3L3-2

Repair Topic Model : 大規模系列データのための系列トピックモデル

Repair Topic Model : A new topic model for huge sequential data

石畠 正和　　岩田 具治
Masakazu Ishihata　Tomoharu Iwata

NTT コミュニケーション科学基礎研究所
NTT Comunication Science Laboratories

Topic modeling is a fandamental technique to analize symblic data using latent variables called topics. Latent Dirichlet allocation (LDA) is one of the most famous topic models and assumes that a given data set is a set of symbols. Because of this assumption, LDA cannot exploit sequential information such as the order of words. A hidden Markov model (HMM) is a famous probabilistic model for sequential data and several topic models based on HMM have been proposed. However, it is known that HMM-based topic models are too slow to apply huge sequential data. In this paper, we propose the repair topic model (RTM) which is a new topic model for modeling huge sequential data. In RTM, sequences are compressed by the Repair algorithm which is a grammar based compression algorithm. Thanks to employing the Repair, RTM exploits sequential information as a compressed grammar and is much faster than HMM-based topic models because RTM works on the compressed grammar.

1. はじめに

トピックモデリングは文書データなどの記号データを、トピックと呼ばれる潜在変数を用いた混合分布で表現することで、似た使われ方をする単語やパターンを抽出する手法として広く利用されている。Latent Dirichlet Allocation (LDA) [2] は最も有名なトピックモデルの一つである。LDA では文書データを Bag of Words (BoW) と呼ばれる記号の集合として表現し、文書内での単語の共起関係を考慮することで単語をいくつかのトピックに分類する。トピック数を K、 各文書毎の語彙数の総和を M とすれば LDA の計算量は $O(MK)$ と高速だが、BoW を仮定することで文書中の系列情報は失われてしまう。系列情報が失われると、単語の連続しやすさなどの局所的な共起パターンを抽出することができない。

系列データを扱う確率モデルとして Hidden Markov Model (HMM) が知られている [10]。HMM では観測変数は現在の隠れ状態を表現する潜在変数にのみ依存して生成されると仮定し、現在の隠れ状態は直前の隠れ状態のみに依存する。この HMM と LDA を組み合わせることで、系列情報を考慮できるトピックモデルがいくつか提案されている [5, 1]。系列長を S、トピック数を K としたとき、 HMM をベースにしたトピックモデルの計算量は $O(SK^2)$ である。一方、LDA の計算量は文書長 S ではなく文書毎の語彙数 M に比例し、トピック数 K に対しても比例である。そのため LDA の学習は数時間で行えるデータでも、HMM をベースにしたトピックモデルでは、学習に数日から数週間かかる場合がある。

本稿では文書圧縮技術を利用したトピックモデリング法である Repair Topic Model (RTM) を提案する。RTM は文法圧縮法である Repair とトピックモデリング手法である LDA を組み合わせた手法である。文法圧縮とは文書をコンパクトに表現する文法を生成する手法であり、Repair は最頻の連続した 2 単語を新たな単語に置き換えることを繰り返すことで文法を生成する。Repair によって生成された文法情報を LDA に与えることで構造的な情報を加味したトピックモデリングを実現する。LDA は文書内での単語の共起関係に基づいてトピックを生成するのに対し、RTM は文書内の共起だけでなく、得られた構文木内での共起も加味する。これにより、2 語以上で意味を成すようなフレーズや、文書内での単語の利用され方を考慮したトピックが抽出される。構造情報は Repair によりコンパクトな文法として表現されるため、RTM の計算量は HMM に基づくトピックモデリング手法や、確率的文脈自由文法 (PCFG) [7] や Syntactic Topic Model (STM) [3] などの圧縮を目的としない構文木を利用した手法に比べて小さい。

本稿の構成は以下の通りである。2. では準備として Repair と LDA について簡単に述べる。次に 3. では提案手法である Repair Topic Model について述べる。4. では提案法を実データに適用し、提案法により構造情報を加味したトピックモデリングが行えることを確認する。最後に 5. でまとめを述べる。

