CESSA WP 2026-04

社会経済研究のためのテキスト分析 (2): ロシア中央銀行通貨政策ガイドラインのトピック分析

中村靖

岡山香

横浜国立大学

2026 年 7 月

Center for Economic and Social Studies in Asia (CESSA) Working Paper

Downloadable from:

https://www.econ.ynu.ac.jp/cessa/publication/workingpaper/

Center for Economic and Social Studies in Asia, Department of Economics
Yokohama National University

社会経済研究のためのテキスト分析 (2):

ロシア中央銀行通貨政策ガイドラインのトピック分析*

中村靖 (横浜国立大学)
岡山香 (横浜国立大学)

要旨

文書数が少なく，単語種数が多く，トピック数が少ない専門文献に対するLatent Dirichlet Allocation (LDA) トピック分析で生じる問題を検討した．標準的前処理とRのStanzaパッケージで標準英語パイプライン処理の2つのデータを作成し，Rの3つのパッケージの4つのアルゴリズムで分析した．ldaパッケージ以外では，トピック構造の時間変化パターンは，データ前処理，アルゴリズム，Dirichletパラメータ設定にかかわらず大枠で共通であり，LDA分析の頑健さを示す結果となった．一方，どの場合でもトピックの自然言語的意味の同定は困難だった．stanza処理データは，分析対象外単語種を追加的に除去し，Part of Speechタグを有効に使うことができればトピックの自然言語的意味の同定を容易にする可能性がある．

1. はじめに

大量の文書からテキスト分析によって整理された形で情報をとりだす方法は，ビジネス，アカデミックを問わず，標準的かつ必須の分析方法となりつつある．この方法はText mining, Information retrieval, より広い意味では自然言語処理(Natural Language Processing: NLP)など多くの言い方でよばれるが，本稿では「テキスト分析」を使う．

本研究ノートの目的は，文書が取り扱っているトピックを抽出するトピック分析手法を，相対的にデータ量が少なく専門用語が多用される専門文献に適用する場合の有効性と問題点を検討することである．具体的には，Latent Dirichlet Allocation (LDA) モデルを，本研究ノートの第1部(岡山，中村 2026. 以下，研究ノート第1部)で使ったロシア中央銀行「通貨政策ガイドライン」(Monetary Policy Guideline: MPG)に適用する．研究ノート第1部に引き続き，本研究ノートも単語あるいはテキスト分析での分析の基本単位であるtokenをone-hot vector表現で取り扱う分析手法の範囲内にとどまり，単語あるいは文の分散表現(distributed representation)を用いるトピック分析手法は検討しない．Word EmbeddingとNeural Network (NN) モデルを使ったトピック分析は自然言語的意味に近いトピックを抽出するための有効なツールであるが，他方で，16年分のMPGで取り扱っているトピックの構造がど

---

* Text Analysis for Socio-Economic Research, Part 2: LDA Topic Analysis of Central Bank of Russia Monetary Policy Guidelines.
NAKAMURA, Y., Professor Emeritus, Yokohama National University.
OKAYAMA, K., Research Associate, Center for Economic and Social Studies in Asia, Department of Economics, Yokohama National University.

のように変わったかといった問題を簡単に扱うことはできない．NNモデルベースのトピック分析については別の機会に検討する．本研究ノートでは，NNモデルベースのツールであるStanzaを使うが，Stanzaの利用は文書前処理に限られる．

本研究ノートでは研究ノート第1部でおこなったtidytextを使った前処理とStanzaで前処理した場合でのトピック分析の結果の比較もおこなう．研究ノート第1部で単語カウントベースのテキスト分析手法による専門文書のテキスト分析では，専門分野特有の専門用語，特にexchange rate，wholesale price indexといった複合専門用語の識別が問題になることがわかった．これら専門用語は，同じ単語でありながら，文脈に依存して日常的な意味とは異なる意味で使われることがある．MPGでは，多くの場合，exchange rateは「交換」と「率」ではなく，「外国為替レート」である．ただし，「多くの場合」であり，常に「外国為替レート」であるとは限らない．Stanzaによるデータ前処理がこの問題にどの程度対処できるかを検討する．

