

混合メンバシップ・ブロックモデルのオンライン学習

小林 知己[†] 江口 浩二^{††}

[†] 神戸大学工学部情報知能工学科 〒 657-8501 兵庫県神戸市灘区六甲台町 1-1

^{††} 神戸大学大学院システム情報学研究科 〒 657-8501 兵庫県神戸市灘区六甲台町 1-1

E-mail: [†]kobayashi@cs25.scitec.kobe-u.ac.jp, ^{††}eguchi@port.kobe-u.ac.jp

あらまし 近年、ソーシャルメディアを実型としてネットワークとして表現できるデータが多く存在する。このネットワークデータを分析することにより、有用な知見を得たり、リンクを予測することなどが重要な課題となっている。この手法として、混合メンバシップブロックモデル (MMSB) があげられる。しかし、この手法は、観測されたネットワーク構造全体を通して学習することを仮定しており、時間とともに情報が増加するという現実的特徴に対応することが容易ではない。そこで本稿では、MMSB に対するオンライン学習手法を紹介し、実行時間と性能を既存手法と比較する。

キーワード 混合メンバシップブロックモデル, パーティクルフィルタ, ギブスサンプリング, オンライン学習

Tomoki KOBAYASHI[†] and Koji EGUCHI^{††}

[†] Department of Computer Science and Systems Engineering, Faculty of Engineering, Kobe University
1-1 Rokkodaicho, Nada, Kobe, Hyogo 657-8501 Japan

^{††} Graduate School of System Informatics, Kobe University
1-1 Rokkodaicho, Nada, Kobe, Hyogo 657-8501 Japan

E-mail: [†]kobayashi@cs25.scitec.kobe-u.ac.jp, ^{††}eguchi@port.kobe-u.ac.jp

1. はじめに

近年、情報量の急激な増加に伴い、大規模データを分析し、有用な知見を得たり、予測に活かすことなどが課題となっている。また、ネットワーク構造として表すことができる情報が少なくないため、ネットワーク分析の需要がますます高くなってきている。

本稿ではベイズ的アプローチによるネットワークのモデリングに着目する。ベイズ的アプローチによるネットワークのモデリングには大きく二つの流れがあると考えられる。一つ目は Newman ら [1] の混合分布を用いたモデルである。これはそれぞれのノードの隣接ノードが帰属する潜在的なグループが確率分布として表現できるという仮定から、ネットワークのモデリングを行ったものである。さらに Zhang ら [2] は Newman ら [1] の混合分布モデルに事前分布を取り入れた。これは文書解析として知られる潜在的ディリクレ配分法 (Latent Dirichlet Allocation: LDA) [3] [4] [5] をネットワークに適用したものと見え、Simple Social Network-LDA: SSN-LDA と呼ばれている。二つ目は、ブロックモデルと呼ばれるモデルをベイズ的に拡張した確率論的ブロックモデル (Stochastic Block Models: SBM) [6] である。これはノードが単一のクラスタに割り当た

るようにクラスタリングし、同じクラスタのノードはすべて相手クラスタにリンクを張る確率が同じであるという仮定のもとにネットワークのモデリングを行ったものである。また、上記で述べたモデルではクラスタの数を所与のものと仮定している。これを回避したモデルとして無限関係モデル (Infinite Relational Model: IRM) がある [7]。IRM は SBM におけるクラスタ数を Chinese Restaurant Process (CRP) によって無限個に拡張したモデルである。

これらの流れをふまえて、Airoldi らは混合メンバシップブロックモデル (Mixed Membership Stochastic Blockmodels: MMSB) を提案した [8]。このモデルでは、ノードに対する潜在的なグループの分布と、グループ間のリンク生成確率の2つを考慮している。前者は、ディリクレ事前分布を仮定した多項分布で表現する。後者は、ベルヌーイ分布で表現し、さらに、これにベータ事前分布を導入することもできる。これによってリンクによって各ノードが異なるグループに属することを許容する柔軟なモデリングが可能となる。さらに、観測情報ではないグループ同士のつながりやすさを考慮することで、どのようなネットワークに対してもより正確なモデリングを実現できる。

ネットワーク構造にこのモデルを適用するためには、これらの未知パラメータを推定しなければならない。パラメータの推

定方法としては、変分ベイズ法やギブスサンプリング法などがしばしば用いられる。既存の推定方法は、観測されたネットワーク構造全体を通してグループ対を決定する。これをバッチアルゴリズムと呼び、全リンクに関するグループ対について考えるので処理時間が大きくなる。ところで、ネットワーク構造として表せるものの多くは、時間経過とともに情報が増加していく。例えば、映画のレビューサイトの例で考えると、ユーザ・アイテム（ノード）や評価（リンク）は逐次的に与えられ、未来に追加されるネットワーク情報を待たずに、逐次的な判断を要請されることが少なくない。このような状況で、バッチアルゴリズムを毎度行うことは現実的であるとは言えず、オンライン学習がより適切である。

また、文書コレクションなど多様なデータ分析に適用されている潜在的ディリクレ配分法 (Latent Dirichlet Allocation: LDA) に対するオンライン手法が、最初に Song ら (2005) [9] や Banerjee ら (2007) [10] によって提案され、さらに Canini ら [11] や Hoffman ら [12] によって拡張されている。一方、前述のように、MMSB に対してもオンライン学習が重要であると考えられるが、十分に検討されていない。

本稿では、Canini らの LDA のオンライン学習 [11] を参考にしつつ、MMSB に関して、ネットワーク情報が追加されると各パラメータを逐次更新するようなオンラインアルゴリズムを 3 つ (oMMSB, incremental Gibbs sampler, particle filter) 提案し、その推定時間と推定精度のトレードオフを考える。また、particle filter では通常の手法に加えて、その手法の特徴を考えてデータセットの追加方法を工夫することにより、ノイズ的な情報を軽視し必要な情報により注目することを目指す。

2. 混合メンバシップブロックモデル

混合メンバーシップブロックモデル (MMSB) は、Airoldi らによって提案された。このモデルはノードに潜在的なグループを割り当て、あるグループ対に対してリンクが生成される尤度を推定するモデルである。そして、ノード間のリンクは、それぞれのノードの潜在的なグループの多項分布と、グループ対に関するベルヌーイ分布から生成される。

