ADMM-based Distributed Stochastic Variational Inference

Hamza Anwar * **Quanyan Zhu** *

* *Electrical and Computer Engineering Department, New York University, Brooklyn, NY, 11201, USA (e-mail: ha1082@nyu.edu, qz494@nyu.edu)*

Abstract:
Owing to the recent advances in "Big Data" modeling and prediction tasks, variational Bayesian estimation has gained popularity due to their ability to provide exact solutions to approximate posteriors. One key technique for approximate inference is stochastic variational inference (SVI), Hoffman et al. (2013). SVI poses variational inference as a stochastic optimization problem and solves it iteratively using noisy gradient estimates. It aims to handle massive data for predictive and classification tasks by applying complex Bayesian models that have observed as well as latent variables. This paper aims to decentralize it allowing parallel computation, secure learning and robustness benefits. We use ADMM in a top-down setting to decentralize SVI algorithms such that independent learners running inference algorithms only require sharing the estimated model parameters instead of their private datasets. We illustrate the results on latent Dirichlet allocation (LDA) topic model in large document classification, compare performance with the centralized algorithm, and use numerical experiments to corroborate the analytical results.

Keywords: Bayesian methods, Nonparametric methods, Machine learning, Variational inference

1. INTRODUCTION

The explosive influx of data and information for modern day technological systems has opened doors to revolutionary possibilities. One of the most vital uses of this data is for modeling, visualizing, and analyzing large datasets through probabilistic tools. Statistical machine learning is at the core of numerous such applications in what is becoming known as Internet of Things (IoT). Such iterative learning mechanisms improve control performance for many cyber-physical-systems in which parameter estimations and system identifications are required. Probabilistic graphical modeling is one key research area that has played an important role in data analysis in inference and prediction tasks (see Koller and Friedman (2009)). These models visually express the assumptions about data and its hidden structure. Posterior inference algorithms have been proven to exploit such models in explaining this hidden structure while being adaptive, robust, parallelizable, and scalable.

Variational inference, from the late 90s, is a method that transforms complex inference problems into high dimensional optimization problems. In contrast to Monte-Carlo sampling methods (that simply aim to find the exact answer to an approximate problem), the variational Bayesian approach solves for an optimal solution under constraints to the right inference problem, Jordan et al. (1999). On the same lines, stochastic variational inference (SVI) was developed recently that extends variational inference to be solved using stochastic optimization under certain assumptions, Hoffman et al. (2013). SVI works iteratively in gradient ascent fashion using noisy gradient estimates. It provides approximate model posteriors with only a few passes through a large data collection, making it highly scalable. We propose ADMM-based SVI – a distributed stochastic variational inference technique that builds upon standard SVI, retaining most of its benefits, based on the highly parallelizable alternating direction method of multipliers (ADMM).

Related Work Numerous extensions have been proposed for the SVI framework in its application to more model classes (by Foti et al. (2014) and Johnson and Willsky (2014)), different underlying processes (by Hensman et al. (2013) and Gal et al. (2014)), and structural exploitations (by Hoffman and Blei (2015)) making it faster and widely deployable. A variety of works focuses on making variational methods distributed to enhance parallelizability. The work on distributed Bayesian nonparametric models by Campbell et al. (2015) is commendable in making variational inference updates distributed, asynchronous, and 'streaming' (online) and they have shown it to outperform standard SVI. However, their work is only specific to the Dirichlet process mixture and lacks in generalizability to the class of probabilistic models that SVI can deal with. Similar to Campbell et al. (2015), another work D-MFVI by Babagholami-Mohamadabadi et al. (2015) uses ADMM for decentralizing, like us, but lacks in being extendable to online updates, fast convergence rate, and other desirable properties of SVI. Distributed VBA (Hua and Li (2016)) also uses ADMM however their approach lacks in scalability to large data without demanding adequate computational resources.

None of these works use stochastic optimization methods to speed up inference and hence they are fundamentally different from standard SVI itself. One recent work on the extended-SVI by Raman et al. (2016) retains the benefits

of SVI while making it distributed and asynchronous. They employ a rather simple algorithmic change to SVI however their work remains unexplored in terms of depth because they particularly focus on Gaussian mixture models and do not provide how it is extendable to all other probabilistic models for which SVI in general works.

In contrast to all related works highlighted, our approach extends the general SVI framework to a distributed stochastic optimization consensus problem. We tackle the issue of generalizability to all graphical models by providing a general solution and make use of the stochastic gradient updates that make it fast. We run ADMM updates along with stochastic gradient ascent for variational objective to reach consensus among a number of distributed learners. We also discuss convergence properties and linear-time computational benefits for the complexity that ADMM adds to standard SVI. SVI itself being a non-convex stochastic optimization problem makes the distributed problem trickier. Proving convergence for this non-convex stochastic consensus problem is not trivial. Convergence for non-convex structured ADMM problems is a hot topic and recent works on it are fruitful to our analysis (Magnússon et al. (2014) and Hong et al. (2015)). In this paper, we argue for the theoretical existence of locally optimal equilibrium solution and show almost-sure algorithm convergence through numerical experiments. Our work aims to show that independent learners that use SVI for similar applications, can collaborate by exchanging their results (not the data itself) to benefit from each other improving overall accuracy of results. This approach makes our work unique and applicable to wider distributed large-scale inference problems. Furthermore, this kind of an approach poses a game problem with multiple agents interested in performing their inference tasks, and simultaneously benefiting from each other through reinforcement learning and cooperation.

The paper is organized as follows: firstly, we provide a summary of the working of stochastic variational inference followed by our proposed distributed SVI using ADMM in Section 2. We formulate the problem, pose conditions for optimality and present the SVI–ADMM algorithm in Section 3. Afterwards, we give insights to the algorithm's convergence properties in detail in Section 4. Finally, the implementations for distributed SVI along with numerical experiments and results for a massive document-classification problem are presented in Section 5. We conclude the paper in Section 6.