StanzaはUniversal Dependencies (UD)標準に従った多言語対応，UDコーパスデータによる事前学習，python APIによる使いやすさと追加学習を含むモデル拡張の柔軟性といった優れた特徴をもつ標準的NLPパッケージである(Qi et al. 2020)．UD標準は，多言語間で，分析単位や品詞体系について同一の定義，ラベルを与え，同じ形式のデータを作り，同じ自然言語処理の適用を可能にし，それにより分析結果を多言語間で比較可能にすることを目的としている (Asahara et al. 2018; Nivre et al. 2020)．Stanzaは，2026年1月現在で70言語に対応している．Stanzaが，UDの用語法に準拠してdocument，text，token，word等の用語を用い，さらにコード中でもdoc，text，token，word等を変数名，ラベル名等を使っているため，本稿でのこれらの用語の意味は日常的な意味とも，研究ノート第1部における意味とも若干異なる．理解の妨げとなるような問題は無いが，tokenとwordの関係はやや注意を要する．この点については，Stanzaの処理内容とともに検討した方がわかりやすいため，第2節でみる．分析対象文書が多言語の文書である場合，Stanzaは前処理ツールの力な候補である．

LDAモデルは，単語のone-hot vector表現の範囲内では現時点でもっともよく使われるモデルである(Hanker, Kasri and Beni-Hssane 2025)．LDAモデルは，単語は単語(語彙)の集合から確率的に取り出されたものとするBag of wordsを仮定する(Blei, Ng and Jordan 2003; Blei 2012)．Bag of words仮定により，LDAモデルは単語間の前後関係を無視し，したがって文章の自然言語的意味を保持しないことになる．Bag of words仮定はトピック分析を柔軟，容易，普遍的におこなうことを可能にする一方で，抽出されたトピックが自然言語的意味をもつトピックなのかという問題が常につきまとう(Chang et al. 2009)．

本稿の以下の構成は次のとおりである．第2章は，Stanzaについて簡単な解説をおこない，Stanza処理データを作成する．通常処理データは研究ノート第1部で作成したデータと同じである．続いて，LDAの理論とLDAを実際におこなうためのR用パッケージとそれらが用いるLDA計算用アルゴリズムの違いについて簡単な解説をおこなう．R用パッケージのみである理由は，本研究ノートがRで作成されているという以外に理由はない．最後に，通常処理データとStanza処理データをLDAで分析した結果を比較し，LDA計算アルゴリズムの違いとデータ前処理の違いがLDAの結果に与える影響を検討する．

本研究ノートは第1部と同様にR Quarto (https://quarto.org)で作成され，QuartoのqmdファイルはMPGデータとともにAppendixのURLからダウンロードできる．本研究ノートは研究ノート第1部のような

コードへの参照指示を紙幅の制約によりおこなわない．qmdファイルは，本研究ノートの文書，図表とともに通常処理用コードを含むすべてのコードを含んでいて，利用しているパッケージの固定的バージョンでのインストール，MPGデータ (16個のcsv files) の読み込み，pythonのインストール等を含むすべてのコードを簡単に実行できる．コードについてはqmdファイルを直接参照していただきたい．利用しているRのバージョンはR 4.5.3である(R Core Team 2025).

2. Stanzaを使ったテキスト分析データの作成

2.1 自然言語処理ツールとしてのStanza

StanzaはStanford大学の自然言語処理グループ(Stanford NLP Group)が開発している自然言語処理用ツールである(Qi et al. 2020)．Stanzaの主な機能として，(1) 分析対象textをtoken化 (tokenize)，(2) 複数語の結合表現を分解するmulti-word token expansion (mwt)，(3) tokenに品詞を付与するpart-of-speech tagging (pos)，(4) is，areをbeにまとめる語形統一 (lemma)，(5) 主語述語，修飾被修飾といったtoken間の係り受けを決めるdependency parser (depparse)，(6) 人名，組織名，地名，日付，数量など固有名詞，固有表現を識別するNamed Entity Recognition (ner) がある．これらの機能はprocessorと呼ばれるそれぞれpythonで書かれたツールがおこなう．カッコ内はStanzaがprocessorに与えている名称である．それぞれのprocessorはpython apiを用意しており，個々のprocessorを独立に使うことも，いくつかを組み合わせて使うことも，関数と同等の形式で簡単におこなえる．複数のprocessor のシリアルな組合せをpipelineと呼ぶ．text file形式のtextから分析を開始する場合は，各言語により使うべきprocessorの種類と適用順はほぼ標準的に決まる．英語の場合は，tokenize, mwt, pos, lemma, depparse, nerの順でprocessorを適用する以外の設定はあまり考えられないため，Stanzaは，このpipelineを英語用標準pipelineとしてデフォルトで用意している．

