EDGE CONSENSUS COMPUTING FOR HETEROGENEOUS DATA SETS

Kenta Niwa,[1,2] *Guoqiang Zhang*[3] *and W. Bastiaan Kleijn*[2]

[1]NTT Media Intelligence Laboratories, Japan
[2]Victoria University of Wellington, New Zealand
[3]University of Technology Sydney, Australia

ABSTRACT

Edge consensus computing is a framework to optimize a cost function when distributed nodes have distinct data sets available to them. The primal-dual method of multipliers (PDMM) is an optimization algorithm that forms a consensus among nodes by exchanging latent variables rather than the data sets. PDMM often has a high rate of convergence. However, when the nodes see statistically data sets then the performance of PDMM degrades. To overcome this problem, we propose *quadratic PDMM*. In this method, the original cost functions are replaced by their quadratic majorization based on the L2 norm to ensure homogeneous convexity among nodes. We describe a method to set its parameters optimally for fast convergence. Our experiments confirm that the proposed quadratic PDMM provides good performance even when the data sets are heterogeneous.

Index Terms— Edge consensus computing, convex optimization, monotone operator splitting, primal-dual method of multipliers (PDMM)

1. INTRODUCTION

Machine-learning based variable optimization is commonly used in practical applications such as image classification and speech recognition. In many cases, the optimization procedure uses sensor-captured data that are made available to one or more centralized (co-located) processing units. However, it is not always possible to make the data sets available to centralized units when the scale of the data set is very large or the processing units (network nodes) are dispersed over wide areas. When data sets are collected in spatially distributed nodes, it is natural to perform the optimization by exchanging latent variables among the nodes rather than the data sets themselves. This approach can be considered an edge computing procedure [1, 2], and, hence, we call this framework *edge consensus computing*. The goal of this study is to construct a practical edge consensus computing paradigm that (i) is robust to statistical heterogeneity in the data sets available to the nodes and (ii) facilitates low transmission rates between the nodes.

Several algorithms address the edge consensus computing problem for the case that the cost function is convex, including the distributed alternating direction method of multipliers (ADMM) [3, 4, 5, 6, 7] and PDMM [8, 9, 10]. In distributed ADMM, the *primal variables*, which are explicit in the cost function, are exchanged among the nodes. The variables are updated so as to minimize the convex cost while linearly updating its *dual variables* (e.g. [11, 12]), which is used to control differences among nodes with respect to the variables. However, the convergence rate is often relatively slow because it is based on Douglas-Rachford splitting [13, 14] as remarked in [10]. In contrast, PDMM facilitates fast convergence because the constraints on the dual variables are represented by convex form and it is based on Peaceman-Rachford splitting [15, 14, 16] as remarked in [10]. Although the effectiveness of PDMM was shown in experiments, we found that it is a vulnerability to statistical heterogeneity in the data observed by the nodes. This is commonly the case in practical scenarios. The heterogeneity of the data results in the cost functions within the nodes being significantly different from each other.

The contribution of this paper is a new variant of PDMM that we refer to as *quadratic PDMM*. To overcome the problem with heterogeneity in the data sets we replace the original cost with its quadratic majorization using the L2 norm. The majorization minimization is consistent with the original cost minimization. We provide methods to select parameters settings that lead to a fast convergence rate. We show the effectiveness of the proposed method through several numerical experiments with various levels of heterogeneity between the observed data sets.

This paper is organized as follows: the conventional PDMM is explained in Sec. 2. The quadratic PDMM to overcome a drawback of conventional PDMM is proposed in Sec. 3. After conducting experiments in Sec. 4, we conclude this paper in Sec. 5.

2. PROBLEM AND CONVENTIONAL SOLUTION

We now formulate the problem our algorithm solves and briefly review an existing solution. We start with a problem definition and then discuss the conventional PDMM based solution method.

2.1. Problem definition

Let us consider that data captured by sensors are collected in V distributed nodes. The edge structure is described by a graphical model $\mathcal{G}(\mathcal{V},\mathcal{E})$ where $\mathcal{V}$ is the set of V nodes and $\mathcal{E}$ denotes the set of undirected edges. In this paper, the cost functions $F_i(\mathbf{p}_i):\mathbb{R}^M\rightarrow\mathbb{R}\cup\{\infty\}$ are limited to be convex, closed and proper (CCP) (e.g. [11, 12]) and $\mathbf{p}_i\in\mathbb{R}^M$ is the latent variables to be optimized in the i-th node. The constrained optimization for $\mathbf{p}_i$ that we address in this paper is defined as