まずは本稿で用いる定義についてまとめる。グラフを $G = (N, Y)$ と表し、観測データについてノード p, q の隣接行列を $Y(p, q) \in \{0, 1\}$ とする。各ノードはグループに関する多項分布 $\text{Mult}(\pi_p)$ で特徴づけられ、グループ g に関する多項分布パラメータを $\pi_{p,g}$ とすると、 $\pi_{p,g}$ はノード p がグループ g に属する確率である。つまり、ノードはリンクごとに異なるグループをとることができる。グループ間の関係はベルヌーイ分布の行列 $B_{K \times K}$ によって定義される。ここで $B(g, h)$ はグループ g のノードから、グループ h のノードへの辺が存在する確率であり、 K はグループ数を示す。指示ベクトル $\mathbf{z}_{p \rightarrow q}$ はノード p からノード q へリンクについてノード p に割り当てられる潜在グループを表し (該当するグループの成分が 1 であり、他が 0 であるベクトル)、 $\mathbf{z}_{p \leftarrow q}$ はノード q に割り当てられる潜在グループを表す。これら二つのベクトルの集合はそれぞれ $\{\mathbf{z}_{p \rightarrow q} | p, q \in N\} = \mathbf{Z}_{\rightarrow}$ と $\{\mathbf{z}_{p \leftarrow q} | p, q \in N\} = \mathbf{Z}_{\leftarrow}$ である。

以上の定義から、MMSB モデルによってリンクの生成過程は以下ようになる。

(1) すべてのノード p に対して

- $\pi_p \sim \text{Dir}(\alpha)$ 、すなわち、ハイパーパラメータ α_p で特定されたディリクレ分布から多項分布パラメータ π_p をサンプリング

(2) すべてのグループの対 (g, h) に対して

- $B(g, h) \sim \text{Beta}(\psi(g, h))$ 、すなわち、ハイパーパラメータ $\psi(g, h) = (\psi_{1,g,h}, \psi_{2,g,h}) \in \Psi$ で特定されたベータ分布からベルヌーイ分布パラメータ $B(g, h)$ をサンプリング

(3) すべてのノード対 (p, q) に対して

- $\mathbf{z}_{p \rightarrow q} \sim \text{Mult}(\pi_p)$ 、すなわち、多項分布 $\text{Mult}(\pi_p)$ から指示ベクトル $\mathbf{z}_{p \rightarrow q}$ をサンプリング

- $\mathbf{z}_{p \leftarrow q} \sim \text{Mult}(\pi_q)$ 、すなわち、多項分布 $\text{Mult}(\pi_q)$ から指示ベクトル $\mathbf{z}_{p \leftarrow q}$ をサンプリング

- $Y(p, q) \sim \text{Bern}(\mathbf{z}_{p \rightarrow q}^T \mathbf{B} \mathbf{z}_{p \leftarrow q})$ 、すなわち、 $\mathbf{z}_{p \rightarrow q}^T \mathbf{B} \mathbf{z}_{p \leftarrow q}$ によって決まるベルヌーイ分布から $Y(p, q)$ を生成

このとき、データ Y と潜在変数 $\pi_{1:N}, \mathbf{Z}_{\rightarrow}, \mathbf{Z}_{\leftarrow}$ および $\mathbf{B}$ の同時分布 (完全同時分布) は以下ようになる。

$$\begin{aligned} P(Y, \pi_{1:N}, \mathbf{Z}_{\rightarrow}, \mathbf{Z}_{\leftarrow}, \mathbf{B} | \alpha, \Psi) \\ = P(\mathbf{B} | \Psi) \prod_{p, q, p \neq q} P(Y(p, q) | \mathbf{z}_{p \rightarrow q}, \mathbf{z}_{p \leftarrow q}, \mathbf{B}) P(\mathbf{z}_{p \rightarrow q} | \pi_p) \\ P(\mathbf{z}_{p \leftarrow q} | \pi_q) \prod_p P(\pi_p | \alpha) \end{aligned} \quad (1)$$

また、予測分布は以下ようになる。

$$\begin{aligned} P(Y(p, q) = a, \mathbf{z}_{p \rightarrow q} = g, \mathbf{z}_{p \leftarrow q} = h | Y, \mathbf{Z}_{\rightarrow}, \mathbf{Z}_{\leftarrow}, \alpha, \Psi) \\ = \frac{C(p, g) + \alpha_g}{C_p + \sum_g \alpha_g} \frac{C(q, h) + \alpha_h}{C_q + \sum_h \alpha_h} \\ \frac{C(a, g, h) + \psi_{a,g,h}}{C(0, g, h) + C(1, g, h) + \psi_{0,g,h} + \psi_{1,g,h}} \\ = \pi_{p,g} \pi_{q,h} B(g, h) \end{aligned} \quad (2)$$

ここで、 $C(p, g)$ はあるノード p がグループ g に割り当てられた回数であり、 C_p はその合計、つまり $C_p = \sum_g C(p, g)$ である。また、 $\psi_{1,g,h}$ は接続リンクのグループ間の事前分布を表し、 $\psi_{0,g,h}$ は未接続リンクのグループ間の事前分布を表す。 $C(0, g, h), C(1, g, h)$ は、観測されたネットワーク全体におけるノード対において、一方のグループが g であり他方のグループが h であるときにリンクが存在する頻度と存在しない頻度をそれぞれ示す。 $\mathbf{z}_{p \rightarrow q}, \mathbf{z}_{p \leftarrow q}$ はそれぞれリンク (p, q) の始点と終点に割り当たるグループを表す変数であるとする。 $a \in \{1, 0\}$ とし、リンクの有無を表す。

MMSB のグラフィカルモデルは図 1 で示される。

3. アルゴリズム

本章では、MMSB でのパラメータの推定方法としてギブスサンプリングを用いる。まず、観測されたネットワーク構造全体を通して学習・推定するバッチギブスサンプリング (既存手法) について、次にネットワーク情報が追加されると各パラメータを逐次更新するような 3 つのオンラインアルゴリズム (提案手