Stanzaの各processorは，それぞれがNNモデルや確率モデルを組合せて処理をおこなっている．Stanzaが使うNNモデルは，基本的にBiLSTM (Bidirectional Long Short Term Memory)を備えたRecursive Neural Network (RNN) モデルである．文中の単語を分析対象とする場合を例とすると，NNモデルがおこなう処理のイメージは，BiLTSMを用いて単語をencode，つまりvector表現を単語に与え，encodeされた単語情報をもとにして単語列の最初から次に来る単語を予測して生成する．ただし，テキスト分析に使う場合は，分析対象となるオリジナルのテキストが既に存在しているから，次に来る単語を予測する必要はない．decoderの出力は，基本的には単語そのものではなく，単語に付加する情報(annotation)である．posであれば対象の単語に付加する品詞の情報であり，lemmaであればオリジナル文中のareに対して語形統一した場合の情報としてbeを付加する．したがって，StanzaのNNモデルの機能は生成ではなく，翻訳により近い．

LSTM (Long Short Term Memory)は，分析対象単語をスキャンするとき，現在注目している単語の前までの分析結果(隠れ状態ベクトルh_{t-1})をどの程度の量まで保持すべきか決め，現在の分析対象単語の情報を得て，h_{t-1}に付加し，これを現在の分析対象単語の隠れ状態ベクトルh_tとして出力する機構である(Schmid 2019)．これによって，現在注目している単語から離れた単語の情報を考慮できる．Bidirectionalの意味は，文の先頭から分析対象単語までと文の終端から分析対象単語までの両方向でスキャンした結果の隠れ状態ベクトルを結合して，分析対象文字の隠れ状態ベクトルとすることを意味している．BiLTSMによって，現在分析対象としている単語の状態を，分析対象単語の前と後の両方向に離れている単語の情報をともに考慮した上で決定することができる．Decode段階では，当

然であるが，単語列の先端から逐次処理していくため，Bidirectionalではなく，先端から終端へ方向へのLTSM機構のみが働くことになる．

Stanzaの各processorが用いるNNモデルは，それぞれのprocessorの機能に合わせて学習済みであり，processor利用中にNNモデルのパラメータが再調整されることは無い．Stanzaの各processorが使うNNモデルは，UD Treebank corpus を使って学習し，UDモデルに準拠したannotationを出力する(de Marneffe et al. 2021)．UDモデルは，既にのべたように，単語に品詞，係り受け等のannotationを多言語間共通のフォーマットで付与するものであり，UD Treebank dataは，UD annotation付きのcorpus dataである．つまり，各NNモデルはUD Treebank dataのannotationをGold annotation (教師となる正解)として学習済みである．

Stanzaの分析結果の精度は一般的に高いといわれる．しかし，StanzaのNNモデルがUD Treebank dataに対して最適化されている以上，分析対象テキスト，本研究ノートの場合ではMPGに対しても高い精度を必ず達成できるという保証はない．Stanzaは分析者が用意した学習データで各NNモデルを再訓練する学習モードをオプションとして用意している．BiLTSM機構付きRNNの訓練は，TransformerベースのNNモデルよりはハードウェアに対する要求は低いが，最低限でRTX5070クラスのミドルハイレンジ以上のGPUが必要である．GPUなしでは，100万words程度の再訓練に数週間か場合により一か月以上かかるといわれる．なお，後に報告するように，25-40万語程度(前処理により単語数は変わる)であるMPGに対する学習なしのStanza英語pipeline処理にCore i9-10900X，Nvidia Quadro P1000, PCIe 3.0×4 NVMe SSD, DDR4-2666 256GB内部メモリ, 1.3倍overclockのPCで70分以上かかる．ハードウェア以上に問題となるのが正解annotation付きの少なくとも100万単語以上の教師データの準備である．データ作成作業の問題とともに，データ数が40万単語程度のMPG用に訓練するために，100万単語以上のどのような教師データを用意すればよいのかを見極めることが難しい．

