Outliers Detection and Correction for Cooperative Distributed Online Learning in Wireless Sensor Network

Minh N. H. Nguyen, Chuan Pham, Nguyen H. Tran, and Choong Seon Hong,

Department of Computer Science and Engineering, Kyung Hee University, Korea,
Email: {minhnh, pchuan, nguyenth, cshong}@khu.ac.kr.

***Abstract*—The recent distributed online convex optimization framework has developed in Wireless sensor networks (WSN) provide the promising approach for solving approximately stochastic optimization problem over network of sensors follows distributed manner. In practice, most of real environmental sensing activities are highly dynamic where noisy sensory information often appears and affects to the learning performance. However, the original distributed saddle point (DSPA) algorithm is lack of considering about the consequence of falsification in online learning. Based on the simulation observations conducted in this paper, we figure out the fluctuation and the slow convergence rate leads to overall prediction performance reduction of distributed online least square problem. Therefore, we propose an integrated outliers detection, correction mechanism in order to stabilize prediction and improve convergence rate.**

I. Introduction

Wireless sensor networks research has a long history and takes an important role in remote environmental monitoring and target tracking. In these days, Internet of Thing (IoT) and sensor devices that are smaller, cheaper, and intelligent have widely deployed in many new smart applications like smart home, smart city [1]. In fact, individual sensor nodes normally have limited coverage and can only monitor in a small region. While network of sensors proceed collaborated sensing task that can help to increase learning coverage and performance. However, the cooperative strategy does not always help sensor nodes achieve better decisions from the additional information of their neighbors, the sensor nodes are also strongly influenced by noisy information.

In this work, we consider the dynamic environmental sensing tasks where un-predetermined location sensor nodes and target. Therefore, the sensory information can be fluctuation and noisy when these nodes are far away the target location. Centralized mechanism typically requires a centralized server to collect sensory information from all sensors node and perform centralized decisions from collected information. For that reason, it limits the scalable deployment of network of sensors in the unknown environments. Therefore, we are interested in distributed sensing and prediction application over network of sensor nodes. Each sensor node will sense by itself and exchange information with neighbor to improve learning result and make better prediction. This distributed online learning mechanism are well studied in the paper [2]. Moreover, by using the consensus and proximity constraints [2], [3], sensor nodes are incentive to cooperate and make similar prediction results due to they are sensing the same target in the same region. Therefore, some falsification predictions from neighbors affect not only its own prediction but also the other nodes.

In this paper, first, we conduct numerical analysis experiments on the existing DSPA algorithm [2] to show the affect of outliers in cooperative strategy. The algorithm easily faces to significant low accuracy and fluctuation prediction results due to consensus constraint between nodes. Second, we propose the mechanism to detect outliers and correct the prediction by the Outliers Correction Distributed Saddle Point Algorithm (OC-DSPA) in distributed least square problem with two proximity functions. Then the new algorithm shows the stability and faster convergence in learning performance and un-predetermined location target.

The rest of the paper is organized as follows. Related work and background discuss about algorithm DSPA are given in Section II. The formulation distributed online least square problem in Section III. We figure out the affection of outliers and propose the algorithm OC-DSPA in Section IV. Numerical results on the synthetic dataset are given in Section V and conclusions are given in Section VI.

II. Related Work & Background

Most of the works in distributed learning and optimization over graph focus on offline setting that process a full dataset such as alternating direction method of multipliers (ADMM) over network lasso regularization [4], EXTRA algorithm [5]. However, lack of consideration in real time adaptation with the online setting for each agent repeatedly observe and make prediction. In the recent works, distributed saddle point learning mechanism is applied in several machine learning

"This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (B0190-16-2017, Resilient/Fault-Tolerant Autonomic Networking Based on Physicality, Relationship and Service Semantic of IoT Devices)." *Dr. CS Hong is the corresponding author.

problems via distributed online learning mechanism [2], [3], [6]. This approach requires a cheap computation and can be applied in fully distributed manner for all sensor nodes.

Distributed Saddle Point Learning: As a typical setting for distributed optimization over graph of communication, the network of sensors is represented in graph that consists a set of sensor nodes and communication links $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. The set of neighbors of node i is denoted by $n_i = \{j : (i,j) \in \mathbf{E}\}$. The symmetric communication link shows exchange ability of neighbor sensors. The symmetric link between i and j show the neighborhood and they can communicate to share the learning information with each other.

