GDMに基づくインタラクティブトピックモデリングの提案

Proposal for Interactive Topic Modeling based on GDM

小林 賢治[1]　高間 康史[1]　柴田 祐樹[1]
Kenji Kobayashi[1]　Yasufumi Takama[1]　Hiroki Shibata[1]

[1] 東京都立大学 システムデザイン学部
[1] Faculty of System Design, Tokyo Metropolitan University

Abstract: 本稿では，GDM（Geometric Dirichlet Means）に基づくインタラクティブトピックモデリングの手法について提案する．トピックモデルは多数の文書データに含まれるトピックの解析に用いられるが，教師なし学習であるため分析者の期待する結果が得られる保証はない．本稿では，幾何学的計算に基づく GDM を採用し，インタラクティブなモデル変更に必要なパラメータや操作を提案する．各操作の適用事例に基づき，操作意図をモデリング結果に反映できることを示す．

1　はじめに

本稿では，GDM(Geometric Dirichlet Means)[2] に基づくインタラクティブトピックモデリングの手法を提案する．トピックモデリングは，文書集合に存在するトピックを分析する手法として利用される，教師なし学習の一種である [1]．一つの文書には複数のトピックが含まれていることを仮定して，文書に含まれるトピックやトピックに関連する単語などを確率分布として表すことができる．しかしトピックモデルは教師なし学習であるため，分析結果がユーザの期待するものとは異なる場合がある．この問題を解決するため，人間参加型学習を用いたインタラクティブトピックモデリングが提案されている [6]．この手法は，トピックモデリングの学習過程に分析者の知見を介入させることで期待する結果に近づける半教師あり学習である．計算機による出力結果を分析者が確認し，フィードバックを与えて逐次的に修正していく．

トピックモデルとして LDA(Latent Dirichlet Allocation) を用いた既存研究 [6] を参考に，本稿では GDM アルゴリズムを用いたインタラクティブトピックモデリングを提案する．GDM は幾何学的視点からトピックを推定するトピックモデリングの手法である．LDA において，文書はトピックを基底とする潜在トピック座標単体上に射影される．この解釈に基づき，GDM では文書集合に重み付きクラスタリングを適用した結果に幾何学的補正を加えて基底ベクトルを推定する．

提案手法では初期クラスタリングの結果に基づき GDM モデルを作成する．その結果を分析者が確認し，期待とは異なる部分を変更する操作を適用する．操作結果に基づきクラスタリングおよび GDM を再実行し，モデルを更新する．既存研究 [6] ではモデル変更操作として，add word，remove word，change word order，remove document，merge topic，split topic，add to stop words を提案しているが，本稿では文書クラスタリングに基づく GDM の特性を活かし，add word，remove word，change word order，add document，remove document，merge topic，split topic，add to stop words の 8 種類を提案する．

本稿では Livedoor ニュースコーパス[1]から取得した文書集合を用いて，提案する操作により想定する効果が得られるかを検証する．トピック数を指定して GDM を実行し，得られた初期モデルに対して提案する 8 種類の操作のうち紙面の都合により 3 種類のみについてそれぞれ適用した結果，意図通りにモデルが更新されることを示す．

2　関連研究

2.1　トピックモデリング

トピックモデリングは文書集合の潜在トピックを推定する確率モデルである．文書は複数の潜在的なトピックから確率的に生成されていると仮定して，文書に関連するトピックの生成確率を示すトピック分布，トピックに関連する単語の生成確率を示す単語分布を推定する．

トピックモデリングの代表的な手法として LDA[1] や GDM[2] が挙げられる．本節では本稿で利用する GDM について説明する．

[1] https://www.rondhuit.com/download.html

2.2 GDM(Geometric Dirichlet Means)

GDM は文書集合のトピックを推定するトピックモデリングの一手法である．GDM で使用する変数を表 1，アルゴリズムを Algorithm1 に示す．

GDM は幾何学的計算に基づいて単語分布 $\boldsymbol{\phi} = (\boldsymbol{\phi}_{k_1}, ..., \boldsymbol{\phi}_{k_{N_K}})$ を推測する．トピック k は N_V 次元空間上のベクトル $\boldsymbol{\phi}_k$，文書は各トピックを頂点とする単体上に生成される．GDM は主に以下の二つのステップから単語分布 $\boldsymbol{\phi}$ を推測する．