2.2 Stanzaによる前処理

ここでは，通常処理データ作成におけるtoken化との違いを検討するために必要な範囲で，Stanzaの標準英語pipelineでおこなう処理内容を概観する．

2.2.1 tokenizer

標準英語pipelineの最初のprocessorは分析対象文書を単語 (token) に分割するtokenizerである．通常処理データ作成の場合は，与えられたtext中のピリオドやスペース文字をラベル(マーカ)として外形的なルールベースでtoken化する．一方，Stanzaのtokenizerは与えられたtextを文字(character)単位で処理する．tokenizerは，各文字にその文字がtokenの終りか，sentenceの終りか，形態的に1単語だが文法的には複数語から成る縮約語等であるMWT (Multi-Words Token) の分割点かを示すラベルをNNモデル(BiLSTM)を使って与える．各文字にラベルがあたえられると，そのラベルを使ってNNモデル(LSTM)でテキスト分析用のtokenおよびsentenceを再構築する．tokenizerは文字単位のencode，decodeによりtokenを決定するため，未学習語に強いといわれる(Qi et al. 2020)．新造語，ミスタイプ，読み込み書き込みエラー等による文字単位の異常にも対応できる可能性が高い．例えば，“tokenizar”の入力に対して正しく”tokenizer”と出力する可能性が高い．なお，tokenizerは，既存token化マーカ利用，sentence分割，トレーニングモードでの実行等を指定するオプションを持っている．本研究ノートではすべてのprocessorはデフォルトのままで利用し，NNモデルの再訓練もおこなわないため，processorのオプションについては説明しない．これらについては，Stanzaのホームページを参照され

たい(stanfordnlp.github.io/Stanza/).

2.2.2 mwt

標準英語Pipelineの次のprocessorはMWT対象語を実際に個々のtokenに分割するmwtである．tokenizerがMWTの分割点となる文字を既に識別しているが，mwtが実際に分割する．MWT処理の重要性は分析対象言語によって大きく変わる．英語では，数字と単位が連結した10kmのような表現を10とkmに分ける，don'tのような否定の省略形をdoとn'tに分ける程度のわずかな場合しかMWTの対象にならないが，わずかながらもMWT処理の対象となる単語が存在するため英語用pipelineはデフォルトでMWT処理をおこなう．MWT処理は，イタリア語，フランス語では必須，ドイツ語では必要性は低く，ロシア語ではほぼ意味が無く，中国語，日本語はそもそも必要性がない．mwtはtokenizerが識別したMWT分割点の文字で機械的に分割するのではなく，適切なMWT分割後のwordをmwtのNNモデルによって探す．例えば，フランス語のduはdeとleに，イタリア語のdellaはdiとlaに分割する．適切な分割後のwordが見つからない場合は，分割点での機械的分割結果を分割後のwordとして保存する．NNモデルを使った文字単位の処理によりmwtは高い精度でMWTを処理ができるといわれる．

なお，tokenとwordの用語法が研究ノート第1部とは若干異なっている．UDに準拠し，Stanzaは，分析対象テキスト中の表記上の分析基本単位をtoken，文法的な基本単位をwordと呼ぶ．MWTの場合はMWT処理で分割した後の各単位をwordとよぶ．Stanzaの処理出力データは，1つのMWT tokenがMWT分割後の複数のwordを持ち，各wordが完全なannotation情報を持つデータ形式になっている．したがって，Stanzaによるテキスト分析の基本単位は，内容上は文法上の単位であるwordということになる．しかし，圧倒的に多くのwordがMWT tokenではなく，token = wordであるためか，Stanzaのマニュアル，解説等でも，分析対象テキスト上の表記としても，Stanza処理用Rコードの表記上としても，分析の基本単位をtokenとみなして記述している場合が多い．「UDに準拠したStanzaは，分析の基本単位としてtokenとwordの両面をみている」と理解するしかない．本研究ノートでは，研究ノート第1部の用語法に合わせて分析の基本単位を指してtokenを使うが，tokenには自然言語上の「単語」の意味も持たせて使うこととする．厳密な区別と定義が必要な場合は，その都度注意して表記する．

