Slow-learner ain't My Problem: Exploiting Structure in high-D, high-V Latent Space Stochastic Learning

Abstract

We pesent here a scheme for exploiting distributable structure present in latent space model for high-dimensional (high-D) and high-volume (high-V) learning. Models like latent dirichlet allocation, mixed membership stochastic blockmodels, dictionary learning have structures that can be exploted for big learning. We present a distributed learning strategy for such models. The scheme is distributed in data as well as parameter space and avoids waiting for slow learners to obtain full distributive leverage. The learning scheme, flexible to slow worker-processors, has provably correct strategy for convergence even with fewer synchronizations. It provides a tunable synchronization startegy that can be set based on the learning needs and system quality with the end user. We provide theoretical bounds on the parameter variance among workers with different synchronization strategies. Empirical results presented for latent space models such as latent dirichlet allocation and dictionary learning show that it scales very well with large data as well as parameter space. The comparative evaluation with other parallel and distributed learning startegies shows better preformance of the model.

1 Introduction

In todays information age there is trillions of bytes of data being generated every moment. By one estimate we generate 5 exabytes of data on the internet every two days [1] and most of it is user generated content. Given this massive amount of data generated every moment large scale machine learning is not just of academic interest anymore but of practical significance too. Large scale learning or big learning has been an active area of research in recent times. But most of this research has been focused around big data and the aspect of distibuting the learnig model has take back seat. While dealing with big data is definitely a must in this massive content generation age the learning task might turn out to be non-trivial when big data meets complex models. Learning models with high dimension or number of parameters becomes non-trivial even with slight increase in data. Hence there is a need for learning scheme which can perform distributed data as well as parameter learning. This problem is highly prevalent in latent space model such as latent dirichlet allocation (LDA [2]), mixed membership stochastic blockmodels (MMSB [1]) as they inflate their parameter space by introducing large number of latent variables. Henceforth these models would be our object of focus in this paper. We develop a stochastic learning scheme for latent space model, specifically for LDA and MMSB, which is distributed in data as well as parameter space. We modify the originial objective to suit our optimization scheme.need to write this in a better way . The structure obtained can be further exploited to minimize the hazardous effects of slow-processors in the distributed system.

expand some more giving an over view of our learning scheme

[1] http://techcrunch.com/2010/08/04/schmidt-data/

2 Related Work

PSGD (only data parallel); Aggarawal and Duci's Distributed Delayed (Again Data partition only; and it's hard to code); Any Other? How do we discuss the original Haas's paper though we do considerable more in terms of theory and multi-iteration Theory work: Stochastic Approximation book; Zinkevich, Noboru Murata

3 Slow-learner Agnostic Distributed Learning Framework

Our approach is to exploit independent cliques of structure present in large scale machine learning models. Probabilistic graphical models introduce massive amount of latent variables to induce the modelers belief regarding the generative process of the data. Though this enables these models with a unique ability to provide an interpretation and a generative story, it also makes the model harder to learn since the parameter as well as variable space increases massively. These models when run on large data have their learning problem become twice difficult.

While it may appear that latent space models' biggest boon of incorporating hidden variables is their biggest bane for large scale learning, it turns out that is not the case. On close examination it appears that these latent variables have another adavantage: they have a structure. They generally have cliques of correlated variable sets. This independence structure can be exploited effectively for a distributed learning framework. Moreover in models such as LDA and MMSB with a modified objective need to put it in a better way one can achieve distributivity in data as well as parameters. We explain this using a basic LDA model. In simple terms, the aim of an LDA model given a *document by vocabulary* matrix (Y) is to split it into two matrices: a *documents by topics* (π, variable set) and a *topics by vocabulary* matrix (β, parameter set). Equation 1 presents this in an optimization framework with simplex and non-negativity constraints. This is an ℓ_p norm which is typecally ℓ_2.

$$\begin{aligned} \text{argmin}_{\pi,\beta} L = ||Y - \pi\beta||_p^p = \sum_{i,j}(Y_{i,j} - \sum_k \pi_{i,k}\beta_{j,k})_p^p \\ s.t. \sum_k \pi_{i,k} = 1, \sum_k \beta_{j,k} = 1, \pi_{i,k} \geq 0, \beta_{j,k} \geq 0 \ \ \forall\, i,j \end{aligned} \tag{1}$$

For MMSB a similar objective can be formulated. Given a *user by user* interaction strength matrix Y it aims to find two matrices π and B, which are *user by topic* and *topic by topic* matrix respectively, such that $\pi^T B\pi$ reconstructs the original matrix interaction matrix Y. Equation 2 presents this in an optimization framework with the usual non-negativity and simplex constraints.

$$\begin{aligned} \text{argmin}_{\pi,B} L = ||Y - \pi^T B\pi||_p^p = \sum_{i,j}(Y_{i,j} - \sum_{g,h} \pi_{i,g}B_{g,h}\pi_{j,h})_p^p \\ s.t. \sum_k \pi_{i,k} = 1, \pi_{i,k} \geq 0, B_{i,k} \geq 0 \ \ \forall\, i,j \end{aligned} \tag{2}$$

For Dictionary Learning [4] the objective is

$$\begin{aligned} \text{argmin}_{\alpha,D} L = \frac{1}{2}||Y - D\alpha||_2^2 + \lambda||\alpha||_1 \\ s.t.\ D_j^T D_j \leq 1 \forall j \end{aligned} \tag{3}$$

where D is the dictionary being learnt and α is the sparse representation of the signal