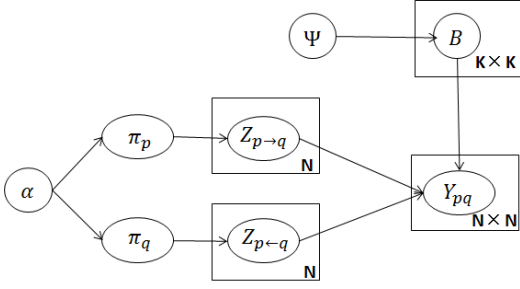


図 1 MMSB のグラフィカルモデル

法)について述べていく。

3.1 BATCH GIBBS SAMPLER

バッチギブスサンプラーでは、前で述べたように、観測されたネットワーク構造全体を通してパラメータを学習・推定する。ここで、グループ対は以下の条件付き確率に基づいてサンプリングされる。注目するノード p, q について、変更後のグループ割り当てをそれぞれ g', h' とする。

$$\begin{aligned}
 P(z_{p \rightarrow q} = g, z_{p \leftarrow q} = h | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{-(p \rightarrow q)}, \mathbf{Z}_{\leftarrow}^{-(p \leftarrow q)}, \alpha, \Psi) \\
 = (C(p, g) - 1 + \Delta(g' \neq g) + \alpha_g)(C(q, h) - 1 + \Delta(h' \neq h) + \alpha_h) \\
 \frac{C(a, g, h) - 1 + \Delta(g' \neq g \wedge h' \neq h) + \psi_{a, g, h}}{C(0, g, h) + C(1, g, h) - 1 + \Delta(g' \neq g \wedge h' \neq h) + \psi_{0, g, h} + \psi_{1, g, h}}
 \end{aligned}
 \quad (3)$$

Algorithm batch Gibbs sampler for $N \times N$

- 1: initialize group assignment randomly for $N \times N$
 - 2: for iter=1 to L_{iter} do
 - 3: for p=1 to N do
 - 4: for q=1 to N do
 - 5: draw $z_{p \rightarrow q}$ from $P(z_{p \rightarrow q} | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{-(p \rightarrow q)}, \mathbf{Z}_{\leftarrow}^{-(p \leftarrow q)}, \alpha, \Psi)$
 - 6: draw $z_{p \leftarrow q}$ from $P(z_{p \leftarrow q} | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{-(p \rightarrow q)}, \mathbf{Z}_{\leftarrow}^{-(p \leftarrow q)}, \alpha, \Psi)$
 - 7: end for
 - 8: end for
 - 9: end for
 - 10: complete the posterior estimates of π and $\mathbf{B}$
-

図 2 ノード数 N のネットワークに対するバッチギブスサンプラーの擬似コード

つまり、注目するグループ対を除いたネットワークによって新しいグループ対を決定する。図 2 に略述されているこの手法は、求めたい事後分布に収束する。このバッチギブスサンプラーは様々な方法で拡張でき、効率的な MMSB のオンライン手法のアルゴリズムにつなげることができる。

3.2 oMMSB

LDA に関して、バッチギブスサンプラーに単純な変更を加えて生み出されたのが、Song らによって提案されたもの (2005) [9] と、Banerjee and Basu(2007) [10] による”oLDA”と呼ばれているものである。ここでは、oLDA を MMSB に適用したものを”oMMSB”と呼ぶこととする。図 3 に略述されているこの手法は、データの最初の部分についてはバッチギブスサンプリングし、その後新たなリンクの有無に関する事象が観測されると、

既に観測されているネットワークの条件付き確率に基づいてそのグループ対を決定する。もし、それが新規ノードを含むリンクであるならば (図 3 の 6 行目, 11 行目), 既に決定したグループ対以外のグループ対を新たに決定する。

Algorithm oMMSB

- 1: batch Gibbs sampler for $N_{firstPart} \times N_{firstPart}$
 - 2: while(add link $p' \rightarrow q'$) do
 - 3: draw $z_{p' \rightarrow q'}$ from $P(z_{p' \rightarrow q'} | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{-(p' \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{-(p' \leftarrow q')}, \alpha, \Psi)$
 - 4: draw $z_{p' \leftarrow q'}$ from $P(z_{p' \leftarrow q'} | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{-(p' \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{-(p' \leftarrow q')}, \alpha, \Psi)$
 - 5: if(p' is new node) then
 - 6: for $q = 1$ to $N_{currentNode}$ (if $q \neq q'$) do
 - 7: draw $z_{p' \rightarrow q}$ from $P(z_{p' \rightarrow q} | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{-(p' \rightarrow q)}, \mathbf{Z}_{\leftarrow}^{-(p' \leftarrow q)}, \alpha, \Psi)$
 - 8: draw $z_{p' \leftarrow q}$ from $P(z_{p' \leftarrow q} | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{-(p' \rightarrow q)}, \mathbf{Z}_{\leftarrow}^{-(p' \leftarrow q)}, \alpha, \Psi)$
 - 9: end for
 - 10: end if
 - 11: if(q' is new node) then
 - 12: for $p = 1$ to $N_{currentNode}$ (if $p \neq p'$) do
 - 13: draw $z_{p \rightarrow q'}$ from $P(z_{p \rightarrow q'} | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{-(p \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{-(p \leftarrow q')}, \alpha, \Psi)$
 - 14: draw $z_{p \leftarrow q'}$ from $P(z_{p \leftarrow q'} | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{-(p \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{-(p \leftarrow q')}, \alpha, \Psi)$
 - 15: end for
 - 16: end if
 - 17: end while
 - 18: the posterior estimates of π and $\mathbf{B}$
-

図 3 oMMSB の擬似コード

oMMSB は、バッチギブスサンプリング後については、各新規リンクについて逐次的に式 (3) を適用し、一度決定されたグループ対は二度と変更されない。このため、その性能はバッチギブスサンプリングの精度に大きく依存し、ここで用いられたデータがネットワーク全体をうまく表現したものでなければ、精度の悪い推定につながってしまう。こうした問題に対応するため、一度決定したグループ割り当てを変更することのできるオンライン手法を考える。

3.3 INCREMENTAL GIBBS SAMPLER

ここでは、incremental Gibbs sampler という手法を紹介する。図 4 に略述されているこの手法は、新たなリンクの有無に関する事象が観測されると、oMMSB と同様に既に観測されているネットワークの条件付き確率に基づいてそのグループ対を決定するのに加えて、 $|R(i)|$ 対のグループ対について過去に決定したグループ割当を変更する。 $|R(i)|$ が大きいほど事後分布は理想値に近づくが、 $|R(i)|$ が線形的に増加すると実行時間は二次関数的に増加する。

3.4 PARTICLE FILTER

3.4.1 particle filter

ここでは、particle filter という手法を紹介する。particle filter は逐次モンテカルロ法 [13] であり、様々な場面で応用されている。図 5 に略述されているこの手法は、ネットワーク構造 (パーティクル) を複数同時に維持する手法である。各パーティクルは異なるグループ割り当てを持ち、各パーティクルで先述の oMMSB と似たような操作を行う。