2. STOCHASTIC VARIATIONAL INFERENCE

SVI is an algorithm for stochastic optimization of mean-field variational inference. We will explain it as a review.

2.1 Model setup

We denote $x = x_{1:N}$ as observations, each observation x_n being a random vector, $x_{i,n} : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X}$ is the probability space corresponding to the actual unknown marginal distribution of the model $p(x)$. Likewise, $z = z_{1:N,1:J}$ are local hidden random variables (that govern the underlying relationship between data and model locally, i.e., we have J hidden variables for each of N observations) and $\beta = \beta_{1:K}$ as K global hidden variables, where random vector β_k has $\beta_{i,k} : \mathcal{B} \to \mathbb{R}$. α are fixed and known hyperparameters to start with – the particular choice of probability distributions of the latent random variables governs corresponding number of hyperparameters, for instance a one-dimensional Gaussian process exhibits two types of hyperparameters, the vertical and horizontal length-scales. The standard problem is to determine the posterior distribution that relates the data and the model:

$$p(z, \beta|x, \alpha) = \frac{p(z, x, \beta|\alpha)}{\int_{z,\beta} p(z, x, \beta|\alpha)}.$$

In general, the denominator term (marginal likelihood) is intractable to compute for many models especially when dataset is large.

A natural choice of tractable family of distributions Q is the exponential family. Moreover, the sufficient statistics do not increase in size with data points which makes it very stable and tractable. Distributions in exponential family are of the form:

$$p(\beta|x, z, \alpha) = h(\beta) \exp\left(\eta_g(x, z, \alpha)^\top t(\beta) - a_g(\eta_g(x, z, \alpha))\right),$$

where the global natural parameter η_g relates linearly with the sum of the local sufficient statistics of local parameters $t(x_n, z_n)$, and a_g, function of natural parameter, is the log-normalizer.

2.2 Variational inference

Standard variational inference finds an approximate posterior to the original posterior inference problem. It minimizes a measure of dissimilarity between two posteriors. The Kullback-Leibler divergence is used as that measure of dissimilarity. The idea is to choose a class of tractable distributions Q and find the probability distribution closest to the posterior distribution of the actual model given the observations:

$$\arg\min_{q \in Q} \quad \mathrm{KL}[q(z, \beta)||p(z, \beta|x)].$$

Since we have no knowledge about the form of $p(x, z, \beta)$, so we cannot directly minimize the objective. To tackle this, we find the evidence lower bound (ELBO), $\hat{\mathcal{L}}(q)$, of the posterior:

$$\begin{aligned}\log p(x) &= \log \int p(x, z, \beta) \frac{q(z, \beta)}{q(z, \beta)} dz d\beta = \log\Big(\mathbb{E}_q\Big[\frac{p(x, z, \beta)}{q(z, \beta)}\Big]\Big) \\ &\geq \mathbb{E}_q[\log p(x, z, \beta)] - \mathbb{E}_q[\log q(z, \beta)] =: \hat{\mathcal{L}}(q) \quad (1)\end{aligned}$$

ELBO gives a lower bound on the log-marginal likelihood. Maximizing ELBO is equivalent to minimizing the KL-divergence. The KL-divergence relates to ELBO through the following:

$$\begin{aligned}\mathrm{KL}[q(z, \beta)||p(z, \beta|x)] &= \mathbb{E}_q[\log q(z, \beta)] - \mathbb{E}_q[\log p(z, \beta|x)] \\ &= \mathbb{E}_q[\log q(z, \beta)] - \mathbb{E}_q[\log p(x, z, \beta)] + \log p(x) \\ &= -\hat{\mathcal{L}}(q) + \text{const.}\end{aligned}$$

With this formulation, the variational inference problem can be solved using the steps highlighted ahead.

Mean-field assumption: For a feasible solution to the above problem, we require that each variable in the distribution is independent of others and only depends on

its own governing parameters. Hence, the global and local variables have their respective global and local 'free' parameters (λ and ϕ),

$$p(x, z, \beta) = p(\beta|\alpha) \prod_{n=1}^{N} p(x_n, z_n|\beta),$$
$$q(z, \beta) = q(\beta|\lambda) \prod_{n=1}^{N} \prod_{j=1}^{J} q(z_{nj}|\phi_{nj}).$$

Think of $\lambda \in \Gamma$ (feasible parameter space) as global parameters because β, being a random variable parameterized by λ, affects $p(x, z|\beta)$ globally; and $\phi_{nj} \in \Phi^D$ as local parameters because each latent variable z_{nj} contains hidden structure that governs j-th variable in n-th (local) observation. Owing to the exponential form for $q(\beta|\lambda)$ and $q(z_{nj}|\phi_n)$, their log inside the objective simplifies manipulation and finally the objective function can be re-written as function of λ and ϕ,

$$\hat{\mathcal{L}}(\lambda, \phi) = \mathbb{E}_q[\log p(\beta|x, z)] - \mathbb{E}_q[\log q(\beta|\lambda)] + \sum_{n=1}^{N} \sum_{j=1}^{J} \Big(\mathbb{E}_q[\log p(x_n, z_{nj})] - \mathbb{E}_q[\log q(z_{nj}|\phi_{nj})] \Big).$$

