Trust-Region Based Stochastic Variational Inference for Distributed and Asynchronous Networks[*]

FU Weiming · QIN Jiahu · LING Qing · KANG Yu · YE Baijia

DOI: 10.1007/s11424-022-2085-5
Received: 7 February 2022 / Revised: 7 May 2022

Abstract Stochastic variational inference is an efficient Bayesian inference technology for massive datasets, which approximates posteriors by using noisy gradient estimates. Traditional stochastic variational inference can only be performed in a centralized manner, which limits its applications in a wide range of situations where data is possessed by multiple nodes. Therefore, this paper develops a novel trust-region based stochastic variational inference algorithm for a general class of conjugate-exponential models over distributed and asynchronous networks, where the global parameters are diffused over the network by using the Metropolis rule and the local parameters are updated by using the trust-region method. Besides, a simple rule is introduced to balance the transmission frequencies between neighboring nodes such that the proposed distributed algorithm can be performed in an asynchronous manner. The utility of the proposed algorithm is tested by fitting the Bernoulli model and the Gaussian model

FU Weiming
Department of Automation, University of Science and Technology of China, Hefei 230027*, China.*
Email: fwm1993@ustc.edu.cn.
QIN Jiahu (Corresponding author)
Department of Automation, University of Science and Technology of China, Hefei 230027*, China; Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei* 230088*, China.*
Email: jhqin@ustc.edu.cn.
LING Qing
School of Computer Science and Engineering, and Guangdong Province Key Laboratory of Computational Science, Sun Yat-Sen University, Guangzhou 510006, China. Email: lingqing556@mail.sysu.edu.cn.
KANG Yu
Department of Automation, University of Science and Technology of China, Hefei 230027*, China; Institute of Advanced Technology, University of Science and Technology of China, Hefei* 230027*, China.*
Email: kangduyu@ustc.edu.cn.
YE Baijia
Department of Automation, University of Science and Technology of China, Hefei 230027*, China.*
Email: yebj@mail.ustc.edu.cn.

[*]This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 61922076, 61873252, 61725304, and 61973324; in part by Guangdong Basic and Applied Basic Research Foundation under Grant No. 2021B1515020094; and in part by the Guangdong Provincial Key Laboratory of Computational Science under Grant No. 2020B1212060032.

[◇] *This paper was recommended for publication by Editor WU Zhengguang.*

to different datasets on a synthetic network, and experimental results demonstrate its effectiveness and advantages over existing works.

Keywords Asynchronous networks, Bayesian inference, distributed algorithm, stochastic variational inference, trust-region method.

1 Introduction

Bayesian inference provides an elegant way to dig hidden information in data. It has a wide range of applications, for example, finding topics for text corpora[1], clustering unlabeled data[2], and predicting unknown data[3]. Stochastic variational inference (SVI)[4] has made Bayesian inference more efficient on massive dataset by using noisy estimates of the objective's natural gradient based on subsamples. It is highly scalable because the update of parameters can be performed without passing through the whole dataset.

Traditionally, to perform SVI, one needs to centralize the data to a single machine. However, in the big data era, data is often distributed over different locations. Sometimes it is infeasible to send the complete data to a central machine due to reasons like heavy communication costs, memory limitations of the central machine, and data privacy or security issues[5–7]. Therefore, it is crucial to investigate the distributed implementation of SVI algorithms to suit the settings where data is collected by multiple nodes.

There have already existed some works devoted to developing distributed SVI algorithms[8–13]. Specifically, the authors in [8] proposed a distributed stochastic variational Bayesian algorithm (dSVB), where the global parameters are approximated by applying the stochastic gradient method followed by a diffusion step[14]. In [9], the alternating direction method of multipliers (ADMM)[15] is used to develop a distributed SVI algorithm, called ADMM-based networked SVI, by introducing redundant variables to decouple the local duplicates of global variables. While in [10], a new distributed SVI algorithm is proposed by applying the symmetric and doubly stochastic matrix to fuse the local parameters obtained from the natural gradient method. Note that the above distributed SVI algorithms[8–10] are designed for synchronous networks. To execute them in asynchronous networks, clock synchronization procedures are essential, which would, however, bring additional communication and computation costs[16]. The waiting time of each node is also inescapable in this case. In view of this, the asynchronous SVI (ASYSVI) algorithm with the master-slave architecture is proposed in [11] by adopting the asynchronous parallel stochastic gradient method[17]. The ASYSVI algorithm may suffer from a single point of failure since it is only the master that is in charge of maintaining the global parameters and pushing them to the slavers. In [12], a token-passing-based asynchronous SVI algorithm is developed without the aid of the central node, where a token containing the global parameters is passed through neighboring nodes. Considering that only the node possessing the token updates according to the SVI procedures, this algorithm has the drawbacks of low computing efficiency and huge wastes of resources. As a comparison, in the extreme stochastic variational inference (ESVI) proposed in [13], global parameters, though partial coordinates, are updated by each node and mixed through a message passing scheme. However, the same drawbacks remain especially when the number of mixture components is small. Therefore, it is still meaningful to develop effective techniques to desynchronize the SVI algorithms. In addition, the above-mentioned distributed SVI algorithms[8–13] are all developed based on the standard SVI[4], which is sensitive to the choice of hyperparameters and is prone to local optima[18, 19]. Thus, appropriate measures can be taken to enhance the performance of distributed SVI algorithms.

With the above considerations, a novel asynchronous distributed trust-region based SVI algorithm (TR-dSVI) for a general class of conjugate-exponential models is developed in this paper. The Metropolis rule[20] is used to diffuse the global parameters over the network and the trust-region method[21] is applied to update local parameters. Besides, we introduce a simple rule to balance the transmission frequencies between neighboring nodes such that the proposed distributed algorithm can be performed in an asynchronous manner. The superiorities of the TR-dSVI compared with existing ones[8–13] are summarized as follows. First, the TR-dSVI offers a more effective way to asynchronously and simultaneously perform parameter updates by applying the asynchronous diffusion method. In comparison with the asynchronous algorithms proposed in [12, 13], the whole global parameters in TR-dSVI are shared and updated without involving waiting times. Second, the trust-region updates, a more robust optimization algorithm, are applied to obtain the refined local parameters, which can shift the load from network communications to local computations. In this way, the performance of TR-dSVI can be improved in terms of saving resources and jumping out of local optima without sacrificing the speed and convenience of SVI. Third, simulation results of the proposed algorithm on different datasets with the Bernoulli model and the Gaussian model show that the proposed algorithm can get better performance than the centralized trust-region based SVI (TR-cSVI)[22], the dSVB[8], and the case without the trust-region updates (NG-dSVI).