2.2.3 pos

pos processorは，BiLSTM機構とbiaffine classifierを使ったNNモデルにより，各tokenにuposタグ，xposタグ，featタグを付ける．uposは各言語共通の品詞ラベル，xposは各言語固有の品詞ラベル，featは時制，格，性などの形態素情報である．BiLSTMにより注目しているtokenの前後のtokenのpos情報を考慮することで，例えば，“動詞，動詞”と続くことは少ないというように，pos系列の合理性を保つ．posは個々のtokenのupos，xpos，featの3つのタグ相互間の不一致を避けるためにbiaffine classifierを使っている．biaffine classifierは，2つの要素のペアを入力として要素間関係を分類するための標準的方法である (Dozat and Manning 2017)．StanzaのbiaffineclassifierはStanza用に実装されているが，一般的なbiaffine classifierと大きな違いはないため，ここではbiaffine classifierの説明は省略する． pos processorのNNモデルは，UD Treebank dataから正しいタグ付けを学習済みである．

なお，Stanzaはpos processorのタグ最適化にConditional Random Field (CRF)を当初は使っていたが，現在はbiaffine classifierに替えている．やや古い説明ではCRFで示している場合もある．

2.2.4 lemma

lemma processorは語形の統一をおこなう．英語の場合はルールベースによるlemma化で十分であるため，英語lemma processorはNNモデルを使っていない．他の多くの西欧言語でも，語形の変化規則が相対的に単純であり，UD lemma辞書が充実しているため，ルールベースのlemma化にとどまる．ただし，Stanzaのlemmaは，posで付けたtag情報も使っているので，NNモデルを間接的に使っているといえる．日本語の場合はtoken化を語彙単位(dictionary word)でおこなうため，UDは日本語lemmaを用意しておらず，lemma processorによる語形統一もない．日本語の場合は，表記の揺れ等の別の次元での語形統一が問題となる．対照的に，ロシア語の語形変化規則は複雑すぎるため，ロシア語lemma processorはNNモデルを使っている．本研究ノートでは英語lemma processorしか使わないため，他の言語のlemma processorについての説明は省略する．

2.2.5 depparse

depparse processorは各tokenにUDの依存関係ラベルを与え，各tokenデータのdeprel属性に格納する．deprelは，主語，目的語，斜角補語，助詞，従属節，複合語，主要述語といった，現在注目しているtokenと他のtokenとの依存関係の種類をあらわす．deprelとともに，現在注目しているtokenの依存関係の対象となっているtokenを識別し，対象となっているtokenの情報(tokenのID番号)を各tokenデータのhead属性に格納する．headが示しているものは，deprel依存関係において分析対象tokenが従属しているtoken，あるいは分析対象tokenが修飾しているtokenを指している．2つのtokenのどちら側からtoken間の関係をみているかという見方の違いであるが，混同しやすいので注意する必要がある．例えば，主語のheadは文の主要な内容である述語である．主語が述語を修飾し，主語は述語に従属している．一方，述語は何も修飾しないので述語にはheadはなくrootである．依存関係は，ツリー型データ構造における親ノード(head)と子ノード(分析対象token)の関係とみなすとわかりやすい．

depparseの処理は，BiLSTM NNモデルで得られた各tokenの隠れ状態ベクトルを入力として，子tokenのhead用，親tokenのhead用，子tokenのdeprel用，親tokenのdeprel用の4種類の特徴量を抜き出すための多層パーセプトロン(MLP)を用意している．より正確には隠れ状態ベクトルだけではなく各tokenの言語特有品詞情報であるxposラベルの情報も使っている．4つのMLPの出力である4つの特徴量を使ってaffine scoreにより，あるtoken iがheadである確率を最大とするtoken jのhead-dependentペアを探す．その結果，すべてのtokenに対して最適なdeprel，headが与えられる．