$$\min_{\mathbf{p}_i}\sum_{i\in\mathcal{V}}F_i(\mathbf{p}_i)\quad \text{s.t.}\;\; \mathbf{A}_{i|j}\mathbf{p}_i+\mathbf{A}_{j|i}\mathbf{p}_j=\mathbf{0}\quad \forall(i,j)\in\mathcal{E}, \quad (1)$$

where the $\mathbf{A}_{i|j}\in\mathbb{R}^{M\times M}$ are parameters that specify the edge constraints. If we aim to attain consensus for the variables $\mathbf{p}_i$ across the V nodes then the $\mathbf{A}_{i|j}$ are

$$\mathbf{A}_{i|j}=\begin{cases}\boldsymbol{I} & (i>j,\; j\in\mathcal{N}(i))\\ -\boldsymbol{I} & (j>i,\; j\in\mathcal{N}(i))\\ \boldsymbol{O} & (\text{otherwise})\end{cases},$$

where $\boldsymbol{I}$ denotes an identity matrix and $\mathcal{N}(i)=\{j\in\mathcal{V}\mid(i,j)\in\mathcal{E}\}$ is the set of neighbors of the i-th node.

The constraint optimization problem (1) is generally solved with the method of Lagrange multipliers (e.g. [11, 12]). The Lagrangian function is defined by

$$L = \sum_{i \in \mathcal{V}} F_i(\mathbf{p}_i) + \frac{1}{2} \sum_{j \in \mathcal{N}(i)} \mathbf{x}_{i|j}^{\mathrm{T}} \big(\mathbf{A}_{i|j}\mathbf{p}_i + \mathbf{A}_{j|i}\mathbf{p}_j\big)\,, \tag{2}$$

where $\mathbf{x}_{i|j} \in \mathbb{R}^M$ and $^{\mathrm{T}}$ denote the dual variables associated with the constraint along edge (i, j) and transposition, respectively. Under some conditions the strong duality theorem (e.g. [11, 12]) holds

$$\min_{\mathbf{p}_i} \max_{\mathbf{x}_{i|j}} L = \max_{\mathbf{x}_{i|j}} \min_{\mathbf{p}_i} L \qquad \forall (i, j) \in \mathcal{E}. \tag{3}$$

Then the dual problem on the right side of (3) is generally solved instead of the primal problem on the left side of (3).

Before continuing our argumentation, we first define some symbols to simplify notation as

$$\mathbf{p} = \Big[\mathbf{p}_1^{\mathrm{T}}, \ldots, \mathbf{p}_V^{\mathrm{T}}\Big]^{\mathrm{T}},$$

$$\mathbf{x} = \Big[\mathbf{x}_{1|2}^{\mathrm{T}}, \ldots, \mathbf{x}_{1|V}^{\mathrm{T}}, \mathbf{x}_{2|1}^{\mathrm{T}}, \ldots, \mathbf{x}_{2|V}^{\mathrm{T}}, \ldots, \mathbf{x}_{V|1}^{\mathrm{T}}, \ldots, \mathbf{x}_{V|V-1}^{\mathrm{T}}\Big]^{\mathrm{T}},$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1|2}, \ldots, \mathbf{A}_{1|V}, & & \mathbf{O} \\ & \mathbf{A}_{2|1}, \ldots, \mathbf{A}_{2|V}, & \\ & & \ddots \\ \mathbf{O} & & \mathbf{A}_{V|1}, \ldots, \mathbf{A}_{V|V-1} \end{bmatrix}^{\mathrm{T}}.$$

We can now refine our definition of the dual problem. Let us first rewrite the dual problem by taking the undirected edges into account:

$$\begin{aligned} &\max_{\mathbf{x}_{i|j}} \min_{\mathbf{p}_i} \sum_{i \in \mathcal{V}} F_i(\mathbf{p}_i) + \sum_{j \in \mathcal{N}(i)} \mathbf{x}_{i|j}^{\mathrm{T}} \mathbf{A}_{i|j} \mathbf{p}_i \\ &= \max_{\mathbf{x}} \min_{\mathbf{p}} F(\mathbf{p}) + \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{p} \\ &= -\min_{\mathbf{x}} F^\star(-\mathbf{A}^{\mathrm{T}} \mathbf{x}), \end{aligned} \tag{4}$$

where $F : \mathbb{R}^{MV} \to \mathbb{R} \cup \{\infty\}$ denotes the sum over all local cost function and where $F^\star$ is the convex conjugate (the Legendre transformation) of F (e.g. [11, 12])

$$F^\star(-\mathbf{A}^{\mathrm{T}}\mathbf{x}) = \max_{\mathbf{p}} \Big(-\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{p} - F(\mathbf{p})\Big)\,. \tag{5}$$

When F is CCP, $F^\star$ is also CCP and $F^{\star\star} = F$ [11].