連絡先: 石畠正和 ishihata.masakazu@lab.ntt.co.jp

2. 準備

2.1 Repair Algorithm

文法圧縮とは、与えられた文字列を表現する決定的な文脈自由文法 (CFG) を生成することで文書を圧縮する手法である [6] 。与えられた文字列を表現する最小の CFG を発見することは NP-hard であり、経験的によい CFG を生成する手法が数多く提案されている [4]。Repair Algorithm [8] は最も有名な文法圧縮手法の一つである。Repair は文字列中で最も多く表れる文字のペア (bigram) を新たな非終端文字に置き換えることを繰り返すことで CFG を生成する。 圧縮前の文書集合を $\boldsymbol{S} = \{S_1, \ldots, S_D\}$ とし、Repair により得られる CFG を $\boldsymbol{G} = \langle \boldsymbol{V}, \boldsymbol{N}, \boldsymbol{R}, \boldsymbol{S}^* \rangle$ と書く。ここで $\boldsymbol{V} = \{a_1, \ldots, a_V\}$ は終端文字の集合、$\boldsymbol{N} = \{A_1, \ldots, A_N\}$ は非終端文字の集合、$\boldsymbol{R} = \{R_1, \ldots, R_N\}$ はルールの集合、$\boldsymbol{S}^* = \{S_1^*, \ldots, S_D^*\}$ は圧縮後の文書集合である。本稿では得られる CFG がチョムスキー標準形となるよう、すべての終端記号 a_i はルール $R_i = A_i \to a_i$ を持つとする。よって Repair によって置き換えられた bigram の種類数 R は $N - V$ である。$S = \sum_{i=1}^{D} |S_i|$ を元文のサイズ、$S^* = \sum_{i=1}^{D} |S_i^*|$ を圧縮後のサイズとする。また $N_{i,v}$ を S_i 中の終端文字 a_v の出現回数、$N_{i,n}^*$ を S_i^* 中の非終端文字 A_n の出現回数とする。つまり $S = \sum_{i,v} N_{i,v}$、$S^* = \sum_{i,n} N_{i,n}^*$ である。また $M = |\{(i,v) \mid N_{i,v} > 0\}|$、 $M^* = |\{(i,n) \mid N_{i,n}^* > 0\}|$ はそれぞれ元文書と圧縮文書の文書毎の語彙数の総和である。

Step	Text	Rules
0.	abcababc	A_1→a, A_2→b, A_3→c
1.	$A_1A_2A_3A_1A_2A_1A_2A_3$	A_4→A_1A_2
2.	$A_4A_3A_4A_4A_3$	A_5→A_4A_3
3.	$A_5A_4A_5$	

図 1: “abcababc” に Repair を適用した様子。

図 1 に $\boldsymbol{S}=\{abcababc\}$ が与えられた時の Repair の動作を示す。結果として得られる CFG $\boldsymbol{G}$ は $\boldsymbol{V}=\{a,b,c\}$, $\boldsymbol{N}=\{A_1,A_2,A_3,A_4,A_5\}$, $\boldsymbol{R}=\{A_1\to a, A_2\to b, A_3\to c, A_4\to A_1A_2, A_5\to A_4A_3\}$, $\boldsymbol{S}^*=\{A_5A_4A_5\}$ となる。

2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) は最も有名なトピックモデルの一つである [2]。LDA では文書中の各単語はトピックと呼ばれる潜在変数を持ち、各潜在変数は文書毎に異なる分布（トピック分布）より確率的に生成され、各単語は対応するトピックに依存した分布（単語分布）より確率的に生成さる。文書集合 $\boldsymbol{S}$ が与えられたとき、文書 S_i 中の j 番目の単語を $x_{i,j}$ とし、 $x_{i,j}$ に対応する潜在変数を $z_{i,j}$ とする。このとき LDA の生成過程は以下である。

1. $k=1,\dots,K$ に対し、単語分布 $\boldsymbol{\phi}_k\sim \mathrm{Dir}(\boldsymbol{\beta})$ を生成
2. $i=1,\dots,D$ に対し、
   (a) トピック分布 $\boldsymbol{\theta}_i\sim\mathrm{Dir}(\boldsymbol{\alpha})$ を生成
   (b) $j=1,\dots,|S_i|$ に対し、
      i. トピック $z_{i,j}\sim\mathrm{Cat}(\boldsymbol{\theta}_i)$ を生成
      ii. 単語 $x_{i,j}\sim\mathrm{Cat}\left(\boldsymbol{\phi}_{z_{i,j}}\right)$ を生成