The rest of this paper is arranged as follows. Section 2 introduces the SVI and its trust-region extension. Section 3 presents the TR-dSVI algorithm. Numerical experiments are given in Section 4 and conclusions are drawn in Section 5.

2 Preliminaries

In this section, we introduce the stochastic variational inference (SVI) and its trust-region extension briefly.

2.1 Basic Model

Consider N conditionally independent pairs of local hidden variables $\boldsymbol{y}_n$ and their corresponding observations $\boldsymbol{x}_n$, $n = 1, \cdots, N$, whose distribution is determined by global hidden variables β with fixed model parameters α. Their graphical model is shown in Figure 1.

We will restrict our attention to the conjugate-exponential models, which include many useful statistical models in the machine learning and statistics literature such as multivariate Bernoulli models, Bayesian mixture models, latent Dirichlet models, and hierarchical linear regression[23, 24]. More specifically, suppose that the prior distribution of β and the pairs of $\boldsymbol{y}_n$

and $\boldsymbol{x}_n$ belong to the following exponential families,

$$p(\beta) \propto \exp(\alpha^{\mathrm{T}} u(\beta) - A(\alpha)), \quad p(\boldsymbol{x}, \boldsymbol{y}|\beta) \propto \prod_n \exp(u(\beta)^{\mathrm{T}} f(\boldsymbol{x}_n, \boldsymbol{y}_n)), \tag{1}$$

where α is called the natural parameters, $u(\cdot)$ is a sufficient statistic function, $A(\cdot)$ is the log normalizer, $f(\cdot,\cdot)$ is a vector-valued function, $\boldsymbol{x} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$, and $\boldsymbol{y} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N\}$. In addition, the above two exponential-family distributions are assumed to satisfy the conjugacy condition, that is, the conditional distributions and their conjugate priors are in the same exponential family. More detailed properties of this model can be found in [4, 23, 24].

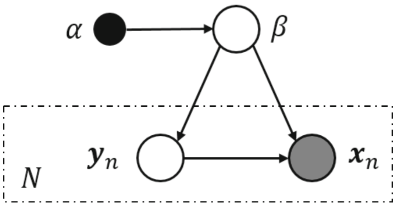


Figure 1 Graphical model considered in this paper

2.2 Stochastic Variational Inference

The variational inference aims to approximate the posterior distribution of the hidden variables given the observations $p(\boldsymbol{y}, \beta|\boldsymbol{x})$ by a factorial distribution $q(\boldsymbol{y}, \beta)$, which is computed by minimizing the Kullback-Leibler (KL) divergence between $q(\boldsymbol{y}, \beta)$ and $p(\boldsymbol{y}, \beta|\boldsymbol{x})$. Considering that the complexity of $q(\boldsymbol{y}, \beta)$ determines the complexity of this optimization, it is usually simplified by imposing the mean-field approximation assumption[23, 24], namely, $q(\boldsymbol{y}, \beta)$ is fully factorized over hidden variables as follows

$$q(\boldsymbol{y}, \beta) = q(\beta; \lambda) \prod_n q(\boldsymbol{y}_n; \phi_n) \tag{2}$$

with

$$q(\beta; \lambda) \propto \exp\left(\lambda^{\mathrm{T}} u(\beta) - A(\lambda)\right), \tag{3}$$

where ϕ_n denotes the local variational parameters of $\boldsymbol{y}_n$ and λ denotes the global variational parameters. Then, it is equivalent to maximizing the evidence lower bound (ELBO) defined as follows to achieve the variational inference,

$$\mathcal{L}(\lambda, \phi) = \mathbb{E}_q\left[\log \frac{p(\beta)}{q(\beta; \lambda)}\right] + \sum_{n=1}^{N} \mathbb{E}_q\left[\log \frac{p(\boldsymbol{x}_n, \boldsymbol{y}_{\boldsymbol{n}}|\beta)}{q(\boldsymbol{y}_n; \phi_n)}\right], \tag{4}$$

where $\phi = \{\phi_1, \cdots, \phi_N\}$.

Variational inference algorithms maximize the ELBO by updating iteratively between ϕ_n and λ. As a comparison, SVI applied the stochastic natural gradient algorithms to update the global variational parameters λ. Specifically, for a uniformly random chosen data $\boldsymbol{x}_n$, considering that

$$\mathcal{L}_n(\lambda, \phi_n) = \; N\mathbb{E}_q\left[\log \frac{p(\boldsymbol{x}_n, \boldsymbol{y}_n|\beta)}{q(\boldsymbol{y}_n; \phi_n)}\right] + \mathbb{E}_q\left[\log \frac{p(\beta)}{q(\beta; \lambda)}\right] \tag{5}$$

gives an unbiased estimation of $\mathcal{L}(\lambda, \phi)$, by applying the natural gradient of $\mathcal{L}_n(\lambda) \triangleq \max_{\phi_n} \mathcal{L}_n$ (λ, ϕ_n), SVI[4] updates λ as follows

$$\lambda^{t+1} = (1-\rho_t)\lambda^t + \rho_t \left(\alpha + N\mathbb{E}_{\phi_n^*}\left[f(\boldsymbol{x}_n, \boldsymbol{y}_n)\right]\right), \tag{6}$$

where ρ_t denotes the learning rate and $\phi_n^* = \arg\max_{\phi_n} \mathcal{L}_n(\lambda^t, \phi_n)$.

Note that the updating of the above procedure is based on a single data point, an extension to batches of multiple data points is straightforward and can be found in [4].