For the LDA objective, figure 3 shows the way parameters and variable sets are divided in the distributive scheme. The distinction between parameter and variables comes in the projection step. While the simplex projection for variables set (π) is done at each update iteration i.e. each SGD update, the simplex projection for the parameters in LDA is done at synchronization with other workers. We perform parameter/variable updates of different blocks parallely. For SGD we perform updates for π, β, which we will collectively refer to as $\mathbf{\Psi}$ matrix whereas ψ are the individual components of the matrix. This definition of $\mathbf{\Psi}$ and ψ will come in handy for updates to variable or

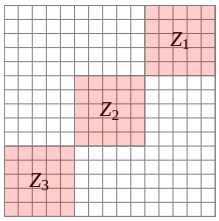

Figure 1: Dividing the *document by vocabulary* matrix fo LDA into blocks such that no two of them share any row or a column.

parameters based on the gradient at individual data points. E.g. the update for the LDA objective L are:

$$\psi^{(t+1)} = \psi^{(t)} - \eta_t \nabla \mathcal{L}_{Y_{i,j}}(\psi^{(t)}) \tag{4}$$

Appropriate projections are applied to after the step in equation 4 based on whether the updated value is of a parameter or a variable.

For these update rules, we list below the differentials for each component σ of ψ where norm use is ℓ_2, and $(\nabla L_{Y_{i,j}}(\psi))_\sigma = \frac{\partial L_{Y_{i,j}}}{\partial \sigma}$:

$$(\nabla L_{Y_{i,j}}(\psi))_\sigma = \begin{cases} -2(Y_{i,j} - \sum_r \pi_{i,r}\beta_{j,r})\beta_{j,\ell} & \text{if } \sigma = \pi_{i,\ell} \\ 0 & \text{if } \sigma = \pi_{i',\ell},\ i \neq i' \end{cases} \tag{5}$$

similarly for $\sigma = \beta_{j,l}$. From this we observe that SGD update for $\pi_{i,l}$ at a particular entry $Y_{i,j}$ depends only on previous $\pi_{i,r}, \beta_{j,r}$ where $r \in 1, \ldots, R$ and R is the number of topics we chose. The updates for each component are similar for the MMSB case.

Input : Y, β, π, sub-epoch size d
$\pi \leftarrow \pi_0$, $\beta \leftarrow \beta_0$
Block Y, π, β into corresponding w blocks
while *not converged* **do**
  Pick step size η_S
  Pick w blocks$(S_1, ..., S_w)$ to form sub-epoch S
  for $b = 0, \ldots, w-1$ ***in parallel*** **do**
    Run SGD on the training points S_b
    /** Do until every block is ready to synchronize **/
    /** So potentially each block S_b runs through each data-point multiple times if slow workers **/
    Apply appropriate constraints (e.g. data-variable constraints in LDA).
  end
  Apply appropriate constraints (e.g. parameter constraints in LDA)
end

Algorithm 1: Algorithm for LDA updates. The w blocks correspond to w worker processors

Given this understanding of our optimization objective and SGD update rules, we would like to segment our data in such a way that certain blocks S_b can be run in parallel, where we define $S_b \subseteq Y$. Figure 3 shows the way we segment our simple matrix Y to enable parallelization. In order to run SGD on our blocks in parallel, we divide them such that no two blocks share common rows or columns. To be more precise, we say that a point $y \in S_b$ is the coordinates in the data, such as $y = (y_i, y_j y) \in Y$. Two blocks S_b and $S_{b'}$ are non-overlapping if for all $y \in S_b$ and $y' \in S_{b'}$, $y_i \neq y'_i$ and $y_j \neq y'_j$. In order to cover all regions of Y, we need multiple sub-epochs. In our algorithm, we run the sub-epochs sequentially, but for each sub-epoch we run SGD on the blocks in parallel. Different blocks in a sub-epoch can make different number of passes through the data. This is where the algorithm is robust against slow processors as the worker keep running instead of waiting until every body is ready to synchronize. (Note, the order in which you run the sub-epochs does not matter, as long as they are each run once per epoch.) Algorithm 1 explain the steps more formally. We next offer a proof that this multi-iteration per block distributed parameter learning converges appropriately.

4 Theoretical Analysis

1) Convergence (with projection)
2) Epoch variance bound
3) Sub-epoch variance bound
4) Show that waiting is not a good idea and workers should keep working till every one is ready

Here we analyse the method presented in algorithm 1 theoretically. We will prove that the multi-iteration per block strategy converges. We provide a bound on the variance accross two blocks with in a sub-epoch running parallely. We show theoretically how the variance varies after each synchronization between two consecutive epochs. In the end we show why working instead of wating for the slowest processor is a better strategy.

4.1 Convergence proof

We introduce $V^{(t)}(\psi^{(t+1)}, \psi^{(t)})$: a state potential function that is defined over the past $\psi^{(t)}$, future $\psi^{(t+1)}$ and the data points $y^{(t)}$ picked at iteration. $V^{(t)}$ encodes the probability of update in the parameter from $\psi^{(t)}$ to $\psi^{(t+1)}$ through updating over a new point $y^{(t)}$. We have the relation

$$p(\psi^{(t+1)}|\psi^{(t)})d\psi^{(t)} = p(V^{(t)}(\psi^{(t+1)}, \psi^{(t)}))dV^{(t)}(\psi^{(t+1)}, \psi^{(t)}) \tag{6}$$

$V^{(t)}(\psi^{(t+1)}, \psi^{(t)})$ and $dV^{(t)}(\psi^{(t+1)}, \psi^{(t)})$ define the data points picked and the volume element of this choice respectively, to get the new update $\psi^{(t+1)}$ from $\psi^{(t)}$. $V^{(t)} = V^{(t)}(\psi^{(t+1)}, \psi^{(t)})$ can be understood as a fucntion that keeps track of the state of $\psi^{(t)}$ and $\psi^{(t+1)}$ and depends upon the data-point ${Y_{i,j}}^{(t)}$ picked in the SGD update.