2.2.6 ner

標準英語pipelineの最後のprocessorは，固有名詞，人名，年，単位等の固有表現を抽出し，tokenにUD固有表現ラベルを与えるとともに，固有表現全体をspanとして格納するner (named entity recognition)である．例えば，“Yokohama National University”は，tokenとしては3つのYokohama，National，Universityであり，それらのtokenはUD NERラベルとして組織名をあらわすORGをもつ．同時に，固有表現としての”Yokohama National University”を1つのspanとして保持する．

処理手順としては，lemma化まで処理された分析対象textをミニバッチ化し(デフォルトで32文長)，BiLSTM機構を持つNNモデルでエンコードする．エンコード結果を入力として，CRFによってtokenの系列全体に最適なBIOES付きのNERタグを与える．BIOESは1つの固有表現内で各tokenがどの位置にあるかを示し，NERタグが固有表現の種類を示す．BIOESは，Begin，Inside，Outside(固有表現で

はない)，End，S(単一tokenの固有表現)である．NERタグは構文についてのannotationの標準を定めるUDの標準化の対象になっていない．人名PER，組織名ORG，地名LOC，その他MISCの4種類のタグを付けるCoNLL-2003が事実上のUD標準になっている．ただし，より細く種類分けしたNERタグを持つコーパスも多い．例えば，国立国語研究所のUD_Japanese-BCCWJ (Asahara et al. 2018) は，CoNLL-2003に作品名，日付，時刻，金額表現，割合表現を加えている．

2.2.7 他のprocessor

Stanzaには，文の文法的構造を表現するタグを与えるconstituency，tokenに感情評価タグを与えるsentiment，分析対象テキスト内で同一の実体を指す表現，例えば，Central Bank of Russiaとロシア中央銀行を指すItを1つのchainにまとめるcoref，言語を判定するlangidといったprocessorがある．nerとcorefの出力結果を使うと，文書ごとの重要な人物，組織，地名をそれらの別表現，代名詞表現をワンセット(chain)として抽出できる．それにdepparseの出力を組み合わせると，文書の意味上の主題によるトピック分析が可能になる．

2.3 Stanza処理データの作成

本研究ノートのqmdファイルはStanzaデータ処理用のコードを含んでいる．Stanzaはpythonで書かれているため，本研究ノートのqmdファイル中では，Rのプロジェクトフォルダにインストールしたpythonを Rのreticulateパッケージを使ってRから利用する設定としている．qmdファイル中のコードを実行する場合は，gmdファイルを置いたR projectフォルダ内にpythonをあらかじめインストールしていただきたい．別の設定でpythonを使っても問題はない．問題が生じるとすれば，特にwindows環境で，Conda等を使ってpythonをインストールした場合，複数のpythonのうち特定のpythonのinstallationを利用する場合である．これは，特にwindows環境では一般的に起こりえる問題である．ここでは特に説明はしないが，特定のpython installationの利用をRに指定する方法ついてはWeb上に多くの解説がある．なお，qmdファイル内のRコードでは，stanz英語パイプラインを動かすmain_trnコードチャンクをコメントアウトして，Stanzaを動かさない設定にしている．MPGのStanza処理だけでCPU処理時間が70分以上かかるためである．main_trnコードチャンクによってStanza処理したデータdf_mpg_stz.rdsは，qmdファイルとともにとしてリポジトリに置いてある．main_trnコードチャンクのコメントアウトを解除してStanza処理をおこなう場合も，コメントアウトの解除以外は特にコードを変更する必要はない．Stanza英語標準pipelineの出力のどの部分をdf_mpg_stzとして保存しているかについては，qmdファイル中のコードを参照していただきたい．

2.4 通常処理データの作成

通常処理の内容は研究ノート第1部とおなじである．研究ノート第1部およびqmdファイルのコードチャンクOHV_MPGを参照されたい．

3. Topic modelling

Topic modellingは，複数のdocumentから成る文書全体を分析対象とし，各documentが扱うトピックを抽出することを目的とする．Topic modellingの手法は大きく2つに分けられる(Hanker Kasri and Beni-Hssane 2025; Laureate, Buntine and Ling 2023)．1つはBag of wordsを仮定する確率モデルによる分析である．Bag of words仮定は，単語種類(word type)の集合(vocabulary)から確率的に取り出された単語の