Next we consider the constraint on the dual variables. As noted in previous works on PDMM [8, 9, 10] each edge has only one constraint in (1) and hence the dual variables must satisfy $\mathbf{x}_{i|j} = \mathbf{x}_{j|i}$, $(i, j) \in \mathcal{E}$. Then, the problem (4) can be written by

$$\min_{\mathbf{x}} F^\star(-\mathbf{A}^{\mathrm{T}}\mathbf{x}) + \delta_{(\boldsymbol{I}-\mathbf{P})}\left(\mathbf{x}\right), \tag{6}$$

where $\delta_{(\boldsymbol{I}-\mathbf{P})}$ is the indicator function defined by

$$\delta_{(\boldsymbol{I}-\mathbf{P})}\left(\mathbf{x}\right) = \begin{cases} 0 & (\boldsymbol{I}-\mathbf{P})\mathbf{x} = \mathbf{0} \\ +\infty & \text{(otherwise)} \end{cases}, \tag{7}$$

and $\mathbf{P}$ is the permutation matrix that exchanges dual variables between each node pair as $\mathbf{x}_{j|i} \leftrightarrow \mathbf{x}_{i|j}$, $\forall (i, j) \in \mathcal{E}$.

2.2. Optimization using Peaceman-Rachford splitting

To solve the optimization problem in (6), monotone operator splitting (e.g. [17, 18]) was utilized in the conventional study [10]. The cost function in (6) is difficult to solve in one step. The operator splitting is a method to overcome this by decomposing it into easier problems to iteratively/alternatively update variables.

Let us rewrite (6) in the following form:

$$\mathbf{0} \in T_1(\mathbf{x}) + T_2(\mathbf{x}), \tag{8}$$

where $T_1 = -\mathbf{A}\partial F^\star(-\mathbf{A}^{\mathrm{T}})$ and $T_2 = \partial\delta_{(\boldsymbol{I}-\mathbf{P})}$ are maximally monotone operators [19, 20, 21] and the inclusion $\in$ facilitates the multi-valued nature of the maximally monotone operator. Because both $F^\star$ and $\delta_{(\boldsymbol{I}-\mathbf{P})}$ are CCP, the stationary point can be found when the subdifferential with respect to $\mathbf{x}$ includes the zero vector as in (8).

Following [10], we use Peaceman-Rachford (P-R) splitting [15, 14, 16] to find a stationary point. Although we omit the derivation, P-R splitting could be obtained by reforming (8). The result of the P-R splitting is that variables are iteratively updated in an alternating manner through Cayley operators (e.g. [17, 18]), which are denoted by $C_n = 2R_n - I$, $(n \in \{1, 2\})$, as

$$\mathbf{z} \in C_2 C_1(\mathbf{z}), \tag{9}$$

where $\mathbf{z}$ is a dual auxiliary variable of $\mathbf{x}$ which is calculated as $\mathbf{x} \in R_1(\mathbf{z})$, $R_n = (I + \sigma T_n)^{-1}$, $(\sigma > 0)$ is the resolvent operator [17, 18], I is the identity operator, and $^{-1}$ is the inverse operator (e.g. [17, 18]). Since P-R splitting can be decomposed as $\mathbf{z} \in C_2 C_1(\mathbf{z}) \Rightarrow \mathbf{z} \in C_2(\mathbf{y})$ where $\mathbf{y} \in C_1(\mathbf{z})$ is a second dual auxiliary variable. Thus, the algorithm can be written as

$$\mathbf{x} \in R_1(\mathbf{z}) = (I + \sigma T_1)^{-1}\mathbf{z}, \tag{10}$$

$$\mathbf{y} \in C_1(\mathbf{z}) = (2R_1 - I)\mathbf{z} = 2\mathbf{x} - \mathbf{z}, \tag{11}$$

$$\mathbf{z} \in C_2(\mathbf{y}) = \mathbf{P}\mathbf{y}, \tag{12}$$

(12) follows from the fact that $\mathbf{P}$ equals C_2, which is proven in [10].

2.3. Conventional PDMM algorithm

As context for our work, we briefly explain the conventional PDMM algorithm (10)-(12) . By reformulating (10) from $\mathbf{0} \in \mathbf{x} - \mathbf{z} + \sigma T_1(\mathbf{x})$ to differential form, the $\mathbf{x}$-update step is obtained as

$$\mathbf{0} \in \partial\bigg(\frac{1}{2\sigma}\ \|\mathbf{x} - \mathbf{z}\|_2^2 + F^\star(-\mathbf{A}^{\mathrm{T}}\mathbf{x})\bigg)\,, \tag{13}$$