We define $\nabla\mathcal{L}(\psi^{(t)})$ as the exact gradient at iteration t. We denote error at iteration t, $\left[\nabla\mathcal{L}(\psi^{(t)}) - \delta L^{(t)}(V^{(t)}, \psi^{(t)})\right]$, as ε_t where $\delta L^{(t)}(V^{(t)}, \psi^{(t)})$ is the SGD gradient at iteration t i.e. $\nabla\mathcal{L}_{Y_{i,j}}(\psi^{(t)})$. We make the assumption that the error $\varepsilon_t = \left[\nabla\mathcal{L}(\psi^{(t)}) - \delta L^{(t)}(V^{(t)}, \psi^{(t)})\right]$ in each step is a martingale difference sequence. i.e we assume that

$$\begin{aligned} E\left[\nabla\mathcal{L}(\psi^{(t)}) - \delta L^{(t)}(V^{(t)}, \psi^{(t)})|\delta L^{(i)}(V^{(i)}, \psi^{(i)}), \psi^{(i)}, i < t, \psi^{(t)}\right] &= 0 \\ E\left[\varepsilon_t|\varepsilon_i, i < t\right] &= 0 \end{aligned} \tag{7}$$

We have to note here that assuming error terms are a martingale difference sequence is a weaker assumption than assuming error terms ε_i are independent of each other. Making the martingale difference assumption means that $\delta L^{(t)}(V^{(t)}, \psi^{(t)})$ conditioned on $\psi^{(0)}$ and $\delta L^{(i)}(V^{(i)}, \psi^{(i)}), i < n$, depends only on $\psi^{(t)}$. This is achieved since the blocking strategy explained earlier has no overlap between two parameters that are updated parallely.

We have w worker processors (algorithm 1) and assume that every worker i touches n_i data point (with repeitition). In other words if the worker i was asigned n points and it touches each point $\kappa_i, \kappa_i \geq 1$, times on an average in its block before syncing then we define n_i and N_w as

$$n_i = \kappa_i n \qquad \text{and} \qquad N_w = \sum_{i=1}^{w} n_i \tag{8}$$

Theorem 1 *The stochastic updates $\psi^{(t+1)} = \psi^{(t)} - \eta_t\nabla\mathcal{L}_{Y_{i,j}}(\psi^{(t)})$ as described in algorithm 1 and the exact updates $\psi^{(t+1)} = \psi^{(t)} - \eta_t\nabla\mathcal{L}(\psi^{(t)})$ (in case of an exact gradient descent) converge to the same set of limit points asymptotically, given that the error terms ε_t are martingale difference sequence, $E[\varepsilon_t^2] < D$ (bounded variance) and $\sum \eta_i^2 < \infty$*

Proof.

From equation 4 we have

$$\begin{aligned} \psi^{(t+1)} &= \psi^{(t)} - \eta_t\delta L^{(t)}(V^{(t)}, \psi^{(t)}) \\ &= \psi^{(t)} - \eta_t\nabla\mathcal{L}(\psi^{(t)}) + \eta_t\left[\nabla\mathcal{L}(\psi^{(t)}) - \delta L^{(t)}(V^{(t)}, \psi^{(t)})\right] \\ &= \psi^{(t)} - \eta_t\nabla\mathcal{L}(\psi^{(t)}) + \eta_t\varepsilon_t \end{aligned} \tag{9}$$

Using n_i and N_w as defined in equation 8

$$\psi^{(t+(\sum_1^w n_i)m)} = \psi^{(t)} + \sum_{i=t}^{t+m(\sum_1^w n_i)} -\eta_i \nabla \mathcal{L}(\psi^{(i)}) + \sum_{i=t}^{t+m(\sum_1^w n_i)} \eta_i \varepsilon_i$$

$$\implies \psi^{(t+mN_w)} = \psi^{(t)} + \sum_{i=t}^{t+mN_w} -\eta_i \nabla \mathcal{L}(\psi^{(i)}) + \sum_{i=t}^{t+mN_w} \eta_i \varepsilon_i$$

$$\text{assuming } \sum_1^w n_i = N_w$$

$$\implies \psi^{(t+mN_w)} = \psi^{(t)} + \sum_{i=t}^{t+mN_w} -\eta_i \nabla \mathcal{L}(\psi^{(i)}) + M_{mN_w} \tag{10}$$

where $M_{mN_w} = \sum_{i=t}^{t+mN_w} \eta_i \varepsilon_i$ is a martingale sequence since it is a sum of martingale difference sequence. mN_w captures the m whole sub-epochs of work done as a whole by all the workers combined. From Doobs martingale inequality ([3], ch. 1, Thm 3.8)

$$P\left(\sup_{t+mN_w \geq r \geq t} |M_r| \geq c\right) \leq \frac{E\left[\left(\sum_{i=t}^{t+mN_w} \eta_i \varepsilon_i\right)^2\right]}{c^2} \tag{11}$$

where $M_r = \sum_{i=t}^{r} \eta_i \varepsilon_i$. Lets look at the RHS of equation 11 above:

$$E\left[\left(\sum_{i=t}^{t+mN_w} \eta_i \varepsilon_i\right)^2\right] = E\left[\sum_{i=1}^{mN_w} (\eta_i \varepsilon_i)^2\right]$$

$$\text{(equation 7} \implies E[\varepsilon_i \varepsilon_j] = 0 \text{ if } i \neq j)$$

$$= \sum_{i=1}^{mN_w} \eta_i^2 E[\varepsilon_i^2] \leq \sum_{i=1}^{mN_w} \eta_i^2 D \to 0$$

$$\text{where } E[\varepsilon_i^2] < D \ \forall i \text{ and assuming } \sum \eta_i^2 < \infty$$

$$\lim_{t \to \infty} \implies P\left(\sup_{i \geq t} |M_i| \geq c\right) = 0 \ \ as \ \ t \to \infty \tag{12}$$

From equation 12 we have

$$\psi^{(t+mN_w)} = \psi^{(t)} + \sum_{i=t}^{t+mN_w} -\eta_i \nabla \mathcal{L}(\psi^{(i)})$$

asymptotically. Hence the algorithm converges to the same set of limit points as the exact gradient descent asymptotically. Note that we do a theoretical analysis of the algorithm without projection steps. Extending the proof to include projection can be done by using Arzela-Ascoli theorem and the limits of converging sub-sequence of our algorithm's SGD updates [5].

