小規模事例に基づく文書クラスタリング技法の実証比較：確率的モデルと非負行列分解とを中心に

Empirical comparison between document clustering techniques based on small test set: Focused on probabilistic models and non-negative matrix factorization

岸田和明（Kazuaki KISHIDA）
慶應義塾大学文学部図書館・情報学専攻
School of Library and Information Science, Keio University
〒 108-8345 東京都港区三田 2-15-45
2-15-45 Mita, Minato-ku, Tokyo 108-8345 JAPAN
kz_kishida@z8.keio.jp

平成 24 年 12 月 26 日

概 要

情報検索にとって重要な文書クラスタリングの技術は，テキストマイニングの研究が盛んになるにつれて高度化しつつある，特に，2000 年代の半ば以降，確率的なモデルに基づく方法と線形代数に依拠する手法とが飛躍的に発展している。本稿は，それらのうちの主な手法を技術的な観点から整理し，さらに，人工的に作成した小規模な文書データベースを使って，それらがどのようなクラスタ集合を生成するかを確認する。対象とした手法は，ベイズ推定に基づく多項混合モデル，von Mises-Fisher 分布による混合モデル，確率的潜在意味索引法（PLSI），Latent Dirichlet Allocation (LDA)，階層的 Dirichlet 過程（HDP）混合モデル，主成分分析，非負行列分解に基づく方法である。小規模データによる実験を通じて，それぞれの手法がそれなりに「妥当な」クラスタ集合を生成することが明らかになった。

Document clustering techniques playing an important role for information retrieval have been recently enhanced as a tool for text mining. Especially, methods based on probabilistic models and on matrix computation have been highly developed since about 2000. This paper tries to review them, and to examine empirically how cluster sets are generated by these methods through experiments using a small document set created artificially for the test. Specifically, multinomial mixture model, mixture model of von Mises-Fisher distributions, probabilistic latent semantic analysis (PLSA), latent Dirichlet allocation (LDA), hierarchical Dirichlet process (HDP) mixture model, principal component analysis (PCA) and non-negative matrix factorization (NMF) were investigated. For the experimental dataset, they generated valid sets of clusters to a certain degree, respectively.

キーワード : 文書クラスタリング，テキストマイニング，確率的混合モデル，潜在的トピックモデル，非負行列分解

keywords: document clustering, text mining, probabilistic mixture mode, latent topic model, non-negative matrix factorization

1　はじめに

N 件の文書から成る集合 $D = \{d_1, d_2, ..., d_N\}$ が与えられたとき，これをクラスタの集合 $\mathcal{C} = \{C_1, C_2, ..., C_L\}$ に分割することを**文書クラスタリング**（document clustering）と呼ぶ[1]。テキスト分類（text categorization）の場合には，既存の分類体系（例：NDC）の中の分類記号を各文書に割り当てるのに対して，文書クラスタリングでは，既存の分類体系を仮定しない。機械学習（machine learning）[35] の用語を使えば，文書クラスタリングは**教師なし**（unsupervised）の分類であって，テキスト分類とは異なり，訓練データ（training data）を必要としない。

情報検索およびその関連領域では，例えば，次のような場合に文書クラスタリングが必要になる。

- 大規模な文献データベースを効果的かつ効率的に検索するために，それを「同質」なグループに前もって分けておく（クラスタ型検索あるいは分散型検索と呼ばれる）。
- 曖昧な検索語（一般的な語や同姓同名など）に対する検索結果を提示する際,「同質」なグループに分割して表示する（クラスタ型検索エンジン，例：Vivisimo）。
- 一定期間のニュース記事から主要な話題や新しい話題を自動検出するために（topic detection と呼ばれる），ニュース記事を話題ごとにグループ化する。

このため，情報検索の領域では，文書クラスタリングの研究が長年に渡って積み重ねられてきた。その手法は岸田（2003）[33] によってレビューされている。

しかしながら，丁度このレビューが執筆された頃から，文書クラスタリングの技術は飛躍的に発展し，特に，確率的モデルに基づく手法と行列の理論（線形代数学）に基づく手法とが高度に発達した。そのひとつの要因は，**データマイニング**（data mining）あるいは**テキストマイニング**（text mining）の領域において，文書クラスタリングの問題が集中的に取り上げられるようになったためであり，これらの手法に関しては，岸田（2003）[33] の内容は大幅に陳腐化してしまった。

そこで本稿では，これらの手法を整理するとともに，その特徴を把握するために，$N = 10$ の小規模な人工的文書集合に対して各手法を実際に適用して，その結果を確認する。この実験用文書集合は「非現実的」であり，十分な評価を与えるものではないが，確率的モデルや行列理論に基づく手法はかなり複雑なので，それらの実際の「動き方」を直感的に理解するには都合がよい。なお，各手法を実行するために，階層的クラスタリングを除いて，それらのプログラムを Java で自作した。

2　文書クラスタリングの特徴とその手法の種類

2.1　文書クラスタリングの特徴

一般に，クラスタリングはさまざまな領域で活用されているが，それらと比較した時の文書クラスタリングの特徴はおおよそ次のようになる。

- 文書 d_i は M 次元ベクトル $\mathbf{d}_i = [w_{i1}, w_{i2}, ..., w_{iM}]^T$ で表現される。M は D 中の異なり語数，w_{ij} は文書 d_i での語 t_j の重みであり，通常，tf-idf で算出される（tf は語の出現頻度（term frequency），idf は出現文書数の逆数（inverse document frequency）を意味する）。

[1] $C_k \cap C_{k'} = \emptyset \quad (k \neq k')$ の場合には「排他的」，$C_k \cap C_{k'} \neq \emptyset \quad (k \neq k')$ は「非排他的」なクラスタリングである。

- 検索結果のクラスタリングなどを除けば，一般に，N と M は大きい（N はおおむね 1 万以上）。
- $N \times M$ 行列 $\mathbf{W} = [w_{ij}]$ $(i = 1, \ldots, N; j = 1, \ldots, M)$ は非常に疎（sparse）である。

このため，M 個の語をそのまま素性（feature）として使用するのは計算量の点で難しいため，何らかの素性選択（feature selection）を実施して，文書ベクトルの次元を減らすことがある。

2.2 文書クラスタリング技法の種類

上記の特徴のため，すべてのクラスタリング法が文書クラスタリングの問題に必ずしも適しているわけではない。文書クラスタリングでよく用いられる手法としては，1) 階層的クラスタリング，2)flat partitioning，3) 確率的モデルに基づく方法，4) 行列の分解に基づく方法などがある。

階層的クラスタリング（hierarchical clustering）では，クラスタが階層的に構成されるため，あたかも図書館での分類体系のように文書が組織化される。この点では，情報検索を応用目的とした場合には理想的な手法であるが，残念ながら，その実行のための計算量が膨大で，大規模文書集合には向かない[2]。

一方，*flat partitioning* では，階層的な構造を得ることはできず，D を単純にその部分集合に分割するのに留まるが（もちろん応用上，それで十分な場合も多い），その計算量は階層的クラスタリングよりも通常はるかに少ない。そのため，その代表である k-means 法や leader-follower 法は，文書クラスタリングに頻繁に適用されてきた。一般には，***k-means* 法**のほうがよく知られている。ここでの「k」はクラスタの個数を意味し（ただし，本稿ではクラスタ個数は L で表記する），*basic k-means* **法**では先験的に与えられたクラスタ個数 L に対して，次の手順が実行される [10]。

1) 各文書ベクトルとクラスタベクトルとの間の距離を計算し，最も近いクラスタに各文書を割り当てる。

2) 割り当ての結果に基づいて，クラスタベクトルを更新する。

通常，すべての文書に対してまず 1) の処理を行い，その後で，2) の更新を一括して実行する（'batch-mode' の k-means 法と呼ばれる）。クラスタベクトルとしては，通常，そのクラスタに属する文書のベクトルの平均

$$\mathbf{m}_k = \frac{1}{\tilde{n}_k} \sum_{i:d_i \in C_k} \mathbf{d}_i, \quad k = 1, \ldots, L \tag{1}$$

が用いられる。ここで $\tilde{n}_k$ はクラスタ C_k に含まれる文書数を意味し，$\mathbf{m}_k$ は**重心ベクトル**（centroid vector）とも呼ばれる。

この手順を実行するには，何らかの方法でクラスタベクトルを初期的に設定しなければならない。このためには，先頭から L 件の文書のベクトルを強制的にクラスタベクトルと見なす方法や無作為に初期クラスタベクトルを生成する方法などがある。いずれにせよ，クラスタリングの結果がその初期設定に大きく影響されてしまうので，上記の 1)2) の手順を，クラスタリングの結果が収束するまで反復的に繰り返す必要がある[3]。なお，上記 1) の手順では「距離」が計算されるが，文書クラスタリングの場合，文書の長さ（document length）の影響を除くため，通常，距離を計算する際にはベクトルは正規化される（例えば，$\tilde{\mathbf{d}}_i = \mathbf{d}_i / \|\mathbf{d}_i\|$）。正規化されたベクトルの長さは 1 であるため，**単位ベクトル**（unit vector）と呼ばれる。

[2] 文書のすべての組（ペア）の間での類似度を計算しなければならず，その組の数は $N(N-1)/2$ になる（すなわち $O(N^2)$）。例えば N が 1 万程度でも，組の総数は約 5000 万になる。文書間の類似度からクラスタを構成していく方法は凝集型（agglomerative）と呼ばれ，幅広く一般的に用いられているが，計算量をより少なく抑えた分割型の階層的クラスタリングを文書集合に適用した例 [29] もある。

[3] $N = 10000$ で 10 回の反復で収束したとすれば，延べ 10 万件の文書の処理で済む（すなわち r を反復回数とすれば $O(Nr)$）。なお，初期クラスタベクトルが異なる場合，反復計算をしたとしても，常に同一の結果に収束するとは限らず，場合によっては不適当なクラスタを生成することがある。この点で，できれば basic k-means 法ではなく，Hartigan-Wong アルゴリズム [12] などのよ

表 1: サンプル DB: $N = 10$

		語の出現頻度 f_{ij}									
語 t_j	n_j	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
t_1	3	4	1	1	0	0	0	0	0	0	0
t_2	6	1	3	6	0	2	1	1	0	0	0
t_3	4	0	0	0	4	2	1	5	0	0	0
t_4	4	0	0	0	3	0	0	0	2	1	3
t_5	6	0	0	0	0	1	5	1	3	5	1
t_6	2	0	0	0	0	0	0	1	0	0	4
文書長 l_i	-	5	4	7	7	5	7	8	5	6	8

注：n_j は語 t_j の出現文書数

ただし，文書クラスタリングの場合には，クラスタ個数 L が前もってわかっていないことも多い。この際には，k-means 法ではなく，*leader-follower* **法**が使用されることがある。この方法では，一般に，上記の手順 1) を「1') 文書を最も類似したクラスタに割り当てる。ただし，その類似度が閾値 θ を超えない場合には，当該文書を新規クラスタとして独立させる」に置き換える。もちろん，閾値 θ が常にうまく設定できるとは限らないが，この方法を使えば，文書集合内の状況に応じて，最終的にクラスタ個数が決まることになる。通常，leader-follower 法では反復計算はなされず，文書集合は 1 回ないし 2 回のみ走査される（先頭文書が自動的に最初のクラスタとなり，文書ごとに手順 1') と 2) を同時に実行し，それを最終文書まで続けていく）。1 回のみの場合には，**単一パスクラスタリング**（single-pass clustering）と呼ばれ，2 回走査する場合には 1 回目で最終的に生成されたクラスタベクトルに対して，2 回目に改めて文書を割り当てていく [16]。この方法はコンピュータの性能が低い時代に，大規模文書集合をクラスタリングするために，よく用いられた（文書のストリームをその到着順にクラスタリングする場合に，単一パスの方法が活用されることもある[4]）。

2.3 階層的クラスタリングと k-means 法の実行例

本稿で使用するサンプルデータベース（サンプル DB）を表 1 に示す。上で述べたように，これは人工的なデータにすぎないが，各クラスタリング法の実際を調べるには小さくて都合がよい。まず，統計ソフトウェア R-2.12.0 の hclust 関数を用いて，階層的クラスタリングを実行して得られたデンドログラム（dendrogram）を図 1 に示す（より具体的には，Ward 法を使用した）。図の左は表 1 の縦列をそれぞれ単位ベクトルに変換して用いた結果であり，右は正規化せずに tf をそのまま使った場合である[5]。

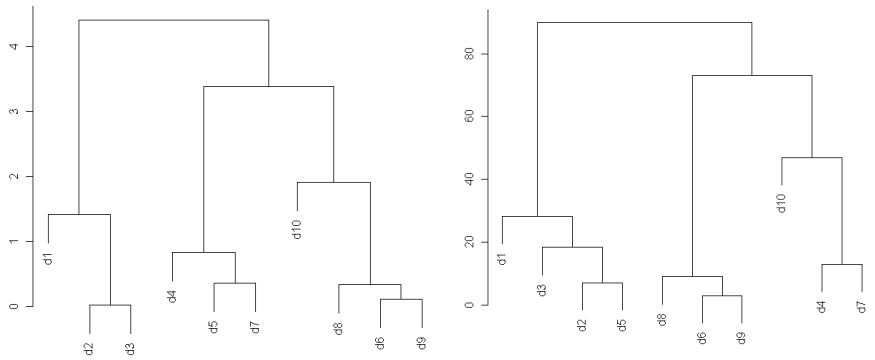
り優れた手法を使うべきである。このアルゴリズムでは，クラスタ内の密集の程度（residual sum of squares : RSS）

$$J = \sum_{k=1}^{L} \sum_{i:d_i \in C_k} \|\mathbf{d}_i - \mathbf{m}_k\|^2 \tag{2}$$