2.3 Trust-Region Method

The trust-region method is applied in [22] to replace the global variational parameters updates of the standard SVI for alleviating the issue of local optima, which replaces Equation (6) with the following trust-region step,

$$\lambda^{t+1} = \arg\max_{\lambda}\{\mathcal{L}_n(\lambda) - \xi_t D_{KL}(\lambda, \lambda^t)\}, \tag{7}$$

where $D_{KL}(\lambda, \lambda^t) = \mathbb{E}_\lambda[\log \frac{q(\beta;\lambda)}{q(\beta;\lambda^t)}]$ denotes the KL divergence between $q(\beta;\lambda)$ and $q(\beta;\lambda^t)$. Note that $\xi_t D_{KL}(\lambda, \lambda^t)$ is the regularization term that prevents the global variational parameters from changing too much in one stochastic update step.

Note that, for fixed ϕ_n^*, the natural gradient of the right-hand side of Equation (7) is given by[4, 22]

$$\alpha + N\mathbb{E}_{\phi_n^*}\left[f(\boldsymbol{x}_n, \boldsymbol{y}_n)\right] - \lambda + \xi_t(\lambda^t - \lambda). \tag{8}$$

Setting the natural gradient to zero yields

$$\lambda = (1-\rho_t)\lambda^t + \rho_t \left(\alpha + N\mathbb{E}_{\phi_n^*}\left[f(\boldsymbol{x}_n, \boldsymbol{y}_n)\right]\right), \tag{9}$$

where $\rho_t = (1+\xi_t)^{-1}$. Thus, Equation (7) can be solved approximately via alternating coordinate ascent by updating iteratively between λ according to Equation (9) and ϕ_n according to $\phi_n^* = \arg\max_{\phi_n} \mathcal{L}_n(\lambda, \phi_n)$.

3 Distributed Trust-Region Based SVI

In this section, we extend the trust-region based SVI to the distributed and asynchronous networks.

3.1 Problem Formulation

Consider a network consisting of J nodes, whose communication topology is presented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the node set $\mathcal{V} = \{1, \cdots, J\}$ and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Note that $(i, j) \in \mathcal{E}$ if nodes i and j can communicate with each other. We denote by $\mathcal{B}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ the neighbor set of node i. Suppose that $\mathcal{G}$ is connected, i.e., there exists a path connecting any two distinct nodes. In addition, each node j stores a set of N_j observations $\boldsymbol{x}_j = \{\boldsymbol{x}_{j1}, \cdots, \boldsymbol{x}_{jN_j}\}$, whose responding local hidden variables are $\boldsymbol{y}_j = \{\boldsymbol{y}_{j1}, \cdots, \boldsymbol{y}_{jN_j}\}$. Then the full data set is $\boldsymbol{x} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_J\}$, the full local hidden variable set is $\boldsymbol{y} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_J\}$, and the total number of observations is $N = \sum_j N_j$. We suppose that the model is the same as that in Subsection 2.1 with global hidden variable β and fixed model parameters α.

We consider a similar mean-field approximation model in Subsection 2.1, then the ELBO can be written as

$$\mathcal{L}(\lambda, \phi) = \mathbb{E}_q \left[\log \frac{p(\beta)}{q(\beta; \lambda)}\right] + \sum_{j=1}^{J} \sum_{i=1}^{N_j} \mathbb{E}_q \left[\log \frac{p(\boldsymbol{x}_{ji}, \boldsymbol{y}_{ji}|\beta)}{q(\boldsymbol{y}_{ji}; \phi_{ji})}\right], \tag{10}$$

where ϕ_{ji} is the local variational parameters of $\boldsymbol{y}_{ji}$. In addition, we let j be a variable taken from $\{1, \cdots, J\}$ uniformly at random and $\boldsymbol{x}_{jn}$ be a uniformly random chosen data in node j, then it is obvious that the following random function

$$\mathcal{L}_{jn}(\lambda, \phi_{jn}) \triangleq N\mathbb{E}_q \left[\log \frac{p(\boldsymbol{x}_{jn}, \boldsymbol{y}_{jn}|\beta)}{q(\boldsymbol{y}_{Rjn}; \phi_{jn})}\right] + \mathbb{E}_q \left[\log \frac{p(\beta)}{q(\beta; \lambda)}\right] \tag{11}$$

also gives an unbiased estimation of $\mathcal{L}(\lambda, \phi)$. Thus, the distributed SVI can be implemented by selecting a data uniformly at random in each node for updating.

3.2 Distributed Sampling and Diffusion

To perform the uniform sampling in distributed networks, the token-passing-based asynchronous SVI algorithm[12] is proposed according to the Metropolis Rule[20]. Specifically, the process can start with any node. The first node initializes the global variational parameters and updates them based on its local data, then packs them into a package and transfers the package to one of the neighbors following the transition probability matrix $P = [p_{ij}]$ defined by using the Metropolis rule:

$$p_{ij} = \begin{cases} \dfrac{1}{\max(|\mathcal{B}_i|, |\mathcal{B}_j|)}, & j \in \mathcal{B}_i, \\ 0, & j \notin \mathcal{B}_i \text{ and } j \neq i, \\ 1 - \displaystyle\sum_{k=1, k\neq i}^{J} p_{ik}, & j = i. \end{cases} \tag{12}$$

When a node receives the package, it updates the global variational parameters in the package based on local data, and then sends the package to a next node.

Remark 3.1 This approach actually defines a Markov chain Monte Carlo method[25] on the network and the probability of receiving the package for each node converges to $1/J$[26]. Thus, the uniform sampling of the nodes can be achieved. However, this approach is inefficient since only one node is selected at a time.

Motivated by the above method, a diffusion scheme, where all the nodes are involved in the parameter update, is applied to firstly obtain the distributed trust-region based SVI for synchronous networks. Letting $\mathcal{L}_{jn}(\lambda) = \max_{\phi_{jn}} \mathcal{L}_{jn}(\lambda, \phi_{jn})$, then one update of parameters for each node j can be presented in the following four steps:

1) Receive global variational parameters λ_i^t from the neighbors $i, \forall i \in \mathcal{B}_j$.