where $\|\cdot\|_p$ denotes L_p norm and update procedures with respect to both $\mathbf{p}$ and $\mathbf{x}$ may be included in (13). To solve (13) previous studies [8, 9, 10], updated these variables were alternatively updated. This is achieved by adding a penalty term to the $\mathbf{p}$-update procedure derived from (5) as

$$\mathbf{p}^{(t+1)} = \arg\min_{\mathbf{p}} \Big(F(\mathbf{p}) + \mathbf{z}^{(t)\,\mathrm{T}}\mathbf{A}\mathbf{p} + \frac{1}{2\sigma}\|\mathbf{A}\mathbf{p}\|_2^2\Big), \tag{14}$$

where the last term in (14) is a penalty term to limit the feasible region in $\mathbf{p}$-update. By substituting $\mathbf{p}^{(t+1)}$ into (13), dual variable is updated by

$$\begin{aligned} \mathbf{x}^{(t+1)} &= \arg\min_{\mathbf{x}} \Big(\frac{1}{2\sigma}\ \|\mathbf{x} - \mathbf{z}^{(t)}\|_2^2 - \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{p}^{(t+1)} - F(\mathbf{p}^{(t+1)})\Big) \\ &= \mathbf{z}^{(t)} + \sigma\mathbf{A}\mathbf{p}^{(t+1)}. \end{aligned} \tag{15}$$

In the conventional PDMM algorithm (14), (15), (11), (12) are reformulated in a node parallelized computation that is summarized in **Algorithm 1**, where $\mathbf{Node}_j \leftarrow \mathbf{Node}_i(\cdot)$ indicates the data transmission from node i to j. Note that further improvements to reduce calculation cost are provided in [8, 9, 10].

It was confirmed that conventional PDMM algorithm generally provides a high convergence rate [8, 9, 10]. However, when the data sets available to the nodes are statistically heterogeneous, conventional PDMM does not perform well without carefully choosing σ. In this scenario the heterogeneity in the convex functions F_i $(i \in \mathcal{V})$ interferes the learning process. An empirical approach to overcoming this issue is to make σ sufficiently small to ensure homogeneous convexity of F_i. However, a more formal and effective approach to solve the problem is desirable.

Algorithm 1 Conventional PDMM

```
Initialization of $\mathbf{z}_{i|j}^{(0)}, \mathbf{p}_i^{(0)}$
for $t \in \{0, \ldots, T-1\}$ do
    ▷ Latent variable update
    for all $i \in \mathcal{V}$ do
        $\mathbf{p}_i^{(t+1)} = \arg\min_{\mathbf{p}_i} \Big( F_i(\mathbf{p}_i) + \sum_{j \in \mathcal{N}(i)} \mathbf{z}_{i|j}^{(t)\,\mathrm{T}} \mathbf{A}_{i|j} \mathbf{p}_i$
            $+ \frac{1}{2\gamma} \| \mathbf{A}_{i|j} \mathbf{p}_i + \mathbf{A}_{j|i} \mathbf{p}_j^{(t)} \|_2^2 \Big)$
    ▷ Dual and its auxiliary variables update
    for all $i \in \mathcal{V}, j \in \mathcal{N}(i)$ do
        $\mathbf{x}_{i|j}^{(t+1)} = \mathbf{z}_{i|j}^{(t)} + \sigma \left( \mathbf{A}_{i|j} \mathbf{p}_i^{(t+1)} + \mathbf{A}_{j|i} \mathbf{p}_j^{(t)} \right)$
        $\mathbf{y}_{i|j}^{(t+1)} = 2\mathbf{x}_{i|j}^{(t+1)} - \mathbf{z}_{i|j}^{(t)}$
    ▷ Transmit variables
    for all $i \in \mathcal{V}, j \in \mathcal{N}(i)$ do
        $\mathbf{Node}_j \leftarrow \mathbf{Node}_i(\mathbf{p}_i^{(t+1)}, \mathbf{y}_{i|j}^{(t+1)})$
    ▷ Dual auxiliary variable update
    for all $i \in \mathcal{V}, j \in \mathcal{N}(i)$ do
        $\mathbf{z}_{i|j}^{(t+1)} = \mathbf{y}_{j|i}^{(t+1)}$
end for
```