を目的関数とし，これを最小にするという基準に従ってクラスタが構成される。なお，$\|\mathbf{d}_i\| = \sqrt{\sum_j w_{ij}^2}$。

[4] この際，クラスタ個数が決まっていれば，k-means 法を反復なしで実行することも考えられる。この種のストリームのクラスタリングの研究は最近盛んになっている（例えば，文献 [1] などを参照）。

[5] この例では idf を加味していないので，$w_{ij} = \mathrm{f}_{ij}$ である（f_{ij} は文書 d_i における語 t_j の出現回数。表 1 参照）。通常は，例えば，$w_{ij} = \mathrm{f}_{ij} \times \log N/n_j$ などのように，tf-idf の重み付けにより文書ベクトルを構成する（n_j は語 t_j の出現文書数）。また，文書ベクトル間の類似度は余弦（cosine）係数で算出し，完全連結法あるいは群平均法を用いる場合が多い（計算量の関係で，一般に「性



(a) Unit vectors　　(b) Original tf vectors

図 1: サンプル DB に対する階層的クラスタリングの結果 (R-2.12.0 による)

表 2: Hartigan-Wong アルゴリズムによる k-means 法の結果

RSS J	#	%	クラスタ集合（$L = 3$）
2.485	106	88.3%	$\{d_1, d_2, d_3\}, \{d_4, d_5, d_7\}, \{d_6, d_8, d_9, d_{10}\}$
3.383	14	11.7%	$\{d_1\}, \{d_2, d_3, d_5\}, \{d_4, d_6, d_7, d_8, d_9, d_{10}\}$
合計	120	100%	

注: # は seeds のパターンの数を示す。

表 1 と図 1 左を見比べれば明らかなように，単位ベクトルを使用した場合には「妥当」な結果が得られている。すなわち，デンドログラムの縦軸の「結合のレベル」に適当な閾値を設定すれば，$L = 3$ の場合，

$$\mathcal{C}_v = \{\{d_1, d_2, d_3\}, \{d_4, d_5, d_7\}, \{d_6, d_8, d_9, d_{10}\}\} \tag{3}$$

というクラスタの集合が生成されたことになる。実際的な例ではないので，これが必ずしも「妥当」とは断定できないが，本稿では Ward 法によって得られたこの $\mathcal{C}_v$ を，$L = 3$ の場合の「妥当」な結果として扱う（単位ベクトルを入力データとしているため，単位ベクトルに基づかない手法の結果との比較には若干の注意が必要である）。

次に $L = 3$ の場合の Hartigan-Wong アルゴリズム [12] による k-means 法の結果を表 2 に示す。3 件の文書をあらかじめ選択し，それらの文書ベクトルをそれぞれのクラスタの初期ベクトルとした場合（このような文書は seed と呼ばれる），3 件の文書の組み合わせの数は ${}_{10}C_3 = 120$ 通りである。そこでこの 120 パターンに対して，それぞれクラスタリングを繰り返した結果を表 2 に掲げている。この結果から，サンプル DB においては，無作為に seeds を選んだ場合，Hartigan-Wong アルゴリズムは，約 88% の確率で「妥当」なクラスタ集合を生成することがわかる[6]。

能が悪い」とされている単連結法を用いる場合もある）。

[6] ここでは詳細な結果は示さないが，basic k-means 法の場合，$\mathcal{C}_v$ を生成した seeds のパターンは 120 通りのうちわずか 29（約 24%）であった。これは Hartigan-Wong アルゴリズムが優れていることの例証である。なお, ここでの計算はすべて d_1 から d_{10} の順序で行った。この処理順序が変われば，結果は異なってくる。

3　確率的モデルに基づく文書クラスタリング

この節では，おおよそ 2000 年以降に開発された確率的モデルに基づくクラスタリング手法のうち，代表的なものを概観し，サンプル DB でのその実行結果を確認する。

3.1　確率分布の混合モデルと EM アルゴリズム

確率分布の**混合モデル**（mixture model）は，文書クラスタリングに応用される以前にもさまざまな分野で活用されてきた [23]。文書クラスタリングの問題にこれを応用する場合，混合モデルは一般に

$$P(\mathbf{d}_i) = \eta_1 P(\mathbf{d}_i|C_1) + \eta_2 P(\mathbf{d}_i|C_2) + \ldots + \eta_L P(\mathbf{d}_i|C_L) = \sum_{k=1}^{L} \eta_k P(\mathbf{d}_i|C_k) \tag{4}$$

と表記される。ここで η_k は混合係数であり，文書ベクトル $\mathbf{d}_i$ は，複数のクラスタにおける確率分布 $P(\mathbf{d}_i|C_k)$ の「混合」として生成されると仮定する。この意味で，(4) 式は一種の確率的な**生成モデル**（generative model）である。$P(\mathbf{d}_i|C_k)$ をポアソン分布とした場合の混合の様子を図 2 に示す[7]。この図は 1 つの語のみに関する分布を示しているが，この例から分かるように，$P(\mathbf{d}_i|C_k)$ は k 番目のクラスタに限定した文書ベクトルの確率分布である。実際の文書クラスタリングでは，$P(\mathbf{d}_i|C_k)$ として多項分布や von Mises-Fisher 分布を使うことが多い（詳細は後述）。

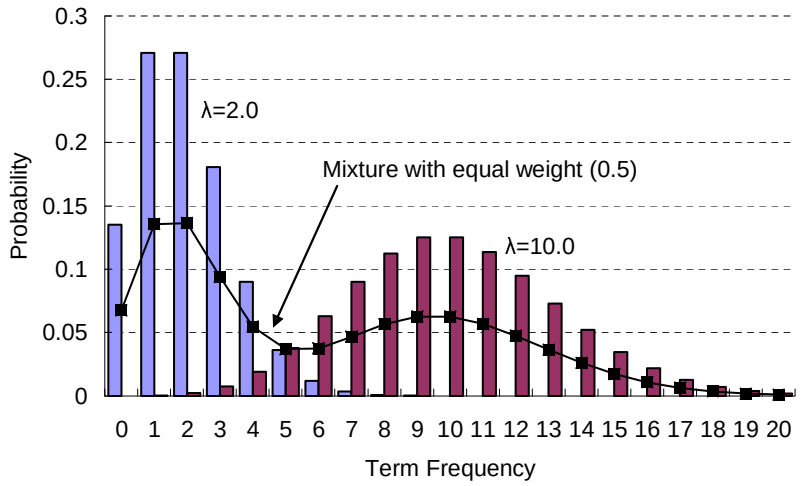


図 2: 2 つのポアソン分布の混合（$\eta_1 = \eta_2 = 0.5$）

これらの確率分布にはいくつかのパラメータが含まれる。混合係数も含めて，これらをデータ（文書クラスタリングの状況では文書集合 D）から推計しなければならない。パラメータを並べたベクトルを $\mathbf{\Psi}$ と表記する。データからパラメータを推計する場合，一般に，最尤推定法が使用される。これはパラメータ $\mathbf{\Psi}$ に基づくデータ $\mathbf{y}$ の確率分布 $P(\mathbf{y};\mathbf{\Psi})$ を，逆に，データが与えられたとき（$\mathbf{y}$ が固定されたとき）の $\mathbf{\Psi}$ の

[7]ポアソン分布は 1960 年代の自動索引法 (automatic indexing) の研究では用いられたが，現在ではほとんど利用されていない。ここでは描画の都合上，ポアソン分布を例示した。図中の λ はポアソン分布のパラメータで，後で出てくる Lagrange の未定係数とは別物である。

確率分布として捉え直し，その確率を最大とする $\mathbf{\Psi}$ を推定量として採用する方法である。この種の推定量を**最尤推定量**（maximum likelihood estimate）と呼ぶ。

文書クラスタリングの場合，$\mathbf{y}$ は，D に含まれるすべての文書中の各語の tf の値を並べたベクトルに相当する。したがって文書間の確率的な独立性を仮定すれば，

$$P(\mathbf{y};\mathbf{\Psi}) = \prod_{i=1}^{N} P(\mathbf{d}_i) = \prod_{i=1}^{N}\sum_{k=1}^{L} \eta_k P(\mathbf{d}_i|C_k) \tag{5}$$

であるから，これを $\mathbf{\Psi}$ に含まれる各パラメータで微分し，それを 0 と置いた方程式を解くことによって，最尤推定量が計算できるはずである。この場合，$P(\mathbf{y};\mathbf{\Psi})$ を**尤度関数**（likelihood function）と呼び，本稿では，$\mathcal{L}(\mathbf{\Psi})$ と表記する。しかし残念ながら，(5) 式は複雑なため，一般にはそのまま微分できない。そこで，通常，*EM* **アルゴリズム**が利用される。

まず,「仮想的」なデータ z_{ki} を導入する（$k = 1,\ldots,L;\ i = 1,\ldots,N$）。これは，文書 d_i が k 番目の混合要素（すなわちクラスタ C_k）から生成されたかどうかを示す 2 値変数で，生成された場合に 1，そうでない場合に 0 となる（LN 個の z_{ki} を並べたベクトルを $\mathbf{z}$ と定義する）。仮に，$\mathbf{z}$ が「観測された」と仮定すると，$\mathbf{x} = [\mathbf{y}^T, \mathbf{z}^T]^T$ に対する確率分布 $P(\mathbf{x};\mathbf{\Psi})$ の対数変換は，$\mathcal{L}_c(\mathbf{\Psi}) \equiv P(\mathbf{x};\mathbf{\Psi})$ と定義して，

$$\log \mathcal{L}_c(\mathbf{\Psi}) = \sum_{k=1}^{L}\sum_{i=1}^{N} z_{ki} \log \eta_k + \sum_{k=1}^{L}\sum_{i=1}^{N} z_{ki} \log P(\mathbf{d}_i|C_k) \tag{6}$$

のように書き下すことができる。ここで，$\mathbf{x}$ を「完全データ」とするならば，$\mathbf{y}$ は不完全データ（incomplete data）である。

完全データの対数尤度 (6) 式ならば，容易に微分でき，その結果，$\mathbf{\Psi}$ の最尤推定量を求めることが可能である。これが EM アルゴリズムの M ステップ（Maximization step）に相当する。ただし，実際には，z_{ki} は観測されない欠損データ（missing data）なので，(6) 式そのものではなく，その期待値（Expectation）を計算して，それを最大化する必要がある。この期待値を計算する段階が E ステップ（Expectation step）であり，E ステップと M ステップを交互に繰り返し反復実行することによって最終的に $\mathbf{\Psi}$ の最尤推定量を数値的に計算する方法を EM アルゴリズムと呼ぶ。

ここでは s 回目の反復計算における $\mathbf{\Psi}$ の推定値を $\mathbf{\Psi}^{(s)}$ と表記する。具体的には，E ステップでは期待値 $\mathcal{Q}(\mathbf{\Psi};\mathbf{\Psi}^{(s)}) \equiv E_{\mathbf{\Psi}^{(s)}}\{\log \mathcal{L}_c(\mathbf{\Psi})|\mathbf{y}\}$ を算出し，次に M ステップで，あらゆる $\mathbf{\Psi}$ に対して $\mathcal{Q}(\mathbf{\Psi}^{(s+1)};\mathbf{\Psi}^{(s)}) \geq \mathcal{Q}(\mathbf{\Psi};\mathbf{\Psi}^{(s)})$ となる $\mathbf{\Psi}^{(s+1)}$ を求めることになる（それが再び E ステップに投入される）。$E_{\mathbf{\Psi}^{(s)}}\{\log \mathcal{L}_c(\mathbf{\Psi})|\mathbf{y}\}$ の計算は通常，複雑であるが[8]，幸い，(6) 式は z_{ki} についての線形関数なので，z_{ki} に対応する期待値をそれぞれ計算して，それを z_{ki} の代わりとしてそのまま使えばよい。そこでその期待値を単に $z_{ki}^{(s)}$ と表記すれば，z_{ki} が 2 値変数であるため，簡単な計算により，

$$z_{ki}^{(s)} = \frac{\eta_k^{(s)} P(\mathbf{d}_i|C_k;\mathbf{\Psi}^{(s)})}{P(\mathbf{d}_i;\mathbf{\Psi}^{(s)})} = \frac{\eta_k^{(s)} P(\mathbf{d}_i|C_k;\mathbf{\Psi}^{(s)})}{\sum_{k'=1}^{L} \eta_{k'}^{(s)} P(\mathbf{d}_i|C_{k'};\mathbf{\Psi}^{(s)})} \tag{7}$$

となる（詳細は文献 [22] の p.13-18 を参照）。

この E ステップの計算結果に基づき，$z_{ki}^{(s)}$ を (6) 式の z_{ki} の代わりに用いれば，M ステップでの $\mathbf{\Psi}^{(s)}$ の更新を実行できる。例えば，混合係数 η_k の場合にはすべて合計すれば 1 という制約があるので，Lagrange の未定乗数 λ を使って，$\mathcal{H} = \log \mathcal{L}_c(\mathbf{\Psi}) - \lambda\left(\sum_k \eta_k - 1\right)$ を微分して 0 と置いたものを解くことになる。

[8] 正確には，$E_{\mathbf{\Psi}^{(s)}}\{\log \mathcal{L}_c(\mathbf{\Psi})|\mathbf{y}\} = \sum_{\mathbf{Z}} \log \mathcal{L}_c(\mathbf{\Psi}) P(\mathbf{Z}|\mathbf{y};\mathbf{\Psi}^{(s)})$ として計算する。ここで $\mathbf{Z}$ はデータ $\mathbf{z}$ に対応した離散確率変数で，また，$\sum_{\mathbf{Z}}$ はそれが取りうるすべての値に対しての合計を意味する。