The local convex loss function for each node is denoted by $f_i : R^p \times \Theta_i \rightarrow R$ to measure the different cost between the estimator with the true observation θ_i. For the uncooperative learning, each node decides by itself without cooperation. The local estimator as follows

$$x_i^L = \operatorname*{argmin}_{x_i \in \mathbf{X}} E_{\theta_i}[f_i(x_i, \theta_i)]. \quad (1)$$

And the aggregation problem over N sensor nodes becomes

$$\mathbf{x}^L = \operatorname*{argmin}_{x \in \mathbf{X_i}} \sum_{i=1}^{N} E_{\theta_i}[f_i(x_i, \theta_i)]. \quad (2)$$

For cooperative learning, the optimization problem (2) needs to add consensus constraints $x_i = x_j$ if node i and j are neighbors. The consensus constraints encourage sensors to share the very similar decision variables x. Specifically, if the communication graph is a connected graph, the expected decisions of all nodes are similar to each other.

In the other work [3], instead of using the consensus constraint, the author propose the proximity constraint (3) to keep neighbor nodes have close decisions rather than similar decisions.

$$h_i(x_i, x_j) \leq \gamma_{ij}, \quad \forall j \in n_i. \quad (3)$$

The proximity constraints still encourage the cooperative learning with small different acceptance in decisions. The proximity function $h(x_i, x_j)$ between node i and j can be measured as norm between two decision variables x_i, x_j and bounded by threshold γ_{ij}. As numerical simulations, DSPA algorithm using the proximity constraints showed the lower intermediate losses in each decision round. In the later analysis, we demonstrate the affection of choosing different proximity functions and threshold values for outliers detection context.

The problem is in form of stochastic optimization over observations where observations are assumed to follow some unknown independent distributions. Then, Lagrangian functions are approximated to instantaneous Lagrangian for each time slot and apply distributed stochastic saddle point algorithm [2]. The algorithm consists of two update steps for each observation that are *primal update* uses gradient descent and *dual update* uses gradient ascent. Under theoretical assumptions, the online algorithm can converge to the saddle point which are the solution of stochastic optimization problem.

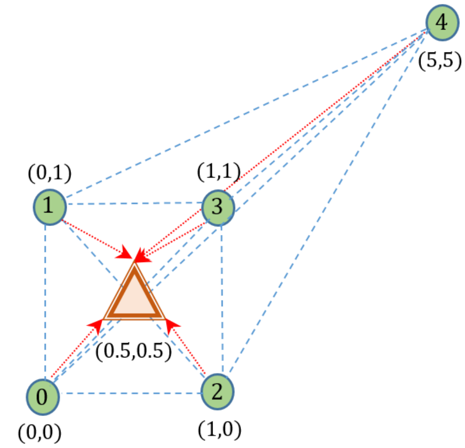


Fig. 1: WSN deployment topology with 5 sensor nodes that are sensing the same target; node 4 is the outliers.

III. Distributed online least square problem

A. Problem Formulation

The least mean square problems are widely used in regression applications to find the parameters for minimizing the error between estimator values and observation values. Signal observations in Gaussian model typically formulate as affine function $\hat{\theta}_i = H_i x_i + w_i$. An agent need to determine the signal x_i that is added with white noise $w_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ from sensing. The loss function of sensor node i is defined as

$$f_i(x_i, \theta_i) = \|\hat{\theta}_i - \theta_i\|^2 = \|H_i x_i - \theta_i\|^2$$

Consider proximity function is L2-norm, the problem becomes

$$x^* := \operatorname*{argmin}_{x \in R^{Np}} \sum_{i=1}^{N} E_{\theta_i} \|H_i x_i - \theta_i\|^2$$
$$s.t. \ \|x_i - x_j\| \leq \gamma_{ij}, \quad \forall j \in n_i \quad (4)$$

