ADMM-based Networked Stochastic Variational Inference

Hamza Anwar, *Member, IEEE,* and Quanyan Zhu, *Member, IEEE*

***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) [1]. 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 Alternating Direction Method of Multipliers in a top-down setting to develop a distributed SVI algorithm such that independent learners running inference algorithms only require sharing the estimated model parameters instead of their private datasets. Our work extends the distributed SVI-ADMM algorithm that we first propose, to an ADMM-based networked SVI algorithm in which not only are the learners working distributively but they share information according to rules of a graph by which they form a network. This kind of work lies under the umbrella of 'deep learning over networks' and our application domain is natural language processing. 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.**

***Index Terms*—variational inference, transfer learning, stochastic optimization, method of multipliers, inference over networks**

I. Introduction

II. ADMM-based Distributed SVI

BUILDING upon the recent work on SVI by Hoffman et al., [1], we consider the SVI problem for a network of learners.

The N observations are $x = x_{1:N}$; the vector of global hidden variables is β; the N local hidden variables are $z = z_{1:N}$, each of which is a collection of J variables $z_n = z_{n,1:J}$; the vector of fixed parameters is α. (Note we can easily allow α to partly govern any of the random variables, such as fixed parts of the conditional distribution of observations. To keep notation simple, we assume that they only govern the global hidden variables.)

This work is partially supported by the grants EFMA-1441140 and SES-1541164 from National Science Foundation.

H. Anwar and Q. Zhu are with the Department of Electrical and Computer Engineering, New York University, New York, NY 10003 USA (e-mail: ha1082@nyu.edu, qz494@nyu.edu).

Manuscript received , 2017; revised September 11, 2017.

A. Optimization problem

$$\begin{aligned} \min_{\lambda_k} \quad & \sum_{k=1}^{K} g_k(\lambda_k) \\ \text{subject to} \quad & \lambda_k - \zeta = 0, \quad k = 1, \cdots, K \\ & \lambda_k \in \Gamma_k \end{aligned}$$

where each λ_k is an m-sized vector and Γ_k indicates the feasible set for the variables λ_k (typically $\Gamma_k = \mathbb{R}_+^m$) and,

$$g_k(\lambda_k) := -\mathbb{E}_{\phi(\lambda_k)}[\eta_g(x,z)]^\top \nabla_{\lambda_k} a_g(\lambda_k) + \lambda_k^\top \nabla_{\lambda_k} a_g(\lambda_k) - a_g(\lambda_k) + \text{const.} \tag{1}$$

which is the standard SVI problem objective function for a single learner. The above optimization problem gives us a solution for K learners when they form a consensus. Using an Augmented Lagrangian approach, as in ADMM, we solve this problem in a distributed iterative fashion for multiple learners.

B. ADMM-based solution

Augmented Lagrangian with a quadratic penalty is used to arrive at the ADMM update iterations. The Lagrange multipliers are denoted by $y_k \in \Gamma_k$. Minimization updates for each processor/agent are given as:

$$\begin{aligned} \lambda_k^{t+1} &= \arg\min_{\lambda_k} \left(g_k(\lambda_k) + (\lambda_k - \zeta^t)^\top y_k^t + \frac{c}{2} ||\lambda_k - \zeta^t||_2^2 \right), \\ \zeta^{t+1} &:= \frac{1}{K} \sum_{k=1}^{K} \left(\lambda_k^{t+1} + (1/c) y_k^t \right), \\ y_k^{t+1} &:= y_k^t + c(\lambda_k^{t+1} - \zeta^{t+1}). \end{aligned}$$

where ζ is called the *central collector*, and c is the quadratic penalty parameter in the augmented Lagrangian which is given as:

$$L_c(\{\lambda_k\}, \{y_k\}, \zeta) = \sum_{k=1}^{K} g_k(\lambda_k) + (\lambda_k - \zeta)^\top y_k + \frac{c}{2} ||\lambda_k - \zeta||_2^2.$$

Here, we note that the λ-minimization update which is actually a solution to $\lambda_k^{t+1} = \arg\min_{\lambda_k \in \Gamma_k} L_c(\{\lambda_k\}, \{y_k^t\}, \zeta^t)$, requires solving a constrained non-convex optimization problem. We solve this in a gradient descent fashion in of itself, as the standard SVI problem was also solved, but the original solution requires inversion of a Hessian matrix. For that we take into account one-step earlier value of λ_k^t — for details about derivation and Hessian inversion approximation used, see Appendix. Thereby, our proposed iterative ADMM methodology

runs along with a gradient-descent iterative update of variables which is completely summarized in algorithm 1.