Co-ordinate ascent: The objective function is first maximized w.r.t. local parameters ϕ. Assume that $\phi(\lambda) := \phi^*$ locally maximizes $\hat{\mathcal{L}}(\lambda, \phi)$ for a given λ. We denote $\mathcal{L}(\lambda)$ as the locally maximized ELBO, i.e., $\mathcal{L}(\lambda) := \mathcal{L}(\lambda, \phi^*)$. For maximizing the locally maximized ELBO w.r.t. global parameters, we manipulate the terms that depend on λ and use first-order-necessary conditions to arrive at local maximizer $\hat{\lambda}$. A similar derivation for local parameters, gives optimum ϕ_{nj}:

$$\mathcal{L}(\lambda) = \mathbb{E}_q[\eta_g(x, z, \alpha)]^\top \nabla_\lambda a_g(\lambda) - \lambda^\top \nabla_\lambda a_g(\lambda) + a_g(\lambda) + \text{const.} \quad (2)$$

$$\nabla_\lambda \mathcal{L}(\lambda) = [\nabla^2_\lambda a_g(\lambda)]^\top (\mathbb{E}_q[\eta_g(x, z, \alpha)] - \lambda),$$
$$\hat{\nabla}_\lambda \mathcal{L}(\lambda) = [\nabla^{-2}_\lambda a_g(\lambda)] \nabla_\lambda \mathcal{L}(\lambda) = \mathbb{E}_q[\eta_g(x, z, \alpha)] - \lambda \quad (3)$$
$$\hat{\lambda} := \mathbb{E}_q[\eta_g(x, z, \alpha)], \quad \phi_{nj} = \mathbb{E}_q[\eta_l(x_n, z_{n,-j}, \beta)].$$

The optimum j-th local parameter in n-th observation context, ϕ_{nj}, depends on the global parameters and other local parameters in the same context. However, the global parameter λ depends on all the local parameters $\phi_{1:N,1:J}$. This structure suggests an iterative co-ordinate ascent update (ρ^t is step-size) solution:

$$\lambda^{t+1} = \lambda^t + \rho^t \hat{\nabla}_\lambda \mathcal{L}(\lambda^t) = \lambda^t + \rho^t(\hat{\lambda} - \lambda^t) = (1 - \rho^t)\lambda^t + \rho^t \hat{\lambda}.$$

2.3 The SVI algorithm

The centralized SVI Algorithm 1 uses variational inference. The global parameter updates rely on information from all local parameters, which is impossible to implement for massive datasets. For this reason, the standard gradient is replaced by the stochastic gradient. One data point is drawn at random from the data set and its local parameters are learned based on prior knowledge of global parameters (initialized). Locally maximized objective is obtained with this step. Then, before the next data point, this data point is replicated N times as if there had been N occurrences of the same data in the dataset. Using this, a noisy estimate of the global parameters is learned. This noisy estimate of global parameters is termed global intermediary parameters, and it is used in determining the stochastic gradient required for the gradient ascent step, which completes one iteration with both local and global updates. Note that the natural gradient [1] in place of Euclidean gradient is used due to ease of computation and the structure of probability space in which the objective is defined (See (3)). For convergence of SVI algorithm, the step-size must be set with the following conditions (Robbins and Monro (1951)), with appropriate constants (delay $\tau \geq 0$ and forgetting-rate $\kappa \in (0.5, 1]$):

$$\sum_t (\rho^t) = \infty, \quad \sum_t (\rho^t)^2 < \infty, \quad \rho^t = (t + \tau)^{-\kappa}.$$

Algorithm 1 Stochastic Variational Inference (Centralized)

1: Initialize $\lambda^{(0)}$
2: Schedule step-size ρ^t routine
3: **repeat**
4: Sample a data point x_i from the data set
5: Compute its local variational parameters,
$$\phi = \mathbb{E}_{\lambda^t}[\eta_l(x_i^{(N)}, z_i^{(N)})].$$
6: Compute intermediate global parameters as if x_i is replicated N times,
$$\hat{\lambda} = \mathbb{E}_\phi[\eta_g(x_i^{(N)}, z_i^{(N)})].$$
7: Update the global variational parameters,
$$\lambda^{t+1} = (1 - \rho^t)\lambda^t + \rho^t \hat{\lambda}.$$
8: **until** forever

3. DECENTRALIZING SVI USING ADMM

Now that we have established the framework and introduced fundamental principles behind the working of stochastic variational inference, we now focus on formulating SVI for a decentralized setting in which independent learners (systems that can run SVI on the data they possess) can contribute to a global learner model. Alternating Direction Method of Multipliers (ADMM), Gabay and Mercier (1976), is a robust technique to decentralize a complex problem by decomposing it into smaller problems solvable in a parallel way. We shall restrict to the use of ADMM (Eckstein (2012)) technique to solve this *consensus* problem. Literature is available for such problems for a variety of settings differing in the particular form of the objective function to be minimized and its corresponding constraints. In this section, at first, we shall formulate the proposed distributed SVI problem as a non-convex stochastic ADMM problem. Then, we shall explain the algorithm followed by analysis and results.

3.1 Problem formulation

Recall that SVI is about stochastic optimization of the evidence lower bound, $\hat{\mathcal{L}}(\lambda, \phi)$. Its solution is has following steps at each gradient ascent iteration:

[1] Amari (1998), while discussing natural gradients for maximum likelihood estimation, showed that the natural gradient relates to the Euclidean gradient through inverse Fisher metric projection.

$$
\begin{aligned}
\text{Sample data point}: &\quad x_n \sim \text{Unif}(x_1, x_2, ..., x_N) \\
\text{E-Step (maximize local)}: &\quad \phi^* = \arg\max_\phi \hat{\mathcal{L}}(\lambda^t, \phi) \\
\textit{Locally-maximized} \text{ objective}: &\quad \mathcal{L}(\lambda) := \hat{\mathcal{L}}(\lambda, \phi^*) \\
\text{M-Step (intermediary params)}: &\quad \hat{\lambda} = \arg\max_\lambda \mathcal{L}(\lambda) \\
\text{Gradient ascent}: &\quad \lambda^{t+1} = \lambda^t + \rho^t(\hat{\lambda} - \lambda^t)
\end{aligned}
$$

First of all, we observe that the objective function $\mathcal{L}(\lambda)$ is not convex in λ, see (2). Decentralizing of the SVI means that K agents use SVI for posterior inference on their own data-sets, such that in the end, the parameters that define the learned models mutually form a consensus, i.e. $\lambda_1 = \lambda_2 = \cdots = \lambda_K$.