2) Select a data $\boldsymbol{x}_{jn}$ uniformly at random from $\boldsymbol{x}_j$.

3) Update the global variational parameters via

$$\lambda_j^{t+1} = \arg\max_{\lambda} \left\{ \mathcal{L}_{jn}(\lambda) - \xi_t \sum_{i \in \{j\} \cap \mathcal{B}_j} p_{ij} D_{KL}(\lambda, \lambda_i^t) \right\}. \tag{13}$$

4) Send new global variational parameters to the neighbors.

Similar to the centralized case, one can solve Equation (13) in Step 3) through multiple natural gradient updates. For fixed ϕ_{jn}^*, the natural gradient of the right-hand side of Equation (13) is given by

$$\alpha + N\mathbb{E}_{\phi_{jn}^*}\left[f(\boldsymbol{x}_j, \boldsymbol{y}_j)\right] - \lambda + \xi_t \sum_{i \in \{j\} \cap \mathcal{B}_j} p_{ij}(\lambda_i^t - \lambda). \tag{14}$$

Setting (14) to zero, then one has

$$\lambda = (1 - \rho_t)\Lambda_j^t + \rho_t(\alpha + N\mathbb{E}_{\phi_{jn}^*}[f(\boldsymbol{x}_j, \boldsymbol{y}_j)]), \tag{15}$$

where $\Lambda_j^t = \sum_{i \in \{j\} \cap \mathcal{B}_j} p_{ij}\lambda_i^t$ and $\rho_t = (1 + \xi_t)^{-1}$. Then, Equation (13) can be solved approximately by updating iteratively between λ_j according to Equation (15) and ϕ_{jn} according to $\phi_{jn}^* = \arg\max_{\phi_{jn}} \mathcal{L}_{jn}(\lambda, \phi_{jn})$.

Remark 3.2 In addition to addressing the issue of local optima[22], applying the trust-region method in communication networks can also save resources. This is because the main consumptions in executing distributed algorithms are the network communications, which are shifted to the local computations through multiple natural gradient updates in the trust-region method. Moreover, it is worth noting that additional overhead caused by multiple updates is often smaller than one might expect[22], since Equation (13) can be solved quickly for many models when λ is near convergence.

Remark 3.3 It can be seen that Λ_j^t fuses all the received variational parameters based on the Metropolis rule. Note that this approach can be viewed as the Markov chain Monte Carlo method with multiple chains, where each node sends the package, in the sense of expectation according to the transition probability matrix P, to its neighbors. Furthermore, since the network is connected and $\sum_i p_{ij} = \sum_i p_{ij} = 1$, one has $\lim_{t\to\infty} P^t = \frac{1}{J}\mathbf{1}\mathbf{1}^{\mathrm{T}}$[27]. Thus, by applying the diffusion scheme, each node can fuse the variational parameters of all the nodes.

3.3 Asynchronous Mechanism

Note that the global variational parameters of neighbors may not be received at each update for asynchronous networks. In this case, the network considered may be unconnected from the communication perspective. Nevertheless, it can be ensured that the network is jointly connected, that is, there exists some T such that the union of the interaction network across every interval $[t, t+T]$ is connected.

According to characteristics of the asynchronous communication, we define a new diffusion matrix $Q_t = [q_{ij}^t]$ by slightly modifying the Metropolis rule as follows

$$q_{ij}^t = \begin{cases} r_{ij}^t p_{ij}, & j \neq i, \\ 1 - \displaystyle\sum_{k=1, k\neq i}^{J} q_{ik}^t, & j = i, \end{cases} \tag{16}$$

where $r_{ij}^t = 1$ means that the global variational parameters of node i are received by node j at iteration t and $r_{ij}^t = 0$ otherwise. According to [27], given the jointly connected network, $\lim_{t\to\infty}\prod_{k=t_0}^{t} Q_k = \frac{1}{J}\mathbf{1}\mathbf{1}^{\mathrm{T}}$ if $\sum_i q_{ij}^t = \sum_i q_{ij}^t = 1$. However, it does not hold all the time since it is possible that $r_{ij}^t \neq r_{ji}^t$. Thus, we introduce a simple asynchronous mechanism to balance the transmission frequencies between neighboring nodes such that this condition can be satisfied on average. Specifically, for each node j, denoting the numbers of sending and receiving information to/from its neighbor i respectively as n_{ji} and n_{ij}, then node j will send the new parameters to neighbor i with probability 1 if $n_{ji} \leq n_{ij}$, with probability $\frac{0.5 n_{ij}}{n_{ji}}$ otherwise. Then, the TR-dSVI algorithm with the asynchronous mechanism can be summarized as shown in Algorithm 1.

Algorithm 1 Distributed trust-region based SVI: For node j

```
Input: Dataset $\boldsymbol{x}_j$, neighbor set $\mathcal{B}_j$, model parameters $\alpha$.
Output: Variational parameters $\lambda_j$ and $\phi_j$.
Compute $p_{ij} = \frac{1}{\max\{|\mathcal{B}_i|,|\mathcal{B}_j|\}}, i \in \mathcal{B}_j$.
Initialize $n_{ij} = r_{ij} = 0, i \in \mathcal{B}_j$.
Initialize $\lambda_j$.
Send $\lambda_j$ to neighbor $i$ and set $n_{ji} = 1$ for all $i \in \mathcal{B}_j$.
repeat
  for $i \in \mathcal{B}_j$ do
    if Received $\lambda_i$ from the neighbor $i$ then
      $n_{ij} = n_{ij} + 1$.
      $r_{ij} = 1$.
    end if
  end for
  Compute $\Lambda_j = \lambda_j + \sum_{i\in\mathcal{B}_j} r_{ij} p_{ij}(\lambda_j - \lambda_i)$.
  Reset $r_{ij} = 0$ for all $i \in \mathcal{B}_j$.
  Select $\boldsymbol{x}_{jn}$ uniformly from $\boldsymbol{x}_j$ and initialize $\phi_{jn}^*$.
  repeat
    $\lambda = (1-\rho_t)\Lambda_j + \rho_t\left(\alpha + N\mathbb{E}_{\phi_{jn}^*[f(\boldsymbol{x}_j,\boldsymbol{y}_j)]}\right)$.
    $\phi_{jn}^* = \arg\max_{\phi_{jn}} \mathcal{L}_{jn}(\lambda, \phi_{jn})$.
  until Convergence
  $\lambda_j = \lambda$.
  for $i \in \mathcal{B}_j$ do
    if $n_{ji} \leq n_{ij}$ then
      Send $\lambda_j$ to neighbor $i$ and update $n_{ji} = n_{ji} + 1$.
    else
      Send $\lambda_j$ to neighbor $i$ and update $n_{ji} = n_{ji} + 1$ with probability $\frac{0.5 n_{ij}}{n_{ji}}$.
    end if
  end for
until Convergence
```

Remark 3.4 Note that it is also straightforward to extend Algorithm 1 to the case with batches of multiple data points according to [4], which we omit due to space limitations.