Algorithm 1 ADMM-based distributed SVI for K players

1: Initialize $c, \lambda_1^{(0)}, \lambda_2^{(0)}, \ldots, \lambda_K^{(0)}$
2: Schedule step-size ρ_t routine
3: **repeat**
4: **for** $k \in \mathcal{K}$ **do**
5: Sample separate data points for all learners
6: Use x_k to compute its local variational parameters,

$$\phi = \mathbb{E}_{\lambda_k^t}[\eta_l(x_k^{(N)}, z_k^{(N)})].$$

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

$$\hat{\lambda}_k = \mathbb{E}_{\phi}[\eta_g(x_k^{(N)}, z_k^{(N)})],$$

$$\hat{\nabla}_{\lambda_k^t} L_c = (\lambda_k^t - \hat{\lambda}_k) - \nabla^{-2} a_g(\lambda_k^t)(y_k^t + c(\lambda_k^t - \zeta^t)).$$

8: Update the global variational parameters using gradient ascent,

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

9: **end for**
10: Update the central collector

$$\zeta^{t+1} = \frac{1}{K}\sum_{k=1}^{K} \lambda_k^{t+1} + (1/c) y_k^t.$$

11: Update all the Lagrange multipliers

$$y_k^{t+1} = y_k^t + c(\lambda_k^{t+1} - \zeta^{t+1}).$$

12: **until** forever

C. Experimental results

New set of experiments for the distributed problem was performed with multiple learners. Here, we show the results with 4 learners. Figure 1, shows the convergence properties of our distributed learners, for a metric of the estimated model's fitness, known as the 'held-out perplexity'. This same metric has been used by Hoffman et al. [2] to show convergence of the algorithm. A comparison of centralized versus distributed two-player SVI algorithms is depicted in Figure 2. We conclude that all the learners not only converge to higher precision in estimates (evident from the Figure 1), but also achieve accuracy of estimates (evident from Table I), while simultaneously maintaining consensus.

From Table I, we see that the highly probable words for a given topic learned by any of the four learners all direct to a similar kind of subject. For example, in Topic#98, the learners understood it to represent the names of months — even though for this topic, the distribution of word occurrences is different for the four players but it is evident that they all point to the same abstract class of words. Similarly, in other topics as well we see similarity in estimates. Sometimes, we even see that the descending order of the words is exactly the same, e.g., in Topic#72 and Topic#38 all learners have the same ordering

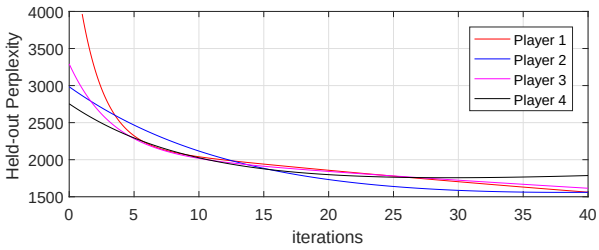


Fig. 1. Four players running independent learners and collaborating. This plot shows that the perplexity is decreasing over time.

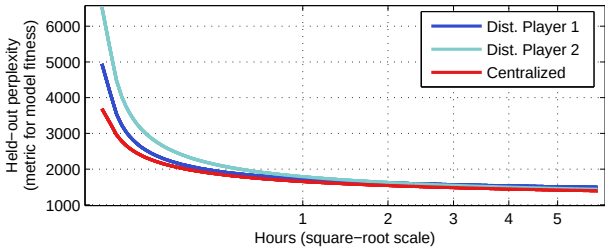


Fig. 2. Working for a two player network versus centralized algorithm.