Objective for distributed SVI: With the above example in mind, we propose distributed-optimization problem as:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{K} g_i(\lambda_i) \\
\text{subject to} \quad & \lambda_i = \lambda_1, \forall i \in \{1, 2, \dots K\} \\
& \lambda_i \in \Gamma
\end{aligned}
$$

where Γ indicates the feasible set for all the variables, and $g_i(\lambda_i) := -\mathcal{L}(\lambda_i)$. Note that $\mathcal{L}(\lambda)$ denotes the locally maximized objective, hence ϕ is omitted. Basically, we want to perform the M-step of the actual algorithm with consensus among the agents whereas the initial part of the SVI algorithm remains the same. The objective function comprises of the following, $\forall i \in \{1, \dots, K\}$:

$$
\begin{aligned}
g_i(\lambda_i) = -\mathbb{E}_{\phi(\lambda_i)}[\eta_g(x, z)]^\top \nabla_{\lambda_i} a_g(\lambda_i) + \lambda_i^\top \nabla_{\lambda_i} a_g(\lambda_i) \\
- a_g(\lambda_i) + \text{const.}
\end{aligned}
$$

Note that only the $\mathbb{E}_{\phi(\lambda)}$ terms encode the information from local (sampled) observation at current iteration, we call it sufficient statistics, which means that if we were to provide the same data to the parallel computational nodes (each of which runs its own instance of SVI), we would have similar sufficient statistics vectors. The log-normalizer a_g is only a function of its corresponding parameter λ. And "const." terms are unimportant in all proceeding analysis.

3.2 Proposed distributed solution

ADMM for SVI: We use the augmented Lagrangian with a quadratic penalty. The Lagrange multipliers are denoted by $y_i \in \Gamma$. Augmented Lagrangian and minimization updates for each processor/agent are given as:

$$
L_c = \sum_{i=1}^{K} g_i(\lambda_i) + y_i^\top (\lambda_i - \lambda_1) + (c/2)||\lambda_i - \lambda_1||_2^2 + \text{const.}
$$

$$
\lambda_i^{t+1} \in \arg\min_\lambda L_c(\lambda, \lambda_{-i}^t, y_i^t), \tag{4}
$$

$$
y_i^{t+1} = y_i^t + c(\lambda_i^{t+1} - \lambda_1^{t+1}).
$$

where λ_{-i}^t denotes all the variables at time t except the i-th i.e. $\lambda_{-i}^t = \lambda_{j \in \{1,\dots,i-1,i+1,\dots,K\}}^t$, and c is the penalty parameter in the augmented Lagrangian. Since the minimzation of the SVI objective g_i is itself an iterative algorithm, so to perform the iterate-minimization step in (4), we perform one iteration of gradient ascent using a noisy stochastic gradient estimate, as in SVI, in each

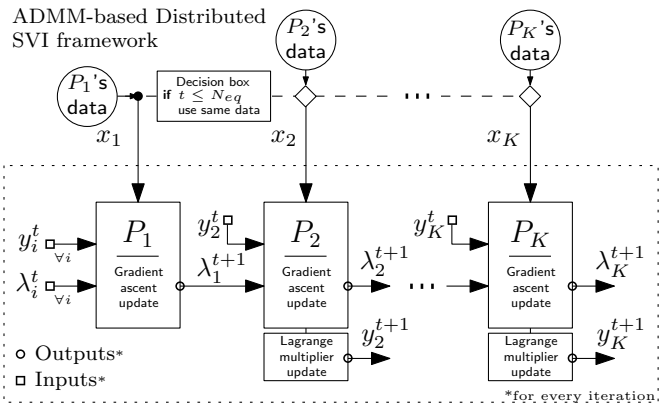

Fig. 1. System diagram showing steps at each iteration

ADMM iteration. To get a noisy estimate, we use global intermediary parameters (as used in SVI), by sampling a data point and repeating it N times (See Algorithm 2). ADMM for iterative solvers such as SVI that rely on gradient ascent has been discussed by Boyd et al. (2011).

Computing the minimum: To minimize the augmented Lagrangian we use its natural gradient at time t in gradient ascent,

$$
\hat{\nabla}_{\lambda_i^t} L_c = [\nabla^{-2} a_g(\lambda_i^t)] \nabla_{\lambda_i^t} L_c,
$$

and from (3),

$$
\hat{\nabla}_{\lambda_i^t} L_c = (\lambda_i^t - \hat{\lambda}_i) + [\nabla^{-2} a_g(\lambda_i^t)](y_i^t + c(\lambda_i^t - \lambda_1^t)). \tag{5}
$$

Note that the for $i = 1$, there is a slight trivial difference in the above update equation. Here, we have used the intermediary global parameters, $\hat{\lambda}_i = \mathbb{E}_q[\eta_g(x_i^{(N)}, z_i^{(N)})]$ obtained by repeating a data point N times and using locally-maximized objective. To use gradient ascent we require computing the Hessian inverse matrix $[\nabla^{-2} a_g(\lambda_i^t)]$. If we have an estimate of this inverse matrix, we can use (5) and the following gradient ascent update in place of (4) in ADMM:

$$
\lambda_i^{t+1} = \lambda_i^t + \rho_t(-\hat{\nabla}_{\lambda_i^t} L_c).
$$

Hessian inverse computation: Equation (5) requires computing the inverse of a Hessian matrix for every learner i at each iteration t. Such a computation pops out in many machine learning estimation tasks. This is computationally expensive especially when the number of local variables grows as in large data sets. We exploit structural properties for this computational hurdle. First of all, the matrix $\nabla_\lambda^2 a_g(\lambda)$ is the Fisher-information matrix of the probability distribution $q(\beta|\lambda)$ – it is symmetric positive definite. Moreover, (as we show in our latent Dirichlet allocation example later) for many of the commonly used probability distributions this Hessian has the form of a diagonal matrix added to a rank-1 matrix:

$$
\nabla_\lambda^2 a_g(\lambda) = \mathbf{diag}(d) + aa^T.
$$

The algorithm LiSSA by Agarwal et al. (2016) addresses the fast computation of Hessian inverse through stochastic sampling. They propose a novel method of estimating the large matrix inverse in linear-time for the similar class of Hessian matrices that we have encountered here. Their work successively generates samples of the Hessian

matrix and feeds the samples to an unbiased matrix-inverse estimator. This estimation problem is solved in linear time based on the size of the parameters' vector λ. In our example, the λ vector had roughly 7702 entries. Thus, the computational burden of large matrix inversion is decreased greatly, and we can estimate the Hessian inverse at each iteration. For the implementation of latent Dirichlet allocation, we show in the appendix, how we employ a fast approximate matrix inversion technique that exploits the structure of Hessian matrix making it solvable in linear time.

Algorithm 2 SVI-ADMM for two players i.e. $K = 2$

Initialize $\lambda_1^{(0)}, \lambda_2^{(0)}, c$
Schedule step-size ρ_t routine
repeat
  if $t > N_{eq}$ **then**
    Sample two data points x_1 and x_2
  else
    Sample one data point for both $x = x_1 (= x_2)$
  end if
  for $m \leftarrow 1$ **to** 2 **do**
    Use x_i to compute its local variational parameters,

$$\phi = \mathbb{E}_{\lambda_i^t}[\eta_l(x_i^{(N)}, z_i^{(N)})].$$

    Apply ADMM λ-minimization-update by computing intermediate global parameters $\hat{\lambda}_i$ and natural gradient,

$$\hat{\lambda}_i = \mathbb{E}_{\phi}[\eta_g(x_i^{(N)}, z_i^{(N)})],$$
$$\hat{\nabla}_{\lambda_i^t} L_c = (\lambda_i^t - \hat{\lambda}_i) + \underbrace{\nabla^{-2} a_g(\lambda_i^t)}_{\text{Hessian Inverse}} (y^t + c(\lambda_i^t - \lambda_{-i}^t)).$$

    Update the global variational parameters using gradient ascent,

$$\lambda_i^{t+1} = \lambda_i^t + \rho_t(-\hat{\nabla}_{\lambda_i^t} L_c).$$

  end for
  Update the Lagrange multipliers

$$y^{t+1} = y^t + c(\lambda_2^{t+1} - \lambda_1^{t+1}).$$

until forever

ADMM-based SVI Algorithm: System level diagram for the working of ADMM-based distributed SVI is shown in Fig. 1. The complete algorithm for two distributed learners is summarized in Algorithm 2. For the first N_{eq} iterations, we provide all the learners with the same data – this is part of initialization phase and the intuition behind this is that online learning algorithms are aimed to operate near equilibrium. This initialization helps in faster convergence at the cost of data sharing.

4. CONVERGENCE PROPERTIES OF SVI-ADMM

The proposed algorithm is an alternating direction Lagrangian method simultaneously running stochastic gradient ascent. We discuss two necessary notions of convergence in this regard. First of all, the whole methodology can be thought of as standard ADMM with a quadratic penalty. Here, the asymptotic convergence of ADMM is required. The important point to note while studying the asymptotic convergence of ADMM is that the iterate-minimization steps at each ADMM iteration are themselves stochastic optimization steps in nature. Thus, to discuss any guarantees on the convergence of ADMM, we require the convergence of stochastic gradient ascent of the augmented Lagrangian. This leads us to the second notion of convergence that of almost-sure convergence for the simultaneous stochastic optimization sub-problem.

4.1 Asymptotic convergence of ADMM

The convergence of ADMM is guaranteed for convex problems. Since our problem is a structured non-convex problem, so we use one of the recent works in regards to non-convex ADMM by Magnússon et al. (2014). More detailed work on the convergence of non-convex consensus problem using ADMM is done by Hong et al. (2015).

Magnússon et al. (2014) (Sec. IV-B) show that there are certain sufficient conditions that if they hold, the FON (the first-order-necessary conditions) for a class of non-convex ADMM problems, are satisfied (Proposition 3). They have given four conditions for this sufficiency. We enumerate them in the context of a two-player SVI-ADMM problem given by:

$$\begin{aligned} \text{minimize}_{\lambda_1, \lambda_2 \in \mathbb{R}^n} \quad & g_1(\lambda_1) + g_2(\lambda_2) \\ \text{subject to} \quad & \lambda_1 = \lambda_2,\ \lambda_1, \lambda_2 \in \Gamma \end{aligned}$$

where, g_1 and g_2 are non-convex, and Γ is the feasible set of the variables. The four conditions to satisfy FON are:

(1) The objective functions g_1 and g_2 should be continuously differentiable.
(2) The set Γ should be closed and expressible in the form of finite equality and inequality constraints of certain form (this trivially holds in our case).
(3) The iterate-minimization steps (e.g. (4)) that are part of the standard ADMM algorithm, should have a local *or* global optimal for all t.
(4) Set of gradient vectors of the constraints evaluated at limit points for both λ_1 and λ_2 should be linearly independent – *regularity* assumption.

For our case in SVI-ADMM, the first two conditions hold trivially. The fourth condition holds straight-forwardly because the constraints are linear functions (being a consensus problem) and thus the constraint gradients are constant vectors. As for the third condition, guarantees on a local or global optimal computed at each iteration cannot be easily stated. We see that at each ADMM iteration, our algorithm goes one step in the direction of a noisy estimate of the stochastic gradient. To show the existence of local optimal, we argue using the work by Bottou (1998) on stochastic gradient learning in the following sub-section.