トピック分布 $\boldsymbol{\theta}$ と単語分布 $\boldsymbol{\phi}$ が与えられたとき、文書 S_i 中で単語 a_v が得られる確率は $\psi_{i,v}=\sum_{k=1}^K\theta_{i,k}\phi_{k,v}$ である。つまり $\boldsymbol{\theta}$ と $\boldsymbol{\phi}$ を与えられたとき、各単語は他の単語と独立であり、単語の並びなどの構造的な情報は考慮されない。

LDA の学習法は数多く提案されているが [11]、本稿では変分ベイズ法を採用する [2]。変分ベイズ法では、計算が困難である真の事後分布 $p(\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\phi}\mid\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta})$ を求める代わりに、これをよく近似する変分事後分布を学習する。変分事後分布のパラメータは以下の更新式により推定される。

$$\tilde{\alpha}_{i,k}=\alpha_{i,k}+\sum_{v=1}^{V}\frac{N_{i,v}\tilde{\theta}_{i,k}\tilde{\phi}_{k,v}}{\tilde{\psi}_{i,v}},\quad \tilde{\theta}_{i,k}=\exp(\mathrm{E}_k[\tilde{\boldsymbol{\alpha}}_i])$$

$$\tilde{\beta}_{k,v}=\beta_{k,v}+\sum_{i=1}^{D}\frac{N_{i,v}\tilde{\theta}_{i,k}\tilde{\phi}_{k,v}}{\tilde{\psi}_{i,v}},\quad \tilde{\phi}_{k,v}=\exp\left(\mathrm{E}_v\left[\tilde{\boldsymbol{\beta}}_k\right]\right)$$

ここで $\mathrm{E}_k[\alpha]=\Psi(\alpha_k)+\Psi\left(\sum_{k'=1}^K\alpha_{k'}\right)$, $\Psi(x)$ は digamma 関数である。$N_{i,v}$ を文書 i 中の単語 v の出現回数、$M=|\{(i,v)\mid N_{i,v}>0\}|$ とすれば、$\boldsymbol{S}$ に対する LDA の変分ベイズ法の 1 ステップ当たりの計算量は $O((M+D+V)K)$ となる。

3. 提案法

3.1 Repair Topic Model

Repair Topic Model (RTM) は Repair により得られた CFG $\boldsymbol{G}$ を元に、文書の生成モデルを定義する。ただし本稿では Repair は文字単位ではなく、単語単位で圧縮を行うとする。T_i を文書 S_i を表現する $\boldsymbol{G}$ より得られる構文木集合とする。LDA では各単語が潜在トピックを持つのに対し、RTM では T_i の各内点が潜在変数を、T_i の各葉が観測変数を持つ。T_i 中の内点 j に対応する潜在変数を $z_{i,j}$, 葉節点 l に対応する観測変数を $x_{i,l}$ と書く。T_i 中の内点 a,b,c がそれぞれ非終端記号 A_n,A_l,A_r に対応し、ルール $A_n\to A_lA_r$ により展開されたとする。このとき b を a の **左側の子**、c を a の **右側の子** と呼び、それぞれの内点に対応する潜在変数 $z_{i,a},z_{i,b},z_{i,c}$ は以下のような依存関係を持つとする。

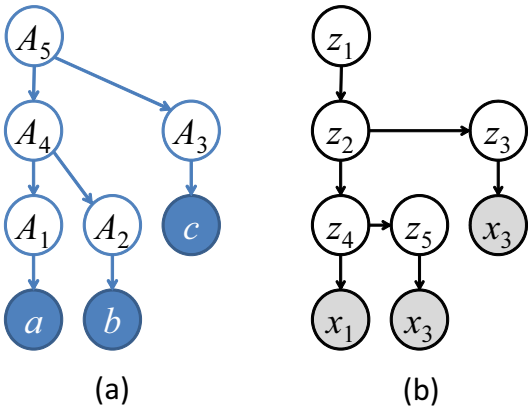

図 2: Repair によって得られる構造の例。Repair によって得られた構文木 (a) と対応するグラフィカルモデル (b) 。

$$p(z_{i,b},z_{i,c}\mid z_{i,a})=p(z_{i,b}\mid z_{i,a})\,p(z_{i,c}\mid z_{i,b})$$