加えて、各パーティクルで重みを考え、重みが大きいほどネットワーク全体をうまく表現した精度の高いパーティクルを表す

Algorithm incremental Gibbs sampler
1: while(add link $p' \rightarrow q'$) do
2: draw $z_{p' \rightarrow q'}$ from $P(\mathbf{z}_{p' \rightarrow q'} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p' \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{\neg(p' \leftarrow q')}, \alpha, \Psi)$
3: draw $z_{p' \leftarrow q'}$ from $P(\mathbf{z}_{p' \leftarrow q'} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p' \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{\neg(p' \leftarrow q')}, \alpha, \Psi)$
4: if(p' is new node) then
5: for $q = 1$ to $N_{currentNode}$ (if $q \neq q'$) do
6: draw $z_{p' \rightarrow q}$ from $P(\mathbf{z}_{p' \rightarrow q} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p' \rightarrow q)}, \mathbf{Z}_{\leftarrow}^{\neg(p' \leftarrow q)}, \alpha, \Psi)$
7: draw $z_{p' \leftarrow q}$ from $P(\mathbf{z}_{p' \leftarrow q} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p' \rightarrow q)}, \mathbf{Z}_{\leftarrow}^{\neg(p' \leftarrow q)}, \alpha, \Psi)$
8: end for
9: end if
10: if(q' is new node) then
11: for $p = 1$ to $N_{currentNode}$ (if $p \neq p'$) do
12: draw $z_{p \rightarrow q'}$ from $P(\mathbf{z}_{p \rightarrow q'} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{\neg(p \leftarrow q')}, \alpha, \Psi)$
13: draw $z_{p \leftarrow q'}$ from $P(\mathbf{z}_{p \leftarrow q'} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{\neg(p \leftarrow q')}, \alpha, \Psi)$
14: end for
15: end if
16: for($p'' \rightarrow q''$ in $\mathcal{R}(p' \rightarrow q')$)
17: draw $z_{p'' \rightarrow q''}$ from $P(\mathbf{z}_{p'' \rightarrow q''} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p'' \rightarrow q'')}, \mathbf{Z}_{\leftarrow}^{\neg(p'' \leftarrow q'')}, \alpha, \Psi)$
18: draw $z_{p'' \leftarrow q''}$ from $P(\mathbf{z}_{p'' \leftarrow q''} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p'' \rightarrow q'')}, \mathbf{Z}_{\leftarrow}^{\neg(p'' \leftarrow q'')}, \alpha, \Psi)$
19: end for
20: end while
21: the posterior estimates of π and $\mathbf{B}$

図 4 incremental Gibbs sampler の擬似コード

と考える．ここで，重みの偏りが大きくなった時，パーティクルが分裂消滅してリサンプリングにより重みが大きい文書に注目する新しいパーティクル集合を生成する．そして，各パーティクルの重みを P^{-1} にリセットし，さらに各パーティクルで incremental Gibbs sampler と同様の操作を行う．

リサンプリングの簡単な手法としては，重みによって定義された多項分布から決定することが考えられる．また，Canini らの LDA での particle filter [11] では，リサンプリング回数をコントロールするパラメータとして ESS 閾値を用いていたが，今回は重みの分散が大きくなりやすいという問題の性質に対処するために， u リンク追加ごとに重みを更新するようにしてリサンプリング回数を調節する．ここで，重みの更新の際には u リンクの尤度平均を用いる．

particle filter の予測分布 $P_{particle}$ は

$$P_{particle} = \sum_k (P^{(k)} \times \omega^{(k)}) \quad (4)$$

である．ここで， $P^{(k)}$ はパーティクル k における予測分布を表しており，式 (2) で求めることができる． $\omega^{(k)}$ はパーティクル k における予測分布を表している．

particle filter ではパーティクル数が大きいほど精度は高くなるが，より時間がかかってしまう．ここで，計算時間のほぼ全てをパーティクルのリサンプリング（図 5 の 22 行目）で占めていることが分かっている [11]．これは，各パーティクルでネットワークのコピー作っているからである．ここで，図 5 についてパーティクル毎に計算している部分（3～19 行目，23～28 行目）に注目すると，この部分は OpenMP などを用いて並列計算することが出来る．このことにより，実行時間を抑える

Algorithm particle filter
1: initialize weights $\omega^{(k)} = P^{-1}$ for $k = 1, \dots, P$
2: while(add link $p' \rightarrow q'$)
3: linkNum ++
4: for $k = 1$ to P do
5: $sum^{(k)} += P^{(k)}(Y(p', q') = 1 \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p' \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{\neg(p' \leftarrow q')}, \alpha, \Psi)$
6: if(linkNum % $u == 0$)
7: $\omega^{(k)} *= sum^{(k)} / u$
8: $sum^{(k)} = 0$
9: end if
10: draw $z_{p' \rightarrow q'}$ from $P^{(k)}(\mathbf{z}_{p' \rightarrow q'} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p' \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{\neg(p' \leftarrow q')}, \alpha, \Psi)$
11: draw $z_{p' \leftarrow q'}$ from $P^{(k)}(\mathbf{z}_{p' \leftarrow q'} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p' \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{\neg(p' \leftarrow q')}, \alpha, \Psi)$
12: if(p' is new node)
13: for $q = 1$ to $N_{currentNode}$ (if $q \neq q'$)
14: draw $z_{p' \rightarrow q}$ from $P^{(k)}(\mathbf{z}_{p' \rightarrow q} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p' \rightarrow q)}, \mathbf{Z}_{\leftarrow}^{\neg(p' \leftarrow q)}, \alpha, \Psi)$
15: draw $z_{p' \leftarrow q}$ from $P^{(k)}(\mathbf{z}_{p' \leftarrow q} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p' \rightarrow q)}, \mathbf{Z}_{\leftarrow}^{\neg(p' \leftarrow q)}, \alpha, \Psi)$
16: end for
17: end if
18: if(q' is new node)
19: for $p = 1$ to $N_{currentNode}$ (if $p \neq p'$) do
20: draw $z_{p \rightarrow q'}$ from $P^{(k)}(\mathbf{z}_{p \rightarrow q'} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{\neg(p \leftarrow q')}, \alpha, \Psi)$
21: draw $z_{p \leftarrow q'}$ from $P^{(k)}(\mathbf{z}_{p \leftarrow q'} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{\neg(p \leftarrow q')}, \alpha, \Psi)$
22: end for
23: end if
24: end for
25: normalize weights ω to sum to 1
26: if $\ \omega\ ^{-2} \leq \text{ESS threshold}$ then
27: resample particles
28: for $k = 1$ to P do
29: for($p'' \rightarrow q''$ in $\mathcal{R}(p' \rightarrow q')$) do
30: draw $z_{p'' \rightarrow q''}$ from
31: $P^{(k)}(\mathbf{z}_{p'' \rightarrow q''} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p'' \rightarrow q'')}, \mathbf{Z}_{\leftarrow}^{\neg(p'' \leftarrow q'')}, \alpha, \Psi)$
32: draw $z_{p'' \leftarrow q''}$ from
33: $P^{(k)}(\mathbf{z}_{p'' \leftarrow q''} \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{\neg(p'' \rightarrow q'')}, \mathbf{Z}_{\leftarrow}^{\neg(p'' \leftarrow q'')}, \alpha, \Psi)$
34: end for
35: end for
36: $\omega^{(k)} = P^{-1}$ for $k = 1, \dots, P$
37: end if
38: end while
39: the posterior estimates of π and $\mathbf{B}$