$\partial\mathcal{H}/\partial\eta_k = \sum_i z_{ki}/\eta_k - \lambda$ なので，$\partial\mathcal{H}/\partial\eta_k = 0$ から $\eta_k = \lambda^{-1}\sum_i z_{ki}$ を得る $(k = 1, \ldots, L)$。z_{ki} の定義により，$\sum_k \sum_i z_{ki} = N$ であるため，最終的に M ステップにおける更新式は，以下のようになる。

$$\eta_k^{(s+1)} = \frac{1}{N}\sum_{i=1}^{N} z_{ki}^{(s)}, \quad k = 1, \ldots, L \tag{8}$$

なお文書クラスタリングの場合，$\mathbf{\Psi}^{(s)}$ が収束した後に，各 z_{ki} の値を使えば，各クラスタへの文書の割り当てが可能となる。つまり，クラスタを次のように構成できる。

$$C_k = \{d_i | \arg\max_{k'} z_{k'i} = k\}, \qquad k = 1, \ldots, L \tag{9}$$

3.2 ベイズ推定に基づく多項混合モデル

具体的に $P(\mathbf{d}_i|C_k)$ として何らかの確率分布を設定し，混合係数の場合と同様に，完全データの対数尤度をそのパラメータで微分することにより，M ステップの更新式を求めることができる。この確率分布として，ここでは教師付きのテキスト分類でもよく用いられる**多項分布**（multinomial distribution）

$$P(\mathbf{d}_i|C_k) = \mathcal{A}_i \prod_{j=1}^{M} p_{j|k}^{\mathrm{f}_{ij}} \propto \prod_{j=1}^{M} p_{j|k}^{\mathrm{f}_{ij}} \tag{10}$$

を仮定する [25, 26]。ここで $p_{j|k}$ はパラメータ，$\mathcal{A}_i$ は多項係数である（多項係数はデータから計算される定数であることに注意）。$p_{j|k}$ はクラスタ C_k において語 t_j が使用される確率に相当する（$j = 1, \ldots, M$; $k = 1, \ldots, L$）。

この式に基づいて，完全データの対数尤度から直接 $p_{j|k}$ の更新式を求めることももちろん可能であるが，ここでは，多項分布がディリクレ分布の共役（conjugate）であることを利用して，**ベイズ推定**（Bayesian inference）に基づく**多項混合モデル**（multinomial mixture model）を考える。この場合には，パラメータ $p_{j|k}$ 自体が確率変数となり，その**ディリクレ分布**（Dirichlet distribution）は，

$$P(\mathbf{p}_k|\boldsymbol{\beta}) = \frac{\Gamma(\sum_{j=1}^{M}\beta_j)}{\prod_{j=1}^{M}\Gamma(\beta_j)} \prod_{j=1}^{M} p_{j|k}^{\beta_j - 1} \propto \prod_{j=1}^{M} p_{j|k}^{\beta_j - 1} \tag{11}$$

である。ここで，$\mathbf{p}_k$ は特定の k についての $p_{j|k}$ を M 個並べたベクトルであり，$\boldsymbol{\beta} = [\beta_1, \ldots, \beta_M]^T$ はディリクレ分布のパラメータである[9]（$\Gamma(\cdot)$ はガンマ関数）。同様に，混合係数 $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_L]^T$ についても，$\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_L]^T$ をパラメータとするディリクレ分布を仮定する。

$\boldsymbol{\eta}, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_L$ の間の独立性を仮定すれば，ベイズの定理より，$P(\mathbf{\Psi}|\mathbf{y}) = P(\boldsymbol{\eta}, \boldsymbol{p}|\mathbf{y}) = P(\mathbf{y}|\boldsymbol{\eta}, \boldsymbol{p})P(\boldsymbol{\eta})P(\boldsymbol{p})/P(\mathbf{y})$ であるから（$\boldsymbol{p} = [\boldsymbol{p}_1^T, \ldots, \boldsymbol{p}_L^T]^T$），

$$P(\mathbf{\Psi}|\mathbf{y}) \propto \prod_{i=1}^{N}\left[\sum_{k=1}^{L}\left(\eta_k \prod_{j=1}^{M} p_{j|k}^{\mathrm{f}_{ij}}\right)\right] \times \prod_{k=1}^{L}\eta_k^{\alpha_k - 1} \times \prod_{k=1}^{L}\prod_{j=1}^{M} p_{j|k}^{\beta_j - 1} \tag{12}$$

となる。これはパラメータの事前分布 $P(\boldsymbol{\eta})$，$P(\boldsymbol{p})$ に対して，観測データ $\mathbf{y}$ が与えられた後の事後分布であるが，これを最大化すれば最尤推定量を計算できる[10]。そこで，不完全データ $\mathbf{z}$ を導入して，その対数を

[9] パラメータ $p_{j|k}$ が従う分布のパラメータであり，**ハイパーパラメータ**（hyperparameter）と呼ばれる。
[10] 事後分布（posteriori) を最大にするので，ベイズ推定の枠組みでは，MAP（maximum a posteriori）推定量である。

とれば，

$$\log \mathcal{L}_c(\mathbf{\Psi}) = \sum_{k=1}^{L}\sum_{i=1}^{N} z_{ki} \log \eta_k + \sum_{k=1}^{L}\sum_{i=1}^{N} z_{ki} \log \prod_{j=1}^{M} p_{j|k}^{\mathrm{f}_{ij}} + \sum_{k=1}^{L} (\alpha_k - 1) \log \eta_k + \sum_{k=1}^{L}\sum_{j=1}^{M} (\beta_j - 1) \log p_{j|k} \quad (13)$$

となり，これならば微分が可能である。実際，上と同様に，η_k および $p_{j|k}$ に関する制約条件[11]を設定して，$\partial\mathcal{H}/\partial\eta_k = 0$ と $\partial\mathcal{H}/\partial p_{j|k} = 0$ をそれぞれ解けば，

$$\eta_k^{(s+1)} = \frac{\alpha_k - 1 + \sum_i z_{ki}^{(s)}}{\sum_{k'} \left(\alpha_{k'} - 1 + \sum_i z_{k'i}^{(s)} \right)} \quad (14)$$

$$p_{j|k}^{(s+1)} = \frac{\beta_j - 1 + \sum_i \mathrm{f}_{ij} z_{ki}^{(s)}}{\sum_{j'} \left(\beta_{j'} - 1 + \sum_i \mathrm{f}_{ij'} z_{ki}^{(s)} \right)} \quad (15)$$

を得る。この式から明らかなようにハイパーパラメータの導入により，一種の**平滑化**（smoothing）が組み込まれたことになる[12]。また，すべてのハイパーパラメータが 1 の場合には，標準的な（ベイズ推定でない）多項混合モデルに帰着する。

この結果，ベイズ推定に基づく多項混合モデルによる文書クラスタリングの手順は次のようになる。

(0) クラスタ個数 L およびハイパーパラメータ α_k，β_j をあらかじめ決めておく（$k = 1, \ldots, L$; $j = 1, \ldots, M$）。

(1) 一様乱数を発生させ，η_k および $p_{j|k}$ の初期値を設定する（$k = 1, \ldots, L$; $j = 1, \ldots, M$）。

(2)[E ステップ](7) 式と (10) 式で z_{ki} を算出する（$k = 1, \ldots, L$; $i = 1, \ldots, N$）。

(3)[M ステップ](14) 式と (15) 式でパラメータをそれぞれ更新する（$k = 1, \ldots, L$; $j = 1, \ldots, M$）。

(4) 推計値が収束すれば，(9) 式によりクラスタを構成し，終了する。そうでなければ (2) に戻る。

このアルゴリズムをサンプル DB に適用した結果を表 3 に示す。初期値によっては，局所的な最大値（local maximum）に収束することがあるため，初期値を変えて，1000 回ずつ実行した。なお，ハイパーパラメータはすべての k および j で同一の値を使っている（すなわち，対称的なディリクレ分布を仮定）。表から明らかなように，$\alpha_k = \beta_j = 1.0$ の標準的な多項混合モデルでは数多くの局所最大値が出現してしまう。それに対して，$\alpha_k = \beta_j = 6.0$ の場合には，平滑化の効果により，今回の場合には 1000 回すべて大域的な最大値に収束した。この $\alpha_k = \beta_j = 6.0$ の場合には，常に「妥当」なクラスタ集合 $\mathcal{C}_v$ が得られており，この点で k-means 法と比べれば，今回の実験では，ベイズ推定に基づく多項混合モデルのほうが優れていたことになる。

3.3 von Mises-Fisher 分布による混合モデル

確率的な混合モデルにおける構成要素として，文書クラスタリングでは，*von Mises-Fisher* **分布**（vMF 分布）

$$P(\tilde{\mathbf{d}}_i | C_k) = c_M(\kappa_k) \exp[\kappa_k \boldsymbol{\mu}_k^T \tilde{\mathbf{d}}_i] = \frac{\kappa_k^{M/2-1}}{(2\pi)^{M/2} \mathcal{B}_{M/2-1}(\kappa_k)} \exp[\kappa_k \boldsymbol{\mu}_k^T \tilde{\mathbf{d}}_i] \quad (16)$$

もよく利用される [4]。ここで $\boldsymbol{\mu}_k$ はクラスタベクトル（ただし $\boldsymbol{\mu}_k^T \boldsymbol{\mu}_k = 1$），$\kappa_k$ は「concentration parameter」であり（$k = 1, \ldots, L$），$\mathcal{B}_{M/2-1}(\kappa_k)$ は $(M/2-1)$ 次の第 1 種変形ベッセル関数を意味する。この分布は

[11] 特定の k について $\sum_j p_{j|k} = 1$。

[12] すなわち，通常の多項混合モデルに「ラプラス型」の平滑化を組み込んだことと形式的には同等である。

表 3: ベイズ推定に基づく多項混合モデルの実験結果 ($L = 3$ での 1000 試行)

	$\alpha_k = 1.0$		$\alpha_k = 2.0$		$\alpha_k = 4.0$		$\alpha_k = 6.0$	
	$\beta_j = 1.0$		$\beta_j = 2.0$		$\beta_j = 4.0$		$\beta_j = 6.0$	
	$\log \mathcal{L}$	#	$\log \mathcal{L}$	#	$\log \mathcal{L}$	#	$\log \mathcal{L}$	#
1	-46.8	68	-52.9	442	-60.8	482	-66.1	1000
2	-47.2	266	-53.4	357	-61.3	331		
3	-47.3	153	-54.2	185	-62.4	187		
4	-47.6	106	-60.9	8				
5	-48.5	102	-63.0	8				
6	-48.8	32						
7	-49.7	3						
8	-50.7	18						
9	-51.6	67						
10	-51.8	23						
11	その他	162						
合計	-	1000	-	1000	-	1000	-	1000

注: 「その他」には 23 の局所最大値が含まれる。

directional statistics[19] の分野で重要な役割を果たしているが，この分布の基本部分が内積の計算であることから（すなわち $\boldsymbol{\mu}_k^T \tilde{\mathbf{d}}_i$），文書クラスタリングに応用しやすい。例えば，2 つの文書ベクトル間の距離を求めるにはどちらかのベクトルに出現する語すべてに対する計算が必要であるが，内積は，共通する語のみで算出できる。そのためプログラムの工夫次第では，内積を計算したほうが効率的である[13]。

パラメータ推計のための EM アルゴリズムでは，まず，E ステップにおいては，当然，(7) 式の $P(\mathbf{d}_i|C_k)$ として (16) 式を使う。次に M ステップでは，η_k の更新には (8) 式を使い，$\boldsymbol{\mu}_k$ についてはその制約条件を組み込んだ対数尤度

$$\mathcal{H} = \sum_{k=1}^{L} \left[\sum_{i=1}^{N} z_{ki} \log c_M(\kappa_k) + \sum_{i=1}^{N} z_{ki} \kappa_k \boldsymbol{\mu}_k^T \tilde{\mathbf{d}}_i + \lambda_k (1 - \boldsymbol{\mu}_k^T \boldsymbol{\mu}_k) \right] \tag{17}$$

を微分することにより，$\boldsymbol{\mu}_k = \sum_i z_{ki} \tilde{\mathbf{d}}_i / \| \sum_i z_{ki} \tilde{\mathbf{d}}_i \|$ を得るので，これに基づいて更新する（λ_k はクラスタ C_k のための Lagrange 乗数）。また，κ_k に関しては，$A = \| \sum_i z_{ki} \tilde{\mathbf{d}}_i \| / \sum_i z_{ki}$ から $\hat{\kappa}_k = (AM - A^3)/(1 - A^2)$ として推計する [3]。この EM アルゴリズムによるクラスタリングの結果を表 4 に示す。ここでもまた，パラメータの初期値をそれぞれ無作為に決めて 1000 回の試行を繰り返したところ，「妥当」なクラスタ集合 $\mathcal{C}_v$ が構成された試行は，そのうちの 26% に留まったものの，それとほぼ同じ集合である $\{\{d_1, d_2, d_3\}, \{d_4, d_5, d_7, d_{10}\}, \{d_6, d_8, d_9\}\}$ が約 64% の実行で得られているので，今回の実験では，それらを併せれば約 90% の確率でほぼ良好な結果が得られたと判断できる。

[13] このため，basic k-means 法においても，単位ベクトルに対する内積を使うことがあり，*spherical k-means* **法**と呼ばれる [9]。なおこの場合，重心ベクトルも正規化する。