$p(z_{i,b}\mid z_{i,a})$ を **左側遷移分布**、$p(z_{i,c}\mid z_{i,b})$ を **右側遷移分布** と呼び、パラメータ $\boldsymbol{\pi}^L,\boldsymbol{\pi}^R$ を用いて以下のように定義する。

$$p(z_{i,b}\mid z_{i,a})\equiv\pi^L_{z_{i,a},z_{i,b}},\qquad p(z_{i,c}\mid z_{i,b})\equiv\pi^R_{z_{i,b},z_{i,c}}$$

つまり遷移分布は文書 S_i に依存しない。図 2 に構文木の例と対応する潜在トピックの依存関係を表すグラフィカルモデルを示す。構文木中で c が a の右側の子であるとき、グラフィカルモデル中では $z_{i,c}$ は $z_{i,a}$ ではなく、$z_{i,b}$ の子となる。

RTM の生成過程全体を以下に示す。

1. 各トピック $k=1,\dots,K$ に対し、
   (a) 単語分布 $\boldsymbol{\phi}_k\sim\mathrm{Cat}(\boldsymbol{\beta})$ を生成
   (b) 左側遷移分布 $\boldsymbol{\pi}^L_k\sim\mathrm{Cat}\left(\boldsymbol{\gamma}^L\right)$ を生成
   (c) 右側遷移分布 $\boldsymbol{\pi}^R_k\sim\mathrm{Cat}\left(\boldsymbol{\gamma}^R\right)$ を生成
2. 各文書 $i=1,\dots,D$ に対し、
   (a) トピック分布 $\boldsymbol{\theta}_i\sim\mathrm{Cat}(\alpha)$ を生成
   (b) T_i の各内点 j に対し、トピック $z_{i,j}$ を以下にしたがって生成

$$z_{i,j}\sim\begin{cases}\mathrm{Cat}(\boldsymbol{\theta}_i) & j\text{ は根節点}\\ \mathrm{Cat}\left(\boldsymbol{\pi}^L_k\right) & j\text{ は左側の子}\\ \mathrm{Cat}\left(\boldsymbol{\pi}^R_k\right) & j\text{ は右側の子}\end{cases},$$
$$\text{where } k=pv(z_{i,j})$$

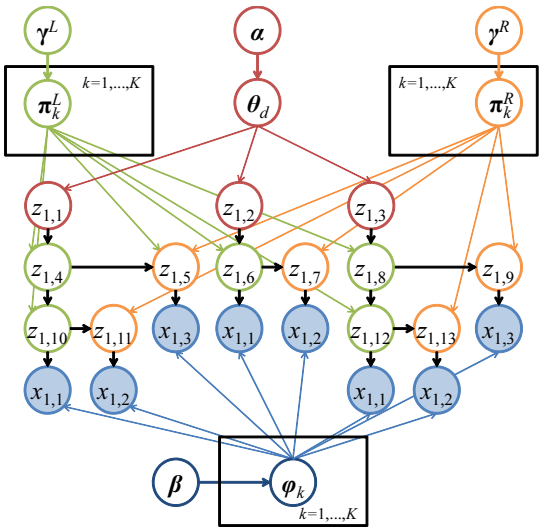

図 3: 文字列 “abcababc” を表現する CFG $\boldsymbol{G}$ が与えられた場合の Repair Topic Model のプレート表記。赤は根節点、緑は左側の子節点、オレンジは右側の子節点、青は葉節点。

(c) T_i の各葉 l に対し、単語 $x_{i,l} \sim \mathrm{Cat}(\boldsymbol{\phi}_k)$ $(k = pv(x_{i,l}))$ を生成

ここで $pv(z_{i,j})$ は $z_{i,j}$ のグラフィカルモデルにおける親潜在変数の値であり、グラフィカルモデルの構造は事前に与えられると仮定する。学習の際にはこのグラフィカルモデルは CFG $\boldsymbol{G}$ より自動的に生成される。図 3 に RTM のグラフィカルモデルの例を示す。潜在変数の数やそれらの依存関係は実際に得られる CFG によって変化する。