4 Numerical Experiments

In this section, we apply the TR-dSVI to the Bernoulli model and the Gaussian model, and demonstrate its utility on different datasets. The asynchronous network under consideration is generated with 50 nodes located in a 5 × 5 sized square randomly by setting the communication distance to 1.2. The constructed network is depicted in Figure 2, which is apparently connected.

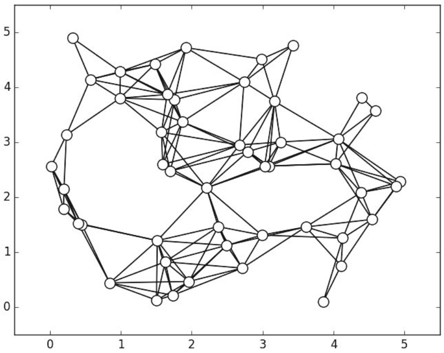


Figure 2 Network topology

4.1 Bernoulli Model

A mixture of multivariate Bernoulli distributions with K components is considered. The global parameters consist of component parameters $\beta_k \in [0,1]^D$ and mixture probabilities π, which satisfy Beta and Dirichlet distributions, with parameters a, b, and α, as follows,

$$p(\beta) \propto \prod_k \prod_d \beta_{kd}^{a-1}(1-\beta_{kd})^{b-1} \quad \text{and} \quad p(\pi) \propto \prod_k \pi_k^{\alpha-1},$$

where $\beta = \{\beta_1, \cdots, \beta_K\}$, $\beta_k = [\beta_{k1}, \cdots, \beta_{kD}]$, and $\pi = \{\pi_1, \cdots, \pi_K\}$. Under this model, the probability density function of any data $\boldsymbol{x}_{jn}$ can be written as

$$p(\boldsymbol{x}_{jn}|\boldsymbol{y}_{jn}, \beta, \pi) = \prod_k p(\boldsymbol{x}_{jn}|\beta_k)^{y_{jnk}} \propto \prod_k \prod_d \beta_{kd}^{y_{jnk}x_{jnd}}(1-\beta_{kd})^{y_{jnk}(1-x_{jnd})},$$

where

$$p(\boldsymbol{y}_{jn}|\pi) \propto \prod_k \pi_k^{y_{jnk}}.$$

Then the factorial distribution has the following form,

$$q(\boldsymbol{y}, \beta, \pi) = q(\beta)q(\pi)\prod_j \prod_n q(\boldsymbol{y}_{jn}; \phi_{jn}),$$

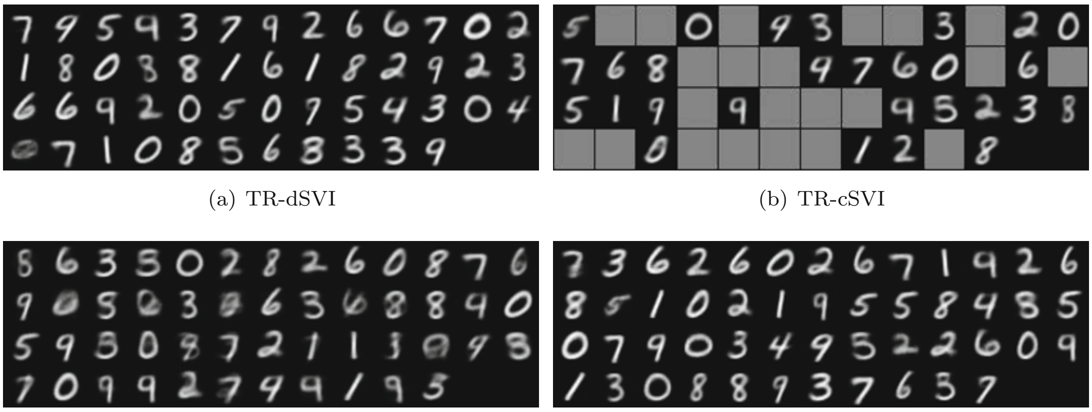
where

$$q(\beta) \propto \prod_k \prod_d \beta_{kd}^{a_{kd}-1}(1-\beta_{kd})^{b_{kd}-1}, \quad q(\pi) \propto \prod_k \pi_k^{\gamma_k - 1}, \quad \text{and} \quad q(\boldsymbol{y}_{jn}|\phi_{jn}) \propto \prod_k \phi_{jn}^{y_{jnk}}$$

with global variational parameters $\{a_{kd}\}, \{b_{kd}\}, \{\gamma_k\}$ and local variational parameters $\{\phi_{jn}\}$.

We apply the Bernoulli model to the Modified National Institute of Standards and Technology (MNIST) dataset, which is a dataset of handwritten digits containing 60000 training instances. Each instance is an image of 28 × 28 pixels with 256 gray levels per pixel. To fit the Bernoulli model, each pixel is binarized. We consider 50 components and set $a = b = \alpha = 1$. The complete dataset is randomly divided into 50 parts and each part is stored in one node. Each node initializes a_{kd} and b_{kd} by sampling them from a Gamma distribution with shape parameter 100 and scale parameter 0.01, and initializes γ_k to 1. We use multiprocess to run the TR-dSVI with 100 epochs and 10 inner loop iterations, and also run the TR-cSVI[22], the dSVB[8], and the case without the trust-region updates (NG-dSVI) for comparisons. We use a batch size of 200 for three distributed algorithms and 10000 for the centralized algorithm, and use $\rho_t = (\tau + t)^{-\kappa}$ as the learning rate for all the four algorithms, where $\kappa = 0.5$ and $\tau = 100$.