4.2 Almost-sure convergence of gradient ascent

Authors of Hoffman et al. (2013) also required certain conditions on the SVI objective to make the stochastic optimization converge in the end. For non-convex objective functions to have a local optimum, three-times differentiability and other mild conditions are required, according to Bottou (1998). The variational objective satisfies those conditions. In our case, the objective function is an augmented Lagrangian having the sum of affine terms and variational objectives. The conditions required for online

Table 1. Top words for five topics (centralized)

music	church	university	party	division
song	district	department	minister	army
single	holy	research	government	war
songs	churches	science	political	forces
tracks	catholic	education	prime	battle
artist	parish	national	power	military
number	bishop	development	parliament	regiment
performed	tower	health	leader	force

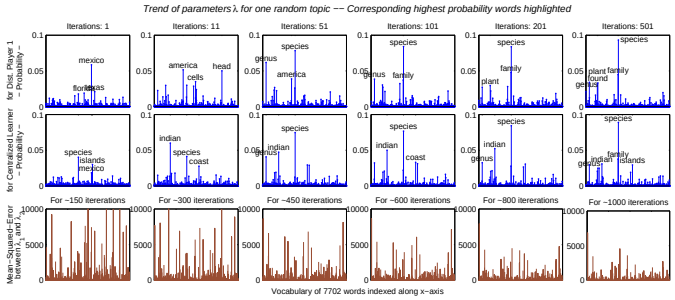

Fig. 2. The plots in the first two rows show that the learned parameters for a randomly chosen topic, for a distributed learner and a centralized learner progressively converge to similar outputs (i.e., showing convergence to an optimum); the third row shows that as the algorithm iterates, the MSE between the estimates of the distributed learners decreases (i.e., consensus is slowly attained).

stochastic gradient algorithms to converge almost-surely, are all satisfied [2] by the augmented Lagrangian as well.

Thus, we establish that our proposed algorithm ADMM-based SVI converges almost-surely to locally optimal λ_1^* and λ_2^* having $\nabla(g_1(\lambda_1^*) + g_2(\lambda_2^*)) \approx 0$. In Fig. 4 we show numerically that perplexity (a measure of model-fitness) for distributed learners approaches that for a centralized learner – even though ADMM slows down the rate of convergence because two iterative algorithms iterate simultaneously (ADMM iterates and stochastic gradient iterations).

5. EXPERIMENTS AND RESULTS

We implement latent Dirichlet allocation (LDA) topic model to test our algorithm. Blei et al. (2003) introduced probabilistic topic models for classification and prediction tasks for large corpora of documents. In latent Dirichlet allocation, each item of document collection is modeled as a finite mixture over an underlying set of topics. Whereas, each topic is modeled as an 'infinite' mixture over topic probabilities. The main idea is to learn the hidden structure that can help in distinguishing documents and articles that have similar themes. In Table 1, we reproduce the work by Hoffman et al. (2010) on online LDA for document classification (which essentially is a centralized learner), and later on, compare its performance with our distributed ADMM-based solution. LDA-specific techniques we used to make ADMM-based SVI fast are given in the Appendix.

[2] The augmented Lagrangian is simply the sum of the variational objective and a quadratic function of primal variables – we use this observation and the fact that the variational objective already satisfies the four conditions given in Section 5.1 of Bottou (1998).

Table 2. LDA–ADMM Exp.1 (Top eight words for four topics)

λ_1	λ_{central}	λ_1	λ_{central}
song	music	roman	roman
music	song	bishop	catholic
rock	songs	catholic	bishop
original	rock	jersey	church
musical	piano	austin	christian
best	musical	cathedral	cathedral
songs	dance	andrew	holy
love	performed	missionary	pope
λ_1	λ_{central}	λ_1	λ_{central}
democratic	mexico	war	war
mexico	governor	battle	army
seats	democratic	force	battle
mexican	republican	attack	forces
elections	senate	action	regiment
senate	mexican	operations	military
governor	senator	flying	british
state	seats	crew	attack

We use two-learner 100-topic LDA in all our experiments and the ADMM penalty parameter value $c \approx 10^{-7}$. The first set of experiments (Table 2) infer topic distributions by analyzing 115,200 random *Wikipedia* articles. Aim for this experiment is to compare estimates of distributed LDA learners with a centralized one λ_{central} that is provided with complete data from both learners, P_1 and P_2. The initialization parameter $N_{eq} \leq 40$. Depicted results are at equilibrium, i.e., when $||\lambda_1 - \lambda_2||_2$ is minimized, and hence we have just shown λ_{central} and $\lambda_{1,\text{distributed}}$. We show words sorted by relevance to the topic in each of the four topics (columns). The word on top of a column indicates the highest likelihood for that word to belong to the corresponding topic in comparison with other words in the vocabulary. This experiment showed that the equilibrium results are comparable. In Table 2, you can observe that both the centralized and distributed learners agree on Topic 1 (column one) to be about *music*, Topic 2 about *religion*, Topic 3 about *government* and so on. In another similar run, the trend of parameters λ for distributed and centralized learners was noted, see Fig. 2. The learned parameters started very differently, but after nearly 500 iterations the distributed and centralized learners give similar distribution parameters. Fig. 2 also shows that the sum of squared differences between the Dirichlet distribution parameters (λ_1 and λ_2) has decreased greatly after 1000 iterations. The choice of penalty parameter c is responsible for the rate of convergence – if increased, the algorithm converges faster, but this makes it risky in avoiding overflows such as making λ negative.