LDA と RTM の違いは、RTM では潜在トピックが図 2 (b) に示すように、$A_n \to A_l A_r$ の形のルールによって階層的な構造を持つ点である。この構造により、RTM は単語の局所的な共起関係や順序関係を考慮したトピックを抽出可能である。仮に得られた CFG $\boldsymbol{G}$ が $A_n \to A_l A_r$ の形のルールを全く持たないとき、RTM は LDA と完全に一致する。

3.2 RTM の学習

RTM の学習は 2.2 節と同様に変分ベイズ法で行う。RTM を愚直に学習すると、各文書を表す構文木の総節点数に比例した計算時間を要する。しかし HMM に対する Forward-Backward アルゴリズム [9] や PCFG に対する Inside-Outside アルゴリズム [7] と同様に動的計画法を用いれば、与えられた CFG のサイズに比例する時間で高速に学習可能である。ここでは RTM を効率的に学習するために **内側確率** $\mathrm{I}_{k,n}$ と **外側期待値** $\mathrm{O}_{n,k}$ という新たな量を導入する。

内側確率 $\mathrm{I}_{k,n}$ とは、ある内点の潜在トピックが値 k のとき、そこから非終端記号 A_n より得られる終端文字列を得る確率である。$\mathrm{I}_{k,n}$ は以下のように再帰的に計算可能である。

$$\mathrm{I}_{k,n} \equiv \begin{cases} \phi_{k,v} & A_n \to v \in \boldsymbol{R} \\ \sum_{k_l=1}^{K} \boldsymbol{\pi}^L_{k,k_l} \mathrm{I}_{k_l,l} \sum_{k_r=1}^{K} \boldsymbol{\pi}^R_{k_l,k_r} \mathrm{I}_{k_r,r} & A_n \to A_l A_r \in \boldsymbol{R} \end{cases}$$

ここで $\mathrm{I}^R_{a,r} \equiv \sum_{b=1}^{K} \boldsymbol{\pi}^R_{a,b} \mathrm{I}_{b,r}$ を中間的に保持すれば、内側確率は $O(VK + RK^2)$ で計算可能である。内側確率 $\mathrm{I}_{k,n}$ より、非終端文字 A_n が圧縮文書 S^*_i に含まれる確率は $\psi_{i,n} \equiv \sum_{k=1}^{K} \theta_{i,k} \mathrm{I}_{k,n}$ となる。

外側期待値 $\mathrm{O}_{n,k}$ とは、文書が与えられたとき、A_n でラベル付けられた内点の潜在トピックが値 k で表れる回数の期待値である。圧縮文書 S^*_i 中の非終端文字 A_n の出現回数を $N^*_{i,n}$ とすれば、A_n を右辺に持つルールが存在しないとき、その外側期待値は以下となる。

$$\mathrm{O}_{n,k} \equiv \sum_{i=1}^{D} \frac{N^*_{i,n} \theta_{i,k}}{\psi_{i,n}}$$

ここで $M^* = |\{(i,n) \mid N^*_{i,n} > 0\}|$ とすれば、この計算量は $O(M^* K)$ である。しかし、$N_{i,n} > 0$ かつ $A_n \to A_l A_r \in \boldsymbol{R}$ であるとき、A_l と A_r は A_n を通して間接的に文書 S_i 内に表れることになる。つまり A_n の外側期待を A_l と A_r の外側期待に伝播させる必要がある。A_n, A_l, A_r に対応する潜在変数の値をそれぞれ k, k_l, k_r とし、A_n の外側期待値 $\mathrm{O}_{n,k}$ が既に計算済みであるとする。このとき、A_l と A_r の外側期待値 O_{l,k_l}, O_{r,k_r} には以下のような加算が行われる。

$$\mathrm{O}_{l,k_l} \mathrel{+}= \mathrm{O}^L_{n,k_l} \mathrm{I}^R_{k_l,r}, \qquad \mathrm{O}_{r,k_r} \mathrel{+}= \mathrm{O}^L_{n,k_l} \mathrm{I}_{k_l,l} \pi^R_{k_l,k_r}$$

ここで $\mathrm{O}^L_{n,k_l} \equiv \mathrm{O}_{n,k} \pi^L_{k,k_l}$ である。この加算は RK^2 回行われるため、外側期待値の計算量は $O(MK + RK^2)$ となる。

