IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 6, 2020, accepted October 15, 2020, date of publication October 22, 2020, date of current version November 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033138

Distributed Boosting Variational Inference Algorithm Over Multi-Agent Networks

XIBIN AN[1], CHEN HU[1], GANG LIU[1], AND MINGHAO WANG[2]
[1]Department of Electronic Engineering, Rocket Force University of Engineering, Xi'an 710025, China
[2]Institute of Survival Technology Effectiveness Evaluation of Flying Vehicle, Beijing 100094, China

Corresponding author: Chen Hu (chenh628@hotmail.com)

ABSTRACT Distributed Bayesian estimation over multi-agent networks has received much attention due to its broad applications, where each agent has its private data that is unavailable to other agents. For efficient inference over multi-agent networks, we develop a distributed boosting variational inference (DBVI) algorithm with limited communication. We first decompose the global cost function into a *sum-of-costs* form, where each local cost only relates to its own dataset. Then, the global posterior distribution is approximated by a gradient decent at each boosting step, followed by a consensus protocol for cooperation with the neighbors. Moreover, we derive DBVI with Gaussian mixture model (DBVI-GMM) in detail. Finally, simulations on the synthetic and real datasets illustrate the effectiveness of the proposed algorithm.

INDEX TERMS Multi-agent networks, distributed machine learning, posterior probability approximation, boosting variational inference.

I. INTRODUCTION

Multi-agent networks have been applied in many fields, such as traffic, communication and military, due to robustness and low cost [1]–[4]. Considering the communication cost and privacy, networks are well-suited to perform distributed data processing and decisions [3]. Distributed algorithms exhibit flexibility and provide robustness to node or link failures in networks [4], [5]. Many distributed algorithms over networks for data analysis and inference have been proposed, such as distributed estimation [5], [6] and inference [7], [8].

In Bayesian framework, statistical inference is used to compute the posterior probability distributions [9], [10]. Variational inference (VI) is a popular method to estimate an intractable posterior distribution by minimizing the Kullback-Leibler (KL) divergence between the tractable distribution and intractable distribution [11], [12]. Mean-field variational inference (MFVI), a widely used variant of VI, assumes that the tractable distribution factorizes across the parameters of the model and obeys the certain family, such as the exponential distribution family [13]. This assumption leads to a convenient and an efficient coordinate-ascent algorithm [12]. In contrast to its computational advantage, MFVI fails to approximate complicated distribution such as multi-modality and heavy-tails [12], [14]–[16]. In practical application, the posterior is usually a non-unimodal distribution. Therefore, it is necessary to study the variational method to approximate the complicated posterior.

An alternative and flexible category for variational approximations is to use mixture of simple model to approximate the multi-modality posterior, such as nonparametric variational inference (NPV) [17]. NPV needs to train more parameters than MFVI method, especially when the number of mixture components is large. Since the loss of NPV is non-convex, a joint optimization of mixture model will be extremely slow. It may need to rerun NPV with different initializations, which have limited the application of NPV. In order to overcome this issue, boosting variational inference (BVI) is proposed to train the mixture components one-at-a-time. BVI starts with the traditional VI and keeps improving the approximation by adding new component. However, in distributed setting, agent cannot access the entire dataset such that BVI cannot be directly applied in networks.

Some distributed variational inference (DVI) algorithms have been proposed to estimate the posterior over networks [18]–[20]. DVI with Gaussian mixture model (DVI-GMM) [18], [20] is proposed to approximate the multi-modality posterior by distributed optimization methods. Although DVI-GMM can successfully capture the multi-modality posterior over networks, it still faces some challenges from both

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

the computational complexity and theory. First, the cost of DVI-GMM is non-convex so that it needs to rerun the algorithm with different initializations and choices for the number of components. Second, there is no theoretical guarantee that the approximation error vanishes with an increasing number of mixture components. It is urgent to develop a distributed variational inference with higher efficiency and guaranteed convergence over networks.

In this paper, we consider the posterior approximation problem over networks, where each agent cannot access the global dataset. Our goal is to design a distributed boosting variational inference algorithm that can perform almost as well as the centralized BVI and ameliorate the computational complexity of DVI-GMM. By decomposing the centralized cost into a sum of local ones, the posterior approximation over multi-agent networks is formulated as a distributed parameters optimization problem. With the help of distributed optimization methods [20]–[23], each agent obtains the centralized solution based on its local dataset and some information from neighbors. The only transmitted message is the parameters information, which can efficiently reduce communication cost. The main contributions of this paper are summarized as follows.

- We propose a distributed boosting variational inference (DBVI) algorithm over networks. By exchanging parameters between neighbors, each agent can effectively obtain the centralized solution. Compared with the BVI algorithms [24]–[26], DBVI can solve the posterior approximation problem caused by distributed data.
- We decompose the centralized cost into a sum of local ones, which only relates to its local dataset. The posterior probability approximation over networks is converted into a distributed optimization problem. With the help of distributed stochastic gradient descent method, local solution can effectively converge to the centralized one.
- We derive DBVI with Gaussian mixture model (DBVI-GMM) algorithm in detail. Compared with DVI-GMM [18], DBVI-GMM can effectively deal with the difficulties caused by the sensitive initialization and uncertain number of mixture components, where DBVI-GMM trains the mixture components one-at-a-time. DBVI-GMM starts with the traditional DVI algorithm and improves the approximation by adding new component until the maximum number of mixture components is reached.

The rest of this paper is organized as follows. Section II states problem formulation. Section III provides distributed boosting variational inference framework. Section IV derives DBVI with Gaussian mixture model. Section V presents some numerical simulations to test the performance. Finally, conclusions are drawn in Section VI.

II. PROBLEM FORMULATION

We consider a network with N agents. The communication between agents can be described by a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a set of nodes $\mathcal{V} = \{1, 2, \cdots, N\}$, a set of edges $\mathcal{E}$ and an adjacent weighted matrix A. For each agent $i \in \mathcal{V}$, denote $\mathcal{E}_i = \{j|(i,j) \in \mathcal{E}\}$ as a set of neighbors of agent i (including anget i itself). The adjacent matrix A is defined as follows,

- $a_{i,j} > 0$ for any $(i,j) \in \mathcal{E}_i$ and $a_{i,i} > 0$ if $j = i$;
- $a_{i,j} = 0$ for any agent j that is not the neighbor of agent i;
- A is a doubly-stochastic matrix, that is $\sum_{j=1}^{N} a_{i,j} = 1$ and $\sum_{i=1}^{N} a_{i,j} = 1$.

Denote the global dataset as $X = \{X_i\}_{i=1}^{N}$, and agent i has its local dataset $X_i = \{x_{ij}\}_{j=1}^{N_i}$. We consider the problem of estimating the posterior of hidden variables θ given observed data X. By Bayesian formula, the posterior distribution $p(\theta|X)$ can be given by

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)}, \tag{1}$$

where $p(\theta)$ is the prior and $p(X|\theta)$ is the likelihood. In most cases, the normalizing constant $p(X)$ is an intractable integral, so an approximation is needed for the normalized posterior distribution $p(\theta|X)$.