In Fig. 3, we have presented learned parameters for distributed vs. centralized learners after running for one hour. Top-left and top-right plots show that the two distributed learners λ_1 and λ_2 are almost exactly same. Bottom-left and bottom-right plots show that the centralized learners (with-all and with-half data respectively) also have similar outputs. Notably, the peaks in the bottom-right plot have larger magnitude compared to those in bottom-left – because bottom-left learner had access to twice more data.

In the second set of experiments (Table 3), the centralized learner that ran in parallel was only provided with one set of data that P_1 had access to. Whereas, P_1 and P_2 oper-

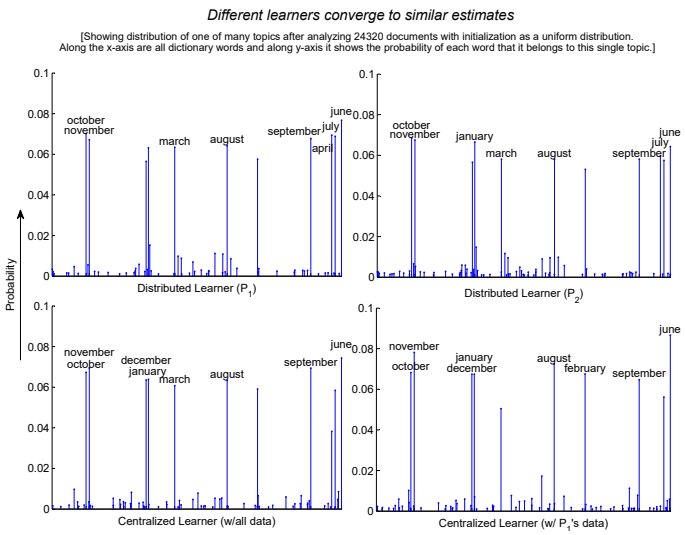

Fig. 3. Plot showing the distribution of word occurrences for some topic and the performance comparison of distributed vs. centralized learners. This shows that given equal amount of different datasets to two SVI learners (if the data is not mutually shared but only the model estimates are shared), the overall system can learn the model as good as how much a joint (centralized) learner can estimate when it has access to all the data (bottom-left) and when it has access to only half of the data (bottom right).

Table 3. LDA–ADMM Exp.2 (Top seven words for two topics)

λ_1	λ_{central}	λ_1	λ_{central}
party	liberal	theatre	theatre
election	color	stage	helena
color	party	opera	shakespeare
council	election	plays	magic
elected	elected	play	kiss
democratic	democratic	productions	jungle
liberal	vote	musical	laughing

ated as distributed SVI-ADMM learners. This experiment showed that the decentralized learner P_1 learned better than the centralized learner giving better estimates. For example, in Table 3, estimates of Topic 1 from decentralized and centralized learners reinforce each other, moreover, in Topic 2 the decentralized learner has learned more words that belong to the topic *theatre* such as *play*, *plays* and *stage* implying stronger learning. The conclusion here is insightful – if a learner has access to updates by other learners, it can improve its own accuracy profoundly. Thus, we conclude that the distributed ADMM-based setting for SVI can even improve the performance. Next, we analyze model-fitness of our estimates and argue for convergence.

We use a held-out perplexity metric as a measure of model fitness. This metric was used by Hoffman et al. (2010). It is defined as the geomteric mean of inverse marginal probability of each word from the set of documents used for testing model fit:

$$\text{perp}(w^{\text{test}}, \lambda, \alpha) := \exp\Big(-(\Sigma_i \log p(w_i^{\text{test}}|\alpha, \beta))/(\Sigma_{i,n} w_{i,n}^{\text{test}})\Big)$$

where $w_{i,n}$ is the number of occurrences of the n-th word in i-th document and w_i is the vector of word occurrences

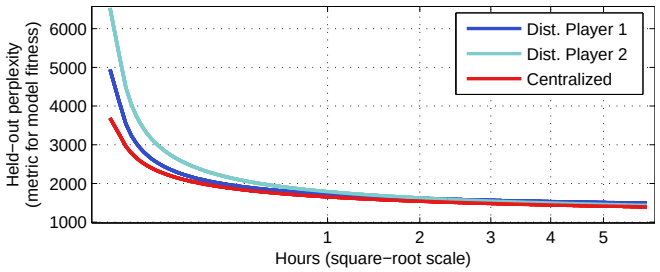

Fig. 4. Metric of model-fitness for randomly chosen articles from *Wikipedia* corpus as function of number of documents is analyzed. Batch size is of 128 documents. Nearly 22k documents are processed every hour.

for i-th document. A low value of this metric for a test-set of documents, indicates better model fit. Since the log-marginal likelihood in the above expression can not be evaluated, so a lower bound on perplexity is used. Fig. 4 shows that with time held-out perplexity values decrease for both the centralized and distributed learners, converging to similar equilibria.

6. CONCLUSION

We have presented a distributed ADMM-based SVI – a decentralized algorithm that solves separable stochastic optimization problems and merges their results to achieve optimal consensus solution. Applications of distributed learning agent systems are common in IoT framework especially when different learning systems do not want to share data with each other but still agree on partial collaboration and transfer learning. A well-trod example of latent Dirichlet allocation for probabilistic topic models is implemented to show comparative results for the centralized and distributed settings. Results show that through collaboration without having to share private data, two or more independent model posterior learners for SVI can improve their learning capabilities. Due to the use of stochastic optimization, this algorithm is considerably fast, scalable, and accurate. Moreover, its distributed learning methodology enhances security and robustness aspects that underpin modern deep learning goals. The accuracy of estimates is consistent with standard SVI and our convergence analysis as well as numerical experiments show sufficiency for the almost-sure existence of optimal equilibrium solution.

