大規模データのための階層ディリクレ過程の並列推定

大元　司†　東羅翔太郎†　江口　浩二†

† 神戸大学　〒 657–8501 神戸市灘区六甲台町 1-1

E-mail: †omoto@cs25.scitec.kobe-u.ac.jp，tora@cs25.scitec.kobe-u.ac.jp，eguchi@port.kobe-u.ac.jp

あらまし　ノンパラメトリックベイズアプローチの一つである階層ディリクレ過程をトピックモデルに応用することにより，与えられたテキストデータ集合から事前知識なしに潜在トピック数を推定しつつ，潜在トピックを推定することができる．しかしながら，これには 空間計算量と時間計算量の二つの面で多大な計算コストが必要となるため，推定手法の効率性が課題であった．本研究では，Chinese Restaurant Franchise に基づく階層ディリクレ過程の推定手法を分散並列化することにより，より大規模なデータに対してより効率的な推定の実現を目指す．

キーワード　階層ディリクレ過程，トピックモデル，並列計算

Tsukasa ŌMOTO†, Shotaro TORA†, and Koji EGUCHI†

† Kobe University

1-1 Rokkodaicho, Nada, Kobe 657–8501 Japan

E-mail: †omoto@cs25.scitec.kobe-u.ac.jp，tora@cs25.scitec.kobe-u.ac.jp，eguchi@port.kobe-u.ac.jp

1. はじめに

今日，我々は情報技術の発展により多種多様なデータを扱っている．最近ではビックデータと呼ばれる膨大なデータを扱うこともある．そのような大規模で多様なデータから我々は有益な情報を得たいと思うことがしばしばある．

その分析手法の 1 つに，潜在的ディリクレ配分法（Latent Dirichlet Allocation，LDA）[2] を典型とするトピックモデルが挙げられる．トピックモデルとは，ある文書集合に対して，文書と単語の間にトピックという潜在変数を仮定し，文書の単語はそのトピックの混合分布から生成される，と仮定した生成モデルである．トピックモデルを様々なデータ解析に応用することで情報検索や情報推薦などに役立てることができる．

LDA はパラメトリックベイズモデルのためモデルの学習にトピック数を事前に与える必要がある．我々がそれぞれのデータに対しこのトピック数を決定することはしばしば困難である．この問題を解決するためにノンパラメトリックベイズアプローチである階層ディリクレ過程（Hierarchical Dirichlet Processes，HDP）[6] を LDA に適用する(注1)ことでデータからトピック数を推定することができる．

しかしながら，HDP-LDA は LDA に比べ多大な計算コストがかかるので，我々が現在扱っているような大規模なデータを学習するためには膨大な計算時間が必要となる．この問題を解決するために本研究では HDP-LDA の分散並列化手法を提案し，大規模なデータに対する学習においてその有効性を示すことを目指す．

以降，本論文は次のような構成になっている．第 2 章で提案手法に欠かすことのできない既存手法を説明する．さらに，並列化の分野で先行する研究を紹介する．第 3 章で我々の並列化手法を紹介する．実験の結果を第 4 章で与え，第 5 章で本論文をまとめる．

(注1)：我々は HDP-LDA と呼ぶ

2. 関連研究

この章では基礎となるモデルとそのサンプリングスキーム，および先行研究を紹介する．

2.1 Hierarchical Dirichlet Processes

階層ディリクレ過程は Teh らによって提案されたノンパラメトリックベイズアプローチである [6]．階層ディリクレ過程はディリクレ過程 [4] を 2 段階に用いることで構成され，生成過程は以下のようになる．