表 4: vMF 分布に基づく混合モデルでのクラスタリングの実験結果（1000 試行）

	試行数	クラスタ集合 ($L=3$)
1	637	$\{d_1,d_2,d_3\}\{d_4,d_5,d_7,d_{10}\}\{d_6,d_8,d_9\}$
2	260	$\{d_1,d_2,d_3\}\{d_4,d_5,d_7\}\{d_6,d_8,d_9,d_{10}\}$
3	39	$\{d_1,d_2,d_3,d_5\}\{d_4,d_7\}\{d_6,d_8,d_9,d_{10}\}$
4	23	$\{d_1,d_2,d_3,d_{10}\}\{d_4,d_5,d_7\}\{d_6,d_8,d_9\}$
5	23	$\{d_1,d_2,d_3,d_5,d_{10}\}\{d_4,d_7\}\{d_6,d_8,d_9\}$
6	15	$\{d_1,d_2,d_3\}\{d_4,d_6,d_8,d_9,d_{10}\}\{d_5,d_7\}$
7	1	$\{d_1,d_2,d_3,d_4,d_5,d_7\}\{d_6,d_9\}\{d_8,d_{10}\}$
8	1	$\{d_1,d_2,d_3,d_4,d_{10}\}\{d_5,d_7\}\{d_6,d_8,d_9\}$
9	1	$\{d_1,d_{10}\}\{d_2,d_3,d_4,d_5,d_7\}\{d_6,d_8,d_9\}$
合計	1000	

3.4 PLSA

文書クラスタリングに適用可能な，混合モデル以外の確率的モデルとして，Hofmann（1999）[14] により考案された *probabilistic latent semantic analysis*（PLSA）がある[14]。このモデルでは，L 個の潜在的なトピック $\tau_1,\ldots,\tau_L$ を設定し，文書 d_i において語 t_j が出現する確率を

$$P(t_j|d_i)=\sum_{k=1}^{L}P(t_j|\tau_k)P(\tau_k|d_i)=\frac{1}{P(d_i)}\sum_{k=1}^{L}P(t_j|\tau_k)P(d_i|\tau_k)P(\tau_k) \tag{18}$$

と仮定する。例えば, ある特定の τ_k に対して，$P(\text{“family”}|\tau_k)$, $P(\text{“home”}|\tau_k)$，$P(\text{“kid”}|\tau_k)$ の値が大きければ，τ_k はトピック「家族・家庭」についての潜在クラスと考えることができる。ここで，文書集合 D から $P(\tau_k|d_i)$ の値を推計すれば，各文書をその値が最も大きなトピック τ_k に割り当てることにより，文書クラスタリングを実行できる（τ_k を 1 つのクラスタと見なす)。多項係数を省略すれば，$\mathcal{L}(\mathbf{\Psi})=\prod_i P(\mathbf{d}_i;\mathbf{\Psi})\propto\prod_i\prod_j P(t_j|d_i)^{\mathrm{f}_{ij}}$ であるから，これに (18) 式を代入すれば，パラメータ推計のための対数尤度は,

$$\log\mathcal{L}(\mathbf{\Psi})\propto\sum_{i=1}^{N}\sum_{j=1}^{M}\mathrm{f}_{ij}\log\sum_{k=1}^{L}P(t_j|\tau_k)P(\tau_k|d_i) \tag{19}$$

となる。

パラメータ $\mathbf{\Psi}=[P(t_1|\tau_1),\ldots,P(t_M|\tau_L),P(\tau_1|d_1),\ldots,P(\tau_L|d_N)]^T$ は EM アルゴリズムによって推計できる[15]。初期値を変えて, サンプル DB に対して 1000 回試行した結果を表 5 に示す。この表では，確率の値が同じだったため，1 件の文書が 2 つのクラスタに同時に属している場合がある。表の上位では「妥当」なクラスタ集合がそれなりに得られていることは明らかであるが，標準的な多項混合モデルと同様に，平

[14]PLSI (probabilistic latent semantic indexing) とも呼ばれる。

[15]観測されないデータ $z_{k|ij}$ を導入する（語 t_j と文書 d_i の組がトピック τ_k に関連付けられていれば 1，そうでなければ 0）。Hofmann(1999)[14] とはやや異なるが，完全データの対数尤度を $\log\mathcal{L}_c(\mathbf{\Psi})=\sum_j\sum_i\mathrm{f}_{ij}\sum_k z_{k|ij}\log[P(t_j|\tau_k)P(\tau_k|d_i)]$ とすれば，多項混合モデルの場合と同様の手順から，以下のように計算できる。

$$z^{(s)}_{k|ij}=\frac{P^{(s)}(t_j|\tau_k)P^{(s)}(\tau_k|d_i)}{\sum_{k=1}^{L}P^{(s)}(t_j|\tau_k)P^{(s)}(\tau_k|d_i)},\quad P^{(s+1)}(\tau_k|d_i)=\frac{\sum_{j=1}^{M}\mathrm{f}_{ij}z^{(s)}_{k|ij}}{\sum_{k'=1}^{L}\sum_{j=1}^{M}\mathrm{f}_{ij}z^{(s)}_{k'|ij}},\quad P^{(s+1)}(t_j|\tau_k)=\frac{\sum_{i=1}^{N}\mathrm{f}_{ij}z^{(s)}_{k|ij}}{\sum_{j'=1}^{M}\sum_{i=1}^{N}\mathrm{f}_{ij'}z^{(s)}_{k|ij'}}$$

表 5: PLSA でのクラスタリングの実験結果（1000 試行）

$\log \mathcal{L}$	#	クラスタ集合
-32.26	331	$\{d_1, d_2, d_3, d_5\}\{d_4, d_5, d_7, d_{10}\}\{d_6, d_8, d_9\}$
-32.42	247	$\{d_1, d_2, d_3, d_5\}\{d_4, d_5, d_7\}\{d_6, d_8, d_9, d_{10}\}$
-32.68	21	$\{d_1, d_2, d_3, d_5\}\{d_4, d_7, d_{10}\}\{d_6, d_8, d_9\}$
-33.41	86	$\{d_1, d_2, d_3\}\{d_4, d_5, d_7\}\{d_6, d_8, d_9, d_{10}\}$
-34.00	33	$\{d_1, d_2, d_3\}\{d_4, d_5, d_6, d_7\}\{d_8, d_9, d_{10}\}$
-34.81	53	$\{d_1, d_2, d_3\}\{d_4, d_5, d_6, d_7, d_9\}\{d_8, d_{10}\}$
-35.74	51	$\{d_1, d_2, d_3\}\{d_4, d_5, d_6, d_7, d_8, d_9\}\{d_{10}\}$
-37.77	57	$\{d_1, d_2, d_3, d_4, d_5, d_7\}\{d_6, d_8, d_9\}\{d_{10}\}$
-38.09	16	$\{d_1, d_2, d_3, d_{10}\}\{d_4, d_5, d_7\}\{d_6, d_8, d_9, d_{10}\}$
-41.00	10	$\{d_1, d_{10}\}\{d_2, d_3, d_4, d_5, d_7\}\{d_6, d_8, d_9, d_{10}\}$
その他	95	-
Total	1000	

注：‘その他’ には出現回数 10 回未満の 40 の局所最大値を含む。

滑化がなされていないため，最適ではない局所最大値が数多く出現してしまうことがわかる。このため潜在的なトピックをデータから推計する場合には，現在では，ベイズ推定に基づく LDA（次節参照）を利用することが多い。

3.5 LDA と Gibbs サンプリング

Blei, Ng & Jordan (2003) [5] により考案された *Latent Dirichlet Allocation* (LDA) は，ベイズ推定の枠組みでの確率的な文書生成モデルである。例えば，文書 d_4 において先頭に出現する語（トークン）を w_{41} と表記すれば，これは次のように生成されると仮定する。

$$d_4 \rightharpoonup \boldsymbol{\theta}_4 \to \pi(\cdot|\boldsymbol{\theta}_4) \mapsto \tau_2 \rightharpoonup \boldsymbol{\phi}_2 \to \pi(\cdot|\boldsymbol{\phi}_2) \mapsto t_3 \to \mathrm{w}_{41}$$

まず，文書 d_4 に対する確率分布 $\boldsymbol{\theta}_4$ がディリクレ分布から選択される。ここで，$\boldsymbol{\theta}_i = [\theta_{i1}, \ldots, \theta_{iL}]^T$ は文書 d_i に k 番目のトピックが割り当てられる確率分布に相当する。次に，この $\boldsymbol{\theta}_4$ から 1 つのトピックが無作為抽出される。上では，これを「$\pi(\cdot|\boldsymbol{\theta}_4) \mapsto \tau_2$」と表記しており，この例では，2 番目のトピック τ_2 が選ばれたことになる。この τ_2 に対して，ディリクレ分布から選択された確率分布を $\boldsymbol{\phi}_2$ と書く。ここで $\boldsymbol{\phi}_k = [P(t_1|\tau_k), \ldots, P(t_M|\tau_k)]^T$ である。この $\boldsymbol{\phi}_2$ からの無作為抽出により t_3 が選択され（$\pi(\cdot|\boldsymbol{\phi}_2) \mapsto t_3$），この語が d_4 のトークン w_{41} として現れることになる[16]。この操作を文書の長さ分繰り返すことによって，文書 d_i が $\mathbf{w}_i = [\mathrm{w}_{i1}, \ldots, \mathrm{w}_{il_i}]^T = [t_{\omega(1|i)}, \ldots, t_{\omega(l_i|i)}]^T$ として生成されると考えるわけである。ここで，l_i は文書 d_i の長さ（すなわちトークンの総数）を表し，$\omega(h|i)$ は d_i における h 番目の位置に出現している語の番号を返す関数とする（$h = 1, \ldots, l_i$）。例えば，先頭から 8 番目のトークンとして t_4 が出現しているならば，$\omega(8|i) = 4$ である。

[16]語 t_j は文書中で複数出現するので，例えば，$\mathrm{w}_{41} = t_3$ かつ $\mathrm{w}_{48} = t_3$ のような場合もあり得る。LDA では，この例が示すようにトークン（token）に基づいて文書の生成がモデル化される。

w_{ih} に対する操作 $\pi(\cdot|\boldsymbol{\theta}_i) \mapsto \tau_k$ の結果を $\tilde{z}_{ih} = k$ と書くことにすれば，PLSA における (18) 式は，

$$P(\mathrm{w}_{ih}|\boldsymbol{\phi}, \boldsymbol{\theta}_i) = \sum_{k=1}^{L} P(\mathrm{w}_{ih}|\tilde{z}_{ih} = k, \boldsymbol{\phi}) P(\tilde{z}_{ih} = k|\boldsymbol{\theta}_i) \tag{20}$$

となる。ここで，$\boldsymbol{\phi} = [\boldsymbol{\phi}_1^T, \ldots, \boldsymbol{\phi}_L^T]^T$ である。$\boldsymbol{\theta}_i$ と $\boldsymbol{\phi}_k$ とが従うディリクレ分布のパラメータ（ハイパーパラメータ）をそれぞれ $\boldsymbol{\alpha}$ と $\boldsymbol{\beta}$ とする。文書 d_i に対して表現 $\mathbf{w}_i$ が生成される確率は，トークン間の独立性の仮定の下に，$\boldsymbol{\theta}_i$ と $\boldsymbol{\phi}_k$ を積分して消去すれば，

$$P(\mathbf{w}_i|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int\int P(\boldsymbol{\phi}|\boldsymbol{\beta}) P(\boldsymbol{\theta}_i|\boldsymbol{\alpha}) \left[\prod_{h=1}^{l_i} \sum_{\tilde{z}_{ih}=1}^{L} P(\tilde{z}_{ih}|\boldsymbol{\theta}_i) P(\mathrm{w}_{ih}|\tilde{z}_{ih}, \boldsymbol{\phi}) \right] d\boldsymbol{\theta}_i d\boldsymbol{\phi} \tag{21}$$

となる（$P(\boldsymbol{\phi}|\boldsymbol{\beta})$ と $P(\boldsymbol{\theta}_i|\boldsymbol{\alpha})$ はディリクレ分布）。

このための LDA モデルのパラメータ推計はかなり複雑であるが[17]，Griffiths & Steyvers(2004)[11] によって，実装がより容易な Gibbs サンプリングの方法が提示され，この方法が広く活用されるようになった。Gibbs サンプリングは，*Markov chain Monte Carlo* (MCMC) [27] の一種であり，同時確率変数 $x_1, \ldots, x_n$ に対する確率分布 $P(x_1, \ldots, x_n)$ を推計するための数値的なシミュレーションである。具体的には，解析的に求められた n 個の条件付き確率 $P(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$（$i = 1, \ldots, n$）から，各 x_i の値を順に無作為抽出する過程を反復的に繰り返し，その抽出結果を集積して，分布 P をそこから経験的に構成する。

LDA の場合のこの条件付き確率は，

$$P(\tilde{z}_{ih} = k|\tilde{\mathbf{z}}^{\neg ih}, \mathbf{w}) = \frac{P(\tilde{z}_{ih} = k, \tilde{\mathbf{z}}^{\neg ih}|\mathbf{w})}{P(\tilde{\mathbf{z}}^{\neg ih}|\mathbf{w})} = \frac{P(\mathbf{w}|\tilde{z}_{ih} = k, \tilde{\mathbf{z}}^{\neg ih}) P(\tilde{z}_{ih} = k, \tilde{\mathbf{z}}^{\neg ih})}{P(\mathbf{w}|\tilde{\mathbf{z}}^{\neg ih}) P(\tilde{\mathbf{z}}^{\neg ih})} \tag{22}$$