RTM は内側確率と外側確率を利用した変分ベイズ法により学習される。各変分分布のパラメータは以下の式により繰り返し更新することで推定される。

$$\tilde{\alpha}_{i,k} = \alpha_{i,k} + \sum_{n=1}^{N} \frac{N^*_{i,n} \tilde{\theta}_{i,k} \mathrm{I}_{k,n}}{\tilde{\psi}_{i,n}}, \ \tilde{\theta}_{i,k} = \exp(\mathrm{E}_k[\tilde{\boldsymbol{\alpha}}_i]),$$

$$\tilde{\beta}_{k,v} = \beta_{i,k} + \mathrm{O}_{v,k} \tilde{\phi}_{k,v}, \ \tilde{\phi}_{k,v} = \exp\left(\mathrm{E}_v\left[\tilde{\boldsymbol{\beta}}_k\right]\right),$$

$$\tilde{\gamma}^L_{k,k_l} = \gamma^L_{k,k_l} + \sum_{n=1}^{N} \mathrm{O}_{n,k} \tilde{\pi}^L_{k,k_l} \mathrm{I}_{k_l,l} \mathrm{I}^R_{k_l,r}, \ \tilde{\pi}^L_{k,k_l} = \exp\left(\mathrm{E}_{k_l}\left[\tilde{\boldsymbol{\gamma}}^L_k\right]\right),$$

$$\tilde{\gamma}^R_{k_l,k_r} = \gamma^R_{k_l,k_r} + \sum_{n=1}^{N} \mathrm{O}^L_{n,k_l} \mathrm{I}_{k_l,l} \tilde{\pi}^R_{k_l,k_r} \mathrm{I}_{k_r,r}, \ \tilde{\pi}^R_{k_l,k_r} = \exp\left(\mathrm{E}_{k_r}\left[\tilde{\gamma}^R_{k_l}\right]\right)$$

ここで上式の内側確率と外側期待値は $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\pi}$ の代わりに $\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\pi}}$ を用いて計算される。RTM に対する変分ベイズ法の 1 ステップ当たりの計算量は $O((M^* + D + V)K + RK^2)$ である。一方、LDA の変分ベイズ法の 1 ステップ当たりの計算量は $O((M + D + V)K)$ である。つまり RTM は LDA に $O(RK^2)$ の計算を追加することで構造情報を加味したトピックモデリングを実現する。一方、HMM をベースにしたトピックモデルは $O(SK^2)$ の計算が必要であり、一般的に $S \gg R$ であるため、RTM の方が遥に高速である。

4. 実験

RTM を実データに適用し、LDA と比べてテストデータに対する対数尤度が改善する事を確認する。本実験では表 1 に示す 5 つのデータセットに対して RTM と LDA を適用し、テスト尤度と計算時間を比較する。ここで LDA は元文書 $\boldsymbol{S}$ を直接入力した場合 (LDA1) と圧縮文書 $\boldsymbol{S}^*$ を入力した場合 (LDA2) の 2 種類の実験を行った。系列情報を破壊しないため、高頻度語（ストップワード）や低頻度語の除去という前処理は行っていない。トピック数 K を $5, 10, \ldots, 100$ と変化させながらを学習行い、変分自由エネルギーを最大とする K を最適なトピック数とし、文書長で正規化されたテスト対数尤度を計算した。

Dataset	D	V	R	S	S^*	M	M^*
Brown	500	47,341	5,382	1,045,281	775,131	352,725	459,409
Reuters	19,065	81,731	17,553	2,514,931	1,600,113	1,452,767	1,362,720
Mail(u)	50	37,399	17,153	2,277,440	1,504,763	219,132	393,690
Mail(m)	5,000	37,399	17,051	2,279,665	1,511,619	1,165,259	1,222,030
MovieLens	942	1,682	459	90,252	87,337	90,252	87,337

表 1: 各データセットの文書数 D, 語彙数 V, $A_n \to A_l A_r$ の形のルール数 R, 元文の総文長 S^*, 圧縮後の総文長 S, 元文の各文書の異なり語彙数の総和 M^*, 圧縮後の各文書の異なり語彙数の総和 M。