$$G_0|\gamma, H \sim DP(\gamma, H) \tag{1}$$

$$G_j|\alpha, G_0 \sim DP(\alpha, G_0) \tag{2}$$

$$\theta_j|G_j \sim G_j \tag{3}$$

$$x_{ji}|\theta_j \sim F(\theta_{ji}) \tag{4}$$

ここで，H は基底分布，α,γ はハイパーパラメータであり，DP はディリクレ過程からのサンプリングを表す．図 1 はこの生成

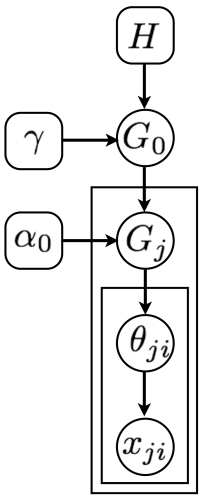

図 1　HDP のグラフィカルモデル

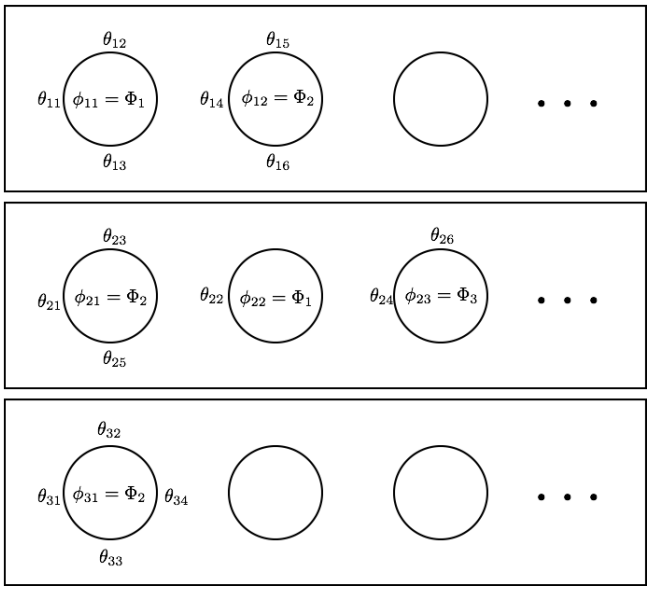

図 2　Chinese Restaurant Franchise．各レストランは長方形で示され，テーブルが円で示されている．

過程に対応するグラフィカルモデルである．階層ディリクレ過程は複数のグループデータ（例えば文書）が存在する問題にしばしば用いられる．

2.2　Chinese Restaurant Franchise

ディリクレ過程からのサンプリングは無限次元からのサンプリングになるため，階層ディリクレ過程の構築には工夫が必要である．Teh らは提案論文において 3 つの手法，Chinese restaurant franchise, augmented representation, direct assignment を紹介した [6]．本研究では比較的直観的に理解がしやすい Chinese restaurant franchise（CRF）を使用した．

CRF は Chinese restaurant process [6]（CRP）を共通の料理を持つ複数のレストランへ拡張したサンプリング手法である．トピックモデルにおいて，中華料理店が文書，料理がトピック，客が単語に対応する．図 2 はこのメタファを示す．また，この論文で用いられる記号の説明を表 1 で与える．

CRF による HDP の構築は次のようになる [6] [9]．

表 1　記号の説明

記号	説明
θ_{ji}	レストラン j の i 番目の客が食べている料理
Φ_k	全てのレストランで共通の料理
ϕ_{jt}	レストラン j，テーブル t の料理
t_{ji}	レストラン j の客 i が座るテーブル
k_{jt}	レストラン j の，テーブル t の料理
$n_{jt\cdot}$	レストラン j のテーブル t に座っている客の数
$n_{\cdot\cdot k}$	全てのレストランで料理 k を食べている客の数
$m_{\cdot k}$	全てのレストランで料理 k が置かれているテーブルの数

a）　sampling t_{ji}

j 番目のレストランの i 番目の客は以下の式に従ってサンプリングされる．

$$p(t_{ji} = t|\boldsymbol{t}^{-ji}, \boldsymbol{k}) \propto \begin{cases} n_{jt\cdot} & \text{if } t \text{ is previously used.} \\ \alpha_0 & \text{if } t = t^{new} \end{cases} \quad (5)$$