■

4.2 Intra sub-epoch variance

We assume for simplicity that the parameter being updated in block-i is univariate. This analysis can be easily extended to a multivariate parameter updation case in each block of a sub-epoch. $\psi^{(t)}$ is the value of parameter theta at iteration t. η_t is the step size at iteration t and $L^{(t)}$ is the loss at point $y^{(t)}$ in iteration t. We define $v^t = V^{(t)}$ the potential function defined at iteration t as in equation 6

$$\psi^{(t+1)} = \psi^{(t)} - \delta\psi^{(t)}(V^{(t)}, \psi^{(t)}) \tag{13}$$

where $\delta\psi^{(t)}(V^{(t)}, \psi^{(t)}) = \eta^{(t)}\delta L^{(t)}(V^{(t)}, \psi^{(t)}) \Rightarrow \psi^{(t+1)} = \psi^t - \eta_t \delta L^t(V^t, \psi^t)$

Summing equation 13 over n_i, the number of points updated in block i of a sub-epoch

$$\psi^{t+n_i} = \psi^t - \sum_{i=1}^{n_i} \eta_{t+i}\delta L^{t+i}(V^{t+i}, \psi^{t+i}) \tag{14}$$

Let V denote denote the joint potential for all the n_i points encountered in block i. The equation 6 can be extended as

$$\begin{aligned} &p(\psi^{(t+n_i)}|\psi^t)d\psi^{(t+n_i)} = p(V(\psi^{(t+n_i)}, \psi^t))dV \\ &\Rightarrow p(\psi^{(t+n_i)})d\psi^{(t+n_i)} = \int_{\psi^t} p(\psi^{(t+n_i)}|\psi^t)p(\psi^t)d\psi^t d\psi^{(t+n_i)} = \int_{\psi^t} p(V(\psi^{(t+n_i)}, \psi^t))dV p(\psi^t)d\psi^t \end{aligned} \tag{15}$$

Lemma 1 *Let $u(\psi^{(t+n_i)})$ be a function of $\psi^{(t+n_i)}$ then*

$$\mathbb{E}^{\psi^{(t+n_i)}}[u(\psi^{(t+n_i)})] = \mathbb{E}^{\psi^t}[\mathbb{E}^V[u(\psi^{(t+n_i)})]]$$

Proof. From equation 15

$$\begin{aligned} \mathbb{E}^{\psi^{(t+n_i)}}[u(\psi^{(t+n_i)})] &= \int_{\psi^{(t+n_i)}} u(\psi^{(t+n_i)})p(\psi^{(t+n_i)})d\psi^{(t+n_i)} \\ &= \int_{\psi^{t+i}} u(\psi^{(t+n_i)})P(\psi^{(t+n_i)})d\psi^{(t+n_i)} \\ &= \int_V \int_{\psi^t} u(\psi^{(t+n_i)})P(V(\psi^{(t+n_i)}, \psi))dV P(\psi^t)d\psi^t \\ &= \mathbb{E}^{\psi^t}[\mathbb{E}^V[u(\psi^{(t+n_i)})]] \end{aligned}$$

■

Lemma 2

$$\mathbb{E}^V[\delta L^{t+i}(v^{t+i}, \psi^{t+i})] = \frac{d\mathbb{E}^V[L^{t+i}(v^{t+i}, \psi^{t+i})]}{d\psi^{t+i}}$$

Proof. Due to randomness in picking the point to be updated in iteration $t+i$ We have

$$\begin{aligned} &\mathbb{E}^V[L^{t+i}(v^{t+i}, \psi^{t+i})] = \int L(y, \psi^{t+i})dy \\ &\Rightarrow \frac{d\mathbb{E}^V[L^{t+i}(v^{t+i}, \psi^{t+i})]}{d\psi^{t+i}} = \mathbb{E}^V[\frac{dL^{t+i}(v^{t+i}, \psi^{t+i})}{d\psi^{t+i}}] = \mathbb{E}^V[\delta L^{t+i}(v^{t+i}, \psi^{t+i})] \end{aligned}$$

■

Lemma 3

$$\mathbb{E}^V[L^{t+i}(v^{t+i}, \psi^{t+i})] = \mathbb{E}^{v^{t+i}}[L^{t+1}(v^{t+i}, \psi^{t+i})]$$

Proof.