VI is used to approximate the posterior of unobserved variables by a tractable distribution q, which can be found by minimizing the Kullback-Leibler (KL) divergence between p and q. Specifically,

$$q = arg\min_q D_{KL}(q||p) = arg\min_q \int q \log \frac{q}{p}, \tag{2}$$

Under mean-field assumption, $q(\theta) = q(\theta_1)q(\theta_2)\cdots q(\theta_d)$, where d is the dimension of θ.

Traditional VI is difficult to capture the complicated distribution such as multi-modality and heavy-tails [26]. An applicable way is to employ the mixture model to approximate the complicated posterior, such as NPV [17] and BVI [24]–[26]. Specifically, let $q = \sum_{C=1}^{M} \alpha_C h_C$, where $\alpha_C \in (0, 1]$, $\sum_{C=1}^{M} \alpha_C = 1$, M is the number of mixture components and h_C belongs to some basic family of known distribution with parameters λ_C, denoted as $h_C(\cdot; \lambda_C)$, such as Gaussian distribution. A direct optimization on $\{(\alpha_C, h_C)\}_{C=1}^{M}$ is hard [26]. By introducing gradient boosting, $\{(\alpha_C, h_C)\}_{C=1}^{M}$ can be optimized one by one from $C = 1$ to M in BVI [24]–[26]. In the Cth iteration, the corresponding mixture model is described as

$$q^C = (1 - \alpha_C)q^{C-1} + \alpha_C h_C. \tag{3}$$

where q^{C-1} is the current existing approximation.

Combine (3) and (2), problem now becomes

$$\{\alpha_C, h_C\} = arg \min_{\alpha_C, h_C} D_{KL}(((1 - \alpha_C)q^{C-1} + \alpha_C h_C)||p). \tag{4}$$

A greedy minimization is used to seek a sequence $\{(\alpha_C, h_C)\}_{C\in\mathbb{N}}$. As $C \to \infty$, we pursue $\nabla_{q^C} D_{KL}(q^C||p) \to 0$ such that $\sum_{C=1}^{\infty} \alpha_C h_C = p$. For $C = 1, 2, \cdots$, (α_C, h_C) can be obtained with two steps as shown in Figure 1.

In each iteration, BVI updates h_C based on the gradient $\nabla_{q^{C-1}} D_{KL}(q^{C-1}||p)$, and then updates α_C with fixed h_C.

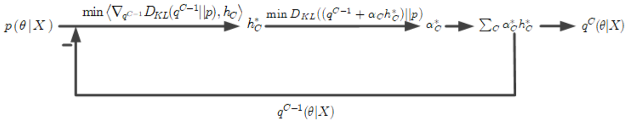
$p(\theta|X)$
$\min\langle\nabla_{q^{C-1}}D_{KL}(q^{C-1}||p),h_C\rangle$
h_C^*
$\min D_{KL}((q^{C-1}+\alpha_C h_C^*)||p)$
α_C^*
$\sum_C \alpha_C^* h_C^*$
$q^C(\theta|X)$
$-$
$q^{C-1}(\theta|X)$

FIGURE 1. The Framework of BVI.

In order to obtain the gradient, we take a first-order Taylor series expansion of $D_{KL}(q^C||p)$ in q^{C-1},

$$\begin{aligned}&D_{KL}((1-\alpha_C)q^{C-1}+\alpha_C h_C||p)\\&\quad=\alpha_C\nabla_{q^{C-1}}D_{KL}(q^{C-1})(h_C-q^{C-1})\\&\qquad+D_{KL}(q^{C-1}||p)+o(\alpha_C^2)\\&\quad=D_{KL}(q^{C-1}||p)+\alpha_C\left\langle\log q^{C-1}-\log p,h_C\right\rangle\\&\qquad-\alpha_C\left\langle\log q^{C-1}-\log p,q^{C-1}\right\rangle+o(\alpha_C^2),\end{aligned}\tag{5}$$

where $\langle q_1,q_2\rangle=\int q_1\, q_2$ is the inner product. It is clear that $\nabla_{q^{C-1}}D_{KL}(q^{C-1}||p)=\log q^{C-1}-\log p$. Since q^{C-1} and p are constant, we can obtain the optimal h_C by minimizing $\langle\log q^{C-1}-\log p,h_C\rangle$ with respect to h_C. A direct minimization of $\langle\log q^{C-1}-\log p,h_C\rangle$ is ill-posed [24], since h_C will degenerate to a point mass. Here, we consider the regularized cost function with respect to h_C [25]. Specifically,

$$h_C^*=arg\min_{h_C}E_{h_C}(\log q^{C-1}-\log p)+\frac{\delta}{2}||h_C||_2^2.\tag{6}$$

where $E_{h(\theta)}(f(\theta))=\int f(\theta)h(\theta)d\theta$. Assume that weak learner h_C belongs to a certain distribution family with parameters λ_C denoted as $h_C(\theta;\lambda_C)$. The 'best' distribution in (6) is represented by its parameters λ_C^*. The problem (6) can be rewritten as

$$\lambda_C^*=arg\min_{\lambda_C}L1^C(\lambda_C),\tag{7}$$

where

$$\begin{aligned}L1^C(\lambda_C)=E_{\theta\sim h_C(\theta;\lambda_C)}[\log q^{C-1}(\theta)-\log p(\theta|X)]\\+\frac{\delta}{2}||h_C(\theta;\lambda_C)||_2^2.\end{aligned}\tag{8}$$

With fixed $h_C^*(\theta;\lambda_C^*)$, we have $q^C=q^{C-1}+\alpha_C h_C^*(\theta;\lambda_C^*)$. The weight α_C can be obtained by minimizing KL divergence between q^C and p, that is

$$\alpha_C^*=arg\min_{\alpha_C}L2^C(\alpha_C),\tag{9}$$

where $L2^C(\alpha_C)=E_{\theta\sim q^C}(\log q^C(\theta)-\log p(\theta|X))$.

Note that the optimal λ_C^* and α_C^* in (7) and (9) can be obtained by each agent if each agent knows the global data set $\{X\}$. However, in our paper, agent i can only access to its local dataset $\{X_i\}$. Therefore, we have to investigate a distributed algorithm to optimize $L1^C(\lambda_C)$ and $L2^C(\alpha_C)$.

III. DISTRIBUTED BOOSTING VARIATIONAL INFERENCE

In this section, we convert problems (7) and (9) into distributed optimization problem, and show how to obtain the solution of (7) and (9) at each agent via distributed gradient descent method.