- 文書集合に対し重み付きクラスタリングを実行しクラスタ中心 $\boldsymbol{\mu} = (\boldsymbol{\mu}_{k_1}, .., \boldsymbol{\mu}_{k_{N_K}})$ を求める (Algorithm1，2 行目)
- 幾何学的補正により $\boldsymbol{\mu}$ から $\boldsymbol{\phi}$ を求める (Algorithm1，5 行目)

GDM ではクラスタ数 (N_K) を指定し，重み付きクラスタリングにより文書集合をクラスタに分割する．本稿では重み付き k-means を利用する．各クラスタがトピックに対応する．次に，求めた $\boldsymbol{\mu}$ に対して式 (1) に示す幾何学的補正を加えて単語分布 $\boldsymbol{\phi}$ を求める (Algorithm1，5 行目)．

$$\boldsymbol{\phi}_k = \boldsymbol{C} + m_k(\boldsymbol{\mu}_k - \boldsymbol{C}), k \in K \quad (1)$$

$$\boldsymbol{C} = \frac{1}{N_D} \sum_{d \in D} \bar{\boldsymbol{w}}'_d \quad (2)$$

$$m_k = \frac{\max_{d \in D_k} \|\boldsymbol{C} - \bar{\boldsymbol{w}}'_d\|_2}{\|\boldsymbol{C} - \boldsymbol{\mu}_k\|_2}, k \in K \quad (3)$$

$$\bar{\boldsymbol{w}}'_d = \frac{\bar{\boldsymbol{w}}_d \circ \boldsymbol{\gamma}_d}{\|\bar{\boldsymbol{w}}_d \circ \boldsymbol{\gamma}_d\|_1} \quad (4)$$

式 (4) の $\circ$ はアダマール積を表す．ただし，式 (1) により計算される単語分布で $\phi_{kv} < 0$ となる場合を考慮して $k \in K$ において式 (5) により ϕ_{kv} を更新する (Algorithm1，7 行目)．

$$\phi_{kv} = \frac{R(\phi_{kv})}{\sum_{v' \in V} R(\phi_{kv'})}, R(x) = \begin{cases} x \ (x > 0) \\ 0 \ (x \leq 0) \end{cases} \quad (5)$$

Algorithm 1 Geometric Dirichlet Means

1: D の中心ベクトル $\boldsymbol{C}$ を求める
2: クラスタ中心 $\boldsymbol{\mu}_k (k \in K)$ を求める
3: $m_k (k \in K)$ を求める
4: **for** $k \in K$ **do**
5: 　$\boldsymbol{\phi}_k$ を求める
6: 　**for** $v \in V$ **do**
7: 　　ϕ_{kv} を更新

表 1: GDM で使用する記号

記号	説明
$D = \{d_1, d_2, ..., d_{N_D}\}$	文書集合
$K = \{k_1, k_2, ..., k_{N_K}\}$	トピック集合
$V = \{v_1, v_2, ..., v_{N_V}\}$	語彙集合
$D_k \subseteq D$	$k \in K$ に割り当てられた文書集合
w_{dn}	$d \in D$ の n 番目の単語
$\epsilon_d (> 0)$	$d \in D$ の重さ
$\bar{w}_{dv} (\geq 0)$	$d \in D$ 内の全単語に占める $v \in V$ の出現割合
$\gamma_{dv} (> 0)$	$d \in D$ における $v \in V$ の重さ
N_d	$d \in D$ の単語数
$\boldsymbol{\theta}_d = (\theta_{dk_1}, ..., \theta_{dk_{N_K}})$	$d \in D$ のトピック分布
$\boldsymbol{\phi}_k = (\phi_{kv_1}, ..., \phi_{kv_{N_V}})$	$k \in K$ の単語分布
$\theta_{dk} (\in [0, 1])$	$d \in D$ での $k \in K$ の生成確率
$\phi_{kv} (\in [0, 1])$	$k \in K$ での $v \in V$ の生成確率

2.3 人間参加型学習 (human in the loop)

人間参加型機械学習 (human in the loop)[3][4] は，機械による学習に人間の知見をインタラクティブに介入させる学習方法で，機械による学習結果の間違いや想定と異なる部分を人間が発見しフィードバックを与える半教師あり学習である．まず初めに，機械は通常の機械学習（分類モデルやクラスタリングなど）を行う．人間が学習結果を見て機械にフィードバックを与え，それに基づき機械は再学習をする．このプロセスを繰り返すことで，分類・予測精度が向上したり，人間が希望する結果に近づくことが期待できる．