TABLE I
TOP THREE WORDS FOR FIVE TOPICS LEARNED BY EACH OF THE FOUR PLAYERS AFTER 35 ITERATIONS I.E. $64 \times 35 = 2240$ INDEPENDENT DOCUMENTS ANALYZED BY EACH PLAYER. WORDS ARE WRITTEN IN DESCENDING ORDER OF PROBABILITY OF OCCURRENCE. PENALTY PARAMETER FOR ADMM $c = 5 \times 10^{-8}$, AND TOTAL TOPICS WERE 100.

	Player 1	**Player 2**	**Player 3**	**Player 4**
Topic#98	june march november	september october november	september october november	september october november
Topic#72	elected democratic republican	elected democratic republican	elected democratic republican	elected democratic republican
Topic#59	functions users file	functions users file	actor functions user	functions actor user
Topic#56	university college education	university college education	university college education	university education college
Topic#38	music song single	music song single	music song single	music song single

of words. Thus, the table shows that despite the fact that all learners use their own independently fetched datasets from *Wikipedia* articles, the consensus between results is achieved among all the learners, due to the central collection constraint.

III. ADMM-BASED NETWORKED SVI

Now, after conclusive results about distributed 'fully-conected' SVI algorithm, we move on to a network of nodes having independent learners residing at each node. The only difference in problem formulation, as we will see is in the equality constraints. We use the network formulation as given in [3].

> The network is modeled by an undirected graph denoted by $\mathcal{G}(\mathcal{K}, \mathcal{E})$ with $\mathcal{K} := \{1, \ldots, K\}$ representing the set of nodes, and $\mathcal{E}$ representing the set of links between nodes. Node $k \in \mathcal{K}$ only communicates with his neighboring

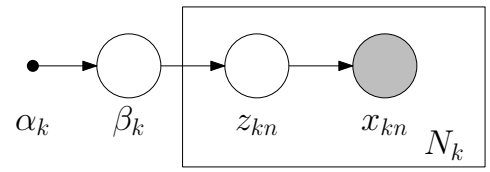


Fig. 3. A graphical probabilistic model for each node k in the graph $\mathcal{G}$

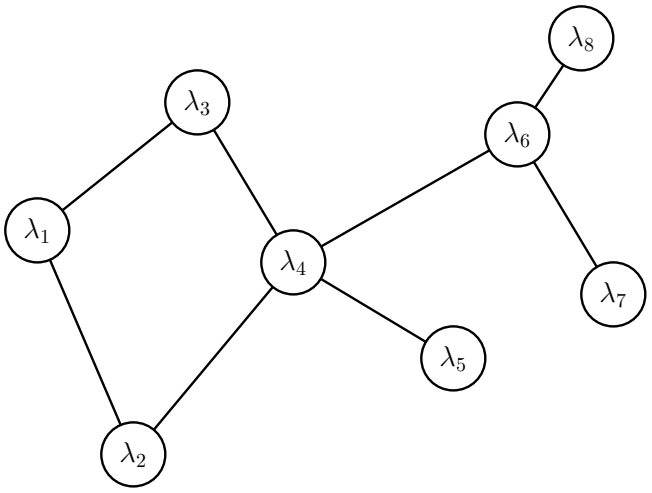


Fig. 4. Example of an undirected graph $\mathcal{G}(8,8)$ having 8 nodes and 8 edges. Here, node 4 has the highest number of neighbors $B_4 = \{2,3,5,6\}$.

nodes $B_k \subseteq \mathcal{K}$. Note that without loss of generality, graph $\mathcal{G}$ is assumed to be connected. The network can contain cycles. An example of such a network is is shown in Figure 4.

At every node $k \in \mathcal{K}$, a set of observations $\mathcal{D}_k := \{x_{kn} : n = 1, \ldots, N_k\}$ of size N_k is available, where x_{kn} denotes the n-th observation for the k-th node. Though not explicitly expressed, each x_{kn} can be a collection of multiple random variables. The vector of global hidden variables for node k is β_k; its N_k local hidden variables are $z_k = z_{k,1:N}$, each of which is a collection of J variables $z_{kn} = z_{kn,1:J}$; the vector of fixed parameters is α_k.