図 5 particle filter の擬似コード

ことが出来る．

3.4.2 ノイズ除去 particle filter

パーティクル毎に異なるネットワーク構造データを与えることができるという particle filter の特徴を用いて，より精度の高いモデル推定をすることを考える．図 6 のアルゴリズムは，図 5 で示される通常の particle filter を少し変更したものである．

4 行目が図 5 から追加された部分である．リンク追加ごとにパーティクルに追加するかどうかを考えている．各パーティクルで 0～1 までの乱数を考え，それが各パーティクルで定められた閾値よりも大きければそのパーティクルにリンクを追加している．つまり，リンク追加確率が各パーティクルごとに異なるので，パーティクルごとに接続しているリンクを変えることができる．3 行目において， $k = 1$ のパーティクルでは全リンクが追加され， k が大きいパーティクルほど保持しているネッ

Algorithm particle filter

```

1: initialize weights  $\omega^{(k)} = P^{-1}$  for  $k = 1, \dots, P$ 
2: while(add link  $p' \rightarrow q'$ )
3:    $linkNum++$ 
4:   for  $k = 1$  to  $P$  do
5:      $sum^{(k)} += P^{(k)}(Y(p', q') = 1 | \mathbf{Y}, \mathbf{Z}_{\rightarrow}^{(p' \rightarrow q')}, \mathbf{Z}_{\leftarrow}^{(p' \leftarrow q')}, \alpha, \Psi)$ 
6:     if( $linkNum \% u == 0$ )
7:        $\omega^{(k)} *= sum^{(k)} / u$ 
8:        $sum^{(k)} = 0$ 
9:     end if
10:    if( $random.uniform(0, 1.0) \geq (k-1)/(P-1)$ )
11:      以降, 図 5 と同じ
12:      draw  $z_{p' \rightarrow q'}^{(k)}$  from ...

```

図 6 ノイズ除去 particle filter の擬似コード

トワーク構造は疎である．対して， $k = P$ のパーティクルではおよそ $\frac{1}{P}$ のリンクしか追加されない．

もしリンクが追加されにくい(ネットワークが疎である)パーティクルの重みが大きくなると，その期間の追加リンクが予測に不適(ノイズ)であるとし，余分とみなしたリンクの追加を省いて通常の particle filter とは異なる結果となる．

3.4.3 ネットワークの疎密を考慮した尤度

3.4.2 では，パーティクルごとにリンクの追加頻度を変えることにより，ネットワークを oMMSB, incremental Gibbs, 通常の particle filter と比べて疎になる場合を考えた．またパーティクルごとのネットワークの疎密も異なる．

ここで，式 (2) について考える． $\pi_{p,g}$ は，ノード p がグループ g をもつ割合を示しており，これはネットワークの疎密によって左右されることはない． $\pi_{q,h}$ も同様である．

$$B(g, h) = \frac{C(a, g, h) + \psi_{a,g,h}}{C(0, g, h) + C(1, g, h) + \psi_{0,g,h} + \psi_{1,g,h}} \quad (5)$$

$B(g, h)$ は，グループ g, h が接続される割合を示している．これはネットワークが密なほど期待値は高くなってしまいうので，つまり追加リンク数が多いほど尤度が大きくなってしまい，3.4.2 のような接続リンクが異なるパーティクル間の比較ができない．また全リンクを追加している oMMSB, incremental Gibbs と尤度を比較できない．

そのため，重みの更新(図 5, 4 行目)や particle filter の予測分布(式 (4))において，予測分布 $P^{(k)}$ を以下のように補正する．

$$P^{(k)} = \pi_{p,g} \pi_{q,h} B(g, h) \times (\gamma_{all} / \gamma_k) \quad (6)$$

ここで， γ は以下の式で示される．

$$\gamma = \frac{\sum_{g,h} (C(1, g, h) + \psi_{1,g,h})}{\sum_{g,h} (C(0, g, h) + C(1, g, h) + \psi_{0,g,h} + \psi_{1,g,h})} \quad (7)$$

γ_k はパーティクル k について， γ_{all} は全リンクが追加された場合についてである．

この補正により，ネットワークの疎密に関わらずそれまでのすべてのデータが追加されたという条件下で，パーティクル間や oMMSB, incremental Gibbs などの手法とも比較できる．

4. 実験

ネットワーク構造を捉えるオンライン学習において，本稿では前述のように主に実行時間と性能のトレードオフを考える．情報が与えられた時に，正しいグループ対をできるだけ短い時間で決定し，高精度の推定を行うことが目的である．そこで，本実験では時系列のデータを用いて各手法を比較した．

4.1 実験設定

4.1.1 データセット

データセットとして，映画のレビューサイトである MovieLens^(注1)のデータを利用した．データはユーザ数 943, アイテム数 1682 の，10 万件の格付けデータである．つまり，ノード数合計 2625, リンク数 10 万のネットワークとなっている．また，各ユーザは少なくとも 20 のアイテムを格付けしている．本稿ではユーザの格付けの値は考慮せず，格付けしたか，していないかの二値でモデルの推定を行った．対象期間は 1997/9/19 ~ 1998/4/22 までの約 7 か月である．本研究では秒単位でタイムスタンプが与えられたこのデータセットをタイムスタンプに基づいて 14 分割した．つまり，約 15 日ごとのデータを 14 セット用意した．各期間のリンク数は下記のとおりである．