Appendix A. HESSIAN INVERSION FOR LDA

First of all, the inverse Hessian term $\nabla_\lambda^{-2} a_{g_1}$ in (5) is discussed. Recall that the global random variables $\beta_{1:K}$ in LDA topic model were such that each one of them represented the probability distribution of one of the K topics over the entire vocabulary of V words. Each β_k is distributed by a Dirichlet distribution on the $V-1$ simplex governed by its respective $\lambda_k \in \mathbb{R}_+^V$ (positive reals). Hence, for each topic, we had a vector of global parameters λ_k of size V. In the implementation, the vocabulary size is $V = 7702$ words. Computation of the Hessian term and its inverse for such a big-sized λ is challenging. We know that, for a Dirichlet distribution $q(\beta|\lambda) \sim \text{Dir}(\lambda_1, ..., \lambda_V)$ the Fisher information matrix is given as a difference of a

diagonal matrix and a matrix of all ones multiplied by a scalar Yang and Berger (1996):

$$\nabla^2_\lambda a_g = \begin{bmatrix} \psi'(\lambda_1) & & \\ & \ddots & \\ & & \psi'(\lambda_V) \end{bmatrix} - \psi'(\Sigma\lambda_i) \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix},$$
$$= \mathbf{diag}(\psi'(\lambda)) - \psi'(\Sigma\lambda_i)\mathbb{1}\mathbb{1}^\top.$$

where, the function $\psi' : \mathbb{R}_+ \to \mathbb{R}_+$ is the well-known polygamma function of order one, i.e., the first derivative of log of gamma function.

For computing the inverse $[\nabla^2_\lambda a_{g_1}]^{-1}$, we use the fact the the polygamma function of order one, $\psi'(t)$ resembles the function $1/t$, and that, the parameters λ are strictly positive. This implies that for large V we have $\psi'(\Sigma\lambda_i) \ll \psi'(\lambda_j)$ for any $j \in \{1, \dots, V\}$. Now, consider the following matrix (with vector $d \in \mathbb{R}^V_+$ and scalar $\kappa \geq 0$):

$$\mathbf{A} = \mathbf{diag}(d) - \kappa\mathbb{1}\mathbb{1}^\top.$$

Assume $\kappa \ll d_i$, and so an approximate inverse of $\mathbf{A}$ is:

$$\mathbf{A}^{-1} = \mathbf{diag}(e) + \kappa(ee^\top) + O(\kappa^2),$$

where, $e = [\frac{1}{d_1} \dots \frac{1}{d_V}]^\top$. This is because,

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{diag}(d)\mathbf{diag}(e) + \kappa\mathbf{diag}(d)(ee^\top) - \kappa\mathbb{1}\mathbb{1}^\top\mathbf{diag}(e) + O(\kappa^2),$$
$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} + O(\kappa^2).$$

With this form of the approximate Hessian inverse, the matrix-vector product $\nabla^{-2}_\lambda a_g b$ for any appropriate vector b is computationally very cheap because it can be computed without finding the whole matrix: a diagonal matrix times a vector is merely element-wise product, and for the second term, we compute the dot product $e^\top b$ which reduces it to scalar multiplications with e. Hence, we use this approximate inverse which does not require computing any matrix for computing $\tilde{\mathbf{A}}^{-1}b$ as given in (5).

REFERENCES

Agarwal, N., Bullins, B., and Hazan, E. (2016). Second order stochastic optimization in linear time. *arXiv preprint arXiv:1602.03943*.

Amari, S.I. (1998). Natural gradient works efficiently in learning. *Neural computation*, 10(2), 251–276.

Babagholami-Mohamadabadi, B., Yoon, S., and Pavlovic, V. (2015). D-mfvi: Distributed mean field variational inference using bregman admm. *arXiv preprint arXiv:1507.00824*.

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

Bottou, L. (1998). Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9), 142.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1), 1–122.

Campbell, T., Straub, J., Fisher III, J.W., and How, J.P. (2015). Streaming, distributed variational inference for bayesian nonparametrics. In *Advances in Neural Information Processing Systems*, 280–288.

Eckstein, J. (2012). Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Research Report, Rutgers University*, RRR, 32–2012.

Foti, N., Xu, J., Laird, D., and Fox, E. (2014). Stochastic variational inference for hidden markov models. In *Advances in Neural Information Processing Systems*, 3599–3607.

Gabay, D. and Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1), 17–40.

Gal, Y., van der Wilk, M., and Rasmussen, C. (2014). Distributed variational inference in sparse gaussian process regression and latent variable models. In *Advances in Neural Information Processing Systems*, 3257–3265.

Hensman, J., Fusi, N., and Lawrence, N.D. (2013). Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*.

Hoffman, M., Bach, F.R., and Blei, D.M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, 856–864.

Hoffman, M.D. and Blei, D.M. (2015). Structured stochastic variational inference. In *Artificial Intelligence and Statistics*.

Hoffman, M.D., Blei, D.M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1), 1303–1347.

Hong, M., Luo, Z.Q., and Razaviyayn, M. (2015). Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 3836–3840. IEEE.

Hua, J. and Li, C. (2016). Distributed variational bayesian algorithms over sensor networks. *IEEE Transactions on Signal Processing*, 64(3), 783–798.

Johnson, M. and Willsky, A.S. (2014). Stochastic variational inference for bayesian time series models. In *ICML*, 1854–1862.

Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., and Saul, L.K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2), 183–233.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Magnússon, S., Chathuranga, P., Rabbat, M., and Fischione, C. (2014). On the convergence of alternating direction lagrangian methods for nonconvex structured optimization problems.

Raman, P., Zhang, J., Yu, H.F., Ji, S., and Vishwanathan, S. (2016). Extreme stochastic variational inference: Distributed and asynchronous. *arXiv preprint arXiv:1605.09499*.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Yang, R. and Berger, J.O. (1996). *A catalog of noninformative priors*. Institute of Statistics and Decision Sciences, Duke University.