With the graph formulation given above, we pose the distributed SVI problem for a network of learners given as:

$$\begin{aligned} \min_{\{\lambda_k\}} \quad & \sum_{k=1}^{K} g_k(\lambda_k) \\ \text{s.t.} \quad & \lambda_k = \lambda_l, \forall k \in \mathcal{K},\ l \in B_k \end{aligned} \tag{2}$$

with $\lambda_1, \cdots, \lambda_K$ as variables. Here, $g_k : \Gamma_k \to \mathbb{R}_+$ is a non-linear function of λ_k re-written here:

$$g_k(\lambda_k) := -\mathbb{E}_{\phi(\lambda_k)}[\eta_g(x,z)]^\top \nabla_{\lambda_k} a_g(\lambda_k) + \lambda_k^\top \nabla_{\lambda_k} a_g(\lambda_k) - a_g(\lambda_k) + \text{const.}$$

Optimization problem (2) is equivalent to the following,

$$\begin{aligned} \min_{\{\lambda_k\}} \quad & \sum_{k=1}^{K} g_k(\lambda_k) \\ \text{s.t.} \quad & \lambda_k = \zeta_{kl}, \quad \zeta_{kl} = \lambda_l, \quad \forall k \in \mathcal{K},\ l \in B_k \end{aligned} \tag{3}$$

where ζ_{kl} are redundant variables that will facilitate the decoupling of variable λ_k at node k from its neighboring nodes $l \in B_k$. This problem will be solved using its dual. We denote the Lagrange multipliers by y_{kl1} (y_{kl2}) for the constraints $\lambda_k = \zeta_{kl}$ ($\zeta_{kl} = \lambda_l$). We observe that for each k we have $2|B_k|$ equality constraints. The augmented Lagrangian with a quadratic penalty is:

$$\begin{aligned} L_c\left(\{\lambda_k\}, \{\zeta_{kl}\}, \{y_{klj}\}\right) = \sum_{k=1}^{K} \Bigg(g_k(\lambda_k) + \sum_{l \in B_k} \Big(& y_{kl1}^\top (\lambda_k - \zeta_{kl}) \\ & + y_{kl2}^\top (\zeta_{kl} - \lambda_l) + \frac{c}{2}\left(||\lambda_k - \zeta_{kl}||^2 + ||\zeta_{kl} - \lambda_l||^2\right) \Big) \Bigg), \end{aligned} \tag{4}$$

The augmented Lagrangian can be iteratively minimized with respect to each variable by keeping others constant, which gives us a set of minimization updates for each variable summarized in the following Proposition.

A. Proposition 1

The distributed iterations solving (3) are as follows:

$$\lambda_k^{t+1} = \arg\min_{\lambda_k} \left\{ \begin{aligned} & g_k(\lambda_k) + \lambda_k^\top \sum_{l \in B_k} \left(y_{kl1}^t - y_{lk2}^t\right) \\ & + \frac{c}{2} \sum_{l \in B_k} \left(||\lambda_k - \zeta_{kl}^t||^2 + ||\zeta_{lk}^t - \lambda_k||^2\right) \end{aligned} \right. \tag{5}$$

$$\zeta_{kl}^{t+1} = \arg\min_{\zeta_{kl}} \left\{ \begin{aligned} & -\zeta_{kl}^\top y_{kl1}^t + \zeta_{kl}^\top y_{kl2}^t + \frac{c}{2}\left(||\lambda_k^{t+1} - \zeta_{kl}||^2\right) \\ & + \frac{c}{2}\left(||\zeta_{kl} - \lambda_l^{t+1}||^2\right) \end{aligned} \right. \tag{6}$$

$$y_{kl1}^{t+1} = y_{kl1}^t + c(\lambda_k^{t+1} - \zeta_{kl}^{t+1}), \quad \forall k \in \mathcal{K},\ l \in B_k \tag{7}$$

$$y_{kl2}^{t+1} = y_{kl2}^t + c(\zeta_{kl}^{t+1} - \lambda_l^{t+1}), \quad \forall k \in \mathcal{K},\ l \in B_k \tag{8}$$

and correspond to the standard ADMM solver discussed in [4].