Using the definition of V^{t+i} in equation 6, the fact that V is a joint variable of each V^{t+i} and an any iteration $t+i$ the chance of picking any data point is completely random and indpendent of any other iteration.

$$\mathbb{E}^V[L^{t+i}(v^{t+i}, \psi^{t+i})] = \mathbb{E}^{v^{t+i}}[L^{t+1}(v^{t+i}, \psi^{t+i})]$$

■

Lemma 4

$$\mathbb{E}^V[\frac{dL^{t+i}(v^{t+i},\psi^{t+i})}{d\psi^{t+i}}\frac{dL^{t+j}(v^{t+j},\psi^{t+i})}{d\psi^{t+j}}] = \frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}}\frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}}$$

Proof. Two different data points picked at iteration $(t+i)$ and $(t+j)$ are independent of each other. Using this fact and the definition of potetntial function V in equation 15

$$\begin{aligned}\mathbb{E}^V[\frac{dL^{t+i}(v^{t+i},\psi^{t+i})}{d\psi^{t+i}}\frac{dL^{t+j}(v^{t+j},\psi^{t+i})}{d\psi^{t+j}}] &= \mathbb{E}^V[\frac{dL^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}}]\mathbb{E}^V[\frac{dL^{t+j}(v^{t+j},\psi^{t+i})}{d\psi^{t+j}}] \\ \left(\text{using lemma 2}\right) &= \frac{d\mathbb{E}^V[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}}\frac{d\mathbb{E}^V[L^{t+j}(v^{t+j},\psi^{t+j})]}{d\psi^{t+j}} \\ \left(\text{using lemma 3}\right) &= \frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}}\frac{d\mathbb{E}^{v^{t+j}}[L^{t+j}(v^{t+j},\psi^{t+j})]}{d\psi^{t+j}}\end{aligned}$$

■

Theorem 2 *We define ψ_* as the global optima and Ω_0 as the hessian of the loss at ψ_* i.e. $\Omega_0 = \frac{d^2\mathbb{E}[L(\psi_*)]}{d\psi_*^2}$ (assuming that ψ is univariate) then*

$$\frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}} = \Omega_0(\psi_t - \psi_* + \delta_i) + \mathcal{O}(\rho_t^2)$$

where $\mathcal{O}(\rho_t^2) = \mathcal{O}(|\psi_{t+i} - \psi_*|^2)$ with the assumption that $\mathcal{O}(\rho_{t+i})$ is small $\forall i \geq 0$ and $\delta_i = \psi_{t+i} - \psi_t$.

Proof. Lets define $\phi(\psi_{t+i}) = \mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]$ Using Taylor's theorem and expanding around ψ_*

$$\begin{aligned}\phi(\psi^{t+i}) &= \phi(\psi_*) + \frac{d\phi(\psi_*)}{d\psi_*}(\psi^{t+i} - \psi_*) + \frac{(\psi^{t+1} - \psi_*)^2}{2}\frac{d^2\phi(\psi_*)}{d\psi_*^2} + \mathcal{O}((\psi^{t+i} - \psi_*)^3) \\ &= \phi(\psi_*) + \frac{(\psi^{t+i} - \psi_*)^2}{2}\frac{d^2\phi(\psi_*)}{d\psi_*^2} + \mathcal{O}((\psi^{t+i} - \psi_*)^3) \left(\text{as } \frac{d\phi(\psi_*)}{d\psi_*} = 0 \text{ at optima}\right) \\ &\Rightarrow \frac{d\phi(\psi^{t+i})}{d\psi^{t+i}} = (\psi^{t+i} - \psi_*)\frac{d^2\phi(\psi_*)}{d\psi_*^2} + \mathcal{O}((\psi^{t+i} - \psi_*)^2)\end{aligned}$$

$$\Rightarrow \frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}} = \Omega_0(\psi^t - \psi_* + \delta_i) + \mathcal{O}(\rho_t^2)\Bigg(\text{with the assumption that } \mathcal{O}(\rho_t) \text{ is small we have } \mathcal{O}(\rho_{t+i}^2) = \mathcal{O}(\rho_t^2) \Bigg)$$

■

Theorem 3 *With ψ_* as defined in theorem 4.2 and assuming that ψ is univariate we have*

$$\mathbb{E}^{v^{t+i}}[(\frac{dL^{t+i}(v^{t+i},\psi^{t+i})}{d\psi^{t+i}})^2] = \Omega_1 + \mathcal{O}(\mathbb{E}[\mathcal{O}(\rho_t)]) + \mathcal{O}(\rho_t^2)$$

where $\mathcal{O}(\rho_t^2)$ and δ_i are as defined in theorem 4.2 and $\Omega_1 = \mathbb{E}^{v^{t+i}}[(\frac{dL^{t+i}(v^{t+i},\psi_*)}{d\psi_*})^2]$

Proof. Expanding $L^{t+i}(v^{t+i}, \psi^{t+i})$ around ψ_* using Taylor's theorem

$$
\begin{aligned}
L^{t+i}(v^{t+i}, \psi^{t+i}) &= L^{t+i}(v^{t+i}, \psi_*) + \frac{dL^{t+i}(v^{t+i}, \psi_*)}{d\psi_*}(\psi^{t+i} - \psi_*) \\
&\quad + \frac{1}{2}\frac{d^2L^{t+i}(v^{t+i}, \psi_*)}{d\psi_*^2}(\psi^{t+i} - \psi_*)^2 + \mathcal{O}((\psi^{t+i} - \psi_*)^3) \\
\Rightarrow \frac{dL^{t+i}(v^{t+i}, \psi^{t+i})}{d\psi^{t+i}} &= \frac{dL^{t+i}(v^{t+i}, \psi_*)}{d\psi_*} + \frac{d^2L^{t+i}(v^{t+i}, \psi_*)}{d\psi_*^2}(\psi^{t+i} - \psi_*) + \mathcal{O}((\psi^{t+i} - \psi_*)^2) \\
&\Rightarrow \mathbb{E}^{v^{t+i}}[(\frac{dL^{t+i}(v^{t+i}, \psi^{t+i})}{d\psi^{t+i}})^2] = \mathbb{E}^{v^{t+i}}[(\frac{dL^{t+i}(v^{t+i}, \psi_*)}{d\psi_*})^2 \\
&\quad + 2\frac{dL^{t+i}(v^{t+i}, \psi_*)}{d\psi_*}\frac{d^2L^{t+i}(v^{t+i}, \psi_*)}{d\psi_*^2}(\psi^{t+i} - \psi_*) + \mathcal{O}((\psi^{t+i} - \psi_*)^2)] \\
&\Rightarrow \mathbb{E}^{v^{t+i}}[(\frac{dL^{t+i}(v^{t+i}, \psi^{t+i})}{d\psi^{t+i}})^2] = \Omega_1 + \mathcal{O}(\mathbb{E}[(\psi_{t+i} - \psi_*]) + \mathcal{O}(\rho_t^2) \\
&= \Omega_1 + \mathcal{O}(\mathbb{E}[\mathcal{O}(\rho_t)]) + \mathcal{O}(\rho_t^2)
\end{aligned}
$$