Algorithm 2 Quadratic PDMM

```
Initialization of $\mathbf{z}_{i|j}^{(0)}, \mathbf{p}_i^{(0)}$
for $t \in \{0, \ldots, T-1\}$ do
    ▷ Dual and its auxiliary variables update
    for all $i \in \mathcal{V}, j \in \mathcal{N}(i)$ do
        $\mathbf{x}_{i|j}^{(t+1)} = \left( \frac{1}{\sigma} \boldsymbol{I} + \eta \mathbf{A}_{i|j} \mathbf{A}_{i|j}^{\mathrm{T}} \right)^{-1}$
            $\cdot \left[ \mathbf{A}_{i|j} \left( p_i^{(t)} - \eta \partial F_i(\mathbf{p}_i^{(t)}) \right) + \frac{1}{\sigma} \mathbf{z}_{i|j}^{(t)} \right]$
        $\mathbf{y}_{i|j}^{(t+1)} = 2\mathbf{x}_{i|j}^{(t+1)} - \mathbf{z}_{i|j}^{(t)}$
    ▷ Latent variable update
    for all $i \in \mathcal{V}$ do
        $\mathbf{p}_i^{(t+1)} = \mathbf{p}_i^{(t)} - \eta \left( \partial F_i(\mathbf{p}_i^{(t)}) + \sum_{j \in \mathcal{N}(i)} \mathbf{A}_{i|j}^{\mathrm{T}} \mathbf{x}_{i|j}^{(t+1)} \right)$
    ▷ Transmit variables
    for all $i \in \mathcal{V}, j \in \mathcal{N}(i)$ do
        $\mathbf{Node}_j \leftarrow \mathbf{Node}_i(\mathbf{y}_{i|j}^{(t+1)})$
    ▷ Dual auxiliary variable update
    for all $i \in \mathcal{V}, j \in \mathcal{N}(i)$ do
        $\mathbf{z}_{i|j}^{(t+1)} = \mathbf{y}_{j|i}^{(t+1)}$
end for
```

3. PROPOSED METHOD

We propose the *quadratic PDMM* method to overcome the vulnerability to the data set statistical heterogeneity. To ensure homogeneous convexity for each node cost, we replace the original cost with their quadratic majorization function using L2 norm (cf. Sec. 3.1). By predicting the convergence rate on the proposed algorithm, we derive parameter settings way for fast convergence (cf. Sec. 3.2).

3.1. Quadratic PDMM algorithm derivation

To ensure properties such as strong convexity (SC) and Lipschitz smoothness (LS) [17, 22, 23, 24] independently of the statistic properties of the data and the structure of F, we define a new cost function G to work with:

$$G(\mathbf{p}) = F(\mathbf{p}^{(t)}) + \left\langle \partial F(\mathbf{p}^{(t)}), \mathbf{p} - \mathbf{p}^{(t)} \right\rangle + \frac{1}{2\eta} \| \mathbf{p} - \mathbf{p}^{(t)} \|_2^2, \quad (16)$$

where $\eta \geq 0$, $\mathbf{p}^{(t)}$ is the latent variables at the update time t and G is then $1/\eta$-SC and $1/\eta$-LS. For the case that F is strictly CCP (e.g. [11, 12]) and hence continuously-differentiable, the original cost function is approximated from the second-order Taylor series expansion about $\mathbf{p}$ by using $\boldsymbol{\delta} \to \mathbf{0}$ as

$$F(\mathbf{p}^{(t)} + \boldsymbol{\delta}) = F(\mathbf{p}^{(t)}) + \left\langle \partial F(\mathbf{p}^{(t)}), \boldsymbol{\delta} \right\rangle + \frac{1}{2} \left\langle \mathbf{H}_F(\mathbf{p}^{(t)}) \boldsymbol{\delta}, \boldsymbol{\delta} \right\rangle + o(\| \boldsymbol{\delta} \|_2^2),$$

where the Hessian $\mathbf{H}_F$ is positive definite and its maximum eigenvalue at any $\mathbf{p}$ is denoted by λ_{max}. When $\eta \leq 1/\lambda_{\text{max}}$, G becomes majorization function of F, i.e., $G(\mathbf{p}) \geq F(\mathbf{p})$ (e.g. [11, 12]). Then, majorization minimization is consistent with minimizing F. For the case that F is not strictly CCP, making η sufficiently small in G is consistent with minimizing F. The convex conjugate of G is defined by

$$G^{\star}(-\mathbf{A}^{\mathrm{T}} \mathbf{x}) = \max_{\mathbf{p}} \left(-\mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{p} - G(\mathbf{p}) \right), \quad (17)$$

where $G^{\star}$ is η-SC and η-LS [17, 22, 23, 24].