b）　sampling k_{jt}

j 番目のレストランのテーブル t に置く料理は以下の式に従ってサンプリングされる．

$$p(k_{jt} = k|\boldsymbol{t}, \boldsymbol{k}^{-jt}) \propto \begin{cases} m_{\cdot k} & \text{if } k \text{ is previously used.} \\ \gamma & \text{if } k = k^{new} \end{cases} \quad (6)$$

c）　sampling x_{ji}

最後に以下の式に従って単語がサンプリングされる．

$$p(\boldsymbol{x}|\boldsymbol{t}, \boldsymbol{k}) = \prod_k f_k(\{x_{ji} : k_{ji} = k\}) \quad (7)$$

$$f_k(\{x_{ji} : k_{ji} = k\} = \frac{\Gamma(V\beta)}{\Gamma(n_{\cdot\cdot k} + V\beta)} \frac{\prod_v \Gamma(n^v_{\cdot\cdot k} + \beta)}{\Gamma^v(\beta)} \quad (8)$$

ここで，V は語彙数，β はディリクレハイパーパラメータである．

2.3　並　列　化

先行研究として，Newman らの Approximate Distributed HDP（AD-HDP）[5] が挙げられる．AD-HDP は本来ならば依存関係にある変数の結びつきは実際には弱いという考えに基づき HDP を近似を用いて並列化している．実際にこの近似を用いる並列化は HDP の非並列推定よりも良い結果を残す．我々と違って，Newman らは direct assignment [6] を用いて HDP を実装した．

Asuncion らの Async-HDP は異なる環境下での分散推定を想定したアルゴリズムである [1]．同時に，Asuncion らは Newman らの AD-HDP と根底は同じであるが，データの集約の部分で異なるアプローチを取った Parallel-HDP も提案した．Parallel-HDP はスレッドに由来する変数のみを通信に用いる．Async-HDP のノード間の通信はこの Parallel-HDP のデータ集約の考えを用いて実現されている．

3.　ハイブリッド型推論

東羅らは SMP クラスター上で LDA の MPI/OpenMP ハイブリッド型並列推定手法を開発した [8]．我々も同様に HDP-LDA の分散並列推定のためにハイブリッド型手法を提案する．

表 2　データセット

	KOS	NIPS
文書数 (D)	3,430	1,500
語彙数 (V)	6,906	12,419
総単語数 (N)	467,714	1,932,365

3.1　Hybrid-AD-HDP

最初に，我々は東羅らと同様のアプローチを取った Hybrid-AD-HDP を提案する．Newman ら [5] は，2.3 節でも言及した AD-HDP を，MPI を使用し各コアに対して通信する形で並列化している．これに対して Hybrid-AD-HDP は，MPI によって各ノードに対して通信し，ノード内ではマルチスレッド処理によって並列化を実現する．ノード内の推論とノード間のデータ集約にもまた AD-HDP のアプローチを用いる．

具体的には，各ノードはギブスサンプリングの前にグローバルモデルの情報を分配し共通の状態でサンプリングを始める．次にサンプリング後の情報をそのまま集約し，1 つ前のモデルの情報との差分を求める．これを収束するまで，あるいは一定回数繰り返すことでグローバルモデルを更新していく．

3.2　Hybrid-Parallel-HDP

次に，我々はノード内の推論とノード間のデータ集約に Parallel-HDP [1] のアプローチを適用した Hybrid-Parallel-HDP を提案する．

Hybrid-AD-HDP の場合，ノード内の各ギブスサンプリングの後に全体の同期を取らなければ，高確率でグローバルモデルの状態が不適当なものになってしまうことがわかった．各ノードで推論を始めるときに，ノード内におけるローカルモデルとグローバルモデルの間に差異がある場合，ノード内の推論がうまくいかないからである．