Consider proximity function is L2-norm square, the problem becomes

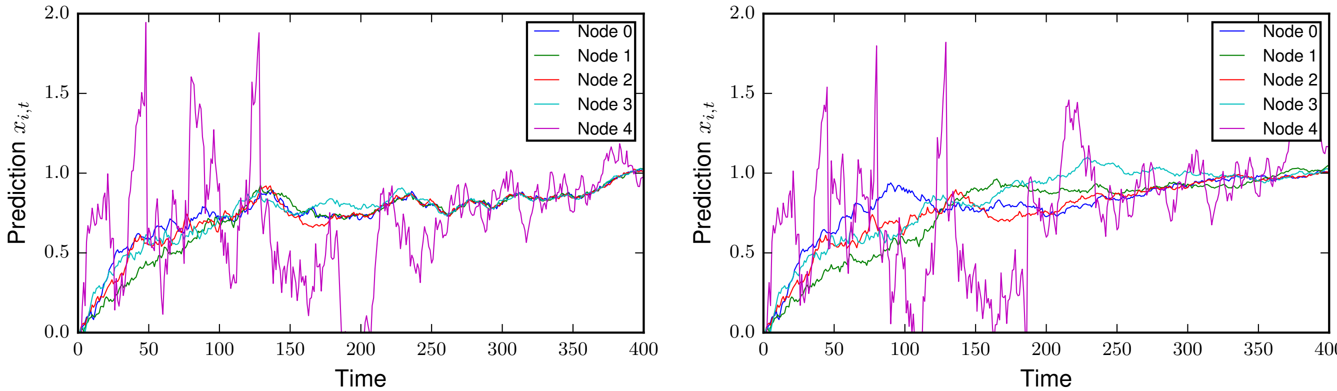
$$x^* := \operatorname*{argmin}_{x \in R^{Np}} \sum_{i=1}^{N} E_{\theta_i} \|H_i x_i - \theta_i\|^2$$
$$s.t. \ \frac{1}{2}\|x_i - x_j\|^2 \leq \gamma_{ij}, \quad \forall j \in n_i \quad (5)$$

Lagrangian functions for the optimization problems as follows:

$$\mathcal{L}(x, \lambda) = \sum_{i=1}^{N} E_{\theta_i} \|H_i x_i - \theta_i\|^2 + \sum_{i=1}^{N} \sum_{j \in n_i} \lambda_{ij} (\|x_i - x_j\| - \gamma_{ij}) \quad (6)$$

$$\mathcal{L}(x, \lambda) = \sum_{i=1}^{N} E_{\theta_i} \|H_i x_i - \theta_i\|^2 + \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in n_i} \lambda_{ij} (\|x_i - x_j\|^2 - \gamma_{ij}) \quad (7)$$

The stochastic Lagrangian functions (6), (7) are approximated to instantaneous Lagrangian functions (8), (9). At each time



(a) L2-norm proximity function

(b) L2-norm square proximity function

Fig. 2: The affection of outliers in DSPA learning with two proximity functions

slot t agent i observes a realization $\theta_{i,t}$ of random variable θ_i and make a prediction x_i.

$$\hat{\mathcal{L}}_t(x,\lambda) = \sum_{i=1}^{N} \|H_i x_i - \theta_{i,t}\|^2 + \sum_{i=1}^{N} \sum_{j\in n_i} \lambda_{ij}(\|x_i - x_j\| - \gamma_{ij}) \tag{8}$$

$$\hat{\mathcal{L}}_t(x,\lambda) = \sum_{i=1}^{N} \|H_i x_i - \theta_{i,t}\|^2 + \frac{1}{2} \sum_{i=1}^{N} \sum_{j\in n_i} \lambda_{ij}(\|x_i - x_j\|^2 - \gamma_{ij}) \tag{9}$$

B. Distributed Saddle Point Algorithms

The stochastic saddle point algorithm provides a stochastic approximation for the stochastic optimization problem. The algorithm uses alternating primal and dual stochastic gradient descent which has primal update step and dual update step as follows:

$$x_{t+1} = \mathcal{P}_X[x_t - \epsilon_t \Delta_x \hat{\mathcal{L}}_t(x_t, \lambda_t)], \text{(Primal update)} \tag{10}$$

$$\lambda_{t+1} = \mathcal{P}_\Lambda[\lambda_t + \epsilon_t \Delta_\lambda \hat{\mathcal{L}}_t(x_{t+1}, \lambda_t)], \text{(Dual update)} \tag{11}$$

We derive the gradients of the instantaneous Lagrangian function (8) and get update steps as follows:

$$x_{i,t+1} = x_{i,t} - \epsilon_t\big(2H_{i,t}^T(H_{i,t}x_{i,t} - \theta_{i,t}) + \sum_{j\in n_i} \lambda_{ij,t} \frac{x_{i,t} - x_{j,t}}{\|x_{i,t} - x_{j,t}\|}\big) \tag{12}$$

$$\lambda_{ij,t+1} = \mathcal{P}_{\Lambda_{ij}}\left[\lambda_{ij,t} + \epsilon_t(\|x_{i,t+1} - x_{j,t+1}\| - \gamma_{ij})\right]. \tag{13}$$