The problem in quadratic PDMM is defined by replacing $F^{\star}$ in (6) with $G^{\star}$ as

$$\min_{\mathbf{x}} G^{\star}(-\mathbf{A}^{\mathrm{T}} \mathbf{x}) + \delta_{(\boldsymbol{I} - \mathbf{P})}(\mathbf{x}). \quad (18)$$

Although the quadratic PDMM algorithm basically follows P-R splitting, the solver in (10) is changed to

$$\mathbf{0} \in \partial \left(\frac{1}{2\sigma} \| \mathbf{x} - \mathbf{z} \|_2^2 + G^{\star}(-\mathbf{A}^{\mathrm{T}} \mathbf{x}) \right). \quad (19)$$

From (17), it follows that the optimal point with respect to $\mathbf{p}$ is dependent of $\mathbf{x}$. Thus, $\mathbf{p}$ is represented as a function of $\mathbf{x}$ as

$$\mathbf{p}^{(t+1)}(\mathbf{x}) = \arg\min_{\mathbf{p}} \Big(\mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{p} + G(\mathbf{p}) \Big) = \mathbf{p}^{(t)} - \eta \Big(\partial F(\mathbf{p}^{(t)}) + \mathbf{A}^{\mathrm{T}} \mathbf{x} \Big). \quad (20)$$

Then, $\mathbf{x}$ is updated as

$$\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}} \left(\frac{1}{2\sigma} \| \mathbf{x} - \mathbf{z}^{(t)} \|_2^2 - \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{p}^{(t+1)}(\mathbf{x}) - G(\mathbf{p}^{(t+1)}(\mathbf{x})) \right)$$

$$= \left(\frac{1}{\sigma} \boldsymbol{I} + \eta \mathbf{A} \mathbf{A}^{\mathrm{T}} \right)^{-1} \left[\mathbf{A} \left(\mathbf{p}^{(t)} - \eta \partial F(\mathbf{p}^{(t)}) \right) + \frac{1}{\sigma} \mathbf{z}^{(t)} \right]. \quad (21)$$

The quadratic PDMM algorithm that reforms (21), (20), (11), (12) into node parallelized computation manner is summarized in **Algorithm 2**. A side effect of the proposed method is that it exchanges only the dual auxiliary variable $\mathbf{y}_{i|j}$ between connected nodes. This halves the data transmission rate compared with conventional PDMM.

3.2. Convergence rate prediction

We now discuss how to set the parameters σ to achieve high convergence rate, assuming that η is properly set as discussed in Sec. 3.1, From the basic property of the Cayley operator, its contractive rate can be determined when $G^{\star}$ is both SC and LS (e.g. [18]). Since $G^{\star}$ is η-SC and η-LS, it is represented for the input/output pairs on $\mathbf{y} \in C_1(\mathbf{z})$ as

$$\| \mathbf{y}^{(t+1)} - \mathbf{y}^{(t)} \|_2^2 \leq \left(1 - \frac{4\sigma\eta}{(1 + \sigma\eta)^2} \right) \| \mathbf{z}^{(t)} - \mathbf{z}^{(t-1)} \|_2^2, \quad (22)$$

Since the Cayley operator C_2 is non-expansive, i.e., $\| \mathbf{z}^{(t+1)} - \mathbf{z}^{(t)} \|_2^2 \leq \| \mathbf{y}^{(t+1)} - \mathbf{y}^{(t)} \|_2^2$ (e.g. [18]), the convergence rate on the quadratic PDMM satisfies

$$\| \mathbf{z}^{(t)} - \mathbf{z}^{*} \|_2^2 \leq \left(1 - \frac{4\sigma\eta}{(1 + \sigma\eta)^2} \right)^t \| \mathbf{z}^{(0)} - \mathbf{z}^{*} \|_2^2, \quad (23)$$

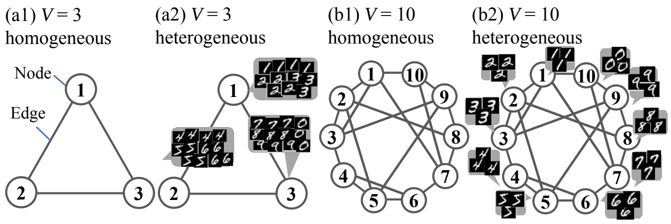

Fig. 1. Combination of edge structure and training data sets

where $\mathbf{z}^*$ denotes the stationary point of $\mathbf{z}$. For fast convergence even when data set available for each node are heterogeneous, good choices are $\sigma = 1/\eta$ [18] while holding $\eta \le 1/\lambda_{\max}$.

4. EXPERIMENTS

4.1. Experimental setup

To confirm the effectiveness of the proposed method, numerical experiments were conducted. We used the MNIST data set [25] composed of handwritten images of $B = 10$ digits (60,000 training and 10,000 evaluating images). The input vector $\boldsymbol{v}_{i,d}$, of dimension is $M = 785$, is composed of 784 ($= 28 \times 28$) pixels and a bias, and the output supervisor $\{s_{i,1,d}, \ldots, s_{i,B,d}\}$ is composed of 1 (ideal respond) and the others are $B-1$ zeros. Ridge regression is used as a cost function F because it is strictly CCP and we can set the parameters η, σ according to our theory:

$$F(\mathbf{p}) = \sum_{i\in\mathcal{V}}\sum_{b\in\mathcal{B}}\sum_{l\in\mathcal{D}(i)} \frac{1}{2VBD(i)} \| s_{i,b,d} - \mathbf{p}_{i,b}^{\mathrm{T}} \boldsymbol{v}_{i,d} \|_2^2, \qquad (24)$$

where $\mathcal{B} \in \{1, \ldots, B\}$ and $\mathcal{D}(i) \in \{1, \ldots, D(i)\}$ denotes the index set of data set included in node i. We prepared two kinds of graph structures $V \in \{3, 10\}$ as shown in Fig. 1. They were designed such that both long and short paths were present. The training data set was divided into V nodes in two ways. In the *homogeneous* case, the training data were divided randomly. In the *heterogeneous* case, the training data set were divided to obtain a bias for each node. Thus, for $V = 3$ we selected $\{1, 2, 3\}$ for node #1, $\{4, 5, 6\}$ for node #2 and $\{7, 8, 9, 0\}$ for node #3. Similarly we assigning each digit to on node when $V = 10$. The number of data for each node $D(i)$ was not necessarily identical. In total, four kinds of edge/data set combinations as shown in Fig. 1 were prepared.

Three algorithms were compared: distributed ADMM (D-ADMM) [3], conventional PDMM (PDMM) described in **Algorithm 1**, and the proposed quadratic PDMM (Q-PDMM) as shown in **Algorithm 2**. Since the maximum eigenvalue of Hessian in all situations was $\lambda_{\max} = 47.3$, we set parameters as $\eta = 0.02 (\le 1/\lambda_{\max})$ and $\sigma = 50$ $(= 1/\eta)$ in the quadratic PDMM. On the other hand, in conventional PDMM, σ was adjusted to be converged in all situations and this strategy resulted in $\sigma = 50$. The variables $\mathbf{p}$ and $\mathbf{z}$ were initialized randomly in which follows Gaussian $\mathcal{N}(0, 0.1)$. To investigate the robustness to asynchronous information exchange among nodes, information exchange was performed randomly at a rate of once per three iterations. Although this exchange frequency rate slows the convergence compared with the synchronous information-exchange case, the differences in the convergence curves due to the edge structure did not change significantly. Thus, we show experimental results only for the asynchronous case.

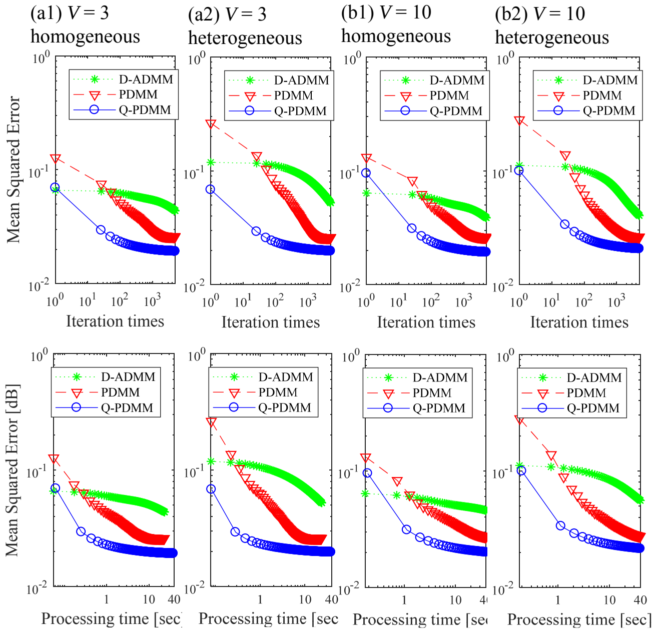

Fig. 2. Convergence curves where variable exchange was exchanged asynchronously/randomly among node at a rate of once per three iterations for iteration times (upper row) and for processing time (lower row)

4.2. Experimental results

Figure 2 shows the averaged mean squared error (MSE) for iteration time (upper row) and for processing time (lower row). In total, $T = 5,000$ iterative updates were computed on GPU (NVIDIA Tesla K40). The evaluation scores were calculated using (24), but test data were utilized rather than training data. The differences with respect to $\mathbf{p}_i$ among the V nodes were quite small in all methods. The convergence curves were changed dependent on the statistical properties on the data sets and the graph structures with conventional methods. On the other hand with the quadratic PDMM, its convergence curve was changed dependent on the graph structure only. This would be because the cost convexity among nodes is homogenized in the proposed method. Its convergence rate was seemed to be the fastest because it reached stationary points in early iteration times. Even when comparing processing time as a standard, common results were obtained.