そこで，我々は AD-HDP ではなく Parallel-HDP の考えに基づき，ノード間の通信にノードに由来する差分のみを通信することで，上記の問題を排除した手法を開発した．第 4 章の実験において，Hybrid-AD-HDP と比べるとわずかな計算コストの増加を確認するが，拡張性の面で利点を確保しながらノード間における無駄な通信を削減することができ，高速な推論が可能となることを後の実験で示す．

4.　実　　験

提案手法の有用性を示すために実験を行った(注2)．本論文の実験で用いたデータセットは KOS blog entries と NIPS full papers である(注3)．データセットの概要を表 2 に示す．

我々は各文書の 90%を訓練データ，残りの 10%をテストデータに分割する 10-fold cross-validation によりモデルを評価した [7]．評価尺度として以下の式で定義されるテストセットパープレキシティを求めた．

(注2)：実験に用いた一部のプログラムのソースコードは https://github.com/henry0312/LDA/ に公開している．

(注3)：これらのデータセットは http://archive.ics.uci.edu/ml/datasets/Bag+of+Words よりダウンロード出来る．

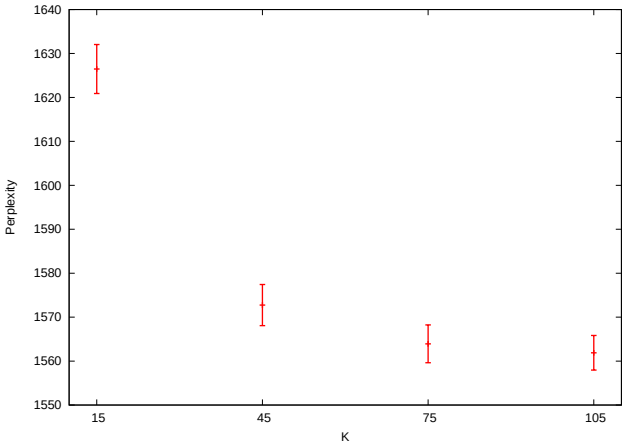

図 3　KOS における初期化実験．(1) の初期トピック数の違いによる結果の比較．

$$perplexity(\boldsymbol{w}) = \exp\left\{-\frac{1}{N}\log p(\boldsymbol{w}|Training\ corpus)\right\}$$

ここで，$\boldsymbol{w}$ はテストセットを構成する単語（延べ語）の集合，N はその総単語数（総述べ語数）である．

4.1　初　期　化

予備実験として HDP の初期化による違いを観察した．HDP の初期化には幾つかの方法が考えられる．この論文では 2 つの初期化方法で実験した．

（1）　初期トピック数を与える初期化

（2）　CRF に従う初期化

(1) の初期トピック数は KOS の時 $K = 15, 45, 75, 105$，NIPS の時 $K = 20, 70, 120, 150, 170, 220$ とした．(1) と (2) 両方において Burn-in を 20 回とした．また，比較のために LDA のトピック数を変化させる実験も行った．LDA のハイパーパラメータは $\alpha = 1/K, \beta = 0.5$ であり，HDP-LDA のハイパーパラメータは $\alpha_0 = E[Gamma(1,1)] = 1, \gamma = E[Gamma(1,0.1)] = 10, \beta = 0.5$（$Gamma$ 分布は rate parameter を使用）とし，Teh らの提案論文に倣って Burn-in 以降の各推論後に更新を行った [6] [3]．

KOS に対する実験結果を図 3，5，NIPS に対する実験結果を図 4，6，7，8 に示す．図 5，6，7 の横軸は LDA のトピック数であり，2 つの HDP はともに最終的に学習したモデルのパープレキシティを示している．図 3，4 の横軸は初期トピック数であり，図 8 の横軸は推論回数である．また，エラーバーは標準誤差を示す．

KOS に対する実験では (1) も (2) も LDA のベストパフォーマンス（図 5 におけるトピック数約 55 前後のときのパープレキシティで約 1,550）と一致しなかった．これは文書数の割に総単語数が少ないことが原因でそれぞれが正確にモデルを学習出来なかったためだと思われる．(1) と (2) の比較では (1) の方が最終的に良い結果となっているがその差はわずかである．