Assume the data of each agent is mutually independent, the log joint posterior can be obtained as

$$\begin{aligned}&\log p(\theta|X)\\&\quad=\log[\prod_{i=1}^{N}p(X_i|\theta)p(\theta)]-\log\prod_{i=1}^{N}p(X_i)\\&\quad=\frac{1}{N}\sum_{i=1}^{N}(N\log p(X_i|\theta)+\log p(\theta)-N\log p(X_i)),\end{aligned}\tag{10}$$

where $p(X_i|\theta)$ and $p(X_i)$ only relate to its local dataset X_i, and θ is the global parameters. Substituting (10) into (8), we have

$$\begin{aligned}&L1^C(\lambda_C)\\&\quad=E_{\theta\sim h(\theta;\lambda_C)}(\log q^{C-1}(\theta)-\log p(\theta|X))+\frac{\delta}{2}||h||_2^2\\&\quad=\frac{1}{N}\sum_{i=1}^{N}\{E_{\theta\sim h(\theta;\lambda_C)}[\log q^{C-1}(\theta)-\log\tilde{p}(X_i|\theta)\\&\qquad-\log p(\theta)]\}+\frac{\delta}{2}||h||_2^2+\log p(X)\\&\quad=\frac{1}{N}\sum_{i=1}^{N}L1_i^C(\lambda_C)\end{aligned}\tag{11}$$

where $\log\tilde{p}(X_i|\theta)=N\log p(X_i|\theta)$ means the likelihood of parameter θ given local data replicated N times. Since $p(X)$ is a constant, $L1_i^C(\lambda_C)$ can be rewritten as

$$\begin{aligned}&L1_i^C(\lambda_C)\\&\quad=E_{\theta\sim h_i^C}[\log q_i^{C-1}(\theta)-\log\tilde{p}(X_i|\theta)-\log p(\theta)]+\frac{\delta}{2}||h||_2^2.\end{aligned}\tag{12}$$

Similarly, local cost $L2_i^C(\alpha_C)$ can be written as

$$\begin{aligned}&L2_i^C(\alpha_C)\\&\quad=E_{\theta\sim q_i^C}[\log q_i^C(\theta)-\log\tilde{p}(X_i|\theta)-\log p(\theta)].\end{aligned}\tag{13}$$

It should be noticed that local cost $L1_i^C(\lambda_C)$ and $L2_i^C(\alpha_C)$ only relate to local dataset and the number of agents. Therefore, our problem becomes

$$\begin{cases}\min_{\lambda_C}\sum_{i=1}^{N}L1_i^C(\lambda_C)\\\min_{\alpha_C}\sum_{i=1}^{N}L2_i^C(\alpha_C).\end{cases}\tag{14}$$

Our goal is to make local parameters approach to the centralized solution, that is $(\alpha_{i,C},\lambda_{i,C})\to(\alpha_C^*,\lambda_C^*)$ for $C=1,2,\cdots$. The structure of (14) is suitable for distributed optimization algorithm, such as distributed stochastic gradient descent (DSGD) [27], [28] and distributed gradient descent (DGD) [29], [30]. Next, we will show how to find the optimal λ_C^* and α_C^* at each agent by DSGD.

For agent i, $q_i^C(\theta) = \sum_C \alpha_{i,C} h_{i,C}(\theta; \lambda_{i,C})$ where $\lambda_{i,C}$ is the parameters of weak learner $h_{i,C}$ and $\alpha_{i,C}$ stands for the weight of $h_{i,C}$. A two iterative steps is used to optimize the parameters $\lambda_{i,C}$ and $\alpha_{i,C}$. For $\lambda_{i,C}$, we define an intermediate variable $\varphi_{i,C}$ updated by gradient descent step. Then, a combination step is used to make the parameters consensus. Specifically, the update of $\lambda_{i,C}$ is

$$\varphi_{i,C}^t = \lambda_{i,C}^{t-1} + \eta_t \nabla_{\lambda_{i,C}^{t-1}} L1_i^C(\lambda_{i,C}^{t-1}) \tag{15a}$$

$$\lambda_{i,C}^t = \sum_{j \in \mathcal{E}_i} a_{ij} \varphi_{i,C}^t, \tag{15b}$$

where $\nabla_{\lambda_{i,C}^{t-1}} L1_i^C(\lambda_{i,C}^{t-1})$ is the gradient of $L1_i^C(\lambda_{i,C}^{t-1})$, and the step size η_t should satisfy the following conditions [18]

$$\eta_t > 0, \ \sum \eta_t = \infty, \ \sum \eta_t^2 < \infty. \tag{16}$$

Here, we consider $\eta_t = 1/(d + \tau t)$ [18], where the forgetting rate τ controls the decreasing speed of η_t, and d is a positive constant, $d \geq 1$.

With fixed $h_{i,C}^*$, $\alpha_{i,C}^*$ can be updated as follows

$$\phi_{i,C}^t = \alpha_{i,C}^{t-1} + \eta_t \nabla_{\alpha_{i,C}^{t-1}} L2_i^C(\alpha_{i,C}^{t-1}), \tag{17a}$$

$$\alpha_{i,C}^t = \sum_{j \in \mathcal{E}_i} a_{ij} \phi_{i,C}^t. \tag{17b}$$

The update of parameters in (15) and (17) is motivated by distributed stochastic gradient descent algorithm over networks [28], [31], [32]. Each agent runs a gradient descent step using its own dataset. Then, the combination step can be considered as a procedure gradually collecting the global information of parameters. Therefore, the convergence value of the procedure (15) and (17) is not a local solution but the centralized one.

The gradient $\nabla_{\lambda_{i,C}} L1_i^C(\lambda_{i,C})$ in (15) can be estimated by the re-parameterization trick [12], [33], [34], that is

$$\begin{aligned}
&\nabla_{\lambda_{i,C}} L1_i^C(\lambda_{i,C}) \\
&\quad = \nabla_{\lambda_{i,C}} E_{\theta \sim h_{i,C}(\theta; \lambda_{i,C})}[\log q_i^{C-1}(\theta) - \log \tilde{p}(X_i|\theta) \\
&\qquad - \log p(\theta)] + \nabla_{\lambda_{i,C}} \frac{\delta}{2} ||h||_2^2 \\
&\quad \approx \frac{1}{L} \sum_{l=1}^{L} \nabla_{\lambda_{i,C}} L1_i^C(\theta_i^l),
\end{aligned} \tag{18}$$

where θ_i^l stands for the lth Monte Carlo sample, and L is the total number of Monte Carlo samples. Similarly, we have

$$\begin{aligned}
&\nabla_{\alpha_{i,C}} L2_i^C(\alpha_{i,C}) \\
&\quad = \nabla_{\alpha_{i,C}} E_{\theta \sim q_i^C(\theta; \alpha_{i,C})}[\log q_i^C(\theta) - \log \tilde{p}(X_i|\theta) \\
&\qquad - \log p(\theta)] \\
&\quad = E_{\theta \sim h_{i,C}} f(\alpha_{i,C}) - E_{\theta \sim q_i^{C-1}} f(\alpha_{i,C}),
\end{aligned} \tag{19}$$

where

$$f(\alpha_{i,C}) = \log \frac{(1 - \alpha_{i,C}) q_i^{C-1}(\theta) + \alpha_{i,C} h_{i,C}^*(\theta; \lambda_{i,C}^*)}{\tilde{p}(x_i|\theta) p(\theta)}.$$

The proposed distributed boosting variational inference (DBVI) algorithm is summarized in Algorithm 1. DBVI begins with a single component, where we use DVI [18] to fit the parameter $\lambda_{i,1}$ and set $\alpha_{i,1} = 1$. If $C > 1$, we fix the previous approximation q_i^{C-1}, and obtain the optimal $\lambda_{i,C}^*$ and $\alpha_{i,C}^*$ by minimizing $L1_i^C$ and $L2_i^C$.