Proof: The first task is to cast the problem (3) into standard ADMM problem form in [4]. The network description adopted here is similar to the one used in [5] and thus we use it to establish equivalence with standard ADMM [4]. Thereby, the remaining form of the minimization updates is directly derived from the augmented Lagrangian given in (4). The λ-minimization update (5) is derived by eliminating the terms that do not affect the minimization in augmented Lagrangian:

$$\begin{aligned} \lambda_k^{t+1} &= \arg\min_{\lambda_k} L_c(\lambda_k, \{\zeta_{kl}^t\}, \{y_{kl1}^t\}, \{y_{kl2}^t\}), \\ &= \arg\min_{\lambda_k} \left\{ \begin{aligned} & g_k(\lambda_k) + \sum_{l \in B_k} \left(\lambda_k^\top y_{kl1}^t + \frac{c}{2}||\lambda_k - \zeta_{kl}^t||^2\right) \\ & + \sum_{s \in B_k} \left(-\lambda_k^\top y_{sk2}^t + \frac{c}{2}||\zeta_{sk}^t - \lambda_k||^2\right) \end{aligned} \right., \end{aligned}$$

which upon merging the two summations reduces to (5). Similarly, the ζ-minimization update (6) comes directly from,

$$\zeta_{kl}^{t+1} = \arg\min_{\zeta_{kl}} L_c(\{\lambda_k^{t+1}\}, \zeta_{kl}, \{y_{kl1}^t\}, \{y_{kl2}^t\}).$$

Equations (7)–(8) are the dual variable updates (cf. [5]). ■

Next we reduce the iteration equations to a simpler form. Here, we observe that the ζ_{kl} update has the following unique solution (by putting the derivative equal to zero and solving),

$$\zeta_{kl}^{t+1} = \frac{1}{2c}(y_{kl1}^t - y_{kl2}^t) + \frac{1}{2}(\lambda_k^{t+1} + \lambda_l^{t+1}) \tag{9}$$

Putting (12) in (7)–(8) gives,

$$y_{kl1}^{t+1} = \frac{1}{2}(y_{kl1}^t + y_{kl2}^t) + \frac{c}{2}(\lambda_k^{t+1} - \lambda_l^{t+1}), \tag{10}$$

$$y_{kl2}^{t+1} = \frac{1}{2}(y_{kl1}^t + y_{kl2}^t) + \frac{c}{2}(\lambda_k^{t+1} - \lambda_l^{t+1}). \tag{11}$$

Now, we assume that both the Lagrange multipliers are identically initialized at every node k, as zero $y_{kl1}^0 = y_{kl2}^0 = 0_{m\times 1} \quad \forall k \in \mathcal{K},\ l \in B_k$. This ensures that $y_{kl1}^1 = y_{kl2}^1$, and $y_{kl1}^2 = y_{kl2}^2$, and son on. We see that only one of the two multipliers per node needs to be updated at each time step. Furthermore, (9) simplifies to,

$$\zeta_{kl}^{t+1} = \frac{1}{2}(\lambda_k^{t+1} + \lambda_l^{t+1}). \tag{12}$$

Finally the ADMM iterations $\forall k \in \mathcal{K}$ simplify, summarized in the following Proposition.

B. Proposition 2

Selecting $y_k^0 := y_{kl1}^0 = y_{kl2}^0 = 0_{m\times 1}$ as initialization $\forall k \in \mathcal{K},\ l \in B_k$, the iterations (5)–(8) reduce to the following,

$$\lambda_k^{t+1} = \arg\min_{\lambda_k} g_k(\lambda_k) + \lambda_k^\top y_k^t + c\sum_{l\in B_k}\left|\left|\lambda_k - \frac{1}{2}(\lambda_k^t + \lambda_l^t)\right|\right|^2 \tag{13}$$

$$y_k^{t+1} = y_k^t + c\sum_{l\in B_k}(\lambda_k^{t+1} - \lambda_l^{t+1}), \quad \forall k \in \mathcal{K} \tag{14}$$

Proof: Substituting (12) into the objective (5) gives the following,

$$\arg\min_{\{\lambda_k\}} L_c(\{\lambda_k\},\{\lambda_k^t\},\{\zeta_{kl}^t\},\{y_{kl1}^t\},\{y_{kl2}^t\}) = \sum_{k=1}^K\left(g_k(\lambda_k) + \lambda_k^\top\sum_{l\in B_k}\left(y_{kl1}^t - y_{lk2}^t\right) + \frac{c}{2}\sum_{l\in B_k}\left(\left|\left|\lambda_k - \frac{1}{2}(\lambda_k^t + \lambda_l^t)\right|\right|^2 + \left|\left|\frac{1}{2}(\lambda_l^t + \lambda_k^t) - \lambda_k\right|\right|^2\right)\right). \tag{15}$$