一方，NIPS に対する実験では (1)($K = 120, 170, 220$ の時) と (2) はともに LDA のベストパフォーマンス（図 6，7 におけるトピック数 130 前後のときのパープレキシティで約 1,450）と一致あるいはそれより良い結果となった．しかし，$K = 20, 70$

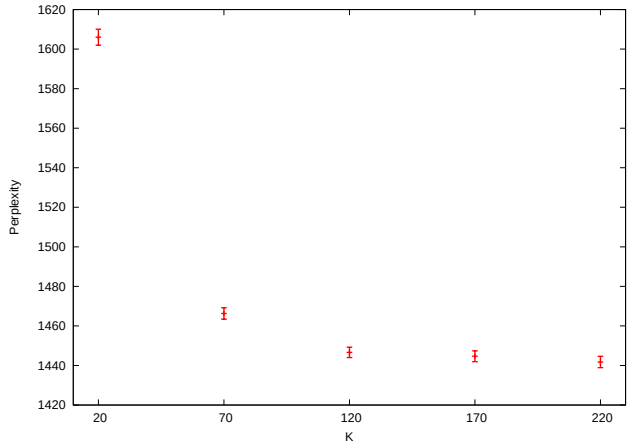

図 4　NIPS における初期化実験．(1) の初期トピック数の違いによる結果の比較．

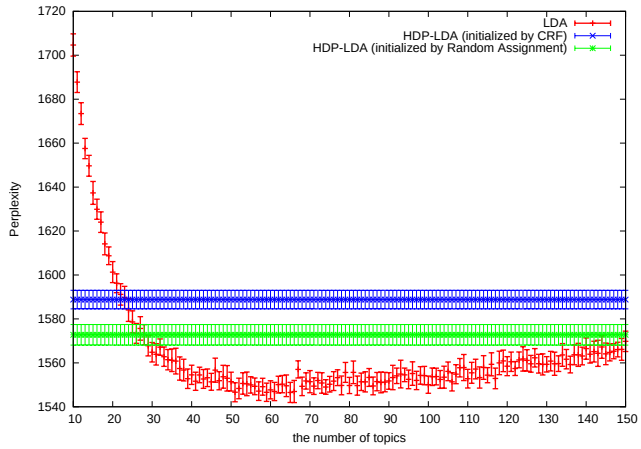

図 5　KOS における初期化実験 ($K = 45$)．LDA のトピック数 10-150

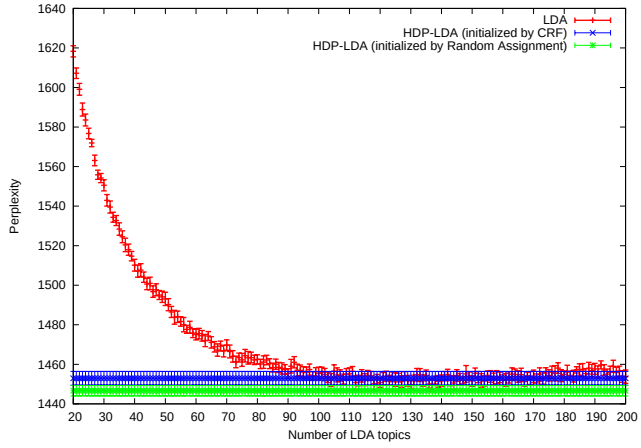

図 6　NIPS における初期化実験 ($K = 120$)．LDA のトピック数 20-200

の時 (1) は LDA のそれには及ばなかった．後者の時，最終的に学習したトピック数が前者より少なかったのでこれが原因だと考えられる．(1) と (2) の比較では (1) の方が好結果であることがわかるが，KOS の時と同じくその差はわずかである．また，この NIPS に対する実験結果は Teh らの実験 [6] における HDP の挙動と一致する．