We derive the gradients of the instantaneous Lagrangian function (9) and get update steps as follows:

$$x_{i,t+1} = x_{i,t} - \epsilon_t\big(2H_{i,t}^T(H_{i,t}x_{i,t} - \theta_{i,t}) + \sum_{j\in n_i} \lambda_{ij,t}(x_{i,t} - x_{j,t})\big), \tag{14}$$

$$\lambda_{ij,t+1} = \mathcal{P}_{\Lambda_{ij}}\left[\lambda_{ij,t} + \frac{1}{2}\epsilon_t(\|x_{i,t+1} - x_{j,t+1}\|^2 - \gamma_{ij})\right]. \tag{15}$$

After equation derivations, the distributed learning process is showed in the DSPA algorithm for each agent i, as Alg. 1.

Algorithm 1 Distributed Saddle Point Algorithm (DSPA) [2]

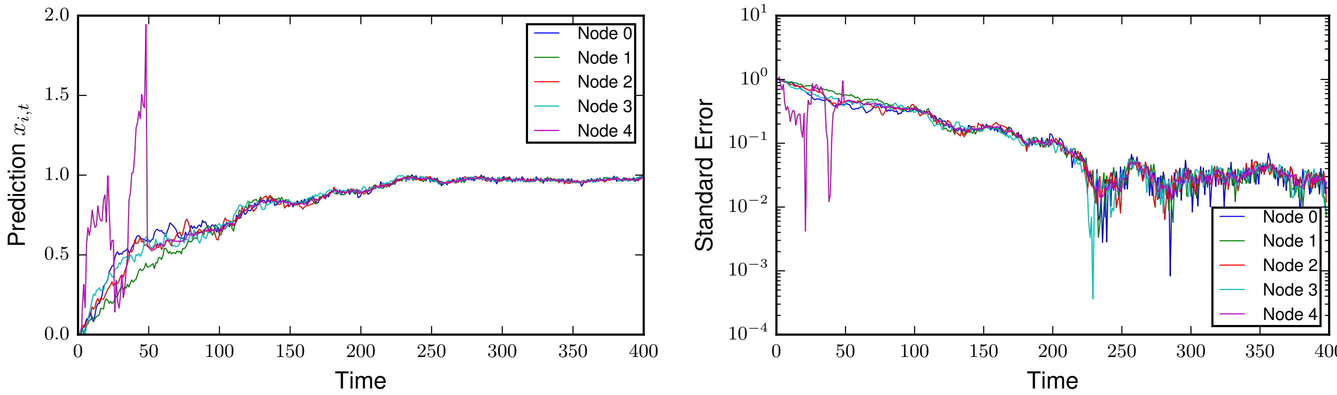
Initialization: Initialize $x_{i,0}$ and $\lambda_{i,0}$
for $t = 1, 2, \ldots, T$ **do**
  Send dual variables $\lambda_{ij,t}$ to neighbors $j \in n_i$ and receive $\lambda_{ik,t}$
  Perform **Primal Update** to predict $x_{i,t+1}$ (12) or (14)
  Send primal variables $x_{i,t+1}$ to neighbors $j \in n_i$ and receive $x_{j,t+1}$
  Perform **Dual Update** to calculate $\lambda_{ij,t+1}$ (13) or (15)
end for

