{ij} = 3$) and fixed-time [11] dynamics (with $\alpha = 4$, $\beta = \gamma = 1$, $a = 2$, $b = 9$) are shown in Fig. 3. Recall that finite/fixed-time dynamics are non-Lipschitz and result in undesirable chattering of the SVM parameters in steady-state.

VII. CONCLUSION AND FUTURE RESEARCH

In this work, a Lipschitz dynamics is proposed to solve D-SVM over a dynamic WB-digraph in a hybrid setting using matrix perturbation analysis. Our CT results can be easily extended to the DT case by adopting, for example, approximate Euler-Forward discretization and replacing matrix M in (12)

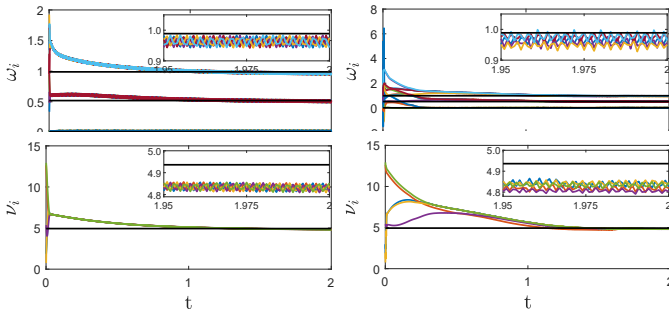


Fig. 3. Time-evolution of ω_i and ν_i under (Left) finite-time [13] and (Right) fixed-time [11] dynamics chatter around the optimal value in steady-state due to non-Lipschitz dynamics.

with $M_d = I + TM$, where T is the sampling time. Then, the explicit upper bound on T such that stable CT dynamics from Theorems 1-2 remains stable after discretization can be defined. On the other hand, implicit discretizations, e.g., Euler-Backward, impose no upperbound on T , but they are more time-consuming and harder to implement. As future directions, extensions to *time-delayed* networks, *online* D-SVM, and *sparse* digraphs are of interest.

REFERENCES

- [1] R. Xin, S. Kar, and U. A. Khan, "Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 102–113, 2020.
- [2] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1869–1889, 2020.
- [3] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 2032–2048, 2020.
- [4] M. Doostmohammadian, A. Aghasi, and T. Charalambous, "Fast-convergent dynamics for distributed resource allocation over sparse time-varying networks," *arXiv preprint arXiv:2012.08181*, 2020.
- [5] Y. Lu, V. Roychowdhury, and L. Vandenbergh, "Distributed parallel support vector machines in strongly connected networks," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1167–1178, 2008.
- [6] E. Y. Chang, K. Zhu, H. Wang, and H. Bai, "Psvm: Parallelizing support vector machines on distributed computers," in *Foundations of Large-Scale Multimedia Information Management and Retrieval*, pp. 213–230. Springer, 2011.
- [7] A. Navia-Vázquez, D. Gutierrez-Gonzalez, E. Parrado-Hernández, and J. J. Navarro-Abellan, "Distributed support vector machines," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 1091, 2006.
- [8] A. Bordes, S. Ertekin, J. Weston, L. Botton, and N. Cristianini, "Fast kernel classifiers with online and active learning," *Journal of Machine Learning Research*, vol. 6, no. 9, 2005.
- [9] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, no. 5, 2010.
- [10] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014.
- [11] B. Ning, Q. Han, and Z. Zuo, "Distributed optimization for multiagent systems: An edge-based fixed-time consensus approach," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 122–132, 2017.
- [12] K. Garg, M. Baranwal, A. O. Hero, and D. Panagou, "Fixed-time distributed optimization under time-varying communication topology," *arXiv preprint arXiv:1905.10472*, 2019.
- [13] S. Rahili and W. Ren, "Distributed continuous-time convex optimization with time-varying cost functions," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1590–1605, 2017.
- [14] M. Doostmohammadian, "Single-bit consensus with finite-time convergence: Theory and applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 3332–3338, 2020.
- [15] Z. Li and Z. Ding, "Time-varying multi-objective optimisation over switching graphs via fixed-time consensus algorithms," *International Journal of Systems Science*, vol. 51, no. 15, pp. 2793–2806, 2020.
- [16] P. Armand and R. Omhenni, "A globally and quadratically convergent primal-dual augmented lagrangian algorithm for equality constrained optimization," *Optimization Methods and Software*, vol. 32, no. 1, pp. 1–21, 2017.
- [17] P. Srivastava and J. Cortés, "Distributed algorithm via continuously differentiable exact penalty method for network optimization," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 975–980.
- [18] F. Mansoori and E. Wei, "A fast distributed asynchronous newton-based optimization algorithm," *IEEE Transactions on Automatic Control*, vol. 65, no. 7, pp. 2769–2784, 2019.
- [19] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.
- [20] B. Van Scoy and L. Lessard, "A distributed optimization algorithm over time-varying graphs with efficient gradient evaluations," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 357–362, 2019.
- [21] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 417–428, 2015.
- [22] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1185–1197, 2013.
- [23] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.
- [24] Y. Sun, A. Daneshmand, and G. Scutari, "Convergence rate of distributed optimization algorithms based on gradient tracking," *arXiv preprint arXiv:1905.02637*, 2019.
- [25] A. Simonetto, A. Koppel, A. Mokhtari, G. Leus, and A. Ribeiro, "Decentralized prediction-correction methods for networked time-varying convex optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5724–5738, 2017.
- [26] A. Koppel, S. Paternain, C. Richard, and A. Ribeiro, "Decentralized online learning with kernels," *IEEE Transactions on Signal Processing*, vol. 66, no. 12, pp. 3240–3255, 2018.
- [27] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.
- [28] R. Goebel, R. Sanfelice, and A. Teel, "Hybrid dynamical systems," *IEEE control systems magazine*, vol. 29, no. 2, pp. 28–93, 2009.
- [29] D. L. Ly and H. Lipson, "Learning symbolic representations of hybrid dynamical systems," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3585–3618, 2012.
- [30] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.
- [31] G. W. Stewart and J. Sun, "Matrix perturbation theory," 1990.
- [32] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.
- [33] O. Chapelle, "Training a support vector machine in the primal," *Neural computation*, vol. 19, no. 5, pp. 1155–1178, 2007.
- [34] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Prentice Hall, 2020.
- [35] P. D. Lorenzo and G. Scutari, "NEXT: in-network nonconvex optimization," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.
- [36] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: accelerated distributed directed optimization," *IEEE Trans. on Autom. Control*, vol. 63, no. 5, pp. 1329–1339, 2017.
- [37] S. Safavi and U. A. Khan, "Asymptotic stability of stochastic ltv systems with applications to distributed dynamic fusion," *IEEE Trans. on Automatic Control*, vol. 62, no. 11, pp. 5888–5893, 2017.
- [38] A. P. Seyranian and A. A. Mailybaev, *Multiparameter stability theory with mechanical applications*, vol. 13, World Scientific, 2003.
- [39] K. Cai and H. Ishii, "Average consensus on general strongly connected digraphs," *Automatica*, vol. 48, no. 11, pp. 2750–2761, 2012.
- [40] R. Bhatia, *Matrix analysis*, Springer Science & Business Media, 2013.
- [41] S. Russell and P. Norvig, *Artificial intelligence*, Prentice Hall, 2010.