Note that $\{\lambda_k\}$ is the set of all variables of optimization and $\{\lambda_k^t\}$ denote constants known from previous iteration. All-zero initialization of the Lagrange multipliers implies that $y_{kl1}^t = -y_{lk1}^t \forall t$ [cf. from (10)–(11)], and so the first double sum in (15) can be rewritten as:

$$\sum_{k=1}^K\sum_{l\in B_k}\lambda_k^\top\left(y_{kl1}^t - y_{lk2}^t\right) = 2\sum_{k=1}^K\lambda_k^\top\sum_{l\in B_k}y_{kl1}^t. \tag{16}$$

The other two double sums in (15) can be simplified to give,

$$\frac{c}{2}\sum_{l\in B_k}\left(\left|\left|\lambda_k - \frac{1}{2}(\lambda_k^t + \lambda_l^t)\right|\right|^2 + \left|\left|\frac{1}{2}(\lambda_l^t + \lambda_k^t) - \lambda_k\right|\right|^2\right) = c\sum_{l\in B_k}\left(\left|\left|\lambda_k - \frac{1}{2}(\lambda_k^t + \lambda_l^t)\right|\right|^2\right). \tag{17}$$

By defining $y_k^t := 2\sum_{l\in B_k} y_{kl1}^t$, and substituting (16) and (17) into (15), gives the final form of the augmented Lagrangian which completes the proof:

$$\arg\min_{\{\lambda_k\}} L_c(\{\lambda_k\},\{\lambda_k^t\},\{y_k^t\}) = \sum_{k=1}^K g_k(\lambda_k) + \sum_{k=1}^K\lambda_k^\top y_k^t + c\sum_{k=1}^K\sum_{l\in B_k}\left|\left|\lambda_k - \frac{1}{2}(\lambda_k^t + \lambda_l^t)\right|\right|^2.$$

■

C. Network solution

Now, we present a solution to the ADMM minimization update (13), which is a non-convex optimization problem, similar to the corresponding update in distributed SVI in section II-B. We make use of stochastic gradient descent like standard SVI algorithm for minimization of augmented Lagrangian (cf. [1]). It is known that the natural gradient of g_k is given as,

$$\hat{\nabla}_{\lambda_k} g_k(\lambda_k^t) = \lambda_k^t - \hat{\lambda}_k.$$

The solution is presented in algorithm 2.

Algorithm 2 ADMM-based networked SVI for K players

1: Initialize $c, \lambda_1^{(0)}, \lambda_2^{(0)}, \ldots, \lambda_K^{(0)}$
2: Schedule step-size ρ_t routine
3: **repeat**
4: **for** $k \in \mathcal{K}$ **do**
5: Sample separate data points x_k for all learners
6: Use x_k to compute its local variational parameters,

$$\phi = \mathbb{E}_{\lambda_k^t}[\eta_l(x_k^{(N)}, z_k^{(N)})].$$

7: Apply ADMM λ-minimization-update by computing intermediate global parameters $\hat{\lambda}_k$ and natural gradient of augmented Lagrangian,

$$\hat{\lambda}_k = \mathbb{E}_\phi[\eta_g(x_k^{(N)}, z_k^{(N)})],$$

$$\hat{\nabla}_{\lambda_k^t} L_c = (\lambda_k^t - \hat{\lambda}_k) - \left[\nabla^{-2} a_g(\lambda_k^t)\right]\left(y_k^t + c\sum_{l\in B_k}\left(\lambda_k^t - \lambda_l^t\right)\right).$$

8: Update the global variational parameters using gradient ascent,

$$\lambda_k^{t+1} = \lambda_k^t + \rho^t\left(-\hat{\nabla}_{\lambda_k^t} L_c\right).$$