IV. Outliers Detection and Correction Scheme

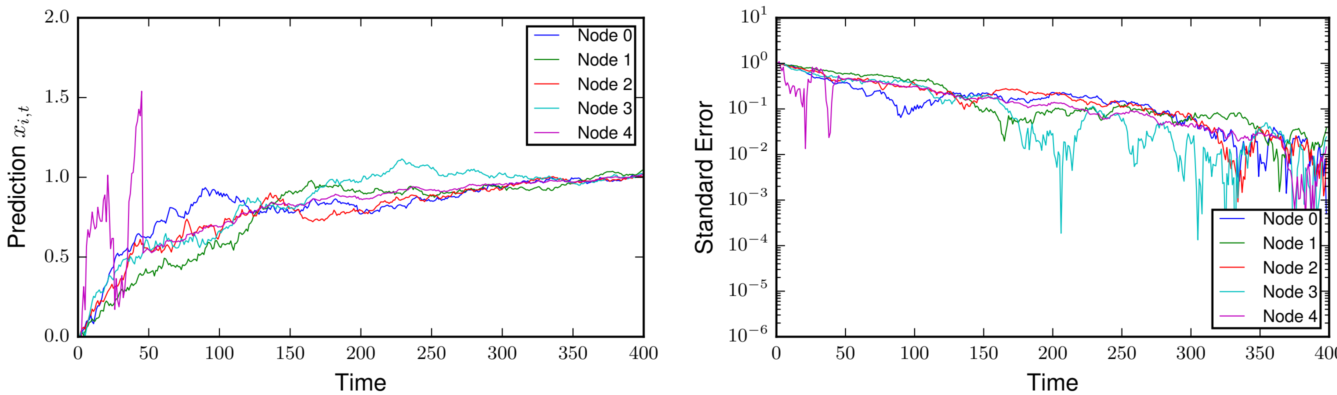
A. Outliers Affect Observation

Simulation Setting: We construct a simulation scenario described in Fig. 1. In the scenario, all sensor nodes are sensing the same target signal within a region via the red lines. While they can communicate with each other and share the learning information via communication links which are the blue lines. Locations of these sensor nodes and target denote by pairs of coordinates. Each random observation at each agent are affected by independent Gaussian noise w_i. The true signal is 1, while all sensors only can get observations θ_i and make prediction x_i. The noise assumes to be linearly increase by the distance between the sensors and the target location. In fact, we assume that the higher distance yields to the higher variance in sensing. In the scenario, the outlier node is node 4 due to it is far from the remaining sensor nodes and target. Then node 4 observation random variable is high variance compare to the others.

Observation: To examine the learning performance at each node, we apply DSPA learning algorithm with two different proximity constraints such as (4) and (5). Fig. 2 illustrates a simulation realization for both proximity constraints DSPA learning. After 400 observations, except node 4, the remaining nodes predict close to the true signal. Although the variance of prediction signal in node 4 has a tendency to reduce by time



(a) L2-norm proximity function.



(b) L2-norm square proximity function.

Fig. 3: Learning performance OC-DSPA learning.

Algorithm 2 Outliers Correction - Distributed Saddle Point Algorithm (OC-DSPA)

Initialization: Initialize $x_{i,0}$, $\lambda_{i,0}$ and $w_{i,0}$
for $t = 1, 2, \ldots, T$ **do**
  Send dual variables $\lambda_{ij,t}$ to neighbors $j \in n_i$ and receive $\lambda_{ik,t}$
  if $\|x_{i,t} - \bar{x}_{n_i,t}\| \leq \beta$ and $w_{i,t} < \underline{w}$ **then**
    $w_{i,t+1} = \max(0, w_{i,t} - 1)$
    Process **Primal Update** to predict $x_{i,t+1}$
  else
    $w_{i,t+1} = \min(\bar{w}, w_{i,t} + 1)$
    $x_{i,t+1} = \bar{x}_{n_i,t}$
  end if
  Send primal variables $x_{i,t+1}$ to neighbors $j \in n_i$ and receive $x_{j,t+1}$
  Process **Dual Update** to calculate $\lambda_{ij,t+1}$
end for

because of proximity constraint, it still does not guarantee the convergence of node 4 to the true signal. Another downside aspect is the scarification intermediate predictions of the other sensor nodes. Therefore, the overall learning performance is low and variance.

B. Outliers Correction - Distributed Saddle Point Algorithm

As we see in the previous section, the DSPA algorithm with both proximity constraints are sensitive when network have outlier nodes. The noisy sensory information affect to the convergence rate of overall learning performance. Each node communicates and gets the prediction from its neighbors then noisy predictions can make unpredictable effects such as fluctuation, very slow convergence and convergence in wrong predictions. In practice, outlier nodes can often appear when we deploy with in arbitrary location of sensor nodes or moving target. In order to detect outlier sensor nodes, we take observe the difference between each node 's prediction with all neighbor 's predictions. As expected, the predictions are primal variables should be close to their neighbors. Then we make criteria for updating the prediction values as

$$\left\|x_{i,t} - \bar{x}_{n_i,t}\right\| \leq \beta, \qquad (16)$$

where $\bar{x}_{n_i,t} = \frac{1}{|n_i|}\sum_{j \in n_i} x_{j,t}$.