人間参加型学習は人間の知見を学習に加えることができるため，学習データが不足している場合や学習結果に高い精度が求められる場合に用いられる．

2.4 COP K-means

COP K-means[5] は通常の K-means に制約を加えるクラスタリング手法である．制約はデータ対に対するものとして与えられ，以下の二種類が定義されている．

- Must-link 制約：2 つのデータが同じクラスタに配置されなければならないことを示す制約
- Cannot-link 制約：2 つのデータが同じクラスタに配置されてはならないことを示す制約

制約はデータをクラスタに割り当てる際に利用される．データに Must-link 制約が適用されている場合，対となるデータがすでにクラスタに割り当てられていれば同じクラスタに割り当てる．Cannot-link 制約が適用されている場合，対となるデータとは異なるクラスタに割り当てる．データは制約に違反しない限り最も近いクラスタに割り当てられ，制約に違反した場合はクラスタリングは失敗する．

3 GDM に基づくインタラクティブトピックモデリング

提案手法では，最初に通常のトピックモデリングを行う．その後，分析者が結果を見てフィードバックを機械に与えることで，トピックモデルを修正する．GDM では重み付き K-means を用いるが，提案手法では分析者のフィードバックを考慮するために，GDM の再実行時には COP K-means を用いる．すなわち，モデル変更操作はクラスタリングにおいて考慮される．本節では，クラスタリングにおいて修正対象となるパラメータを説明した後，8 種類のモデル変更操作について説明する．

3.1 修正対象パラメータ

提案手法では，以下のパラメータをユーザからのフィードバックに基づき修正する．

- γ_{dv}：文書 d における単語 v の重さ
- ϵ_d：d の重さ
- Must-cluster 制約：指定したデータが指定したクラスタに割り当てられなければならないことを示す制約
- Cannot-cluster 制約：指定したデータが指定したクラスタに割り当てられてはならないことを示す制約

γ_{dv} は初期値を 1 とし，これを変更することで各文書における単語の重要度を調整する．ϵ_d は初期値を 1 とし，文書がクラスタ中心に与える影響を調整する．Must-cluster 制約，Cannot-cluster 制約は COP K-means においてデータをクラスタに割り当てる際に利用する．データに Must-cluster 制約が適用されていた場合，指定されているクラスタにデータを割り当てる．Cannot-cluster 制約が適用されている場合，指定されているクラスタ以外の最も近いクラスタに割り当てる．

3.2 改良操作

本稿では 8 種類の改良操作を提案する．操作概要，操作手順，変更するパラメータについての説明を行う．

3.2.1 add word

add word の引数はトピック $k(\in K)$ と単語 $v(\in V)$ であり，ϕ_{kv} が k において他の単語よりも高くなるように修正する操作である．実行手順を以下に示す．

1. k に割り当てられていない文書から v を含む $D_{\text{add}} \subset D$ を選択する
2. k と D_{add} に Must-cluster 制約を適用する
3. $\boldsymbol{\gamma}_{dv}, d \in D_k$ を変更する

手順 2 において，分析者は v を含む各文書の内容を見て D_{add} を選択する．k，v を指定して add word を適用した場合の Newton 法を用いた $\boldsymbol{\gamma}$ 更新式を式 (6) に示す．ただし，$\gamma_{dv} = \gamma_1$ とし，k で最も出現確率の高い単語を v_{H} とする．(n) は更新過程において n ステップ目の値であることを表し，本稿では終了条件を $f(\gamma_1^{(n)}) < 10^{-8}$ とした．A は加速係数である．

$$\gamma_1^{(n+1)} = \gamma_1^{(n)} - Af^{(n)} / \frac{\partial f^{(n)}}{\partial \gamma_1^{(n)}} \quad (6)$$

$$f^{(n)} = \mu_{kv}^{(n)} - \{\mu_{kv_{\text{H}}}^{(n)} + (\mu_{kv_{\text{H}}}^{(n)} - \mu_{kv_{\text{H}-1}}^{(n)})\} \quad (7)$$

$$\mu_{kv} = \frac{1}{\sum_{d \in D_k} \epsilon_d} \sum_{d \in D_k} \frac{\bar{w}_{dv}\gamma_{dv}}{\|\bar{\boldsymbol{w}}_d \circ \boldsymbol{\gamma}_d\|_1} \epsilon_d \quad (8)$$