Figure 3 shows the cluster centers (defined by the expected values of the probabilities under the posterior approximations) found by applying the four algorithms on the MNIST dataset. It can be seen that the TR-dSVI, the dSVB, and the NG-dSVI can find more mixture components than the TR-cSVI, which implies that these distributed SVIs have more potential to identify all the latent patterns than the TR-cSVI. Furthermore, one can also obtain that the cluster centers found by the dSVB are blurrier rather than that found by the TR-dSVI and the NG-dSVI. One possible reason is that the diffusion scheme used in this paper can get better fusing performance than that used in the dSVB.



(a) TR-dSVI

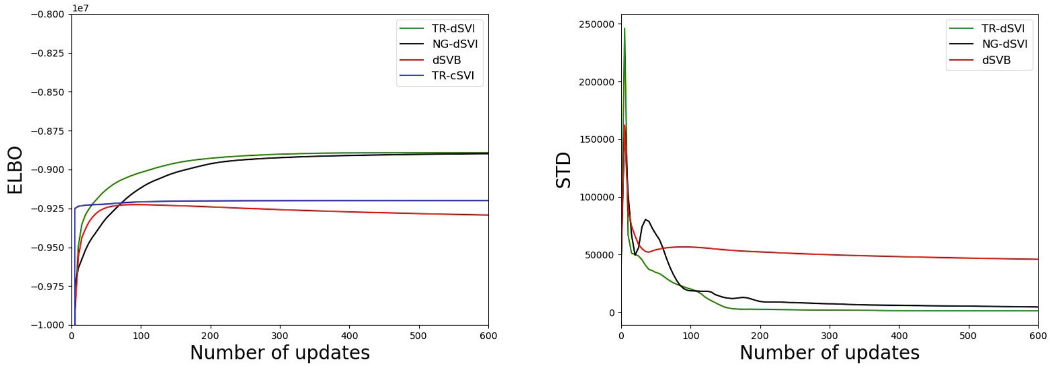
(b) TR-cSVI

(c) dSVB

(d) NG-dSVI

Figure 3 The cluster centers found by applying the TR-dSVI, the TR-cSVI, the dSVB, and the NG-dSVI on the MNIST dataset

Figure 4 depicts the evolution of the ELBO obtained by the four algorithms and the evolution of the standard deviation of the ELBO over all the nodes obtained by the three distributed algorithms in the MNIST dataset. One can observe from Figure 4(a) that although the TR-dSVI converges slower than the TR-cSVI and the dSVB, it can converge to a better value. Besides, through comparing the convergence speeds and the convergence values obtained by the TR-dSVI and the NG-dSVI in Figure 4(a), we can conclude that using trust-region updates can improve the clustering performance in terms of reducing communications and jumping out of local optima. Furthermore, among the three distributed algorithms, the TR-dSVI can achieve the best fusing performance over the network since it obtains the smallest standard deviation of the ELBO over all the nodes according to Figure 4(b).



(a) Evolution of ELBO (b) Evolution of ELBO's standard deviation

Figure 4 The evolution of the ELBO and its standard deviation over all the nodes obtained by applying the TR-dSVI, the TR-cSVI, the dSVB, and the NG-dSVI on the MNIST dataset

4.2 Gaussian Model

A mixture of multivariate Gaussian distributions with K components is considered. The global parameters consist of component mean $\mu = [\mu_1, \cdots, \mu_k]$, component precision matrix $\boldsymbol{T} = [\boldsymbol{T}_1, \cdots, \boldsymbol{T}_K]$, and mixture probabilities $\pi = \{\pi_1, \cdots, \pi_K\}$, which satisfy Normal, Wishart, and Dirichlet distributions, with parameters $\alpha, \beta_0, \mu_0, \boldsymbol{m}_0$, and $\boldsymbol{W}_0$, as follows,

$$p(\mu|\boldsymbol{T}) \propto \prod_{k=1}^{K} \mathcal{N}(\mu_k; \boldsymbol{m}_0, (\beta_0 \boldsymbol{T_k})^{-1}), \quad p(\boldsymbol{T}) \propto \prod_{k=1}^{K} \mathcal{W}(\boldsymbol{T_k}; \boldsymbol{W}_0, \nu_0), \quad \text{and} \quad p(\pi) \propto \prod_{k} \pi_k^{\alpha-1}.$$

Under this model, the probability density function of any data $\boldsymbol{x}_{jn}$ can be written as

$$p(\boldsymbol{x}_{jn}|\boldsymbol{y}_{jn}, \mu, \boldsymbol{T}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_{jn}; \mu_k, \boldsymbol{T}_K^{-1})^{y_{jnk}},$$