Algorithm 1 Distributed Boosting Variational Inference (DBVI)

```
Input: Construct the posterior model $p(\theta|X)$, give a network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and local dataset $X_i$ at agent $i$, and denote the maximum of $C$ as $M$.
Output: Local approximation $q_i^M$ of each agent.
Start with $\lambda_{i,1}$ by DVI [18] with $\alpha_{i,1} = 1$;
for $C = 2 : M$ do
    Optimize $\lambda_{i,C}$ via (15);
    Optimize $\alpha_{i,C}$ via (17);
    $q_i^C = (1 - \alpha_{i,C}^*) q_i^{C-1} + \alpha_{i,C}^* h_{i,C}^*$;
end for
return Local approximation: $q_i^M$
```

Remark 1: Compared with BVI, the proposed algorithm (DBVI) is applicable for distributed data. By decomposing the centralized cost into a sum of local ones, variational boosting over networks is formulated as a distributed optimization problem. With the help of distributed stochastic gradient descent method [27], [28], local solution converges to the centralized one.

IV. DBVI WITH GAUSSIAN MIXTURE MODEL

In variational approximation field [17], [18], [35], [36], Gaussian mixture model has been widely used to approximate the multi-modality distribution. It has been proved that Gaussian mixture model can effectively approximate an arbitrary distribution [15]. In this section, Gaussian mixture model (GMM) is applied to the proposed algorithm.

A network with N agents is considered, and each agent i has N_i observations $x_{ij}(i = 1, 2, \cdots, N; j = 1, 2, \cdots, N_i)$. The Gaussian mixture model is described as

$$q_i^M(\theta) = \sum_{C=1}^{M} \alpha_{i,C} N_{i,C}(\theta; \mu_{i,C}, \sigma_{i,C}), \tag{20}$$

where $0 < \alpha_{i,C} \leq 1$, $\sum_{C=1}^{M} \alpha_{i,C} = 1,$, M is the total number of mixture components, $N_{i,C}(\theta; \mu_{i,C}, \sigma_{i,C})$ means the Cth component, and $\alpha_{i,C}$ is the corresponding weight. We first show how to find $N(\theta; \mu_{i,C}, \sigma_{i,C})$. Then, we optimize $\alpha_{i,C}$ with fixed $N(\theta; \mu_{i,C}^*, \sigma_{i,C}^*)$. Specifically, substituting $h_{i,C} = N(\theta; \mu_{i,C}, \sigma_{i,C})$ into (12) and (13), we have

$$\begin{aligned}
&L1_i^C(\mu_{i,C}, \sigma_{i,C}) \\
&\quad = E_{\theta \sim N(\mu_{i,C}, \sigma_{i,C})}[\log q_i^{C-1}(\theta) - \log \tilde{p}(X_i|\theta) \\
&\qquad - \log p(\theta)] - \frac{\delta}{4} |\sigma^T \sigma|,
\end{aligned} \tag{21}$$

$$L2_i^C(\alpha_{i,C}) \\ = E_{\theta\sim q_i^C}[\log q_i^C(\theta) - \log \tilde{p}(X_i|\theta) - p(\theta)]. \tag{22}$$

Now, our problem becomes

$$\begin{cases} \mu_{i,C}^*, \sigma_{i,C}^* = arg\min_{\mu_{i,C},\sigma_{i,C}} \sum_{i=1}^N L1_i^C(\mu_{i,C},\sigma_{i,C}) \\ \alpha_{i,C}^* = arg\min_{\alpha_{i,C}} \sum_{i=1}^N L2_i^C(\alpha_{i,C}), \end{cases} \tag{23}$$

which can be solved by (15)- (19).

The details of DBVI with Gaussian mixture model is presented in Algorithm 2.

Algorithm 2 DBVI With Gaussian Mixture Model (DBVI-GMM)

```
Start with $q_{i,1}$ by DVI [18];
**for** C = 2:M **do**
    Obtain $\mu_{i,C}^*, \sigma_{i,C}^*$ via minimizing (21);
    Obtain $\alpha_{i,C}^*$ via minimizing (22);
    $q_i^C = (1-\alpha_{i,C}^*)q_i^{C-1} + \alpha_{i,C}^* h_{i,C}^*$;
**end for**
**return** local approximation: $q_i^M$.
```

Next, we present how to start algorithm 2 with the first component, $q_{i,1}(\theta;\lambda_{i,1}) = N(\theta;\mu_{i,1},\sigma_{i,1})$. The algorithm starts by DVI [18] with a single component to approximate to the posterior. Specifically,

$$\mu_{i,1}^*, \sigma_{i,1}^* = arg\min \frac{1}{N}\sum_{i=1}^N E_{\theta\sim q_{i,1}}[\log q_{i,1}(\theta) \\ - \log\tilde{p}(X_i|\theta) - \log p(\theta)]. \tag{24}$$

where $\log\tilde{p}(X_i|\theta) = N\cdot\log p(X_i|\theta)$.

Moreover, we briefly analyze the performance of the sequence $D_{KL}(q_i^C||p)$ generated by DBVI-GMM. From (5), we have

$$D_{KL}(q^C||p) - D_{KL}(q^{C-1}||p) \\ \leq \alpha_C^* \left\langle \log q^{C-1} - \log p, h_C^* - q^{C-1}\right\rangle. \tag{25}$$

Refer to the Triangle Condition in [24], we have

$$\langle \log q^{C-1} - \log p, h_C^* - q^{C-1}\rangle \\ = \int (h_C^* - q^{C-1})(\log q^{C-1} - \log p) \\ = D_{KL}(h_C^*||p) - D_{KL}(h_C^*||q^{C-1}) \\ -D_{KL}(q^{C-1}||p) \leq 0. \tag{26}$$

Then, we have $D_{KL}(q^C||p) \leq D_{KL}(q^{C-1}||p)$. From the properties of DSGD in [30], we know that q_i^C converges to q^C in theory. Hence, $D_{KL}(q_i^C||p)$ decreases continuously as C increases.

Remark 2: Compared with DVI-GMM [18], DBVI-GMM presents a new adaptive variational method, where a joint optimization on $\{\alpha_{i,C},(\mu_{i,C},\sigma_{i,C})\}_{C=1}^M$ is converted into a sequence of optimizations on $\{\alpha_{i,C},(\mu_{i,C},\sigma_{i,C})\}$ from $C=1$ to M. DBVI-GMM ensures that the approximation error decreases continuously with an increasing number of components, so that it is natural to adjust the number of mixture components by monitoring the training error.

V. SIMULATION RESULTS AND ANALYSIS

In this section, we verify the performance of the proposed algorithm on several examples. We mainly focus on whether local solution of DBVI can approach to the centralized one of BVI, and whether DBVI can capture the multi-modality posterior. We firstly consider an one-dimensional multi-modality posterior to test the performance of DBVI-GMM, and then use a real data (frisk dataset [37]) to test the performance of DBVI-GMM.