■

Theorem 4 *The variance of the parameter ψ at the end of a sub-epoch S in block S_i which updated n_i points as defined in equation 8 is*

$$
\begin{aligned}
Var(\psi^{t+n_i}) &= Var(\psi^t) - 2\eta_t n_i \Omega_0 (Var(\psi^t)) - 2\eta_t n_i \Omega_0 CoVar(\psi_t, \bar{\delta_t}) + \eta_t^2 n_i \Omega_1 \\
&\quad + \underbrace{\mathcal{O}(\eta_t^2\rho_t) + \mathcal{O}(\eta_t\rho_t^2) + \mathcal{O}(\eta_t^3) + \mathcal{O}(\eta_t^2\rho_t^2)}_{\Delta_t}
\end{aligned}
$$

Constants Ω_0 and Ω_1 are defined in theorems 4.2 and theorems 3 respectively.

Proof. We start with analysing $\mathbb{E}^V[u(\psi^{(t+ni)})]$ term from lemma 1

$$\mathbb{E}^V[u(\psi^{(t+ni)})] = \mathbb{E}^V[u(\psi^t + (\underbrace{-\sum_{i=1}^{n_i}\eta_{t+i}\delta L^{t+i}(v^{t+i},\psi^{t+i})}_{\nabla}))]$$

$$= \mathbb{E}^V[u(\psi^t) - \frac{du(\psi^t)}{d\psi^t}\nabla + \frac{1}{2}\frac{du^2(\psi^t)}{d(\psi^t)^2}\nabla^2 + \mathcal{O}(\eta_t^3)]$$

$$= u(\psi^t) - \eta_t\frac{du(\psi^t)}{d\psi^t}\mathbb{E}^V[\sum_{i=1}^{n_i}\delta L^{t+i}(v^{t+i},\psi^{t+i})] + \eta_t^2\frac{1}{2}\frac{du^2(\psi^t)}{d(\psi^t)^2}\mathbb{E}^V[(\sum_{i=1}^{n_i}\delta L^{t+i}(v^{t+i},\psi^{t+i}))^2]$$

$$+ \mathcal{O}(\eta_t^3) \qquad \Big(\text{since } \eta_t = \eta_{t+i} \text{ within a block and expanding } \nabla\Big)$$

$$= u(\psi^t) - \eta_t\frac{du(\psi^t)}{d\psi^t}\sum_{i=1}^{n_i}\frac{d\mathbb{E}^V[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}} + \eta_t^2\frac{1}{2}\frac{du^2(\psi^t)}{d(\psi^t)^2}\mathbb{E}^V[(\sum_{i=1}^{n_i}\frac{dL^{t+i}(v^{t+i},\psi^{t+i})}{d\psi^{t+i}})^2]$$

$$+ \mathcal{O}(\eta_t^3)$$

$$= u(\psi^t) - \eta_t\frac{du(\psi^t)}{d\psi^t}(\sum_{i=1}^{n_i}\frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}})$$

$$\Big(\text{using Lemma 3}\Big)$$

$$+ \eta_t^2\frac{1}{2}\frac{du^2(\psi^t)}{d(\psi^t)^2}\left[\mathbb{E}^V[\sum_{i=1}^{n_i}(\frac{dL^{t+i}(v^{t+i},\psi^{t+i})}{d\psi^{t+i}})^2] + \mathbb{E}^V[\sum_{i\neq j}\frac{dL^{t+i}(v^{t+i},\psi^{t+i})}{d\psi^{t+i}}\frac{dL^{t+j}(v^{t+j},\psi^{t+j})}{d\psi^{t+j}}]\right]$$

$$+ \mathcal{O}(\eta_t^3)$$

$$= u(\psi^t) - \eta_t\frac{du(\psi^t)}{d\psi^t}(\sum_{i=1}^{n_i}\frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}}) + \eta_t^2\frac{1}{2}\frac{du^2(\psi^t)}{d(\psi^t)^2}\left[\sum_{i=1}^{n_i}\mathbb{E}^{v^{t+i}}[(\frac{dL^{t+i}(v^{t+i},\psi^{t+i})}{d\psi^{t+i}})^2]\right.$$

$$\left. + (\sum_{i\neq j}\frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}}\frac{d\mathbb{E}^{v^{t+j}}[L^{t+j}(v^{t+j},\psi^{t+j})]}{d\psi^{t+j}})\right] + \mathcal{O}(\eta_t^3) \tag{16}$$

$$\Big(\text{using Lemma 4}\Big)$$

From equation 17 and lemma 1

$$\mathbb{E}^{\psi^{(t+n_i)}}[u(\psi^{(t+n_i)})] = \mathbb{E}^{\psi^{(t)}}\left[u(\psi^t) - \eta_t\frac{du(\psi^t)}{d\psi^t}(\sum_{i=1}^{n_i}\frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}})\right.$$
$$+ \eta_t^2\frac{1}{2}\frac{du^2(\psi^t)}{d(\psi^t)^2}\left[\sum_{i=1}^{n_i}\mathbb{E}^{v^{t+i}}[(\frac{dL^{t+i}(v^{t+i},\psi^{t+i})}{d\psi^{t+i}})^2]\right.$$
$$\left.\left. + (\sum_{i\neq j}\frac{d\mathbb{E}^{v^{t+i}}[L^{t+i}(v^{t+i},\psi^{t+i})]}{d\psi^{t+i}}\frac{d\mathbb{E}^{v^{t+j}}[L^{t+j}(v^{t+j},\psi^{t+j})]}{d\psi^{t+j}})\right]\right] + \mathcal{O}(\eta_t^3) \tag{17}$$