5. CONCLUSION

We proposed *quadratic PDMM* for enhancing the robustness to the statistically heterogeneous data sets in edge consensus computing. By replacing the original cost with its quadratic majorization using L2 norm, homogeneous convexity on the nodes is ensured even when the data sets available are heterogeneous among nodes. Our investigation of the convergence rate on the quadratic PDMM led to a method to set the parameters to obtain fast convergence. Through experiments, it was confirmed that the proposed method works well even if heterogeneous data sets are provided to the nodes.

For future work, further investigations with various data sets and testing with other edge structures are needed ascertain the effectiveness of the proposed method under various conditions. In addition, it is natural to study the use of the method to synchronize deep neural networks in the context of large-scale stochastic optimization.

6. REFERENCES

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” *Proc. of workshop on Mobile cloud computing (MCC’12)*, pp. 13–16, 2012.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: vision and challenges,” *IEEE internet of things journal*, vol. 3, pp. 637–646, 2016.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, vol. 3, Foundations and Trends in Machine Learning, 2011.

[4] R. Zhang and J. Kwok, “Asynchronous distributed admm for consensus optimization,” *Proc. of 31st international conference on machine learning (ICML’14)*, pp. 1701–1709, 2014.

[5] E. Wei and A. Ozdaglar, “Distributed alternating direction method of multipliers,” *Proc. of 51st IEEE conference on decision and control*, 2012.

[6] Q. Ling and A. Ribeiro, “Decentralized linearized alternating direction method of multiplier,” *Proc. of IEEE international conference on acoustics, speech and signal processing (ICASSP’14)*, pp. 5447–5451, 2014.

[7] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, “Decentralized quadratically approximated alternating direction method of multipliers,” *Proc. of IEEE global conference on signal and information processing (GlobalSIP’15)*, pp. 795–799, 2015.

[8] G. Zhang and R. Heusdens, “On simplifying the primal-dual method of multipliers,” *Proc. of international conference on acoustics, speech and signal processing (ICASSP’16)*, pp. 4826–4830, 2016.

[9] G. Zhang and R. Heusdens, “Distributed optimization using the primal-dual method of multipliers,” *IEEE trans. on signal and information processing over networks*, 2017 (accepted).

[10] T. Sherson, R. Heusdens, and W. B. Kleijn, “Derivation and analysis of the primal-dual method of multipliers based on monotone operator theory,” *arXiv: 1706.02654*, 2017.

[11] R. T. Rockafellar, *Convex analysis*, Princeton University Press, 1970.

[12] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[13] J. Douglas and H. H. Rachford, “On the numerical solution of heat conduction problems in two and three space variables,” *Trans. of American Mathematical Soc.*, vol. 82, pp. 421–439, 1956.

[14] P. L. Lions and B. Mercier, “Splitting algorithm for the sum of two nonlinear operators,” *Soc. for industrial and applied mathematics (SIAM) Journal on Numerical Analysis*, vol. 16, pp. 964–979, 1978.

[15] K. B. Kellogg, “A nonlinear alternating direction method,” *Mathematics of computation*, vol. 23, pp. 23–27, 1969.

[16] D. W. Peaceman and H. H. Rachford, “The numerical solution of parabolic and elliptic differential equations,” *Soc. for industrial and applied mathematics (SIAM) Journal of soc. for industrial and applied mathematics*, vol. 3, pp. 28–41, 2017.

[17] R. T. Rockafellar and R. J. B. Wets, *Variational analysis*, Springer, 1998.

[18] E. K. Ryu and S. Boyd, “Primer on monotone operator methods,” *Applied and computational mathematics*, vol. 15, pp. 3–43, 2016.

[19] G. J. Minty, “Monotone networks,” *Proc. of the royal soc. A: mathematical, physical and engineering sciences*, vol. 257, pp. 194–212, 1960.

[20] G. Zames, “On the input-output stability of time-varying nonlinear feedback system part one: conditions derived using concepts of loop gain, conicity, and positivity,” *IEEE trans. on automatic control*, vol. 11, pp. 228–238, 1966.

[21] R. T. Rockafellar, “On the maximality of sums of nonlinear monotone operators,” *Trans. of American mathematical soc.*, vol. 149, pp. 75–88, 1970.

[22] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization*, John Wiley & Sons Ltd., 1983.

[23] Y. Nesterov, *Introductory lectures on convex optimization*, Springer, 2004.

[24] S. M. Kakade, S. Shalev-shwartz, and A. Tewari, “On the duality of strong conveity and strong smootheness: learning applications and matrix regularization,” *Technical report on Toyota technological institute*, pp. 1–10, 2009.

[25] Y. LeCun, C. Cortes, and C. J. C. Burges, “The mnist database of handwritten digits:,” http://yann.lecun.com/exdb/mnist/.