Here, a network with 6 agents is considered, where the topology is shown in Figure 2, and the adjacent weighted matrix A is defined as follows

$$A = \begin{bmatrix} 0.7 & 0.1 & 0 & 0 & 0 & 0.2 \\ 0.1 & 0.5 & 0.3 & 0 & 0 & 0.1 \\ 0 & 0.3 & 0.5 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0.5 & 0.3 & 0 \\ 0 & 0 & 0 & 0.3 & 0.5 & 0.2 \\ 0.2 & 0.1 & 0 & 0 & 0.2 & 0.5 \end{bmatrix}.$$

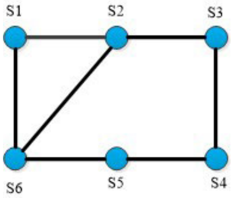

FIGURE 2. The Topology of Sensor Network.

A. ONE-DIMENSIONAL DISTRIBUTION

The target distribution is generated from the mixture of two Gaussian components. The corresponding parameters are setting as follows

$$\alpha = (\alpha_1,\alpha_2) = (0.75, 0.25), \\ \mu = (\mu_1,\mu_2) = (-3, 2), \\ \sigma = (\sigma_1,\sigma_2) = (2.5, 6). \tag{27}$$

In Figure 3, the background (grey shaded area) depicts the target distribution. DBVI-GMM starts by DVI-GMM with $C = 1$ to approximate the target distribution. As shown in Figure 3, local approximation with a single component can effectively capture the mainly part of target distribution.

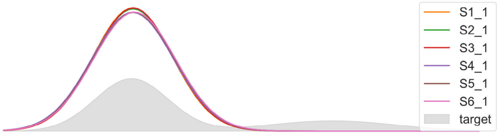

FIGURE 3. The approximation of one-diensional posterior with C = 1.

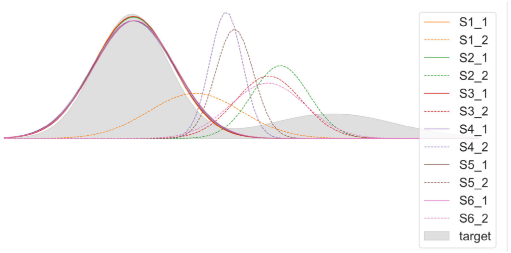

FIGURE 4. Initial new component at C = 2.

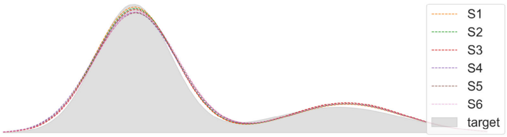

FIGURE 5. The approximation of one-dimensional posterior with C = 2.

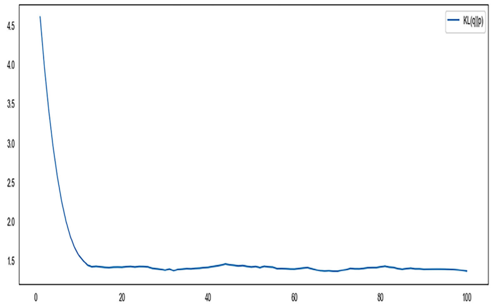

FIGURE 6. The cost (KL divergence) for the first component at S2.

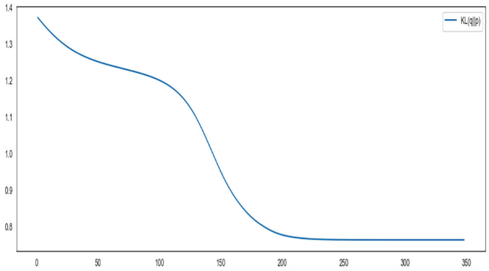

FIGURE 7. The cost (KL divergence) for the second component at S2.

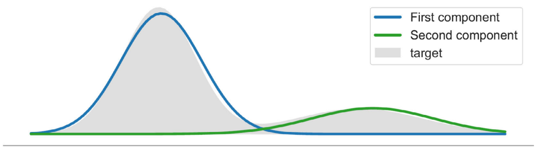

FIGURE 8. The approximation for the BVI with C = 2.

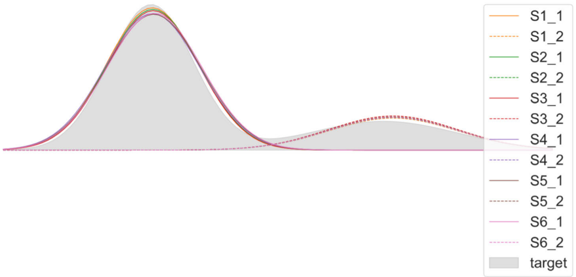

FIGURE 9. The approximation for the DVI-GMM with 2 components.

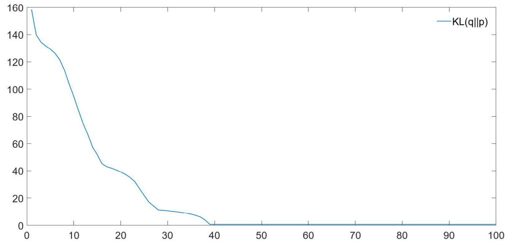

FIGURE 10. The cost (KL divergence) of the DVI-GMM with 2 components.

TABLE 1. The detail results of parameters.

Nodes	S1	S2	S3
α_1	0.747	0.746	0.745
(μ_1, σ_1)	(-2.899,1.964)	(-2.899,1.964)	(-2.899,1.9645)
α_2	0.253	0.254	0.255
(μ_2, σ_2)	(2.598,5.994)	(2.598,5.994)	(2.598,5.994)
Nodes	S4	S5	S6
α_1	0.739	0.739	0.739
(μ_1, σ_1)	(-2.899,1.965)	(-2.899,1.964)	(-2.899,1.965)
α_2	0.261	0.261	0.261
(μ_2, σ_2)	(2.598,5.994)	(2.595,5.994)	(2.598,5.994)
Algorithms	Target	DVI-GMM	
α_1	0.75	0.740	
(μ_1, σ_1)	(-3,2)	(-2.901, 1.97)	
α_2	0.25	0.260	
(μ_2, σ_2)	(2.5,6)	(2.606, 6.01)	

Next, a new component is initialized randomly for $C = 2$, as shown in Figure 4. We find that the value of the first component decreases. The reason is that the mixture weight of the first component decreases due to the addition of a new component. We fit the new component by DBVI-GMM, and the solution with $C = 2$ is shown in Figure 5. It can be noticed that the mixture model with $C = 2$ already achieves a good approximation. Figures 6 - 7 show the evolution of the cost (KL divergence) for the first and second component, respectively. It is noticed that the cost decreases with the introduction of new component. Figures 5-7 show that DBVI-GMM can effectively approximate the multi-modality distribution. As we see from the curve shown in Figure 5 and Figure 8, each agent can obtain the centralized solution.

As a comparison, we compare the performance with DVI-GMM in [18], [20]. Since we do not know the proper number of mixture components, it needs to rerun the DVI-GMM with different C. Figure 9 shows the approximation of DVI-GMM with 2 components, and Figure 10 plots the corresponding cost curve. It can be found that each agent also approximates the target distribution successfully. The detail results of DBVI-GMM, BVI and DVI-GMM are presented in Table 1.