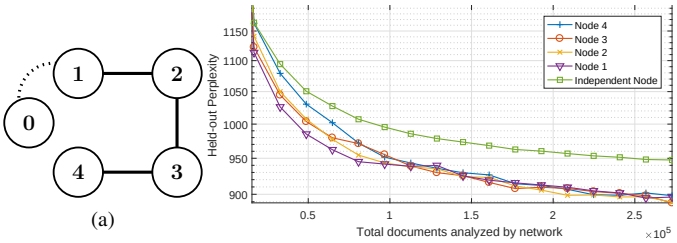
9: **end for**
10: Update all the Lagrange multipliers

$$y_k^{t+1} = y_k^t + c\sum_{l\in B_k}\left(\lambda_k^{t+1} - \lambda_l^{t+1}\right).$$

11: **until** forever

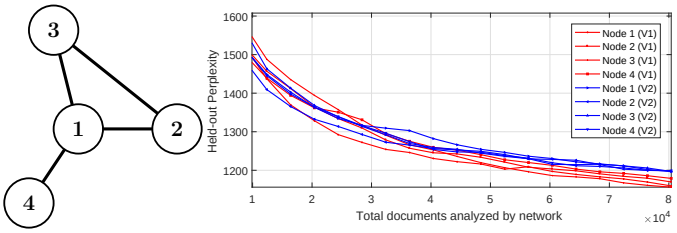
D. Experimental results

The issue of cross-matching the topics between two different players was solved using a correlation metric. Like discussed



(a) (b)

Fig. 5. (a) Line-type graph network. Dotted line between two nodes indicates same dataset supply. Solid line indicates possibility of transfer learning between nodes via ADMM. (b) Perplexity trajectory for a line-type graph.



(a) (b)

Fig. 6. (a) Star-type strongly connected network. (b) Perplexity trajectory for a strongly connected network (V1) versus a weakly connected line-type network (V2). Clearly strongly connected network starts performing better after some iterations.

earlier, SVI relies on random initialization of the global parameters. So, each player initializes with different global parameters, and as they encounter observations, they update the global parameters. Since, in our setting each player has its own independent dataset, so the trajectory of converging to 'true' topics is different for every player. That is why, if two completely independent learners are fed the same data, they converge to similar estimates but with different trajectories (i.e. topic 0 for player A may truly represent the contents of topic 43 for player B, and so on). Thus, in order to match the right topics, a correlation metric was needed. We used the Pearson correlation coefficient for this.

Result for an experiment that used a line-type graph is shown in Fig. 5b. The network is shown in Fig. 5a. In this experiment nodes 0 and 1 were provided with exactly same set of data (limited to a fixed 800 documents offline available that were fed repeatedly). The nodes 2, 3, and 4 were provided with an online data i.e. independent and new data points at each iteration. The purpose of this experiment was to see how the connected nodes corroborate and improve estimation accuracy (of node 1) in contrast to accuracy of the independently running learners (i.e. node 0). The perplexity metric trajectory shown in Fig. 5b, supports our claim that node interaction through ADMM updates certainly benefits. We achieve better accuracy in the estimate of node 1 as compared to that of node 0 because the learning at nodes 2-4 affects that of node 1 due to the consensus constraint.

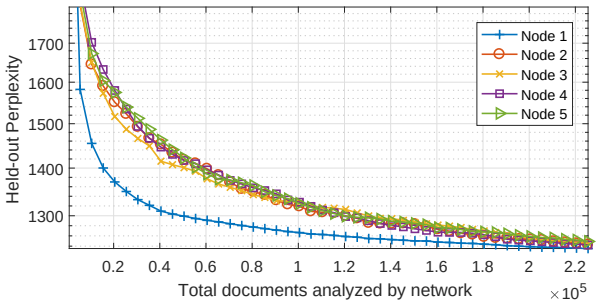


Fig. 7. Independent SVI with complete data versus networked SVI with partitioned data. Node 1 is an independent learner. Nodes 2-5 are connected in a network having partitioned datasets.

IV. Discussion and Conclusion

Acknowledgment

References

[1] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[2] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent dirichlet allocation," in *advances in neural information processing systems*, 2010, pp. 856–864.

[3] R. Zhang and Q. Zhu, "Secure and resilient distributed machine learning under adversarial environments," in *Information Fusion (Fusion), 2015 18th International Conference on*. IEEE, 2015, pp. 644–651.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[5] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, no. May, pp. 1663–1707, 2010.