表 1 データセットの詳細

	対象期間	リンク数
セット 1	1997/9/20/03:05:10 ~	9245
セット 2	1997/10/5/14:24:04 ~	4704
セット 3	1997/10/20/17:58:08 ~	3627
セット 4	1997/11/5/01:51:45 ~	15882
セット 5	1997/11/20/14:19:48 ~	9462
セット 6	1997/12/5/21:03:22 ~	6485
セット 7	1997/12/21/02:55:22 ~	5848
セット 8	1998/1/5/13:35:37 ~	6889
セット 9	1998/1/20/20:34:45 ~	7100
セット 10	1998/2/5/05:54:28 ~	4450
セット 11	1998/2/20/12:37:14 ~	6560
セット 12	1998/3/6/18:26:18 ~	2888
セット 13	1998/3/23/04:29:56 ~	12599
セット 14	1998/4/7/12:51:19 ~ 1998/4/22/23:10:38	4261

4.1.2 実験環境

全実験は，48GB のメモリ，12 コア 24 スレッドでクロック周波数 3.65GHz の CPU (Xeon X5675) 上で行った．また，プログラミング言語は C++ で実装し，並列化の際に OpenMP を用いた．並列数は全て 10 としている．

4.1.3 評価手法

本実験では，評価手法としてテストセット対数尤度を用いる．これは訓練セットを用いて推定されたモデルを用いて，テストセットが予測できているのかを対数尤度で求める手法である．対数尤度はパーレキシティの負の対数に等しく，この値が大きいほど推定モデルの精度が高いといえる．これは，トピックモデルの分野で広く用いられる評価手法である．

(注1): <http://grouplens.org/>

具体的には、テストセットの尤度は以下の式で求められる。

$$p(\mathbf{w}^{test}) = \prod_{p,q} \sum_{g,h} \pi_{p,g} \pi_{q,h} B(g,h) \quad (8)$$

対数尤度は、上式の対数を取ることで求められる。また、 $\pi_{p,g}, \pi_{q,h}, B(g,h)$ は、式 (2) で求めたのと同様に、以下の式で求めることができる。

$$\pi_{p,g} = \frac{C(p,g) + \alpha_g}{C_p + \sum_g \alpha_g}, \quad \pi_{q,h} = \frac{C(q,h) + \alpha_h}{C_q + \sum_h \alpha_h}$$

$$B(g,h) = \frac{C(a,g,h) + \psi_{a,g,h}}{C(0,g,h) + C(1,g,h) + \psi_{0,g,h} + \psi_{1,g,h}}$$

4.1.4 モデル推定手法

本実験では、以下の手法を比較した。

- batch Gibbs sampler

リンク追加時はグループをランダムに割り当てていき、データセットが 1 セット追加される毎にネットワーク全体でのバッチギブスサンプリングを行う。オンライン学習での理想値となる。

- batch to random

oMMSB 以下の手法と条件をそろえるため、初期化段階でのバッチギブスは行い、その後リンク追加時にはグループをランダムに割り当てていく。

- oMMSB
- incremental Gibbs sampler

グループ割り当ての変更回数 $|R(i)| = 1000, 10000$ の 2 通りで実験を行った。

- particle filter

particle filter での尤度を求める際の予測分布については式 (7) を用いている。また、ESS 閾値は $1 + e^{-30}$ 、パーティクル数は 10 に固定している。重みの更新頻度 $u = 1, 50$ の 2 通り、 $|R(i)| = 0, 100000$ の 2 通り、通常のパーティクルフィルタとノイズ除去パーティクルフィルタの 2 通りをそれぞれ実験している。つまり、8 通りの実験を行った。

また、全手法において、グループ数は 5、ハイパーパラメータは α の全成分を 1、 $\psi_{0,g,h} = \psi_{1,g,h} = 1$ とし、バッチ段階の繰り返し回数 (図 2, 2 行目の L_{iter}) は 100 とする。

4.1.5 実行時間の測定

本実験では、主に推定精度と推定時間のトレードオフを考えるとこととなる。そこで、以降では推定時間を、“データの最初の部分の初期化が終わった段階から最後のデータセットの推定結果が計算されるまで”と定義する。本稿ではパラメータの逐次更新するときの所要時間に注目しているためである。

4.1.6 訓練セットとテストセット

本稿では、2 種類の実験を行った。

- 全区間の尤度計算

テストセットとして 7 か月分の全リンクを用いる。訓練セットは 4.1.1 で述べた 7 か月 (14 期間) のデータであり、1 期間が追加されるたびに 7 か月分のリンクをうまく推定できているかを評価した。

- 次区間の予測実験

テストセットとして最後にパーティクルに追加されたデータ

セットの次の 1 期間を用いる。つまり、訓練セットは 4.1.1 で述べた 7 か月 (14 期間) のデータであり、1 期間が追加されるたびに次の 1 期間のリンクをうまく推定できているかを評価した。

4.2 全区間の尤度計算

4.2.1 結果

各手法における推定時間と尤度を以下に示す。表、グラフ中において、b to r は batch to random, inc は incremental Gibbs Sampler を表す。また、pF1 は通常の particle filter を表し、pF2 はノイズ除去 particle filter を表すとする。また、パラメータを括弧内に ($|R(i)|, u(3.4.1$ で定義)) と表す。例えば、pF2(100000, 10) は、グループ割り当ての変更回数 $|R(i)|=100000$ 、重みの更新頻度 $u=10$ のノイズ除去 particle filter である。グラフについては、縦軸がテストセット対数尤度、横軸が予測する月数である。

表 2 全区間の尤度計算におけるモデル推定時間

手法	時間 (s)	手法	時間 (s)
oMMSB	4	batch	3300
pF1(0,1)	68	pF2(0,1)	76
pF1(0,50)	20	pF2(0,50)	30
pF1(100000,1)	426	pF2(100000,1)	454
pF1(100000,50)	30	pF2(100000,50)	47
inc($ R(i) = 1000$)	135	inc($ R(i) = 10000$)	1275
b to r	2		