系列が文書であると仮定する．さらに，個々の単語の取り出しは，基本的に他の要因とは独立であると仮定するため，単語間の関係や文脈は消滅し，考慮されない．もう1つの方法は，単語間の関係あるいは文脈を考慮してトピックを抽出することを目指す大規模言語モデル(LLM)を用いたトピック分析である．本研究ノートでは，前者の確率モデルによる分析のうち，現在主流であるLatent Dirichlet Allocation (LDA)による分析のみを取り上げる．

3.1 LDAによるトピック分析

3.1.1 LDAの基本的アイディア

ここではLDAの基本的アイディアを直感的にのみ説明する．説明にもちいる主要な記号はTable 1のとおりである．まず前提として，D個の文書d ($d \in 1 \cdots D$) に分けることができる分析対象となる文書がある．本研究ノートの例では，16年分のMPG全体がDであり，各年のMPGが個々の文書dであり，D＝16である．LDAは，文書はその文書で用いられている単語種(vocabulary)の集合から一定の確率にしたがって取り出された単語の系列であるとみなすBag of wordsを仮定する(Blei, Ng and Jordan 2003)．つまり，分析対象文書中の単語の順序関係は考慮しない．$w_{d,i}$は観察された文書dのi番目の単語を意味しているが，LDAモデル内では位置(d,i)は計算の便宜上つけたインデックスとしての意味しかなく，自然言語上の単語の並び順という意味はない．単語種と単語の関係についても注意を要する．単語種を重複カウントする単語数N_W，$N_{d,W}$と単語種を重複カウントしない単語種数N_V，$\mathrm{N}_{\mathrm{d,v}}$は通常一致しない．事前にわかる関係は$N_W \leq N_V$だけで，$N_{d,w}$と$N_{d,v}$の大小関係にはあらゆる場合がありえる．

Table 1: Notations

記号	意味	説明
D	分析対象文書の全体あるいは総数	
W	Dに含まれる単語全体の集合	
V	分析対象文書が使う単語種全体あるいは総数	Wの重複なし集合
K	分析対象文書に含まれるトピック全体あるいは総数	
d	Dを構成する個々の文書あるいはそれを指すindex	$d \in \{1, \cdots, D\}$
v	Vを構成する個々の単語種あるいはそれを指すindex	$v \in \{1, \cdots, V\}$
k	Kを構成する個々のトピックあるいはそれを指すindex	$k \in \{1, \cdots, K\}$
$w_{d,i}$	文書dのi番目の単語	文書dのi番目の単語(観察データ)
$z_{d,i}$	$w_{d,i}$に割り当てられたトピック	$z_{d,i} \in \{1{:}K\}$
N_W	文書全体Dの総単語数	
$N_{d,W}$	文書dの単語数	
$N_{d,v}$	文書dのある単語種vの数	$N_{d,w=v}$と同じ
$\theta_d^{(K)}$	文書別トピック分布のパラメータ(K項分布)	K次元. 添え字を省略する場合がある
$\phi_k^{(V)}$	トピック別単語種分布のパラメータ(V項分布)	V次元. 添え字を省略する場合がある
$\alpha^{(K)}$	$\theta^{(K)}$ を与えるDirichlet分布のパラメータ	K次元. 添え字を省略する場合がある
$\beta^{(V)}$	$\phi_k^{(V)}$を与える Dirichlet分布のパラメータ	V次元. 添え字を省略する場合がある
$\hat{x}$	xの推定値	省略する場合がある

LDAモデルの基本的な考え方は，次のとおりである．

(1) 各文書dはK個のトピックのミックスからなり，トピックのミックス，つまり文書別トピック分布は文書ごとに変わる可能性がある．文書別トピック分布はK個のパラメータ$\theta^{(K)}$をもつK項分布であらわせると想定する($P(k \mid d) = multinomial\left(\theta^{(K)}\right)$)．分析前はトピックス自体が未知であるから，文書別トピック分布をデータから直接観察することはできない．なお，コードの多項分布パラメータは各項の出現確率と試行回数を含む場合が多いが，LDAの説明においては多項分布パラメータに試