3.2.2 remove word

remove word の引数はトピック $k(\in K)$ と単語 $v(\in V)$ であり，ϕ_{kv} を 0 に近づける操作である．この操作では，D_k 内の全文書 d について γ_{dv} を非常に小さな値に変更する．本稿ではこれを 10^{-6} とした．

3.2.3 change word order

change word order の引数はトピック $k(\in K)$ と単語 v_1'，$v_2'(\in V)$ であり，k の関連単語リスト内で両単語の位置を入れ替える操作である．これは，$\phi_{kv_1'}$，$\phi_{kv_2'}$ の大小関係を入れ替えることに相当し，D_k 内の全文書 d について両単語に対応した重み γ_1, γ_2 の値をそれぞれ，$i = 1, 2$ について式 (9) に基づき更新することで実現する．計算には 2 変数非線形連立方程式を解くために 1 次近似 Newton 法を使用する．$\boldsymbol{J}$ はヤコビ行列 $(J_{ij} = \partial f_i / \partial \gamma_j)$，$\boldsymbol{\delta}$ はクロネッカーデルタを表す．本稿では終了条件を $|f_1| + |f_2| \leq 10^{-8}$ とした．

$$\gamma_i^{(n+1)} = \gamma_i^{(n)} - \sum_{j=1,2} \left(J^{-1}\right)_{ij}^{(n)} f_j^{(n)} \quad (9)$$

$$f_i = \frac{1}{\sum_{d\in D_k} \epsilon_d} \sum_{d\in D_k} \left(\gamma_{dv_i'} \bar{w}_{dv_i'}\right) - \sum_{j=1,2} (1-\delta_{ij}) \mu_{kv_j'} \quad (10)$$

3.2.4 add document

add document の引数はトピック $k(\in K)$ と文書集合 $D_{\text{add}}(\subset D - D_k)$ であり，k に D_{add} の文書を関連付ける操作である．既存研究 [6] には存在しない操作であるが，文書が操作主体となる GDM の特徴を生かすために本稿で提案する．実行手順を以下に示す．

1. k 以外に割り当てられた文書から追加する D_{add} を選ぶ
2. k と $d \in D_{\text{add}}$ に Must-cluster 制約を適用する
3. $\epsilon_d, d \in D_{\text{add}}$ を変更する

手順 1 において，D_{add} は k 以外に割り当てられた文書の内容を分析者が見ることによって選択する．手順 3 において，ϵ_d の値は式 (11) により求める．ここで，k' は d が所属していたトピック，old, new は d を k' から k へ移動する前後の状態をそれぞれ意味する．$\text{Dis}(k,d)$ は k のクラスタ中心と d のユークリッド距離である．この更新により，$\text{Dis}^{\text{new}}(k,d) < \text{Dis}^{\text{old}}(k,d)$ となる．

$$\hat{\epsilon}_d = \frac{\eta_d}{1-\eta_d} E_k^{\text{old}} \quad (11)$$

$$\eta_d = 1 - \frac{1}{2} \frac{\text{Dis}^{\text{old}}(k',d)}{\text{Dis}^{\text{old}}(k,d)} \quad (12)$$

$$E_k^{\text{old}} = \sum_{d\in D_k^{\text{old}}} \epsilon_d \quad (13)$$

D_{add} 内の d と k の関係を強めるために，ϵ_d は 1 以上の値とする必要があるが，モデル修正操作を何度も繰り返した際に大きくなりすぎることも防ぐ必要がある．そのため，式 (11) で求めた値を以下に従い修正する．

$$\epsilon_d = \begin{cases} \min(\hat{\epsilon}_d, \max(1, 0.9E_k^{\text{old}})) & (\hat{\epsilon}_d > 1) \\ 1 & (\hat{\epsilon}_d \le 1) \end{cases}$$

add word，add document の 2 つの操作において，どちらも文書を追加するという操作を行うが，add word は指定した単語の重さを変更することが目的であるのに対し，add document は文書をクラスタに近づける事が目的である点で異なる．

3.2.5 remove document

remove doucment の引数はトピック k と文書集合 $D_{\text{remove}}(\subset D_k)$ であり，k から D_{remove} の文書を削除する操作である．実行手順を以下に示す．