である。ここで，$\tilde{\mathbf{z}}^{\neg ih}$ は，D 中のすべての文書の $\tilde{z}_{ih}$（$h = 1, \ldots, l_i; i = 1, \ldots, N$）を順に並べたベクトル $\tilde{\mathbf{z}}$ から，ある 1 つの $\tilde{z}_{ih}$ のみを削除したベクトルを意味する。また，$\mathbf{w} = [\mathbf{w}_1^T, \ldots, \mathbf{w}_N^T]^T$ である[18]。例えばもし，s 回目の反復において，$P(\tilde{z}_{32} = 1|\tilde{\mathbf{z}}^{\neg 32}, \mathbf{w}) = 0.4$，$P(\tilde{z}_{32} = 2|\tilde{\mathbf{z}}^{\neg 32}, \mathbf{w}) = 0.1$，$P(\tilde{z}_{32} = 3|\tilde{\mathbf{z}}^{\neg 32}, \mathbf{w}) = 0.5$ のように計算されたとする（$L = 3$）。ここで 0.0 ∼ 1.0 の一様乱数を発生させ，その値が 0.451... だったとすれば，この値は「0.4」と「0.4 + 0.1」との間であるから，s 回目の反復におけるトークン w_{32} のトピックは $k = 2$ となる（すなわち $\tilde{z}_{32}^{(s)} = 2$）[19]。この結果は次のトークン w_{33} のサンプリングのための (22) 式の値に直ちに反映され，同様な割り当てが繰り返される。

ディリクレ分布の場合，確率の合計が 1.0 になることを利用して，$\boldsymbol{\phi}_k$ や $\boldsymbol{\theta}_i$ を積分で消去することは容易である[20]。この結果，対称ディリクレ分布（すべての k について $\alpha_k = \alpha$）を仮定すれば，$\theta_i[k] \equiv \theta_{ik}$ と定義して，

$$P(\tilde{\mathbf{z}}_i) = \int P(\tilde{\mathbf{z}}_i|\boldsymbol{\theta}_i) P(\boldsymbol{\theta}_i|\alpha)\, d\boldsymbol{\theta}_i = \int \prod_{h=1}^{l_i} \theta_i[\tilde{z}_{ih}] P(\boldsymbol{\theta}_i|\alpha)\, d\boldsymbol{\theta}_i \propto \frac{\prod_k \Gamma(\mathrm{f}_{k|i} + \alpha)}{\Gamma(\mathrm{f}_{\cdot|i} + L\alpha)} \tag{23}$$

---

[17] Blei, Ng & Jordan (2003) [5] では，変分ベイズ法によって推計がなされているが，そのしくみはかなり複雑である。ここでは詳細は示さないが，この方法を使って 1000 試行の実験を試みたところ，$\mathcal{C}_v$ およびそれに類似したクラスタ集合 $\{\{d_1, d_2, d_3\}, \{d_4, d_5, d_7, d_{10}\}, \{d_6, d_8, d_9\}\}$ が得られたのは 28.8% に留まった。ただしハイパーパラメータの設定が悪かったのかもしれない。

[18] なお，文書間の独立性から，$P(\mathbf{w}) = \prod_i P(\mathbf{w}_i)$ となる。

[19] 本稿では，D 中の先頭の文書から最後の文書まで，すべてのトークンへのトピックの一連の割り当てを s 回目の反復とし，次に再び先頭の文書に戻った時，この回数を増やして $s + 1$ 回目の反復とする。

[20] ディリクレ分布は連続確率分布なので，例えば，$\mathbf{p}_k$ についての (11) 式を積分して 1 とおけば，以下の式を得る。

$$\int \prod_{j=1}^{M} p_{j|k}^{\beta_j - 1} d\mathbf{p}_k = \frac{\prod_{j=1}^{M} \Gamma(\beta_j)}{\Gamma(\sum_{j=1}^{M} \beta_j)}$$

を得る。ここで，$\mathrm{f}_{k|i}$ は文書 d_i において k 番目のトピックに属するトークンの数を意味し，また，$\mathrm{f}_{\cdot|i} = \sum_k \mathrm{f}_{k|i} = l_i$ である。同様に，$P(\boldsymbol{\phi}) = P(\boldsymbol{\phi}_1) \times \ldots \times P(\boldsymbol{\phi}_L)$ を仮定すれば，$\phi[j|k] \equiv P(t_j|\tau_k)$ と定義して，

$$P(\mathbf{w}|\tilde{\mathbf{z}}) = \int P(\mathbf{w}|\tilde{\mathbf{z}}, \boldsymbol{\phi}) P(\boldsymbol{\phi}|\beta) d\boldsymbol{\phi} = \int \prod_{i=1}^{N} \prod_{h=1}^{l_i} \phi[\omega(h|i)|\tilde{z}_{ih}] P(\boldsymbol{\phi}|\beta) d\boldsymbol{\phi} \propto \prod_{k=1}^{L} \frac{\prod_j \Gamma(\mathrm{F}_{j|k} + \beta)}{\Gamma(\mathrm{F}_{\cdot|k} + M\beta)} \tag{24}$$

となる（すべての j について $\beta_j = \beta$）。ここで，$\mathrm{F}_{j|k}$ は D 中で語 t_j が k 番目のトピックに割り当てられた回数（トークン数）を意味し，また，$\mathrm{F}_{\cdot|k} = \sum_j \mathrm{F}_{j|k}$ である。これらの計算結果を (22) 式に代入してみると，結局，当該トークン w_{ih} 以外の部分は約分されて消えることがわかる。約分の結果として残るのは，例えば，$\mathrm{F}_{j|k}$ については，

$$\frac{\Gamma(\mathrm{F}^{\neg ih}_{j|k} + 1 + \beta)}{\Gamma(\mathrm{F}^{\neg ih}_{j|k} + \beta)} = \mathrm{F}^{\neg ih}_{j|k} + \beta, \;\; j = \omega(h|i) \tag{25}$$

のみである[21]。ここで，$\mathrm{F}^{\neg ih}_{j|k}$ は，$\mathrm{F}_{j|k}$ をある 1 つのトークン w_{ih} を除いて集計した結果を示す。その他の量についても同様な計算を行えば，結局，

$$P(\tilde{z}_{ih} = k|\tilde{\mathbf{z}}^{\neg ih}, \mathbf{w}) \propto \frac{\mathrm{F}^{\neg ih}_{\omega(h|i)|k} + \beta}{\mathrm{F}^{\neg ih}_{\cdot|k} + M\beta} \frac{\mathrm{f}^{\neg ih}_{k|i} + \alpha}{\mathrm{f}^{\neg ih}_{\cdot|i} + L\alpha} \propto \frac{\mathrm{F}^{\neg ih}_{\omega(h|i)|k} + \beta}{\mathrm{F}^{\neg ih}_{\cdot|k} + M\beta} (\mathrm{f}^{\neg ih}_{k|i} + \alpha) \tag{26}$$

のように計算されるので，この分布を使って上記のようにサンプリングを逐次行えばよいことになる。

s 回目の反復におけるこれらの量を使えば，$\hat{\theta}^{(s)}_{ik} = (\mathrm{f}_{k|i} + \alpha)/(\mathrm{f}_{\cdot|i} + L\alpha)$ として，$\boldsymbol{\theta}_i$ を推計できる（同様に，$\hat{\phi}^{(s)}_{j|k} = (\mathrm{F}_{j|k} + \beta)/(\mathrm{F}_{\cdot|k} + M\beta)$）。サンプル DB に対して，1100 回の反復を繰り返し，101 回目以降の各反復での推定値 $\hat{\theta}^{(s)}_{ik}$ および $\hat{\phi}^{(s)}_{j|k}$ の平均値（合計して 1000 で割ったもの）を求めた結果を表 6 に示す[22]。ハイパーパラメータについては，恣意的なルールにより，$\alpha = (l \times 0.05)/(N \times L)$，$\beta = (l \times 0.05)/(M \times L)$ とした（ここで $l = \sum_i l_i$）。この表からは，各文書について，$\hat{\theta}^{(s)}_{ik}$ の平均値が最も大きいものをとってクラスタ集合を構成すれば，「妥当」な $\mathcal{C}_v$ が得られていることがわかる。なお，$\boldsymbol{\theta}_i$ や $\boldsymbol{\phi}_k$ の分布を推計せずに，文書クラスタリングのみを実行するならば，反復ごとに，文書 d_i を $k = \arg\max_{k'} \mathrm{f}_{k'|i}$ となる τ_k に割り当てて，クラスタ集合をそれぞれ構成しておき，最後に，Gibbs サンプリングの実行全体で最頻出するクラスタ集合のパターンを出力結果とすることもできる。

ただし，Gibbs サンプリングは一様乱数の発生に基づいているため，その実施ごとに各反復での割り当て結果は常に変化する。そこで，表 6 の操作をさらに 100 回繰り返す実験を試みた。その結果を表 7 に示す。表 6 は 1100 回の反復による 1 回のシミュレーションであり，初期値から出発した 1 つの Markov chain（連鎖）である。一方，表 7 は 100 chains の結果の要約であり，各 chain で最終的に決定されたクラスタ集合を提示している。なお，この表には，200 回の反復による結果および $\alpha = 0.1$，$\beta = 0.01$ とした場合の結果を併せて表示した。この結果からは，今回の実験では，適切にハイパーパラメータの値を設定すれば，高い確率で「妥当」なクラスタ集合が得られていることがわかる。

3.6 HDP によるクラスタ個数の同時推定

以上の確率的なクラスタリング手法では，k-means 法と同様に，クラスタ個数（または潜在的なトピックの個数）をあらかじめ決めておく必要がある。もしそれが不明な場合には，クラスタの個数を変えつつクラスタリングを何回か試行し，何らかの基準を使って，その中から最適なものを選択しなければならない（一

[21] ガンマ関数では，$\Gamma(x) = (x-1) \times \Gamma(x-1)$ が成り立つ点に注意。

[22] 1 回目から 100 回目での結果には初期値の影響が残っている可能性があるため除いた。この期間を「burn-in period」と呼ぶ。

表 6: Gibbs サンプリングによる LDA モデルのパラメータ推計（1 つの chain）

$\hat{\theta}_{ik}^{(s)}$ の平均値				$\hat{\phi}_{j\|k}^{(s)}$ の平均値			
	τ_1	τ_2	τ_3		τ_1	τ_2	τ_3
d_1	.934	.032	.033	t_1	.317	.005	.005
d_2	.916	.042	.041	t_2	.647	.055	.023
d_3	.950	.026	.023	t_3	.011	.541	.046
d_4	.024	.815	.160	t_4	.007	.187	.218
d_5	.309	.468	.222	t_5	.009	.082	.612
d_6	.107	.219	.674	t_6	.009	.128	.095
d_7	.088	.727	.184				
d_8	.032	.110	.857				
d_9	.027	.075	.897				
d_{10}	.030	.435	.535				

注 1：101 回目から 1100 回目までの反復の平均値。

注 2：$\alpha = .172$ and $\beta = .103$.

表 7: LDA モデルにおける 100 chains の Gibbs サンプリングの結果

		$\alpha = .100$		$\alpha = .172$	
		$\beta = .010$		$\beta = .103$	
	反復回数 =	200	1100	200	1100
1.	$\{d_1, d_2, d_3, d_4, d_5, d_7\}\{d_6, d_8, d_9\}\{d_{10}\}$	9	1	0	0
2.	$\{d_1, d_2, d_3, d_5\}\{d_4, d_7, d_{10}\}\{d_6, d_8, d_9\}$	8	5	8	1
3.	$\{d_1, d_2, d_3, d_5\}\{d_4, d_7\}\{d_6, d_8, d_9, d_{10}\}$	0	1	1	0
4.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_6, d_7, d_8, d_9\}\{d_{10}\}$	3	3	3	1
5.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_6, d_7, d_9\}\{d_8, d_{10}\}$	1	0	1	0
6.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_6, d_7, d_{10}\}\{d_8, d_9\}$	2	0	0	0
7.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_6, d_7\}\{d_8, d_9, d_{10}\}$	3	8	5	1
8.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_7, d_{10}\}\{d_6, d_8, d_9\}$	29	15	33	18
9.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_7\}\{d_6, d_8, d_9, d_{10}\}$	39	60	37	74
10.	$\{d_1, d_2, d_3\}\{d_4, d_7, d_8, d_{10}\}\{d_5, d_6, d_9\}$	0	2	0	1
11.	$\{d_1, d_2, d_3\}\{d_4, d_7, d_{10}\}\{d_5, d_6, d_8, d_9\}$	2	2	4	2
12.	$\{d_1, d_2, d_3\}\{d_4, d_8, d_9, d_{10}\}\{d_5, d_6, d_7\}$	1	0	1	0
13.	$\{d_1, d_2, d_3\}\{d_4, d_{10}\}\{d_5, d_6, d_7, d_8, d_9\}$	0	0	4	0
14.	その他	3	3	3	2
	chains の合計	100	100	100	100

注 1：1〜100 回目の反復はクラスタ構成のための計算に含めていない。

注 2：「その他」は単一の chain にのみ出現したクラスタ集合。

種のモデル選択)。それに対して，LDA モデルの拡張である *hierarchical Dirichlet process* (HDP) の混合モデルでは，クラスタ個数も同時に推計される。これは，このモデルが理論的には「無限個」の分布の混合モデルであり，その分布の個数がデータに応じて，有限個に決まってくることによる。

HDP 混合モデルによって潜在的なトピックを文書集合から推計する方法にはいくつかの種類があるが，本稿では，Teh, Jordan, Beal & Blei (2005) [30, 31] による「*Chinese restaurant franchise*」(CRF) モデルに基づく Gibbs サンプリングの方法のうちの 1 つのみを取り上げる。CRF モデルでは，トークン w_{ih} を「客」，トピック τ_k を「料理」として捉える。ただし，レストランの 1 つのテーブルでは，1 つの料理のみが注文されるとする（料理は途中で変わることがある)。1 つのレストランが 1 つの文書に相当し，フランチャイズ組織のため，料理のメニューは全レストランで同一である。客はどこかのテーブルに着席するが，これがトークン w_{ih} にトピック τ_k が割り当てられたことを意味する（客はテーブルを移る場合がある)。