From equation above the variance of ψ^{t+n_i} is

$$
\begin{aligned}
&Var(\psi^{t+n_i}) = \mathbb{E}^{\psi^{(t+n_i)}}[(\psi^{(t+n_i)})^2] - \left(\mathbb{E}^{\psi^{(t+n_i)}}[\psi^{(t+n_i)}]\right)^2 \\
&= \mathbb{E}^{\psi^t}[(\psi^t)^2] - \eta_t n_i \mathbb{E}^{\psi^t}[2\psi^t(\Omega_0(\psi^t - \psi_* + \bar{\delta_t}) + \mathcal{O}(\rho_t^2))] \\
&\left(\text{using theorem 4.2 and defining } \bar{\delta_t} = \frac{\sum_{i=1}^{n_i} \delta_i}{n_i}\right) \\
&+ \eta_t^2 \frac{1}{2} \mathbb{E}^{\psi^t}[2\{n_i(\Omega_1 + \mathcal{O}(\mathbb{E}[\rho_t]) + \mathcal{O}(\rho_t^2)) \\
&+ \sum_{i \neq j} (\Omega_0(\psi^{t+i} - \psi_*) + \mathcal{O}(\rho_t^2))(\Omega_0(\psi^{t+j} - \psi_*) + \mathcal{O}(\rho_t^2))\}] \\
&- \left(\mathbb{E}^{\psi^t}[\psi^t] - \eta_t n_i \mathbb{E}^{\psi^t}[(\Omega_0(\psi^t - \psi_* + \bar{\delta_t}) + \mathcal{O}(|\psi^t - \psi_*|^2))]\right)^2 \\
&= \mathbb{E}^{\psi^t}[(\psi^t)^2] - 2\Omega_0 \eta_t n_i \mathbb{E}^{\psi^t}[(\psi^t)^2] + 2\Omega_0 \eta_t n_i \psi_* \mathbb{E}^{\psi^t}[\psi^t] - 2\Omega_0 \eta_t n_i \mathbb{E}^{\psi^t}[\psi^t \bar{\delta_t}] - \mathcal{O}(\eta_t \rho_t^2) \\
&+ \eta_t^2 n_i \Omega_1 + \mathcal{O}(\eta_t^2 \rho_t) + \mathcal{O}(\eta_t^2 \rho_t^2) + \mathcal{O}(\eta_t^2 \rho_t^3) + \mathcal{O}(\eta_t^2 \rho_t^4) \\
&- \left(\mathbb{E}^{\psi^t}[\psi^t]\right)^2 + 2n_i \eta_t \mathbb{E}^{\psi^t}[\psi^t](\mathbb{E}^{\psi^t}[\Omega_0 \psi^t] - \Omega_0 \psi_* + \mathbb{E}^{\psi^t}[\Omega_0 \bar{\delta_t}] + \mathcal{O}(\rho_t^2)) - \mathcal{O}(\eta_t^2 \rho_t^2) + \mathcal{O}(\eta_t^3) \\
&= Var(\psi^t) - 2\eta_t n_i \Omega_0 (Var(\psi^t)) - 2\eta_t n_i \Omega_0 CoVar(\psi_t, \bar{\delta_t}) + \eta_t^2 n_i \Omega_1 \\
&+ \underbrace{\mathcal{O}(\eta_t^2 \rho_t) + \mathcal{O}(\eta_t \rho_t^2) + \mathcal{O}(\eta_t^3) + \mathcal{O}(\eta_t^2 \rho_t^2)}_{\Delta_t}
\end{aligned} \tag{18}
$$

■

Make clear that we are simplifying here and making the assumption that ψ is uni-variate

4.3 Inter sub-epoch variance

Two blocks Z_i and $Z_{i'}$ in a given sub-epoch are independent if for each $y \in Z_i$ and $y^{'} \in Z_{i'}$ we have

$$
\begin{aligned}
\nabla L_y(\psi) &= \nabla L_y(\psi - \eta \nabla L_{y'}(\psi)) \\
\text{and } \nabla L_{y'}(\psi) &= \nabla L_{y'}(\psi - \eta \nabla L_y(\psi))
\end{aligned} \tag{19}
$$

From our algorithm (as defined in 1) and equation 5, for any two points $y \in Z_i$ and $y' \in Z_{i'}$ their rows or columns do not overlap (figure 3). From equation (5) we see that y does not modify ψ in positions for which $i \neq y_i$, $j \neq y_j$ and $k \neq y_k$. Therefore, because y and y' do not overlap in any coordinates, an update from $\nabla L_{y'}$ will update different set of parameter in π, β than ∇L_y, where ∇L_y is the gradient at point y Additionally from equation (5) we see that any updates on ∇L_y only use values from $\pi_{y_i,*}$ and $\beta_{y_j,*}$, and thus do not use any values that would be updated by $\nabla L_{y'}$. Because updates from y only effect parameters in y's coordinates, updates from y are only based on parameters in y's coordinates, and y and y' have no overlapping coordinates. Thus equation 19 holds and the updates in block Z_i and $Z_{i'}$ in a sub-epoch s happen on two different sub-set of parameters and are independent of each other. In the synchronize step, at the end of a sub-epoch S_{n+1}, the new parameter set $\Psi_{S_{n+1}}$ is obtained by aggregating the non-overlapping updates $\delta\psi^i_{S_{n+1}}$ and $\delta\psi^j_{S_{n+1}}$ from any two blocks S^i_{n+1} and S^j_{n+1} independently. We can write the variance $Var(\Psi_{S_{n+1}})$ at the