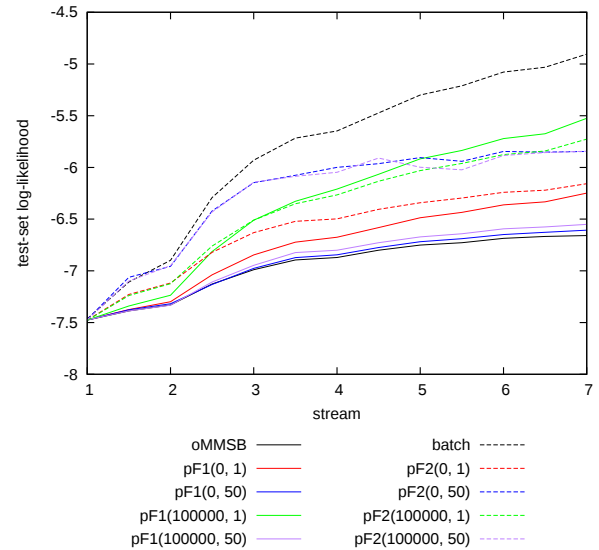


図 7 全区間の尤度計算結果 (パーティクルフィルタ)

4.2.2 考察

4.1.6 で述べたように、この実験では、テストセットは 7 か月分の全リンクである。どの程度のリンクを追加すれば、どの程度ネットワーク全体を把握できるのかということ考えている。また、訓練データは常にテストデータの要素であるので、重みの更新やグループ決定が理想的なものになるであろうということも考慮すべき点である。

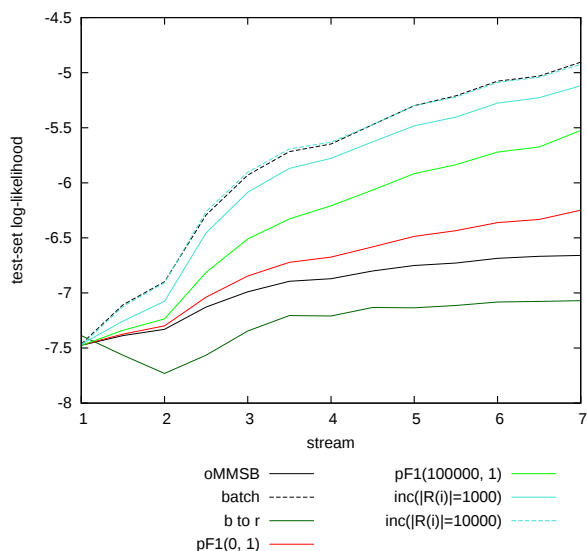


図 8 全区間の尤度計算結果 (まとめ)

まず、図 7 についてであるが、 $pF1(0, 1)$ (赤の実線) が $oMMSB$ を上回っている。 $|R(i)| = 0$ なので、この手法間の差はネットワーク構造データを複数与えられているか 1 つ与えられているかということと、particle filter がリサンプリング (分裂・消滅) を行うということのみである。もちろん、表からも分かるように、推定時間は比較できるものではないが、ネットワーク構造データを複数与えられ、リサンプリングによる分裂・消滅が有効であったと考えられる。

次に、 u の効果、 $|R(i)|$ の効果、ノイズ除去 ($pF2$) の効果について考える。最初に u についてであるが、 $pF1(0, 1)$ (赤の実線) と $pF1(0, 50)$ (青の実線) を比較する。この 2 つは、通常のパーティクルフィルタでグループ対の選り抜きを行わないということは共通していて、 u の大きさ、つまりリサンプリングの回数のみ異なる。リサンプリング回数が多い $u = 1$ が $u = 50$ に比べて高い精度を取っていることが分かる。これは、重みの更新が理想的になるであろうこの実験での特徴といえる。また、 $|R(i)|$ は大きいほど精度が高く、ノイズ除去の効果はパラメータ $u, |R(i)|$ によって異なることが分かった。総じて言えるのは、パーティクルフィルタではどれについても batch と同等の結果を得ることができなかったということである。

ここで、パーティクルフィルタで一番精度が良かったもの ($pF1(100000, 1)$) と一番シンプルなもの ($pF1(0, 1)$) を選抜き、他のオンライン手法と比較した。図 8 を見てみると、batch Gibbs の精度が一番高いが、incremental Gibbs ($|R(i)| = 10000$) も同程度の推定ができていいることである。表 2 より、38 % の推定時間で既存手法の batch Gibbs と同等の推定ができていいる ($3300(s) \rightarrow 1274(s)$)。さらに推定回数が増え、推定のたびにグループ対を決定しないう batch Gibbs は大幅に推定時間が大きくなると考えられるが、逐次的にパラメータを更新するオンライン学習ではそのような心配はない。この点で、このデータでは提案手法の incremental Gibbs が効果的であるといえる。

4.3 次区間の予測実験

4.3.1 結果

各手法における実行時間と尤度を以下に示す。グラフについては、縦軸がテストセット対数尤度、横軸が予測する月数である。

表 3 次区間の予測実験におけるモデル推定時間

手法	時間 (s)	手法	時間 (s)
oMMSB	3	batch	3302
pF1(0,1)	55	pF2(0,1)	64
pF1(0,50)	17	pF2(0,50)	27
pF1(100000,1)	411	pF2(100000,1)	462
pF1(100000,50)	20	pF2(100000,50)	43
inc($ R(i) = 1000$)	136	inc($ R(i) = 10000$)	1291
b to r	1		

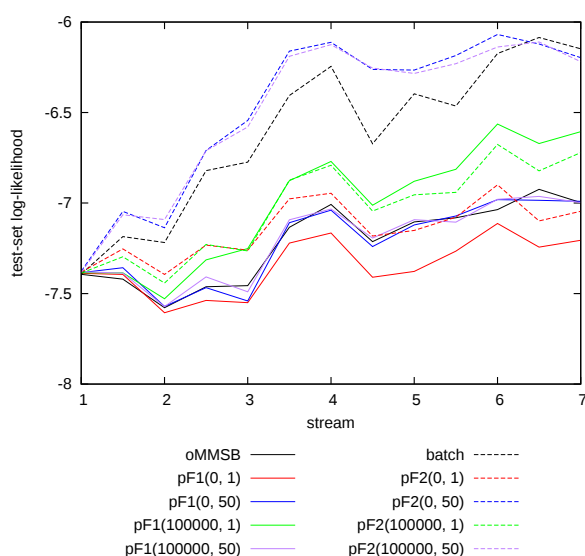


図 9 次区間の尤度計算結果 (パーティクルフィルタ)