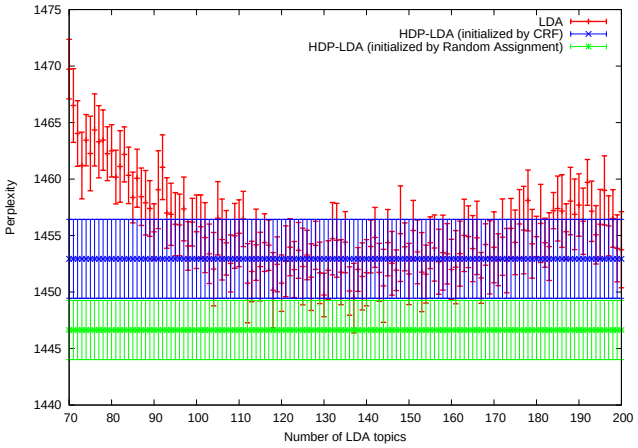

図 7　NIPS における初期化実験 ($K = 120$)．LDA のトピック数 70-200

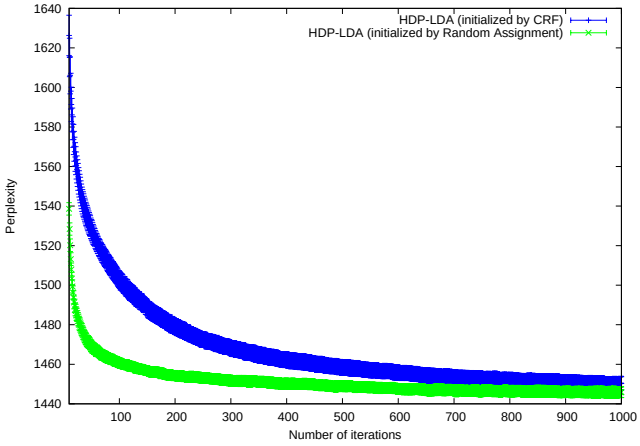

図 8　収束速度の比較 ($K = 120$)．推論回数 10-1000

表 3　実 験 環 境

CPU	Speed (GHz)	Sockets	Mem	Nodes	Network
Xeon E5410	2.33	2	32GB	8	10GbE

表 4　バージョン

OS	GCC	Open MPI	Boost
Debian 6.0.4	4.7.2	1.6.3	1.52

また，我々はメモリ使用量に関しても観察を行ったところ，(1) より (2) の方がメモリ使用量が少ないことがわかった．このことから (2) の方がより無駄のなくテーブル数を推論することが出来ると思われる．この点に着目し，次節で述べる並列化の実験は (2) の初期化に基づいて実施する．

4.2　並　列　化

我々は第 3 章で提案したハイブリッド型実装の高速化実験を行った．実験環境は表 3 で与える SMP クラスター環境である．使用したツールチェインやライブラリ等のバージョンは表 4 で示す．

4.2.1　ハイブリッド型実装による高速化実験

我々はハイブリッド型実装による効果を確かめるために，Hybrid-Parallel-HDP と非ハイブリッド型実装（MPI による単純な Parallel-HDP，MPI-HDP）の比較実験を行った．結果