where

$$p(\boldsymbol{y}_{jn}|\pi) \propto \prod_{k} \pi_k^{y_{jnk}}.$$

Then the factorial distribution has the following form,

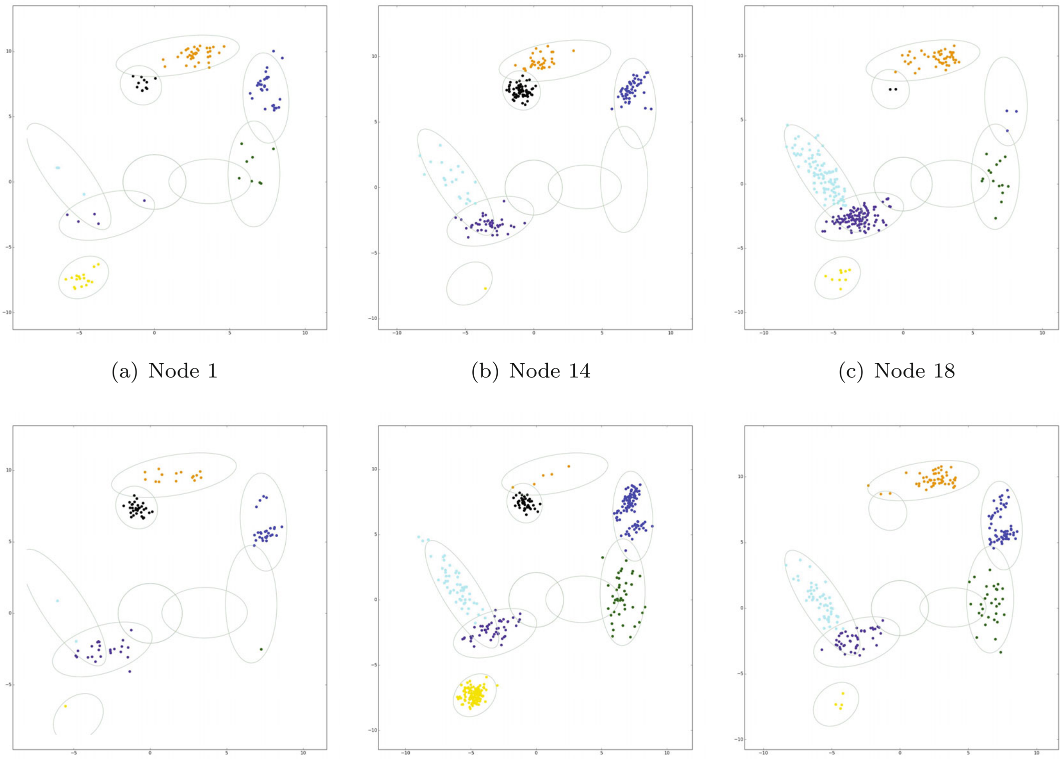
$$q(\boldsymbol{y}, \pi, \mu, \boldsymbol{T}) = q(\pi) \prod_{k} q(\mu_k, \boldsymbol{T}_k) \prod_{j} \prod_{n} q(\boldsymbol{y}_{jn}; \phi_{jn}),$$

where

$$q(\pi) \propto \prod_k \pi_k^{\gamma_k - 1}, \; q(\mu_k, \boldsymbol{T}_k) \propto \mathcal{N}(\mu_k; \boldsymbol{m}_k, (\beta_k \boldsymbol{T}_k)^{-1}) \mathcal{W}(\boldsymbol{T}_k; \boldsymbol{W}_k, \nu_k), \text{ and } q(\boldsymbol{y}_{jn} | \phi_{jn}) \propto \prod_k \phi_{jn}^{y_{jnk}}$$

with global variational parameters $\{\gamma_k\}, \{\boldsymbol{m}_k\}, \{\beta_k\}, \{\boldsymbol{W}_k\}, \{\nu_k\}$ and local variational parameters $\{\phi_{jn}\}$.

We first fit the Gaussian model to a synthetic dataset composed of 10000 2-dimensional instances generated from 10 Gaussian components. The number of data points in each node is not equal and follows a multinomial distribution. We set $\kappa = 0.5$, $\tau = 10$, $\alpha = 0.2$, $\beta_0 = 0.1$, $\nu_0 = 2$, $m_0 = [0, 0]$, and $\boldsymbol{W}_0 = \boldsymbol{I}_2$, and treat all the data in a single node as a subsample. A fixed number of 10 iterations are used for trust-region updates. Figure 5 shows the clustering results obtained in randomly chosen six nodes by applying the TR-dSVI on the Gaussian synthetic dataset after 2000 iterations, which can demonstrate the good clustering performance of the TR-dSVI. Figure 6 shows the evolution of the ELBO obtained by the four algorithms and the evolution of the standard deviation of the ELBO over all the nodes obtained by the three distributed algorithms in the Gaussian synthetic dataset. It can be seen that when being applied to Gaussian models, the same conclusions can be drawn as that to the MNIST dataset, with the exception that the fusing performance improvement is not significant in the TR-dSVI.


(a) Node 1

(b) Node 14

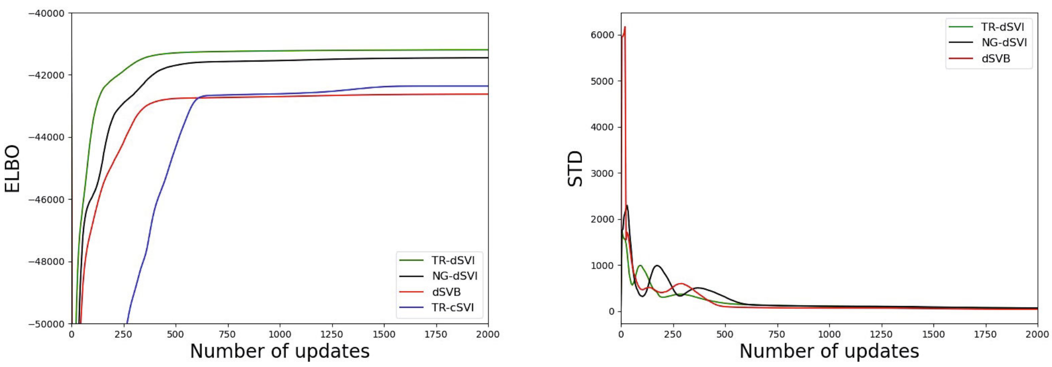
(c) Node 18

(d) Node 25

(e) Node 36

(f) Node 48

Figure 5 Clustering results obtained in six different nodes by applying the TR-dSVI on the Gaussian synthetic dataset



(a) Evolution of ELBO (b) Evolution of ELBO's standard deviation

Figure 6 The evolution of the ELBO and its standard deviation over all the nodes obtained by applying the TR-dSVI, the TR-cSVI, the dSVB, and the NG-dSVI on the Gaussian synthetic dataset.

Second, we fit the Gaussian model to the CIFAR-10 dataset, which consists of 60000 32×32 color images in 10 classes. We extract the RGB hist feature from each image and further apply the principal component analysis (PCA) to reduce the dimension to 32. We use $\kappa = 0.5$, $\tau = 10$, and a fixed number of 5 iterations for trust region updates. A batch size of 100 and 5000 are used respectively for the TR-dSVI and the TR-cSVI. Figure 7 presents the evolution of the ELBO obtained by applying the TR-dSVI and the TR-cSVI on the CIFAR-10 dataset, from which we can see that the TR-dSVI still has better performance than the TR-cSVI on the CIFAR-10 dataset.