Moreover, we use a counter for outliers detection and correction. The counter accounts for the continuously deviate

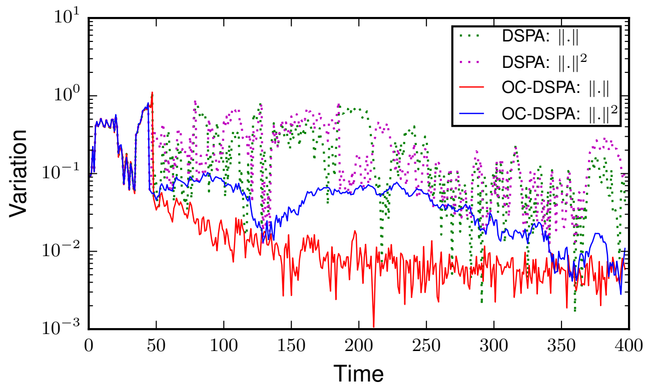


Fig. 4: Variation tracking of DSPA and OC-DSPA algorithm using L2-norm and L2-norm square proximity function.

from the average of neighbor predictions $\bar{x}_{n_i,t}$. When the criteria is hold, each agent reduces its counter $w_{i,t}$; otherwise it increases the counter by 1 and uses the average value of neighbor predictions for the prediction. Using the average values are considered as the error correction and only happen when sensor counters exceed the threshold $\underline{w}$. The counters are limited by the maximum value $\bar{w}$. Finally, the outliers detection and correction mechanism is integrated with DSPA into algorithm OC-DSPA, as Alg. 2.

V. Performance Evaluation

The similar settings from outliers observation simulation settings and the same data generation are used to compare the learning performance with DSPA learning. The prediction values of all sensor nodes converge faster than DSPA to the true signal. Specifically, it takes less than 250 iterations in Fig. 3a and 350 iterations in Fig. 3b to guarantee all predictions are around value 1. After initial fluctuations of Node 4, the variances are controlled by outliers detection and correction scheme that force Node 4 to use average predictions from its neighbors. Besides, L2-norm achieves faster convergence than L2-norm square and encourages all nodes share very similar solutions. L2-norm square still accept little differences between predictions. Notice that, we normalize the threshold γ_{ij} to the equivalent values and the same setting for both. We also take the standard error between prediction values and true value to observe the difference which is denoted as $\|x_{i,t} - x^*\|$. The standard errors obviously shows the higher accuracy, faster convergence when using L2-norm proximity constraint rather than L2-norm square.

Finally, we plot a variation plot to compare the divergence of node 4 with its neighbors, in Fig. 4. The variation of solution $x_{i,t}$ is denote as

$$\frac{1}{N}\sum_{j=1}^{N}\frac{\|x_{i,t} - x_{j,t}\|}{\|x^*\|}.$$

As the results in Fig. 4, OC-DSPA shows very small variation of node 4 solution compare to DSPA. After 50 iterations, the variations of OC-DSPA are less than 0.1 due to node 4 is detected as the outlier node. In addition, OC-DSPA algorithm with L2-norm are more stable than using L2-norm square proximity function. The main reason of the different in stability is L2-norm force all nodes to have very close solutions while L2-norm square accepts small differences in results.

VI. Conclusions & Future Work

In conclusion, we introduce outliers detection and correction mechanism for distributed saddle point algorithm that is a distributed online cooperative learning approach in WSN application. Each node measures the difference between its prediction and average neighbors predictions to detect outliers by the deviation criteria. Then, the average of neighbor predictions are used to correct outliers prediction. Therefore, the advantages of OC-DSPA algorithm are stability and faster convergence compare to the original DSPA algorithm. In addition, two proximity functions are analyzed that also affect to distributed learning performance.

In this work, we do not consider about the mobility of sensor nodes and target. Therefore, in the future work, we advocate to analyze and deal with the moving target and moving sensor nodes. The typical systems are multi-robotic systems such as swarm robotics [7]. These scenarios are more dynamic and unaware of solutions for distributed online learning problem.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[3] A. Koppel, B. M. Sadler, and A. Ribeiro, "Proximity without consensus in online multi-agent optimization," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 3726–3730.

[4] D. Hallac, J. Leskovec, and S. Boyd, "Network lasso: Clustering and optimization in large graphs," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 387–396.

[5] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[6] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "D4l: Decentralized dynamic discriminative dictionary learning," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 2966–2973.

[7] I. Navarro and F. Matía, "An introduction to swarm robotics," *ISRN Robotics*, vol. 2013, 2012.