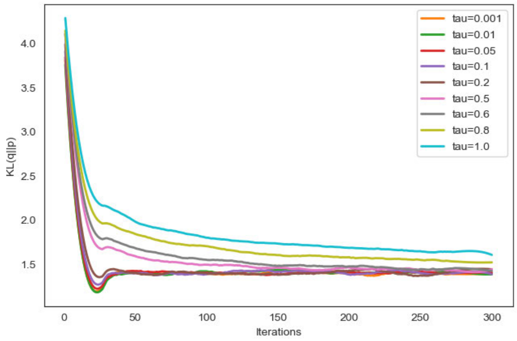

FIGURE 11. The average cost of the DBVI-GMM with different τ.

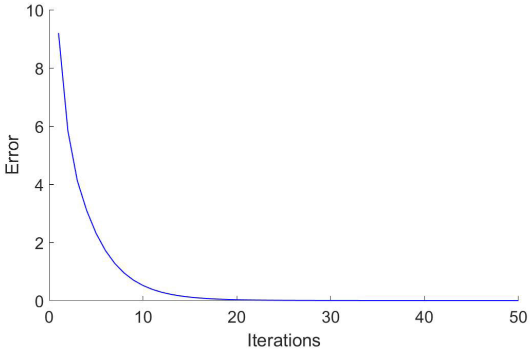

FIGURE 12. The evolution of the consistency error with different iterations.

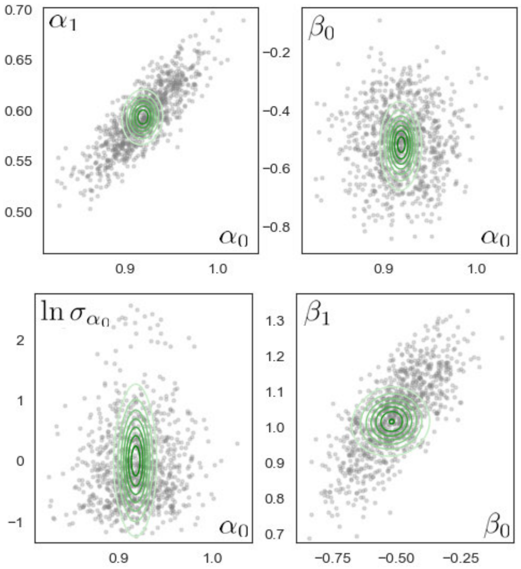

FIGURE 13. The approximations for the BVI with C = 21.

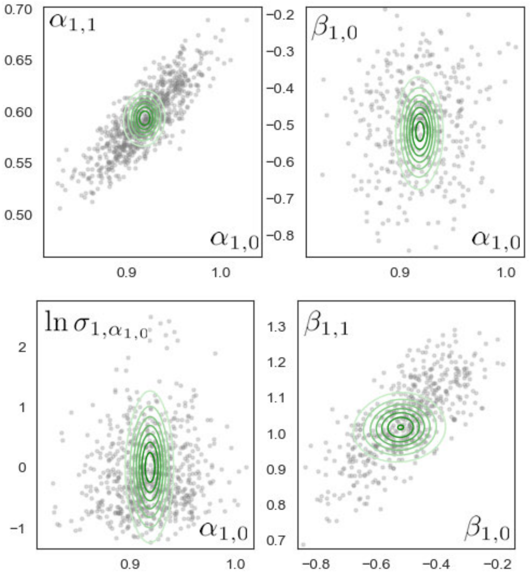

FIGURE 14. Local approximations of agent 1 for the DBVI-GMM with C = 21.

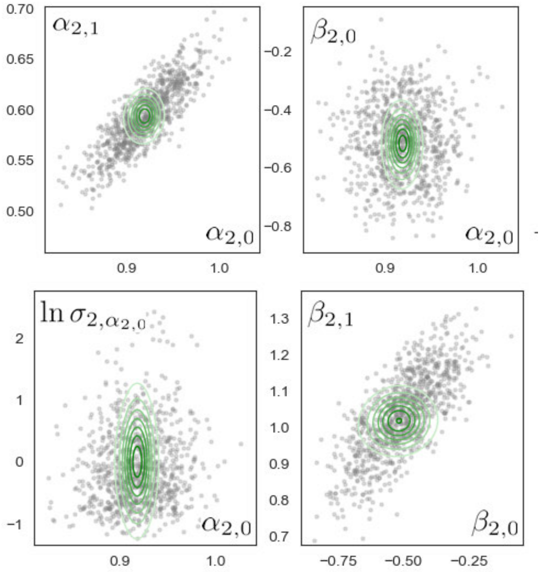

FIGURE 15. Local approximations of agent 2 for the DBVI-GMM with C = 21.

Moreover, we analyze the influence of step size $\eta_t = 1/(d + \tau t)$ to the performance of DBVI-GMM. Figure 11 shows the average cost of all agents for different τ with the same initialization after 300 iterations. It is found that the optimal setting of τ is near $\tau = 0.01$ with $d = 1$. In addition, we analyze the proper iteration of the consistency step in (15) and (17). As we see in Figure 12, the larger iteration is, the smaller the consistency error becomes. The consistency error approach to zero when $Iteration > 20$.

B. MULTI-LEVEL POISSON GLM

In this subsection, we apply DBVI-GMM to approximate the posterior for a hierarchical possion GLM (generalize linear model). We randomly divide the frisk dataset [37] into six sub-datasets, and each agent gets a sub-dataset as its local dataset. We use a hierarchical Poisson GLM to measure the relative rates of stop-and-frisk events in different ethnicities and precincts [26]. Specifically, the model for agent i is

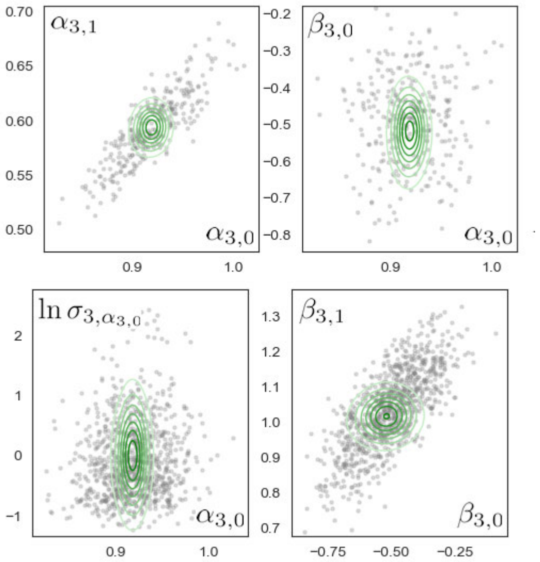

FIGURE 16. Local approximations of agent 3 for the DBVI-GMM with C = 21.