客 w_{ih} が着席したテーブルの「番号」を u_{ih} と表記し，レストラン d_i の u 番目のテーブルにおける料理の「番号」を k_{iu} とする。$m_{i\cdot}$ をレストラン d_i において客が座っているテーブルの総数とすれば，その時点で出されている料理の番号は $\mathbf{k}_i = [k_{i1}, \ldots, k_{im_{i\cdot}}]^T$ であり，LDA モデルと同様にトピック τ_k における語の分布を $\boldsymbol{\phi}_k$ とすれば，各トークン w_{ih} は確率分布 $P(\cdot|u_{ih}, \mathbf{k}_i, \boldsymbol{\phi})$ から生成されることになる。

もし，トークン w_{ih} が着席するテーブルの番号 u_{ih} が，ディリクレ分布 $P(\cdot|\alpha_0/m_{i\cdot}, \ldots, \alpha_0/m_{i\cdot})$ に従う $m_{i\cdot}$ 次元ベクトル $\tilde{\boldsymbol{\theta}}_i$ から無作為抽出されると仮定すれば，$\mathbf{u}_i = [u_{i1}, \ldots, u_{il_i}]^T$ の分布は，ディリクレ分布の積分より，

$$P(\mathbf{u}_i) = \int P(\mathbf{u}_i|\tilde{\boldsymbol{\theta}}_i) P(\tilde{\boldsymbol{\theta}}_i|\alpha_0/m_{i\cdot}, \ldots, \alpha_0/m_{i\cdot}) d\tilde{\boldsymbol{\theta}}_i = \frac{\prod_u \Gamma(\tilde{\mathrm{f}}_{u|i} + \alpha_0/m_{i\cdot})}{\Gamma(\tilde{\mathrm{f}}_{\cdot|i} + \alpha_0)} \tag{27}$$

となる。ここで，$\tilde{\mathrm{f}}_{u|i}$ は u 番目のテーブルに着席しているトークンの数である（$\tilde{\mathrm{f}}_{\cdot|i} = l_i$ はその合計)。この場合，$P(u_{ih} = u|\mathbf{u}_i^{\neg ih}) = P(u_{ih} = u, \mathbf{u}_i^{\neg ih})/P(\mathbf{u}_i^{\neg ih})$ は，約分の結果，$(\tilde{\mathrm{f}}_{u|i}^{\neg ih} + \alpha_0/m_{i\cdot})/(\tilde{\mathrm{f}}_{\cdot|i}^{\neg ih} + \alpha_0)$ となるので，$m_{i\cdot} \to \infty$ と仮定すれば，$P(u_{ih} = k|\mathbf{u}_i^{\neg ih}) \to \tilde{\mathrm{f}}_{u|i}^{\neg ih}/(l_i - 1 + \alpha_0)$ である。この分子を文書 d_i における u_{ih} 以外のすべてのテーブルで合計すれば，$l_i - 1$ になるので，結果的に，

$$P(u_{ih} = u|\mathbf{u}_i^{\neg ih}) \to \begin{cases} \frac{\tilde{\mathrm{f}}_{u|i}^{\neg ih}}{l_i - 1 + \alpha_0} & u \text{ が } \mathbf{u}_i^{\neg ih} \text{ の中に含まれている場合} \\ \frac{\alpha_0}{l_i - 1 + \alpha_0} & u \text{ が新たに着席されたテーブルの場合} \end{cases} \tag{28}$$

が導かれる。つまり，着席している客の数が多いほど，そのテーブルが選択される確率が高くなる。

実際に，テーブル u を Gibbs サンプリングの枠組みで抽出するには，条件付き確率

$$P(u_{ih} = u|\mathbf{u}^{\neg ih}, \mathbf{w}, \mathbf{k}) = \frac{P(u_{ih} = u, \mathbf{u}^{\neg ih}|\mathbf{w}, \mathbf{k})}{P(\mathbf{u}^{\neg ih}|\mathbf{w}, \mathbf{k})} = \frac{P(\mathbf{w}|u_{ih} = u, \mathbf{u}^{\neg ih}, \mathbf{k})}{P(\mathbf{w}|\mathbf{u}^{\neg ih}, \mathbf{k})} \times \frac{P(u_{ih} = u, \mathbf{u}^{\neg ih}|\mathbf{k})}{P(\mathbf{u}^{\neg ih}|\mathbf{k})} \tag{29}$$

を計算する必要がある（$\mathbf{u} = [\mathbf{u}_1^T, \ldots, \mathbf{u}_N^T]^T$ および $\mathbf{k} = [\mathbf{k}_1^T, \ldots, \mathbf{k}_N^T]^T$)。この最右辺第 1 項は，(25) 式の導出と同様な手順を使って，

$$\frac{P(\mathrm{w}_{ih} = t_j, \mathbf{w}^{\neg ih}|u_{ih} = u, \mathbf{u}^{\neg ih}, \mathbf{k})}{P(\mathrm{w}_{ih} = t_j, \mathbf{w}^{\neg ih}|\mathbf{u}^{\neg ih}, \mathbf{k})} = \frac{\mathrm{F}_{j|k[ih]}^{\neg ih} + \beta}{\mathrm{F}_{\cdot|k[ih]}^{\neg ih} + M\beta} \equiv f_k^{\neg ih}(\mathrm{w}_{ih}) \tag{30}$$

となる（各テーブルの料理が決まっているので，β をパラメータとする対称ディリクレ分布に従う $\boldsymbol{\phi}_k$ から語が抽出されると考えることができる)。ここで，$k[ih]$ は w_{ih} が着席しているテーブルの料理の番号を示す。ところが，HDP モデルの場合には，これに加えて，新しいテーブルにトークンが着席した場合の確率 $P(\mathbf{w}|u_{ih} = u^\dagger, \mathbf{u}^{\neg ih}, \mathbf{k})$ を考えなければならない（$u^\dagger = m_{i\cdot} + 1$ は新しいテーブルの番号)。この確率は新

しいテーブルで選ばれる料理に依存するので，$f_k^{\neg ih}(\mathrm{w}_{ih})$ そのものではなく，各料理が選択される確率に基づく期待値として計算しなければならない。

料理（トピック）がディリクレ分布 $P(\cdot|\xi/L,\ldots,\xi/L)$ に従う L 次元ベクトルから抽出されるとすれば，上と同様の議論により，$L\to\infty$ の場合，

$$P(k_{iu}=k|\mathbf{k}^{\neg iu})\to\begin{cases} m_{\cdot k}/(m_{\cdot\cdot}+\xi) & k\text{ が }\mathbf{k}^{\neg iu}\text{ の中に含まれている場合}\\ \xi/(m_{\cdot\cdot}+\xi) & k\text{ が新しい料理の場合}\end{cases} \tag{31}$$

となる（$L\to\infty$ の操作が「無限個」の分布を考えることに相当する [24]）。ここで，$m_{\cdot k}$ は料理 k を注文しているテーブルの全レストランでの総数，$m_{\cdot\cdot}$ は全レストランでのテーブルの総数を意味する。この確率分布を使えば，

$$\frac{P(\mathrm{w}_{ih}=t_j,\mathbf{w}^{\neg ih}|u_{ih}=u^\dagger,\mathbf{u}^{\neg ih},\mathbf{k})}{P(\mathrm{w}_{ih}=t_j,\mathbf{w}^{\neg ih}|\mathbf{u}^{\neg ih},\mathbf{k})}=\sum_{k=1}^{L}\frac{m_{\cdot k}}{m_{\cdot\cdot}+\xi}f_k^{\neg ih}(\mathrm{w}_{ih})+\frac{\xi}{m_{\cdot\cdot}+\xi}\frac{1}{M}\equiv P(\mathrm{w}_{ih}=t_j|\dagger) \tag{32}$$

である。ここで，$1/M$ は語 t_j（$j=1,\ldots,M$）の選択確率の事前分布である（すなわち $f_{k^\dagger}^{\neg ih}(\mathrm{w}_{ih})=M^{-1}$，$k^\dagger$ は新しい料理を指す）。

客がテーブルを選択する際にそこでの料理が影響しないと仮定すれば，(29) 式の最右辺第 2 項としては，(28) 式をそのまま使えばよい。したがって，Gibbs サンプリングのための u に関する条件付き確率は，最終的に，

$$P(u_{ih}=u|\mathbf{u}^{\neg ih},\mathbf{w},\mathbf{k})\propto\begin{cases}\tilde{\mathrm{f}}_{u|i}^{\neg ih}f_k^{\neg ih}(\mathrm{w}_{ih}) & u\text{ がすでに着席されているテーブルの場合}\\ \alpha_0 P(\mathrm{w}_{ih}=t_j|\dagger) & u=u^\dagger\text{ の場合}\end{cases} \tag{33}$$

となる。$u\neq u^\dagger$ の場合のこの条件付き確率は, LDA モデルにおける (26) 式と同じ形をしている。一方，新しいテーブルが選ばれた場合には，続けて，このテーブルで注文する料理 τ_k を無作為抽出する必要がある。これは，(32) 式に基づいて，

$$P(k_{iu^\dagger}=k|\mathbf{u},\mathbf{k},\mathbf{w})\propto\begin{cases} m_{\cdot k}f_k^{\neg ih}(\mathrm{w}_{ih}) & k\text{ がすでに注文されている場合}\\ \xi/M & k=k^\dagger\text{ の場合}\end{cases} \tag{34}$$

に従って抽出すればよい。

次に，Gibbs サンプリングの過程において，既存のテーブルの料理を改めて無作為抽出するための条件付き確率は，上と同様の議論を繰り返せば，

$$P(k_{iu}=k|\mathbf{u},\mathbf{k}^{\neg iu},\mathbf{w})\propto\begin{cases} m_{\cdot k}^{\neg iu}g_k^{\neg iu}(\mathbf{w}) & k\text{ がすでに注文されている場合}\\ \xi(1/M)^{|\Lambda[iu]|} & k=k^\dagger\text{ の場合}\end{cases} \tag{35}$$

となる（$\neg iu$ は文書 d_i の u 番目のテーブルを除くことを意味する）。ここで，

$$\frac{P(\mathbf{w}|\mathbf{u},k_{iu}=k,\mathbf{k}^{\neg iu})}{P(\mathbf{w}|\mathbf{u},\mathbf{k}^{\neg iu})}\propto\frac{\prod_{j\in\Lambda[iu]}\Gamma(\mathrm{F}_{j|k}+\xi)}{\prod_{j\in\Lambda[iu]}\Gamma(\mathrm{F}_{j|k}^{\neg iu}+\xi)}\times\frac{\Gamma(\mathrm{F}_{\cdot|k}^{\neg iu}+L\xi)}{\Gamma(\mathrm{F}_{\cdot|k}+L\xi)}\equiv g_k^{\neg iu}(\mathbf{w}) \tag{36}$$

であり，$\Lambda[iu]$ は d_i における u 番目のテーブルに着席しているトークンに対応する語の「番号」（t_j の添字）の集合を意味する。

結果的に，HDP 混合モデルの Gibbs サンプリングでは，(1) 各トークンが着席するテーブルを (33) 式に基づいて順に割り当てる（新しいテーブルの場合には，続けて (34) 式で料理を決定する），(2) 各テーブル

表 8: HDP 混合モデルでの Gibbs サンプリングの実験結果（単一の chain）

	クラスタ集合	L	サンプル数
1.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_7\}\{d_6, d_8, d_9\}\{d_{10}\}$	4	276
2.	$\{d_1, d_2, d_3, d_5\}\{d_4, d_7\}\{d_6, d_8, d_9\}\{d_{10}\}$	4	194
3.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_7\}\{d_6, d_8, d_9, d_{10}\}$	3	68
4.	$\{d_1\}\{d_2, d_3\}\{d_4, d_5, d_7\}\{d_6, d_8, d_9\}\{d_{10}\}$	5	66
5.	$\{d_1, d_2, d_3, d_5\}\{d_4, d_7\}\{d_6, d_8, d_9, d_{10}\}$	3	55
6.	$\{d_1, d_2, d_3\}\{d_4, d_7\}\{d_5, d_6, d_8, d_9\}\{d_{10}\}$	4	44
7.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_7, d_{10}\}\{d_6, d_8, d_9\}$	3	35
8.	$\{d_1, d_2, d_3\}\{d_4, d_5, d_7\}\{d_6, d_9\}\{d_8, d_{10}\}$	4	21
9.	$\{d_1, d_2, d_3, d_4, d_5, d_7\}\{d_6, d_8, d_9, d_{10}\}$	2	20
10.	$\{d_1\}\{d_2, d_3, d_5\}\{d_4, d_7\}\{d_6, d_8, d_9\}\{d_{10}\}$	5	13
11.	その他（全部で 86 パターン）	-	208
	合計	-	1000

注 1：$\alpha_0 = 0.1$, $\beta = 0.01$, $\xi = 0.1$。1100 回反復の単一 chain。

注 2：「その他」は出現サンプル数が 12 以下のもの。

に対する料理を (35) 式に基づいて順に割り当てる，という手順を反復的に繰り返すことになる。それぞれの反復での割り当てに基づいて，$\mathrm{f}_{k|i}$ を計算すれば，LDA モデルと同様の手順で文書のクラスタ集合を得ることができる。この過程において，新しい料理（トピック）が抽出されれば，その時点でのクラスタは 1 つ増えることになる。逆に，どのテーブルからも姿を消した料理は削除するので，この場合には，クラスタが 1 つ減る（テーブルの数もまた増減する）。この増減の繰り返しの中で，最適なクラスタ個数が最終的に決定されることになるが，実際には，この個数は，あらかじめ設定するパラメータ ξ の大きさに多かれ少なかれ依存する。