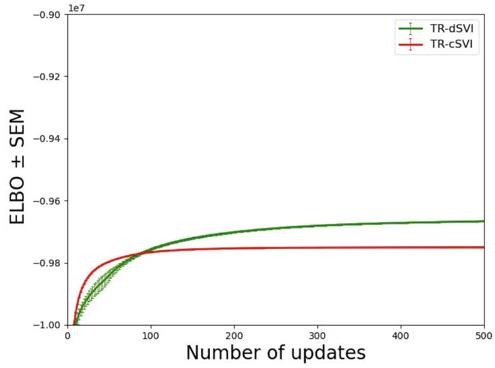


Figure 7 The evolution of the ELBO obtained by applying the TR-dSVI and the TR-cSVI on the CIFAR-10 dataset

5 Conclusions

In this paper, we proposed a trust-region based SVI algorithm for distributed and asynchronous networks and applied it to the Bernoulli model and the Gaussian model. Experiments show that it can obtain better performance than the centralized trust-region based SVI, the dSVB, and the case without trust-region updates.

References

[1] Blei D M, Ng A Y, and Jordan M I, Latent dirichlet allocation, *Journal of Machine Learning Research*, 2003, **3**: 993–1022.

[2] Corduneanu A and Bishop C M, Variational bayesian model selection for mixture distributions, *Artificial Intelligence and Statistics* (Eds. by Jaakkola T and Richardson T), Morgan Kaufmann, Waltham, 2001.

[3] Letham B, Letham L M, and Rudin C, Bayesian inference of arrival rate and substitution behavior from sales transaction data with stockouts, *Proceedings of the* 22*nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, 2016.

[4] Hoffman M D, Blei D M, Wang C, et al., Stochastic variational inference, *Journal of Machine Learning Research*, 2013, **14**(1): 1303–1347.

[5] Hu Y, Niu D, Yang J, et al., FDML: A collaborative machine learning framework for distributed features, *Proceedings of the* 25*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage, USA, 2019.

[6] Liu S, Pan S J, and Ho Q, Distributed multi-task relationship learning, *Proceedings of the* 23*rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax, Canada, 2017.

[7] Fang H, Liu S, and Wei Y, Distributed clustering algorithm for energy efficiency and load-balance in large-scale multi-agent systems, *Journal of Systems Science and Complexity*, 2018, **31**(1): 234–243.

[8] Hua J and Li C, Distributed variational Bayesian algorithms over sensor networks, *IEEE Transactions on Signal Processing*, 2015, **64**(3): 783–798.

[9] Anwar H and Zhu Q, ADMM-based networked stochastic variational inference, arXiv preprint, arXiv: 1802.10168, 2018.

[10] Fu W, Qin J, and Zhu Y, Distributed stochastic variational inference based on diffusion method, *Acta Automatica Sinica*, 2021, **47**(1): 92–99.

[11] Mohamad S, Bouchachia A, and Sayed-Mouchaweh M, Asynchronous stochastic variational inference, *Proceedings of INNS Big Data and Deep Learning Conference*, Sestri Levante, Italy, 2019.

[12] Ye B, Qin J, Fu W, et al., Distributed bayesian inference over sensor networks, *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2021.3106660, 2021.

[13] Zhang J, Raman P, Ji S, et al., Extreme stochastic variational inference: Distributed inference for large scale mixture models, *International Conference on Artificial Intelligence and Statistics*, Naha, Japan, 2019.

[14] Cattivelli F S and Sayed A H, Diffusion LMS strategies for distributed estimation, *IEEE Transactions on Signal Processing*, 2010, **58**(3): 1035–1048.

[15] Boyd S, Parikh N, and Chu E, *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*, Now Publishers Inc, Hanover, USA, 2011.

[16] García-Fernández Á F and Grajal J, Asynchronous particle filter for tracking using non-synchronous sensor networks, *Signal Processing*, 2011, **91**(10): 2304–2313.

[17] Niu F, Recht B, Re C, et al., HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent, *Proceedings of the* 24*th International Conference on Neural Information Processing Systems*, Granada, Spain, 2011.

[18] Hughes M C and Sudderth E, Memoized online variational inference for Dirichlet process mixture models, *Advances in Neural Information Processing Systems*, 2013, **26**: 1133–1141.

[19] Hoffman M D and Blei D M, Structured stochastic variational inference, *Proceedings of the* 18*th International Conference on Artificial Intelligence and Statistic*, San Diego, USA, 2015.

[20] Nedic A, Olshevsky A, and Shi W, Achieving geometric convergence for distributed optimization over time-varying graphs, *SIAM Journal on Optimization*, 2017, **27**(4): 2597–2633.

[21] Nocedal J and Wright S, *Numerical Optimization*, Springer Science & Business Media, New York, USA, 2006.

[22] Theis L and Hoffman M D, A trust-region method for stochastic variational inference with applications to streaming data, *Proceedings of the* 32*nd International Conference on Machine Learning*, Lille, France, 2015.

[23] Hoffman M and Blei D, Stochastic structured variational inference, *Proceedings of the* 18*th International Conference on Artificial Intelligence and Statistics*, San Diego, USA, 2015.

[24] Blei D M, Kucukelbir A, and McAuliffe J D, Variational inference: A review for statisticians, *Journal of the American Statistical Association*, 2017, **112**(518): 859–877.

[25] Betancourt M, The convergence of markov chain monte carlo methods: From the Metropolis method to Hamiltonian Monte Carlo, *Annalen der Physik*, 2019, **531**(3): 1700214.

[26] Durrett R, *Probability: Theory and Examples*, Cambridge University Press, New York, USA, 2019.

[27] Kingston D B and Beard R W, Discrete-time average-consensus under switching network topologies, *Proceedings of the* 2006 *American Control Conference*, Minneapolis, USA, 2006.