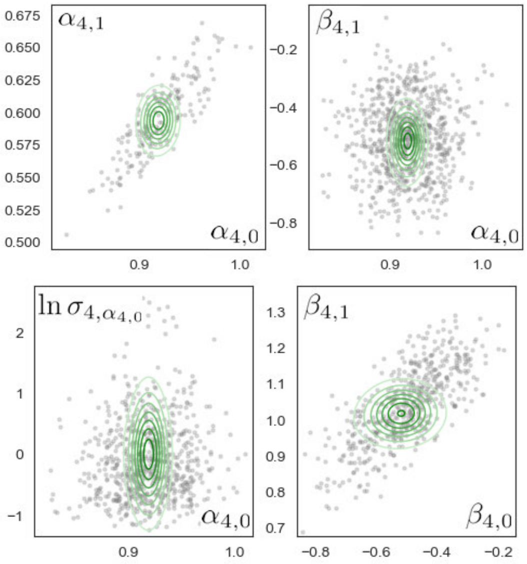

FIGURE 17. Local approximations of agent 4 for the DBVI-GMM with C = 21.

formulated as

$$
\begin{aligned}
\mu_i &\sim N(0, 10^2)\\
\ln\sigma_{i,\alpha}^2, \ln\sigma_{i,\beta}^2 &\sim N(0, 10^2)\\
\alpha_{i,e} &\sim N(0, \sigma_{i,\alpha}^2)\\
\beta_{i,p} &\sim N(0, \sigma_{i,\alpha}^2)\\
\ln\lambda_{i,ep} &= \mu_i + \alpha_{i,e} + \beta_{i,p} + \ln N_{i,ep}\\
Y_{i,ep} &= P(\lambda_{i,ep}),
\end{aligned}
\tag{28}
$$

where $Y_{i,ep}$ are the number of stop-and-frisk events within ethnicity group e and precinct p, $N_{i,ep}$ is the total number of arrests, $\alpha_{i,e}$ and $\beta_{i,e}$ are the ethnicity and precinct effects.

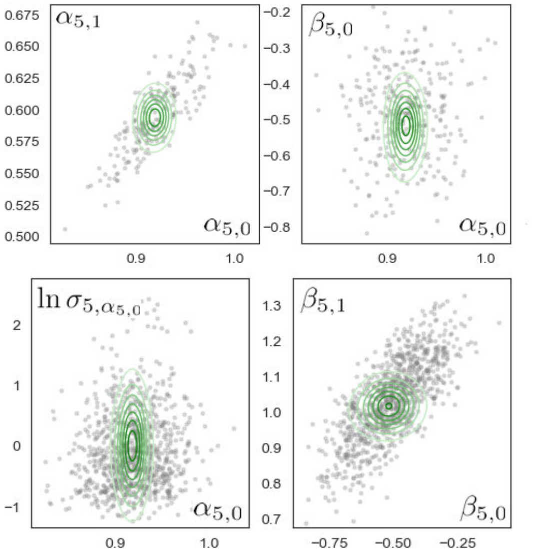

FIGURE 18. Local approximations of agent 5 for the DBVI-GMM with C = 21.

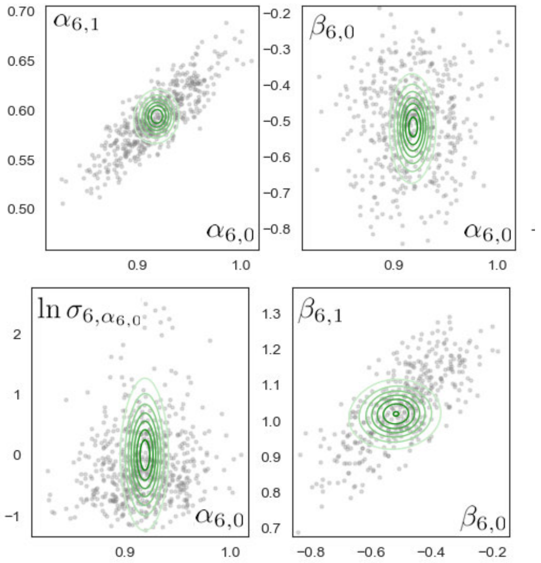

FIGURE 19. Local approximations of agent 6 for the DBVI-GMM with C = 21.

Since the posterior is a 37-dimensional Poisson GLM, we just present a handful of bivariate marginals for the frisk model in this simulations as shown in Figures 13-20, where the different axes represent different attributes pairs including (α_0, α_1), (α_0, β_0), $(\alpha_0, \ln\sigma_{\alpha_0})$, (β_0, β_1).

We set $M = 21$, $\tau = 0.01$, $d = 1$, $iteration = 20$ and train the model using Algorithm 2. For each single component, we take 200 gradient descent steps in (15) and (17), and 100 samples to estimate the gradient in (18) and (19). Figure 13 shows the approximation of BVI with $M = 21$. Figures $14 - 19$ show local approximation of each agent, respectively. It can be found that local approximation is almost the same as the centralized solution.

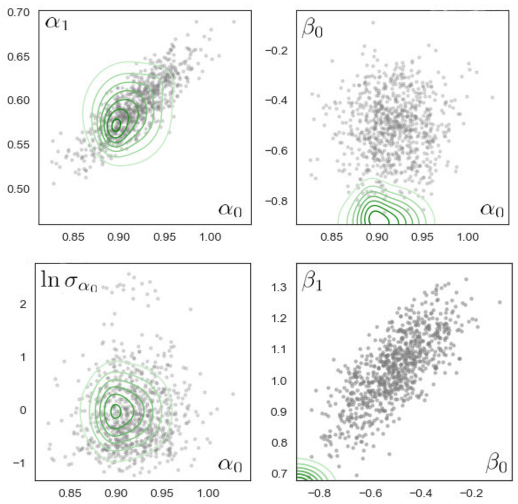

FIGURE 20. The approximation for the DVI-GMM with 3 components.

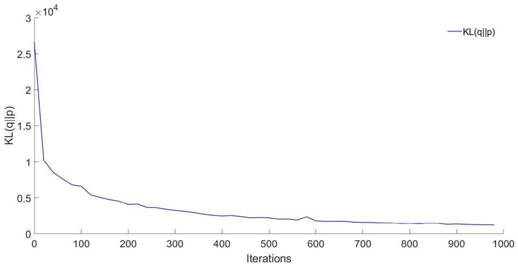

FIGURE 21. The cost(KL divergence) of the DVI-GMM with 3 components.

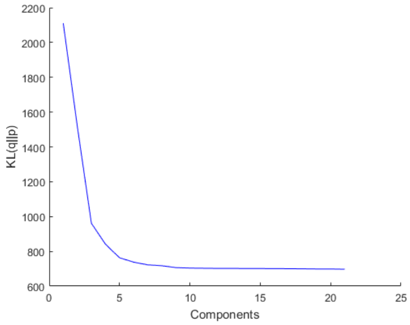

FIGURE 22. The cost of the DBVI-GMM with different components.

Moreover, we present the evolution of the cost ($KL(q||p)$) for DBVI-GMM with an increasing number of mixture components in Figure 22.