図 4 に各データセットに対する RTM, LDA1, LDA2 の負のテスト対数尤度を示す。負のテスト対数尤度は低いほどモデルがデータに合っていることを意味する。グラフより全データセットにおいて RTM の負のテスト対数尤度が最も低いことが分かる。これより構造情報を考慮することでより系列データに合ったモデリングが行える事が確認できた。図 5 に Brown データに対する各手法の計算時間を示す。計算時間は LDA が K に対して線形であるのに対し、RTM は RK^2 の影響で緩やかに非線形に増加している。計算時間の増加は $M+D+V$ に対する R が大きいほど顕著となる。ここで R は S の 1/100 以下であるため、HMM をベースにしたトピックもデルはこの 100 倍以上計算時間必要だと予想される。

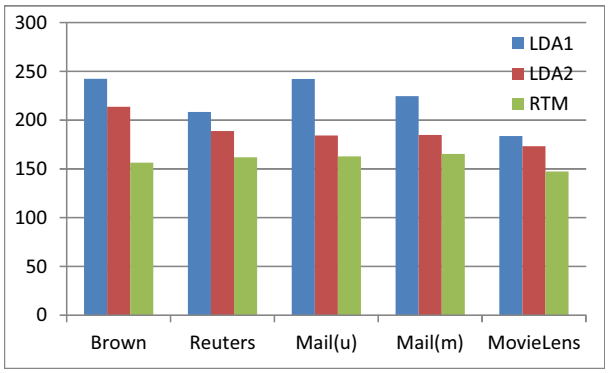

図 4: 各手法の各データに対する負のテスト対数尤度。

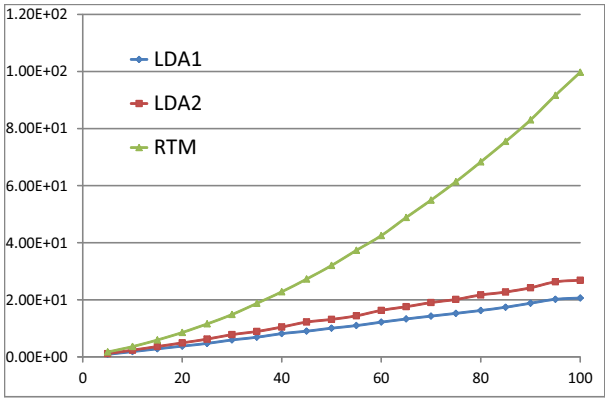

図 5: Brown データに対する学習時間とトピックスの関係。

5. まとめ

本稿では、大規模系列データに対するトピックモデルとして Repair Topic Model を提案した。提案法は文法圧縮アルゴリズムである Repair によって得られた文法と LDA を組み合わせる事で、構造情報を考慮したトピック解析を可能とする。また提案法は圧縮された文法を利用するため、その学習時間は圧縮された文法サイズに比例する。そのため HMM などの系列情報をそのまま利用する手法と比べ、提案法は遥に高速に学習が可能である。本稿では提案方を 5 つの実データに適用し、LDA よりもよいテスト対数尤度を示すことを確認した。

参考文献

[1] Mark Andrews and Gabriella Vigliocco. The Hidden Markov Topic Model: A Probabilistic Model of Semantic Representation. *Topics in Cognitive Science*, 2(1):101–113, January 2010.

[2] DM Blei, AY Ng, and MI Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[3] J Boyd-Graber and DM Blei. Syntactic topic models. In *NIPS*, pages 185–192, 2008.

[4] Moses Charikar, Eric Lehman, April Lehman, Ding Liu, Rina Panigrahy, Manoj Prabhakaran, and Amit Sahai. The Smallest Grammar Problem. *IEEE Transactions on Information Theory*, 2005.

[5] Amit Gruber, Y Weiss, and M Rosen-Zvi. Hidden topic Markov models. In *AISTATS*, 2007.

[6] John C Kieffer and En-hui Yang. Grammar-Based Codes : A New Class of Universal Lossless Source Codes. *IEEE Transactions on Information Theory*, 46(3):737–754, 2000.

[7] K. Lari and S.J. Young. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech & Language*, 4(1):35–56, January 1990.

[8] N Jesper Larsson and Alistair Moffat. Off-Line Dictionary-Based Compression. In *IEEE*, volume 88, 2000.

[9] Norman Weiss Leonard E. Baum, Ted Petrie, George Soules. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[10] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proc. of IEEE*, pages 257–286, 1989.

[11] YW Teh, David Newman, and Max Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. *NIPS*, 2006.