1. D_k から D_{remove} を選ぶ
2. k と $d \in D_{\text{remove}}$ に Cannot-cluster を適用する
3. $d \in D_k - D_{\text{remove}}$ について $\hat{\epsilon}_d = \frac{1}{2} E_k^{\text{old}}$

手順 1 において，分析者は k の各文書の内容を見て D_{remove} を選択する．手順 3 では D_{remove} の各文書が k のクラスタ中心から離れるように，ϵ_d を求める．この更新により，$\text{Dis}^{\text{new}}(k,d) < \text{Dis}^{\text{old}}(k,d)$ となる．また，$\hat{\epsilon}_d$ において 3.2.4 節と同様の修正を行う．

3.2.6 split topic

split topic の引数はトピック k と単語集合 $V_{\text{add}} \subseteq V$ であり，k を二つのトピックに分ける操作である．実行手順を以下に示す．

1. D_k において $v \in V_{\text{add}}$ を含む文書から，新しいトピック k_{new} に移す D_{add} を選択する
2. D_{add} の重心を中心とするクラスタ k_{new} を生成する
3. $d \in D_{\text{add}}$ と k_{new} に Must-cluster 制約を適用する

手順 1 において，D_{add} は $v \in V_{\text{add}}$ を含む文書の内容を分析者が見ることによって選択する．

3.2.7 merge topic

merge topic の引数はトピック k，k' であり，二つのトピックを一つに統合する操作である．実行手順を以下に示す．

1. $d \in D_k \cup D_{k'}$ と k に Must-cluster 制約を適用する
2. K から k' を削除する

3.2.8 add to stop words

単語分布に出現する単語の中には分析者から見て不要な単語が含まれていることがある．add to stop words の引数は $v(\in V)$ であり，v をストップワードに追加する操作である．ストップワードに追加された単語は GDM の分析対象外となる．$\bar{\boldsymbol{w}}$，$\boldsymbol{\gamma}$，$\boldsymbol{\mu}$，$\boldsymbol{\phi}$ から v に関する値を削除する．

4 予備実験

4.1 実験概要

予備実験では，3.2 節で提案した操作によって GDM のモデルが意図通りに変更されることを検証する．紙面の都合により，add word，add document，change word order の適用結果のみ示す．学習データは Livedoor ニュースコーパス[2]を使用した．前処理として Mecab を利用した形態素解析を行い，形態素として名詞 (一般，固有名詞，サ変接続，形容動詞語幹) と，自立形容詞を抽出した．抽出された形態素から，SlothLib[3]のストップワードリストに記載された単語を除去した．文書数は 100，語彙数は 2,402，総単語数は 13,907 となった．

本稿では，モデル変更前後の評価指標として式 (14) で定義される Perplexity (P) を利用する．これは確率モデルの予測性能を評価する指標であり，値が低いほど性能が高いことを示す．

$$P = \exp\left(-\frac{1}{\sum_D N_d}\sum_D\sum_{n=1}^{N_d}\log p(w_{dn})\right) \quad (14)$$

$$p(w_{dn}) = \sum_{k\in K}\theta_{dk}\phi_{kw_{dn}} \quad (15)$$

4.2 実験結果

GDM による初期モデルの各トピックにおける文書割り当て状態を表 2 に，add word(3,‘mac’) を適用した後の文書割り当て状態を表 3 にそれぞれ示す．この操作により，文書 0, 1, 7 がトピック 4 から 3 へ移動している．また，操作前後でのトピック 3 における単語分布 (ϕ_{k_3v} の上位 10 単語) を表 4 に示す．表 4 より，単語 ‘mac’ のトピック 3 における生成確率が操作後に最も高くなっていることがわかる．

初期モデルに対し，add document(2, D_{add}), $D_{\text{add}} = \{81, 85, 86, ..., 98\}$ を適用した結果を表 5 に示す．表 5 より，指定した文書が全てトピック 2 に割り当てられていることがわかる．

初期モデルに対し，change word order(3,‘ノート’,‘書籍’) を適用した後のトピック 3 における単語分布 (ϕ_{k_3v} の上位 10 単語) を表 6 に示す．表 6 より，単語 ‘ノート’，‘書籍’ のトピック 3 での生成確率における大小関係が入れ替わっているのがわかる．

操作前後の Perplexity を表 7 に示す．表 7 より，操作前と比べて add word，add document によって Perplexity は減少しており，モデルが改善されていることがわかる．