恣意的に $\alpha_0 = 0.1$，$\beta = 0.01$，$\xi = 0.1$ と設定し，1100 回の反復による Gibbs サンプリングを行った結果（単一の chain）を表 8 に示す。この表はそれぞれの反復を 1 つのサンプルとして，サンプルごとにクラスタ集合を構成し，その各パターンが出現したサンプル数を集計したものである（101〜1100 回目での集計）。この chain ではクラスタ集合 $\{\{d_1, d_2, d_3\}, \{d_4, d_5, d_7\}, \{d_6, d_8, d_9\}, \{d_{10}\}\}$ が最も頻出し（$L = 4$），これが最終的なクラスタリング結果となる[23]。図 3 は，表 8 の実験におけるクラスタ個数の分布を示したものである（左側）。また，その右側は $\xi = 0.5$ での分布である。この場合には，両方とも，最適なクラスタ個数は $L = 4$ として推計されたことになる。なお，CRF モデルによる別の推計方法もある [30, 31]。

[23]混合モデルのパラメータを Gibbs サンプリングによって推定する場合，label-switching が発生する（文献 [23] の p.129 参照）。例えば，ここでの実験でも，d_1, d_2，d_5 が $s = 515$ では「d_1: $k = 1$, d_2: $k = 1$, d_5: $k = 3$」となり，$s = 570$ では「d_1: $k = 3$, d_2: $k = 3$, d_5: $k = 1$」となった。つまり，同一のクラスタ集合が得られているにも関わらず，その「番号」は変化する。この解決にはいくつかの方法があるが（例えば [28] 参照），文書ベクトルは高次元のため，その適用は難しい。したがって，パラメータの分布の推計が必要なく，単にクラスタ集合を求めるだけならば，各反復ごとにクラスタ集合を構成し，最頻出のパターンを出力結果とする方法は効果的かつ効率的である。

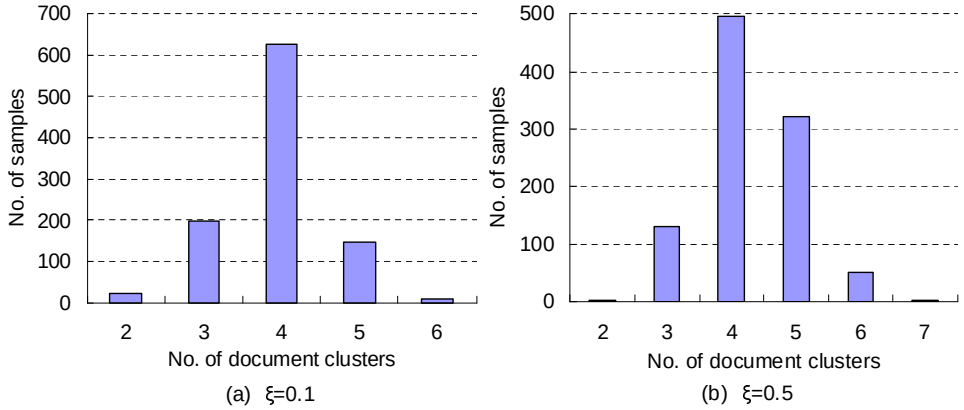


図 3: HDP 混合モデルによる Gibbs サンプリングでのクラスタ（トピック）の個数

4 行列の分解に基づく文書クラスタリング

4.1 特異値分解と主成分分析

線形代数学の理論・技術を用いて,「文書×語行列」$\mathbf{W} = [w_{ij}]$ を分解し，その結果に基づいて文書クラスタリングを行うことが可能である（この節では行列の要素は，特に断らない限り，tf とする。すなわち，$w_{ij} = \mathrm{f}_{ij}$)。この種の分解としては，特異値分解（singular value decomposition: SVD）がよく知られており，この場合，$\mathbf{W} = \mathbf{UQV}^T$ のように，元の行列が 3 つの行列に分解される。ここで $\mathbf{U}$ は $N \times r$ の直交行列，$\mathbf{Q}$ は $r \times r$ の対角行列，$\mathbf{V}$ は $M \times r$ の直交行列である。ただし，ここでは r は行列 $\mathbf{W}$ のランクを表すものとする。SVD は，情報検索における latent semantic indexing(LSI)[8] に応用されたことで有名である[24]。

この特異値分解を直接，文書クラスタリングに応用することもあるが，分解の前に，文書×語行列の各要素から，各語の平均 $\tilde{m}_j = N^{-1} \sum_{i=1}^{N} w_{ij}$ をそれぞれ差し引いておくと，クラスタリング結果がより明快になる。つまり，$\mathbf{m} = [\tilde{m}_1, \ldots, \tilde{m}_M]^T$ を使って，文書×語行列を $N^{-1/2}[\mathbf{W} - \mathbf{em}^T]$ で変換してから（ここで $\mathbf{e} = [1, 1, 1, ..., 1]^T$），

$$\frac{1}{\sqrt{N}}[\mathbf{W} - \mathbf{em}^T] = \mathbf{UQV}^T \tag{37}$$

のように分解する。この結果，例えば，サンプル DB の場合には，

$$\mathbf{U} = \begin{bmatrix} -0.285 & 0.097 & -0.278 & 0.800 & -0.197 & 0.190 \\ -0.350 & 0.110 & -0.059 & -0.023 & 0.086 & -0.561 \\ -0.622 & 0.209 & 0.084 & -0.437 & 0.136 & 0.460 \\ 0.127 & -0.591 & 0.005 & 0.090 & 0.587 & 0.264 \\ -0.110 & -0.093 & 0.223 & -0.071 & -0.024 & -0.525 \\ 0.295 & 0.339 & 0.361 & -0.036 & -0.215 & 0.232 \\ 0.039 & -0.510 & 0.400 & -0.007 & -0.497 & 0.024 \\ 0.280 & 0.181 & -0.116 & 0.033 & 0.413 & -0.182 \\ 0.408 & 0.390 & 0.118 & 0.039 & 0.062 & 0.065 \\ 0.219 & -0.131 & -0.738 & -0.390 & -0.350 & 0.034 \end{bmatrix} \tag{38}$$

[24] この手法では，対角行列の r 個の要素のうち，r' 個を抽出し（$r' < r$），r' 個の要因に基づいて文書ベクトルや質問ベクトルの間の類似度を計算する。この抽出された要因が「潜在的な意味」に相当する。

となるので，この第 1 列と第 2 列を使って，各行（文書）をプロットすると，図 4 のようになる。これは，主成分分析（principal component analysis: PCA）を実行して，その第 1 成分と第 2 成分でサンプルをプロットしたことに相当する[25]。

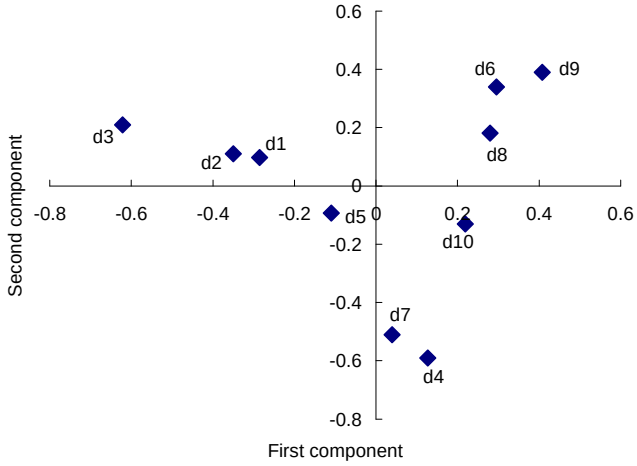


図 4: 主成分分析（PCA）の結果

4.2 非負行列分解

主成分分析の結果を使えば，図 4 のようなプロットを描くことができるものの，これらの成分の値からクラスタ集合を自動的に生成することはやや面倒である。このため，文書クラスタリングでは，非負行列分解（nonnegative matrix factorization: NMF）を用いることがある [32]。慣例に従い，$\mathbf{W}$ ではなく，その転置行列 $\mathbf{Z} = \mathbf{W}^T$ を分解対象とすれば，NMF では，これを $\mathbf{Z} \approx \mathbf{A}\mathbf{C}^T$ のように，2 つの行列に分解する。ただし，$\mathbf{A}$ と $\mathbf{C}$ の要素はすべて非負である。

文書クラスタリングの場合には，クラスタ個数 L を決めておき，$\mathbf{A}$ を $M \times L$ 行列，$\mathbf{C}$ を $N \times L$ 行列とする。その結果，文書 d_i のクラスタの番号を $k = \arg\max_{k'=1,\ldots,L} c_{ik'}$ として，特異値分解に比べて，より単純に求めることができる（ここで $c_{ik'}$ は $\mathbf{C}$ の第 i 行第 k' 列の要素を指す）。ただし，特異値分解とは異なり，非負行列分解はあくまで「近似」なので，この近似を何らかの方法で決める必要がある。

一般に利用されているのは，関数 $f(\mathbf{A}, \mathbf{C}) = \frac{1}{2}\|\mathbf{Z} - \mathbf{A}\mathbf{C}^T\|^2$ を最小にする分解を求める方法である[26]。この関数は，行列 $\mathbf{X}$ のトレースを $\mathrm{tr}[\mathbf{X}]$ で表記すれば，

$$f(\mathbf{A}, \mathbf{C}) = (1/2)\mathrm{tr}[(\mathbf{Z} - \mathbf{A}\mathbf{C}^T)(\mathbf{Z} - \mathbf{A}\mathbf{C}^T)^T] = (1/2)\mathrm{tr}(\mathbf{Z}\mathbf{Z}^T) - \mathrm{tr}(\mathbf{Z}\mathbf{C}\mathbf{A}^T) + (1/2)\mathrm{tr}(\mathbf{A}\mathbf{C}^T\mathbf{C}\mathbf{A}^T) \quad (39)$$

となる。この場合には，$\mathbf{A}$ と $\mathbf{C}$ の行列の要素がすべて非負という条件，すなわちそれぞれの (i, j) 要素について $a_{ij} \geq 0$，$c_{ij} \geq 0$ という不等号を含んだ条件で制約されるため，通常の Lagrange の方法では

[25] 文書×語行列を通常のデータ行列と見なせば，各変数の分散共分散行列は $\hat{\mathbf{\Sigma}} = N^{-1}[\mathbf{W} - \mathbf{e}\mathbf{m}^T]^T[\mathbf{W} - \mathbf{e}\mathbf{m}^T]$ である。ここに，(37) 式を代入すれば，$\hat{\mathbf{\Sigma}} = [\mathbf{U}\mathbf{Q}\mathbf{V}^T]^T\mathbf{U}\mathbf{Q}\mathbf{V}^T = \mathbf{V}\mathbf{Q}\mathbf{U}^T\mathbf{U}\mathbf{Q}\mathbf{V}^T = \mathbf{Q}\mathbf{Q}\mathbf{V}^T$ より，$\hat{\mathbf{\Sigma}}\mathbf{V} = \mathbf{Q}\mathbf{Q}\mathbf{V}$ となる。したがって，(37) 式は，$\hat{\mathbf{\Sigma}}$ の固有値および固有ベクトルを計算したわけであり，主成分分析を実行したことに相当する（主成分分析については文献 [20] などを参照）。

[26] ここで $\|\cdot\|$ は Frobenius ノルムであり，行列 $\mathbf{X} = [x_{ij}]$ に対して $\|\mathbf{X}\| = \left(\sum_i \sum_j x_{ij}^2\right)^{1/2}$ で計算される。

なく，Karush-Kuhn-Tucker(KKT) 定理を使って最適解を求めることになる。この結果，$\mathbf{A}$ については，$\partial f(\mathbf{A},\mathbf{C})/\partial\mathbf{A} = -\mathbf{ZC} + \mathbf{AC}^T\mathbf{C}$ より[27]，Hadamard 積を $\otimes$ で表記すれば[28]，$-\mathbf{ZC}\otimes\mathbf{A} + \mathbf{AC}^T\mathbf{C}\otimes\mathbf{A} - \bar{\mathbf{A}}_1\otimes\mathbf{A} = -\mathbf{ZC}\otimes\mathbf{A} + \mathbf{AC}^T\mathbf{C}\otimes\mathbf{A} = \mathbf{O}$ を得る。ここで，$\mathbf{a}_k$ を M 次元の KKT 乗数ベクトルとして $(k = 1,\ldots,L)$，$\bar{\mathbf{A}}_1 = [\mathbf{a}_1,\ldots,\mathbf{a}_L]$ である[29]。

したがって，$\mathbf{X}$ の (i,j) 要素を $[\mathbf{X}]_{ij}$ で表記すれば，各要素について，$-[\mathbf{ZC}]_{ij}a_{ij} + [\mathbf{AC}^T\mathbf{C}]_{ij}a_{ij} = 0$ という式が得られたことになる。これは $\mathbf{A}$ の要素が，

$$a_{ij} \leftarrow a_{ij}\frac{[\mathbf{ZC}]_{ij}}{[\mathbf{AC}^T\mathbf{C}]_{ij}} \tag{40}$$

による反復計算で推計できることを意味している。同様な手順で計算すれば，$\mathbf{C}$ についても，更新式

$$c_{ij} \leftarrow c_{ij}\frac{[\mathbf{Z}^T\mathbf{A}]_{ij}}{[\mathbf{CA}^T\mathbf{A}]_{ij}} \tag{41}$$

を得る。この推計方法 [17] は，*multiplicative iterative* **アルゴリズム**または Lee-Seung アルゴリズムと呼ばれ，NMF の計算に幅広く利用されている。