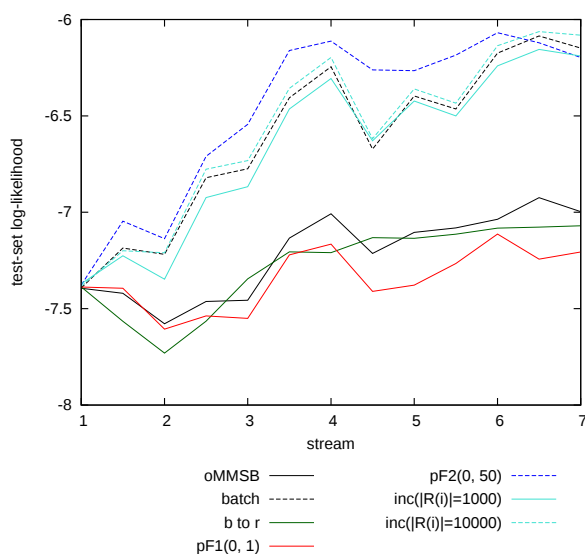


図 10 次区間の尤度計算結果 (まとめ)

4.3.2 考 察

4.1.6 で述べたように、この実験では、テストセットとして最後にパーティクルに追加されたデータセットの次の1期間を用いるので、より現実問題に近い設定であるといえる。しかし、前実験と違い、訓練データとテストデータに重複した要素がないということは考慮しなければならない。

最初に、図9や図10から、データの追加により尤度が必ずしも高くなるわけではないということが、前実験と比較すると明らかである。テストセットに対して有用ではないリンクが存在するということが、原因の一つとして考えられる。

まず、図9についてであるが、 $pF1(0, 1)$ (赤の実線)が $oMMSB$ を下回っている。これは、適切でない重みの際にリサンプリング(分裂・消滅)が起き、以降に必要なであろうグループ割り当てを持つパーティクルが消滅してしまったためであると考えられる。

ここで、重みが大きく偏ることを遅らせることによりリサンプリング回数を少なくしたのが $pF1(0, 50)$ (青の実線)である。これを見てみると、 $pF1(0, 1)$ に比べて精度が良くなっていることが分かる。さらに、この u による工夫に加えて、ノイズ除去を考えたのが $pF2(0, 50)$ (青の破線)であるが、精度が大きく改善していて、全リンクが与えられてさらにグループ対を推定・変更しているbatchと比較しても高い精度であることが分かる。これはノイズの除去が適切に行われていると考えられる。また、 $pF2(0, 50)$ では $|R(i)| = 0$ なのでグループの選び直しを行っていない。このことにより、素早い予測が可能である。

ここで、パーティクルフィルタで一番精度が良かったもの($pF2(0, 50)$)と一番シンプルなもの($pF1(0, 1)$)を選抜し、他のオンライン手法と比較した。図10を見てみると、前実験同様 incremental Gibbs($|R(i)| = 10000$)により batch Gibbs と同等の精度が得られているが、前実験と異なり、 $pF2(0, 50)$ がさらに高い値を取っている。表3より、0.8%の推定時間で既存手法の batch Gibbs よりも精度の高い推定ができていた($3302(s) \rightarrow 27(s)$)。

5. おわりに

本稿では、MMSBのモデル学習に関して、有用であるにもかかわらずこれまで十分に議論されていなかったオンライン学習に関して3つのアルゴリズム($oMMSB$, incremental Gibbs sampler, particle filter)について検討し、精度を維持した速度の向上を図った。また、particle filterに関して、その手法の特徴から、不要な情報を除去することも狙った。

Movie Lensの時系列を考慮したネットワークデータを用いて、各手法に関してテストデータを理想データと現実データでそれぞれ実験し、既存手法との比較や、提案手法の中での比較をテストセット対数尤度を用いて比較した。結果、テストデータに関わらず、提案手法によって既存手法の精度を維持しながらモデル推定時間を抑えられることを確認できた。さらに、次に起こることを予測する現実問題に近い実験では、particle filterの異なるネットワーク構造を複数保持できるという特徴をうまく利用して、既存手法よりも高い精度の予測をすること

ができた。

今後の課題として、particle filter 特徴を利用して、さらなる精度の向上を目指すことが挙げられる。例えば、本実験では逐次的に追加情報を棄却していたが、ある時点において遠い過去のデータを破棄するようにすれば、近い将来をより正確に予測できるかもしれない。この点の検討は今後の課題である。

謝 辞

本研究の一部は、科学研究費補助金基盤研究(B)(20300038, 23300039)による。

文 献

- [1] Newman, M. E. J.: Mixture models and exploratory analysis in networks, *proceedings of the National Academy of Science of the United States of America*, Vol. 104, pp. 9564-9569, 2007.
- [2] Zhang, H., Qiu, B., Giles, C. L., Foley, H. C. and Yen, J.: An LDA-based community structure discovery approach for large-scale social networks, *ISI*, pp.200.207, 2007.
- [3] Blei, D. M., Ng, A. Y. and Jordan, M. I.: Latent Dirichlet allocation, *Journal of Machine Learning Research*, Vol. 3, pp. 993.1022, 2003.
- [4] Steyvers, M. and Griffiths, T.: *Handbook of Latent Semantic Analysis*, chapter 21: Probabilistic Topic Models, Lawrence Erlbaum Associates, Mahwah, New Jersey and London, 2007.
- [5] Griffiths, T. L. and Steyvers, M.: Finding Scientific Topics, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 101, pp. 5228-5235, 2004.
- [6] Nowicki, K. and Snijders, T. A. B.: Estimation and prediction for stochastic blockstructures, *Journal of the American Statistical Association*, Vol. 96, 2001.
- [7] Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T. and Ueda, N.: Learning systems of concepts with an infinite relational model, in *Proceedings of the 21st AAAI conference on Artificial intelligence*, pp. 381-388, 2006.
- [8] Airoldi E. M., Blei D. M., Fienberg S. E., and Xing E. P.: Mixed membership stochastic block models. *The Journal of Machine Learning Research*, Vol.9, pp.1981-2014, 2008.
- [9] Song, X., Lin, C.-Y., Tseng, B. L., and Sun, M.-T.: Modeling and predicting personal information dissemination behavior. In *Proc. 11th ACM SIGKDD Int'l. Conf. on Knowledge Discovery and Data Mining*, 2005.
- [10] Banerjee, A. and Basu, S.: 2007. Topic models over text streams: a study of batch and online unsupervised learning. In *Proc. 7th SIAM Int'l. Conf. on Data Mining*.
- [11] Canini Kevin R., Shi Lei, and Griffiths, Thomas L.: Online Inference of Topics with Latent Dirichlet Allocation. Appearing in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)* 2009.
- [12] Hoffman M., Blei D., Bach F.: Online learning for latent Dirichlet allocation *Neural Information Processing Systems*, 2010.
- [13] Doucet, A., de Freitas, N., and Gordon, N., eds, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.