end of sub-epoch S_{n+1} as

$$
\begin{aligned}
Var(\Psi_{S_{n+1}}) &= \sum_{i=1}^{i=w} Var(\psi^i_{S_{n+1}}) \\
&= \sum_{i=1}^{i=w} \Big[Var(\psi^i_{S_n}) - 2\eta_{S_n} n_i \Omega^i_0 (Var(\psi^i_{S_n})) - 2\eta_{S_n} n_i \Omega^i_0 (CoVar(\psi^i_{S_n}, \bar{\delta}^i_{S_n})) \\
&+ \eta^2_{S_n} n_i \Omega^i_1 + \Delta_{S^i_n} \Big] \Big(\text{Using equation 18} \Big) \\
&= Var(\Psi_{S_n}) - 2\eta_{S_n} \sum_{i=1}^{i=w} n_i \Omega^i_0 Var(\psi^i_{S_n}) - 2\eta_{S_n} \sum_{i=1}^{i=w} n_i \Omega^i_0 CoVar(\psi^i_{S_n}, , \bar{\delta}^i_{S_n}) \\
&+ \eta^2_{S_n} \sum_{i=1}^{i=w} n_i \Omega^i_1 + \mathcal{O}(\Delta_{S_n}) \qquad (20)
\end{aligned}
$$

4.4 Slow learner agnosticism

In equation 20 the variance after each sub-epoch S depends on n_i and η_{S_n} ignoring the higher order terms ($\mathcal{O}(\Delta_{S_n})$). Chosing the step-size η_{S_n} small enough such that

$$
2\eta_{S_n} \sum_{i=1}^{i=w} n_i \Omega^i_0 Var(\psi^i_{S_n}) + 2\eta_{S_n} \sum_{i=1}^{i=w} n_i \Omega^i_0 CoVar(\psi^i_{S_n}, \bar{\delta}^i_{S_n}) > \eta^2_{S_n} \sum_{i=1}^{i=w} n_i \Omega^i_1 \qquad (21)
$$

leads to

$$
Var(\Psi_{S_{n+1}}) < Var(\Psi_{S_n})
$$

The variance decreases when n_i increases given the above conditions about the step size hold which is easy to uphold since the right hand side of inequality is proportinal to second order of step size ($\eta^2_{S_n}$) and the left hand side first order. From equation 1 we have the convergence of the algorithm for a generaic case where different processors are doing different number of iterations over the data points allotted to them. Hence it make sense for the faster ones to do more updates to decrease the variance with appropriate choice of step sizes instead of waiting for the slower ones to finish. By decreasing the variance it's making the parameters move closer to the optima in general because the algorithm is provably convergent. A caveat: doing too many updates (n_i) might increase the variance accross blocks in the sub-epoch if step size η_S is not chosen properly.

5 Experiments

5.1 Scalability experiments in data as well as parameter

Experiments to show how it scales with more processors (Do we have enough processors?) a) Scaling with parameters (number of topics), and b) Scaling with Data (Documents)

Figure 5.2 shows a toy data set LDA objective convergence. The no-wait method is where the processors donot wait for every other processor to finish instead they keep working until time to synchronize. The data set is is a 100 by 1000, document-vocab matrix. The average time take in each iteration for the no-wait case is 40.9375 and for wait case is 41.375 seconds.

5.2 Speed comparison and less sycnhing experiments

Comparison with PSGD on the speed and accuracy of convergence (Do we compare on the original objective or the new objective)

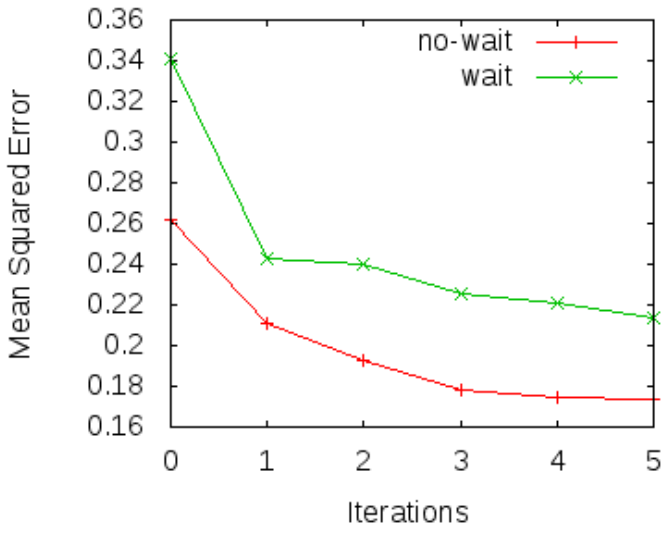

Figure 2: A toy data example run for LDA formulation

5.3 Qualitative results on latent space models

We do both for MMSB and LDA both

Qirong and Alex think that we should include symmetric matrix factorization as well as it shows the scheduling strategy when thinsg are not cleanly seperated.

6 Results Analaysis and Discussion

7 Conclusion

References

[1] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[3] Avner Friedman. Stochastic differential equations and applications. In Jaures Cecconi, editor, *Stochastic Differential Equations*, volume 77 of *C.I.M.E. Summer Schools*, pages 75–148. Springer Berlin Heidelberg, 1975.

[4] Kenneth Kreutz-Delgado, Joseph F. Murray, Bhaskar D. Rao, Kjersti Engan, Te-Won Lee, and Terrence J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Comput.*, 15(2):349–396, February 2003.

[5] H. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.