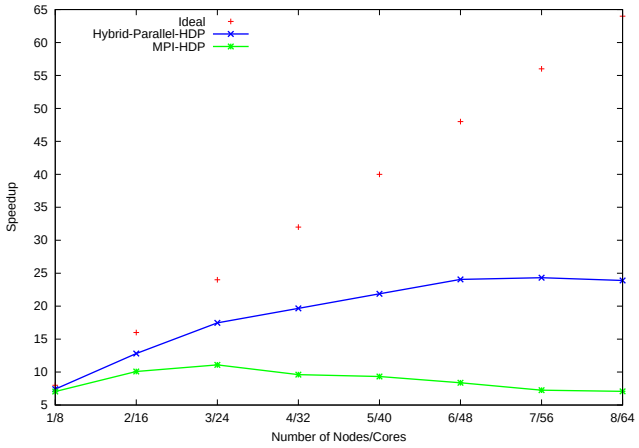

図 9　高速化実験

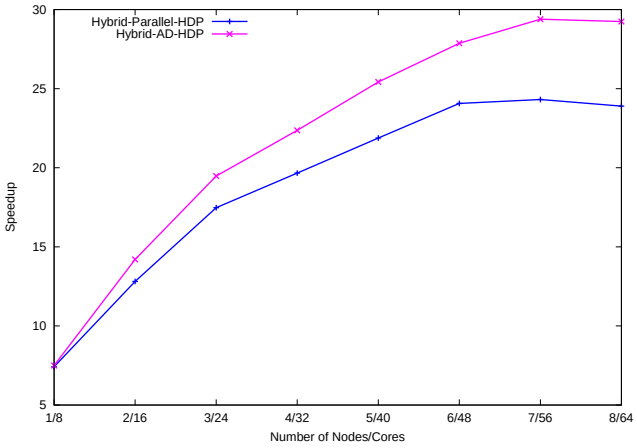

図 11　高速化実験．ハイブリッド型実装同士の比較

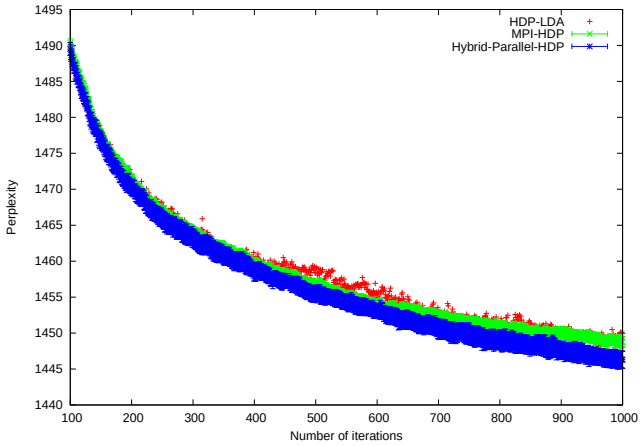

図 10　高速化実験．収束速度の比較．

表 5　高速化実験

コア数＼実装	HDP	MPI-HDP	Hybrid Parallel-HDP	Hybrid AD-HDP
1	64678.061s			
8		9182.572s	8719.837s	8628.472s
16		6409.333s	5048.243s	4554.129s
24		5833.417s	3701.630s	3321.430s
32		6726.878s	3289.321s	2892.103s
40		6931.981s	2955.968s	2544.341s
48		7722.203s	2687.676s	2320.549s
56		8916.456s	2660.224s	2200.594s
64		9142.231s	2706.472s	2211.734s

を図 9，10 および表 5 に示す．図 9 の横軸は並列度（ハイブリッド型ならば ノード数 × コア数 (8 コア)，非ハイブリッド型ならばコア数の総和）であり，縦軸は非並列版 HDP の実行時間を 1 としたときの高速化結果である．

図 9 より，Hybrid-Parallel-HDP は明らかに MPI-HDP よりも高速な推論が実現できていることがわかる．MPI-HDP の場合，3/24 と 4/32 の間で分散による効果を通信と同期コストが上回ってしまい，以降高速化されていない．しかし，Hybrid-Parallel-HDP ではそのような問題は起きず，6/48 までは高速化されているのがわかる．Hybrid-Parallel-HDP の 7/56 と 8/64 で結果が下がっている理由は，データセットのサイズが小さすぎたため分散による効果を通信と同期コストが上回ってしまったためだと考えられる．ゆえに，さらに大きなデータセットに対して実験を行えば期待通りの結果が得られると考えられる．

一見すると理想的な高速化 (Ideal) とはかけ離れているように見える．これは様々な原因が考えられが，まず，実験環境に依るものが大きいと考えられる．今回の実験環境ではアフィニティを考慮していないので，各ノードの能力の 100%を実験に割り当てられたわけではない．本格的な並列化実験を行う場合は，アフィニティを考慮するのが良いであろう．さらに，ノード間の通信や同期コストも無視出来ない(注4)．これらのことを考慮すれば我々のハイブリッド型の手法はうまく機能していると考えられる．