[2]https://www.rondhuit.com/download.html
[3]http://svn.sourceforge.jp/svnroot/slothlib/CSharp/Version1/SlothLib/NLP/Filter/StopWord/word/Japanese.txt

表 2: 初期モデルの文書割り当て状態

トピック id	文書 id
0	9, 17
1	62, 68, 70, 72
2	97
3	2, 8, 16
4	0, 1, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 63, 64, 65, 66, 67, 69, 71, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 98, 99

表 3: add word(3,‘mac’) 適用後の文書割り当て状態 (変化のあったトピックのみ掲載)

トピック id	文書 id
3	0, 1, 2, 7, 8, 16
4	3, 4, 5, 6, 10, 11, 12, 13, 14, 15, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 63, 64, 65, 66, 67, 69, 71, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 98, 99

表 4: add word(3,‘mac’) 適用による単語分布の変化

初期モデル		add word	
単語	生成確率	単語	生成確率
‘サービス’	0.0331	‘mac’	0.0226
‘ノート’	0.0228	‘サービス’	0.0170
‘生産’	0.0201	‘セキュリティ’	0.0170
‘ultrabook’	0.0192	‘リリース’	0.0165
‘レッツ’	0.0190	‘os’	0.0145
‘書籍’	0.0183	‘lion’	0.0134
‘電子’	0.0183	‘ノート’	0.0133
‘mac’	0.0174	‘アップル’	0.0129
‘hp’	0.0173	‘製品’	0.0125
‘終了’	0.0170	‘パソコン’	0.0124

表 5: add document 後の文書割り当て状態

トピック id	文書 id
0	9, 17
1	62, 68, 70, 72
2	81, 85, 86, 87, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98
3	2, 8, 16
4	0, 1, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 63, 64, 65, 66, 67, 69, 71, 73, 74, 75, 76, 77, 78, 79, 80, 82, 83, 84, 88, 99

表 6: change word order(3,‘ノート’,‘書籍’) 適用による単語分布の変化

初期モデル		change word order	
単語	生成確率	単語	生成確率
‘サービス’	0.0331	‘サービス’	0.0328
‘ノート’	0.0228	‘書籍’	0.0225
‘生産’	0.0201	‘生産’	0.0202
‘ultrabook’	0.0192	‘ultrabook’	0.0192
‘レッツ’	0.0190	‘レッツ’	0.0191
‘書籍’	0.0183	‘ノート’	0.0186
‘電子’	0.0183	‘電子’	0.0182
‘mac’	0.0174	‘mac’	0.0173
‘hp’	0.0173	‘hp’	0.0173
‘終了’	0.0170	‘終了’	0.0169

表 7: Perplexity(P) の比較

操作	P
操作前	2645
add word	2538
add document	2286
change word order	2645

5 まとめ

本稿では，GDM に基づくインタラクティブトピックモデリングを提案した．提案手法では，分析者の意図をモデルに反映する 8 種類の操作を定義し，パラメータ更新方法を示した．ニュース記事を対象とした予備実験により，add word，add document により意図通りにモデルが変化し期待したトピックが得られることを示した．スペースの都合上示すことができなかったが，他の操作でも意図通りのモデル変更が可能であることを確認している．今後は，より多様な条件での検証を行う他，実際のユーザによる評価実験を行う予定である．

参考文献

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan: Latent Dirichlet allocation, *Journal of Machine Learning Research 3*, pp. 993-1022, 2003.

[2] M. Yurochkin, and X. Nguyen: Geometric Dirichlet Means algorithm for topic inferene, *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2513–2521, 2016

[3] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park: iVisClustering: An Interactive Visual Document Clustering via Topic Modeling, *Eurographic Conference on Visualization*, vol. 31, no. 3, pp. 1155-1164, 2012

[4] J. Choo, C. Lee, C. K. Reddy, and H. Park: UTOPIAN: User-Driven Topic Modeling on Interactive Nonnegative Matrix Factorization, *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, 2013

[5] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl: Constrained K-means Clustering with Background Knowledge, *18th International Conference on Machine Learning*, pp. 577-584, 2001

[6] A. Smith, V. Kumar, J. Boyd-Graber, K. Seppi, and L. Findlater: Closing the Loop: User-Centered Design and Evaluation of a Human-in-the-Loop Topic Modeling System, *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, pp. 293-304, 2018