Compared with DBVI-GMM in order to seek a good approximation, DVI-GMM needs to have a try with different values of M, such as $M = 5, 10, 15\ldots$. If we consider a DVI-GMM model with 21 components as comparison, it means that 1575 parameters would be optimized together, and it is difficult to obtain the optimal solution. Hence, we just give the solution of DVI-GMM with 3 components in Figure 20, and present the evolution of the cost in Figure 21. It can be noticed that the approximation of DVI-GMM is not as well as that of DBVI-GMM. Compared with DVI-GMM, DBVI-GMM adds the new component one-at-a-time, and it is natural to adjust the number of mixture components adaptively by tracking the training error. Hence, DBVI-GMM can effectively deal with the challenges in DVI-GMM from the computation.

VI. SUMMARY AND CONCLUSION

In this paper, we proposed a distributed boosting variational inference (DBVI) algorithm to approximate the posterior over networks. The mixture model was used to approximate the complicated posterior, and the components of mixture model were trained one-at-a-time in a distributed form. For each new component, we formulated the global cost as a *sum-of-costs* form, and each agent obtained the centralized solution by distributed stochastic gradient descent method. In addition, we derived DBVI with Gaussian mixture model in detail. Simulations demonstrated that DBVI can perform almost as well as the centralized BVI and could effectively deal with the challenges in DVI in computation.

REFERENCES

[1] C. S. Raghavendra, K. M. Sivalingam, and T. Znati, *Wireless Sensor Networks*. New York, NY, USA: Springer, 2006.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] M. Tubaishat and S. Madria, "Sensor networks: An overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, Aug. 2003.

[4] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, 2002, pp. 41–47.

[5] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[6] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[7] V. V. Veeravalli and P. K. Varshney, "Distributed inference in wireless sensor networks," *Philos. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 370, no. 1958, pp. 100–117, 2012.

[8] D. Newman, P. Smyth, M. Welling, and A. U. Asuncion, "Distributed inference for latent Dirichlet allocation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1081–1088.

[9] F. Diez and J. Mira, "Distributed inference in Bayesian networks," *Cybern. Syst., Int. J.*, vol. 25, no. 1, pp. 39–61, 1994.

[10] H. Ge, Y. Chen, M. Wan, and Z. Ghahramani, "Distributed inference for Dirichlet process mixture models," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2276–2284.

[11] D. M. Blei, *Variational Inference*. Princeton, NJ, USA: Princeton Univ. Press, 2002, pp. 1–12.

[12] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, Apr. 2017.

[13] M. Kupperman, "Probabilities of hypotheses and information-statistics in sampling from exponential-class populations," *Ann. Math. Statist.*, vol. 29, no. 2, pp. 571–575, Jun. 1958.

[14] A. Kucukelbir, R. Ranganath, A. Gelman, and D. Blei, "Automatic variational inference in Stan," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 568–576.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[16] B. Wang and D. M. Titterington, "Inadequacy of interval estimates corresponding to variational Bayesian approximations," in *Proc. AISTATS*. Citeseer, 2005.

[17] S. Gershman, M. Hoffman, and D. Blei, "Nonparametric variational inference," 2012, *arXiv:1206.4665*. [Online]. Available: http://arxiv.org/abs/1206.4665

[18] J. Hua and C. Li, "Distributed variational Bayesian algorithms over sensor networks," *IEEE Trans. Signal Process.*, vol. 64, no. 3, pp. 783–798, Feb. 2016.

[19] Y. Gal, M. Van Der Wilk, and C. E. Rasmussen, "Distributed variational inference in sparse Gaussian process regression and latent variable models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3257–3265.

[20] B. Safarinejadian, M. B. Menhaj, and M. Karrari, "Distributed variational Bayesian algorithms for Gaussian mixtures in sensor networks," *Signal Process.*, vol. 90, no. 4, pp. 1197–1208, Apr. 2010.

[21] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[22] S. Mukherjee and H. Kargupta, "Distributed probabilistic inferencing in sensor networks using variational approximation," *J. Parallel Distrib. Comput.*, vol. 68, no. 1, pp. 78–92, Jan. 2008.

[23] B. Safarinejadian and M. B. Menhaj, "Distributed density estimation in sensor networks based on variational approximations," *Int. J. Syst. Sci.*, vol. 42, no. 9, pp. 1445–1457, Sep. 2011.

[24] X. Wang, "Boosting variational inference: Theory and examples," Ph.D. dissertation, Dept. Comput. Sci., Duke Univ., Durham, NC, USA, 2016.

[25] F. Guo, X. Wang, K. Fan, T. Broderick, and D. B. Dunson, "Boosting variational inference," 2016, *arXiv:1611.05559*. [Online]. Available: http://arxiv.org/abs/1611.05559

[26] A. C. Miller, N. J. Foti, and R. P. Adams, "Variational boosting: Iteratively refining posterior approximations," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2420–2429.

[27] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 69–77.

[28] Q. Meng, W. Chen, Y. Wang, Z.-M. Ma, and T.-Y. Liu, "Convergence analysis of distributed stochastic gradient descent with shuffling," *Neurocomputing*, vol. 337, pp. 46–57, Apr. 2019.

[29] E. Ozfatura, D. Gunduz, and S. Ulukus, "Speeding up distributed gradient descent by utilizing non-persistent stragglers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 2729–2733.

[30] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate Bayesian inference," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 4873–4907, 2017.

[31] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. New York, NY, USA: Springer, 2010, pp. 177–186.

[32] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, 2013.

[33] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114

[34] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*. [Online]. Available: http://arxiv.org/abs/1606.05908

[35] D. Gu, "Distributed EM algorithm for Gaussian mixtures in sensor networks," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1154–1166, Jul. 2008.

[36] Y. Weng, W. Xiao, and L. Xie, "Diffusion-based EM algorithm for distributed estimation of Gaussian mixtures in wireless sensor networks," *Sensors*, vol. 11, no. 6, pp. 6297–6316, Jun. 2011.

[37] A. Gelman, A. Kiss, and J. Fagan, "An analysis of the NYPD's Stop-And-Frisk Policy in the context of claims of racial bias," Columbia Public Law Res. Paper 05-95, 2006.

XIBIN AN was born in Tongxu, Henan, China, in 1993. He received the B.S. and M.S. degrees in control science and engineering from the Rocket Force University of Engineering, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Rocket Force University of Engineering. His research interests include ensemble machine learning, distributed optimization methods, and satellite orbit determination.

CHEN HU was born in Baoji, Shaanxi, China, in 1989. He received the B.S. degree in automation from Xiamen University, in 2011, and the M.S. and Ph.D. degrees in control science and engineering from the Rocket Force University of Engineering, in 2014 and 2018, respectively. His research interests include machine learning and distributed optimization methods.

GANG LIU was born in Fuyang, Anhui, China, in 1964. He received the B.S. and M.S. degrees in control science and engineering from the Rocket Force University of Engineering, in 1987 and 1990, respectively, and the Ph.D. degree in automation from Northwestern Polytechnical University, in 1999. His research interests include optimization methods and satellite orbit determinations.

MINGHAO WANG was born in Shandong, China, in 1984. He received the B.S., M.S., and Ph.D. degrees in control science and engineering from the Rocket Force University of Engineering, in 2006, 2009, and 2013, respectively. His research interests include robust control and artificial intelligence.