また，図 10 より，収束速度を比較すると，Hybrid-Parallel-HDP は MPI-HDP よりわずかではあるが，良い結果となっているのがわかる．また，非並列版の HDP と比較しても同等以上の結果であることがわかる．

4.2.2　Hybrid-AD-HDP と Hybrid-Parallel-HDP の比較実験

第 3 章で我々は AD-HDP と Parallel-HDP に基づく 2 つの分散並列手法，Hybrid-AD-HDP と Hybrid-Parallel-HDP を提案した．この章では Hybrid-AD-HDP と Hybrid-Parallel-HDP の違いを比較する実験を行った．その結果を図 11 および表 5 に示す．各ギブスサンプリング後に毎回同期を取ると，理論的に同じモデルを学習するため収束速度に関する実験は行っていない．図の見方は前章と同様である．

図 11 より，Hybrid-Parallel-HDP と Hybrid-AD-HDP を比較すると，後者の方がより高速化されていることがわかる．これは，Parallel-HDP と AD-HDP の違い，つまり各ノードのデータを集約する方法の違いに起因すると思われる．

確かに高速化の点において若干の差が見られるが，Hybrid-Parallel-HDP は拡張面での利点がある．すなわち，Hybrid-AD-HDP を使う限り，各ギブスサンプリング後に毎回同期を

(注4)：アムダールの法則

取る必要があるが，Hybrid-Parallel-HDP ならば容易に非同期な通信も行うことが可能になる．これはさらなる高速化の余地が残されているという事実でもある．

5. おわりに

我々は HDP-LDA の初期化に関する予備実験から CRF に従う初期化のコストパフォーマンスが優れていることを確認した．これは，CRF によって初期化した HDP-LDA を並列化する動機の 1 つとなった．そして，我々はハイブリット型分散推定は，ノード間の無駄な通信コストを削減し，ノードの資源をより有効に活用することができ，大規模なデータに対して計算資源を効率的に使う推定が可能になることを示した．

しかし，本論文の実験で用いたような 10GbE を持つ計算環境はまだ一般的ではない．もし実験環境が 1GbE だった場合，ハイブリッド型実装であっても通信コストが増大していた可能性がある．今後の課題としてより厳しいネットワーク環境下での推定方法やそのために必要であるより効率的な同期手法の開発が挙げられるが，Hybrid-Parallel-HDP はそのような問題に対処する能力を持っていると我々は信じている．

文　献

[1] Arthur Asuncion, Padhraic Smyth, and Max Welling. Asynchronous distributed learning of topic models. *Advances in Neural Information Processing Systems*, Vol. 21, pp. 81–88, 2008.

[2] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, Vol. 3, pp. 993–1022, 2003.

[3] M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, Vol. 90, No. 430, pp. 577–588, 1995.

[4] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pp. 209–230, 1973.

[5] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *The Journal of Machine Learning Research*, Vol. 10, pp. 1801–1828, 2009.

[6] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, Vol. 101, No. 476, pp. 1566–1581, 2006.

[7] Y.W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. *Advances in neural information processing systems*, Vol. 19, p. 1353, 2007.

[8] S. Tora and K. Eguchi. Mpi/openmp hybrid parallel inference for latent dirichlet allocation. In *Proceedings of the Third Workshop on Large Scale Data Mining: Theory and Applications*, p. 5. ACM, 2011.

[9] C. Wang and D.M. Blei. A split-merge mcmc algorithm for the hierarchical dirichlet process. *arXiv preprint arXiv:1201.1657*, 2012.

[10] S.A. Williamson, A. Dubey, and E.P. Xing. Exact and efficient parallel inference for nonparametric mixture models. *arXiv preprint arXiv:1211.7120*, 2012.