なお，任意の正方行列 $\mathbf{S}$ に対して，その逆行列が存在すれば，$\mathbf{Z} \approx \mathbf{AC}^T = [\mathbf{AS}][\mathbf{CS}^{-1}]^T$ であるから，通常，非負行列分解は一意ではない。そこで，この $\mathbf{S}$ を，j 番目の対角要素が $(\sum_i a_{ij}^2)^{-1/2}$ である対角行列と決めておく。これは，$\mathbf{A}$ の各列を単位ベクトルに変換することを意味する。したがって，NMF の手順は，以下のようになる。

(1) 一様乱数を生成し，$\mathbf{A}$ と $\mathbf{C}$ の各要素に割り当てる（初期化）。
(2) すべての要素について $x_{ij} = [\mathbf{ZC}]_{ij}/[\mathbf{AC}^T\mathbf{C}]_{ij}$ を計算する。
(3) すべての要素について $y_{ij} = [\mathbf{Z}^T\mathbf{A}]_{ij}/[\mathbf{CA}^T\mathbf{A}]_{ij}$ を計算する。
(4)$\mathbf{A}$，$\mathbf{C}$ をそれぞれ $a_{ij} \leftarrow a_{ij}x_{ij}$，$c_{ij} \leftarrow c_{ij}y_{ij}$ で更新する。
(5)$\mathbf{A}$ の j 列の各要素を $(\sum_i a_{ij}^2)^{1/2}$ で割る。
(6) 収束すれば終了，そうでなければ (2) に戻る。

となる。サンプル DB に対してこの手順で計算した結果，

$$\mathbf{C}^T = \begin{bmatrix} 2.238 & 3.160 & 5.996 & 0.000 & 1.867 & 0.916 & 0.870 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 4.701 & 1.702 & 0.446 & 4.711 & 0.401 & 0.000 & 1.754 \\ 0.000 & 0.000 & 0.000 & 0.013 & 0.718 & 4.796 & 0.420 & 3.291 & 5.081 & 1.700 \end{bmatrix}$$

が得られたため，この場合には，クラスタ集合は $\{d_1,d_2,d_3,d_5\}$ (第 1 行), $\{d_4,d_7,d_{10}\}$ (第 2 行), $\{d_6,d_8,d_9\}$ (第 3 行) となる。ただし，d_5 と d_{10} については，その最大値が他の値に対して突出しているわけではない。一方，単位ベクトル $\tilde{\mathbf{d}}_i$ を $\mathbf{Z}$ として使用した場合には，「妥当」なクラスタ集合 $\mathcal{C}_v$ が得られた。なお，NMF の場合も，初期値によっては，局所的な最小値に収束することがあるが，今回の実験では，単位ベクトルを

[27]ベクトルや行列での微分については，線形代数学の教科書（[13] など）を参照。

[28]$\mathbf{X}$ と $\mathbf{Y}$ が 2×2 行列ならば以下のようになる。

$$\mathbf{X}\otimes\mathbf{Y} = \begin{bmatrix} x_{11}\times y_{11} & x_{12}\times y_{12} \\ x_{21}\times y_{21} & x_{22}\times y_{22} \end{bmatrix}$$

[29]ベクトル $\mathbf{x}$ に関する条件 $\mathbf{g}(\mathbf{x}) \geq \mathbf{0}$ の下に関数 $f(\mathbf{x})$ を最小とする最適解を $\mathbf{x}^*$ とする（$f:\Re^n\to\Re$ および $\mathbf{g}:\Re^n\to\Re^p$）。KKT 定理によれば，この最適解は，(1)$\mathbf{a}\geq\mathbf{0}$，(2)$Df(\mathbf{x}^*)^T - D\mathbf{g}(\mathbf{x}^*)^T\mathbf{a} = \mathbf{0}$，(3)$\mathbf{g}(\mathbf{x}^*)^T\mathbf{a} = 0$ の条件を満たす（必要条件）[6]。ここで $\mathbf{a}$ は KKT 乗数ベクトル，D は微分を意味する（この場合「データベース」ではない）。NMF の場合，$\mathbf{A}$ の 1 つの列を $\mathbf{x}$ とすれば，$n = p = M$ として，$\mathbf{g}(\mathbf{x}) = \mathbf{x}$ （すなわち $g_1(\mathbf{x}) = x_1,\ldots,g_M(\mathbf{x}) = x_M$）なので，$D\mathbf{g}(\mathbf{x}) = I$（対角要素が 1 の対角行列）より，$D\mathbf{g}(\mathbf{x})^T\mathbf{a} = \mathbf{a}$ である。このため，$\mathbf{x}$ を横に並べて，f の微分結果を使えば，条件 (2) は $-\mathbf{ZC} + \mathbf{AC}^T\mathbf{C} - \bar{\mathbf{A}}_1 = \mathbf{O}$ を意味することになる。この式の右から $\otimes\mathbf{A}$ を適用すれば，$\mathbf{g}(\mathbf{x}) = \mathbf{x}$ の場合には，条件 (3) は $x_j^*\times a_j = 0$ $(j = 1,\ldots,M)$ を意味するので（a_j は $\mathbf{a}$ の要素），$\bar{\mathbf{A}}_1$ を消去できる。

使った場合には，NMF の計算の 100 回の繰り返しすべてにおいて，妥当なクラスタ集合 $\mathcal{C}_v$ が構成された。この結果を見る限りでは，NMF によるクラスタリングの結果は，単位ベクトルを使った場合，妥当かつ安定している（ただし，行列の積の計算に特別な工夫をしない限り，その計算量はかなり多い）。

NMF の拡張として *nonnegative block value decomposition* (NBVD) が提案されている [18]。この場合，分解は

$$\mathbf{Z} \approx \mathbf{A}\mathbf{B}\mathbf{C}^T \tag{42}$$

である。ここで，新たに追加された $\mathbf{B}$ は $L' \times L$ 行列で，それに対応して，$\mathbf{A}$ は $M \times L'$ 行列になる。つまり，$L' \neq L$ ならば，語のクラスタの個数と文書のクラスタ個数とを変えることができる[30]。実際の推計方法は，上記と同様の手順により，

$$[\mathbf{A}]_{ij} \leftarrow [\mathbf{A}]_{ij} \frac{[\mathbf{Z}\mathbf{C}\mathbf{B}^T]_{ij}}{[\mathbf{A}\mathbf{B}\mathbf{C}^T\mathbf{C}\mathbf{B}^T]_{ij}}, \quad [\mathbf{B}]_{ij} \leftarrow [\mathbf{B}]_{ij} \frac{[\mathbf{A}^T\mathbf{Z}\mathbf{C}]_{ij}}{[\mathbf{A}^T\mathbf{A}\mathbf{B}\mathbf{C}^T\mathbf{C}]_{ij}}, \quad [\mathbf{C}]_{ij} \leftarrow [\mathbf{C}]_{ij} \frac{[\mathbf{B}^T\mathbf{A}^T\mathbf{Z}]_{ij}}{[\mathbf{B}^T\mathbf{A}^T\mathbf{A}\mathbf{B}\mathbf{C}^T]_{ij}} \tag{43}$$

となる。ただし，分解の一意性を確保するために，$\mathbf{A}$ だけでなく，$\mathbf{C}$ の要素も正規化する必要がある[31]。

サンプル DB を使って（ただし単位ベクトル），$L' = 4$ かつ $L = 3$ で NBVD を実行した結果，$\mathbf{C}$ からは「妥当」なクラスタ集合 $\mathcal{C}_v$ が得られ，また，

$$\mathbf{A} = \begin{bmatrix} 0.000 & 0.000 & 0.000 & 0.400 \\ 0.015 & 0.009 & 0.081 & 0.915 \\ 0.000 & 0.000 & 0.954 & 0.026 \\ 0.361 & 0.166 & 0.250 & 0.000 \\ 0.926 & 0.985 & 0.000 & 0.030 \\ 0.108 & 0.050 & 0.141 & 0.000 \end{bmatrix} \tag{44}$$

より，語のクラスタ集合は $\{\{t_1, t_2\}, \{t_3, t_6\}, \{t_4\}, \{t_5\}\}$ となった。$L' = L = 3$ の場合には，$\{\{t_1, t_2\}, \{t_3, t_6\}, \{t_4, t_5\}\}$ だったので，クラスタ個数を 4 に増やした場合には，t_4 と t_5 とが分離することになる。

なお，(42) 式の分解は，「three-factor NMF」あるいは「tri-NMF」の特殊形として解釈できる。この他，一般には，行列に関する制約の異なるさまざまな NMF の変形が考案されている（詳細は文献 [7] を参照）。また，通常の文書クラスタリングでは，語と文書の 2 要因から成る「2-way」のデータであるが，*nonnegative tensor factorization* (NTF)[7] の手法を使えば，「3-way」あるいはそれ以上のクラスタリングも可能である（例えば，「著者×語×期間」など）。

5 おわりに

本稿では，文書クラスタリングのための確率的モデルとして，多項混合モデル・vMF 分布の混合モデル・PLSA・LDA・HDP 混合モデル，および行列の分解に基づく文書クラスタリング技法として，特異値分解・NMF・NBVD による手法を議論した。その際，簡単なサンプル DB に対して，それぞれのモデル・手法に基づくクラスタリングを実行し，その結果を確認した。人工的なデータではなく，実際のデータを使って実験を行い，各手法のより詳細な比較を試みることが今後の課題である。

この際に，大規模な文書集合に対して，確率分布に基づくモデルがどれだけ有用かどうかも調べる必要がある。例えば，異なり語数 M が 10,000 を超えるような場合，(10) 式中の確率を正確に計算できるかどう

[30] この点での拡張は PLSA に関しても考案されている [26]。なお，語のクラスタと文書クラスタとを同時に生成することを *co-clustering* と呼ぶことがある。混合モデルや LDA，HDP でもこれは容易であるが（例えば，多項混合モデルならば $p_{j|k}$，LDA や HDP ならば $\phi_{j|k}$ の推計値をそれぞれ使えばよい），語と文書とでクラスタ個数を変えることはできない（拡張は可能かもしれない）。

[31] 文献 [18] では，$c_{ij} / \sum_i c_{ij}$ で正規化されている（a_{ij} も同様）。この場合，$\mathbf{B}$ は，語のクラスタと文書クラスタとの共起頻度行列として解釈できるので，結果を理解しやすい。

かには疑問が残る。Gibbs サンプリングを活用すれば，この種の問題は解決できるかもしれないが，大規模文書集合に対して，サンプリングを多数反復することには計算量の点で問題が生じる。このことは行列の計算に基づく NMF や NBVD にもあてはまり，以上の点を考慮すれば，leader-follower 法のような情報検索分野等で伝統的に考慮されてきた手法のほうが，(現実的な) 大規模文書集合に対しては優位なのかもしれない。

なお，本稿は展望論文ではなく，引用文献は網羅的ではない。この点では，Markou & Singh (2003) [21]，Andrews & Fox (2007) [2]，江口 (2009) [34]，Jain (2010) [15]，Long, Zhang & Yu (2010) [18]，高村 (2010) [35] などにおける引用文献リストが参考になると思われる。

参考文献

[1] C. C. Aggarwal and P. S. Yu. On clustering massive text and categorical data streams. *Knowledge and Information Systems*, 24:171–196, 2010.

[2] N. O. Andrews and E. A. Fox. Recent developments in document clustering. Technical report, TR-07-35, Computer Science, Virginia Tech, 2007.

[3] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Text clustering with mixture of von Mises-Fisher distribution. In A. N. Srivastava and M. Sahami, editors, *Text Mining: Classification, Clustering, and Applications*, pages 121–153. Chapman & Hall, 2009.

[4] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, 2005.

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

[6] E. K. P. Chong and S. H. Żak. *An Introduction to Optimization*. Wiley, third edition, 2008.

[7] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative Martrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. John Wiley and Sons, 2009.

[8] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[9] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42:143–175, 2001.

[10] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, second edition*. Wiley, 2001.

[11] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of National Academic Science of the United States of America*, 101(Suppl.1):5228–5235, 2004.

[12] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[13] D. A. Harville. *Matrix Algebra from a Statistician's Perspective*. Springer, 1997.

[14] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.

[15] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[16] K. Kishida. High-speed rough clustering for very large document collections. *Journal of the American Society for Information Science and Technology*, 61(6):1092–1104, 2010.

[17] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[18] B. Long, Z. Zhang, and P. S. Yu. *Relational Data Clustering: Models, Algorithms, and Applications*. CRC Press, 2010.

[19] K. V. Mardia and P. E. Jupp. *Directional Statistics*. Wiley, 2000.

[20] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[21] M. Markou and S. Singh. Novelty detection: a review - part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, 2003.

[22] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, second edition, 2008.

[23] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

[24] R. M. Neal. Markov Chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[25] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.

[26] J.-F Pessiot, Y.-M. Kim, M. R. Amini, and P. Gallinari. Improving document clustering in a learned concept space. *Information Proceesing & Management*, 46:180–192, 2010.

[27] C. P. Robert and G. Casella. *Introducing Monte Carlo Methods with R*. Springer, 2010.

[28] M. J. Rufo, C. J. Pérez, and J. Martín. Bayesian analysis of finite mixtures of multinomial and negative-multinomial distributions. *Computational Statistics & Data Analysis*, 51:5452–5466, 2007.

[29] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[30] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. Technical report, Department of Statisics, University of Berkeley, 2005.

[31] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[32] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–273, 2003.

[33] 岸田和明. 文書クラスタリングの手法：文献レビュー. *Library and Information Science*, (49):33–75, 2003.

[34] 江口浩二. 文書クラスタリング. In **言語処理学事典**, pages 334–339. 共立出版, 2009.

[35] 高村大也. **言語処理のための機械学習入門**. コロナ社, 2010.