行回数を通常は入れないため，本研究ノートもこれにしたがう．

(2) 各トピックkはそのトピックに固有の単語種出現のパターン，つまりトピック別単語種分布を持つ．トピック別単語種分布はV個のパラメータ$\boldsymbol{\phi}^{(V)}$をもつV項分布であらわせると想定する($P(v \mid k) = multinomial(\boldsymbol{\phi}^{(V)})$)．$P(v \mid k)$の条件に$d$がないことが示すように，トピック別単語種分布は全文書Dで共通であり，個々の文書dには依存しないと想定する．トピックkが知られていないので，$P(v \mid k)$も直接観察することはできない．

(3) 文書d中のi番目の単語$w_{d,i}$の単語種がvであったとき，$w_{d,i}$の事後的な出現確率$P(w_{d,i} = v)$の期待値は，トピック別単語種分布$P(v \mid k)$を文書dのトピック分布$P(k \mid d)$で加重平均した$p(w_{d,i} = v) = \sum_k [p(v \mid k) \cdot p(k \mid d)]$ で決まると想定する．確率$p(w_{d,i} = v)$の期待値は，文書d中の単語種vの単語数$N_{d,w=v}$の文書dの総単語数$N_{w,d}$に対する比$p(w_{d,i} = v) = \frac{N_{d,w=v}}{N_{d,w}}$ であると考えられるので，観察データから直接観察可能である．

(4) 観察データから計算される$p(w_{d,i} = v) = \frac{N_{d,w=v}}{N_{d,w}}$はトピック$k$の情報を持たないため，観察された$p(w_{d,i} = v)$から$p(w_{d,i} = k)$の情報は得られない．つまり，位置($d$,$i$)の単語のトピックがある特定の$k^*$である確率$p(z_{d,i} = k^*)$は観測データからはわからない．$z_{d,i}$は観測できない潜在変数 (Latent variable)である．

(5) LDAモデルの(1)と(2)から$\hat{p}(w_{d,i} = v) = \sum_k [p(v \mid k) \cdot p(k \mid d)]$が計算でき，観察データからは$\mathbb{E}[p(w_{d,i} = v)] = \frac{N_{d,w=v}}{N_{d,w}}$を計算できる．LDAモデルの目的は，単語全体について，観察された$p(w_{d,i} = v)$になるべく近くなる$\hat{p}(w_{d,i} = v)$を与える多項分布パラメータ$\theta^{(K)}, \theta^{(V)}$を得ることである．

(6) 多項分布パラメータ$\theta^{(K)}, \boldsymbol{\phi}^{(V)}$の情報はほぼ無いため，最適なパラメータ値を自動的に計算する．そのため，多項分布パラメータ自体が確率的に決まると想定し，一定の制約のもとで多項分布パラメータの値が動けるようにする．制約がないとパラメータがとり得るあらゆる値を試してみなければならず，計算必要量が無限大になる．LDAでは，多項分布パラメータの標準的な事前分布であるDirichlet分布で多項分布パラメータがとり得る範囲を制約する．N項分布のN個のパラメータはそのN個の各項の出現確率の期待値である．Dirichlet分布は，Dirichlet分布自身のパラメータにしたがって，N個のN項分布パラメータを一定の分散のもとで生成する．Dirichlet分布の分散がゼロでない限り，多項分布のパラメータは変化できる．

サイコロのアナロジーを使うと，1-6までの目をもつ正常なサイコロをN回ふった時にそれぞれの目が何回でるかをNで割った値が6項分布のパラメータである．イカサマ用でない正常なサイコロでは，6項分布の6個のパラメータすべてが均一の1/6である．サイコロが1の目が出やすいイカサマサイコロであるとき，6項分布パラメータは（1/2,1/10,1/10,1/10,1/10,1/10）や（3/4,1/20,1/20,1/20,1/20,1/20）かもしれない．6項Dirichlet分布は，このようなイカサマサイコロの6項分布パラメータを含む6項分布パラメータの候補を生成することができる．Dirichlet分布がどのような多項分布パラメータ候補を生成するかは，Dirichlet分布自身のパラメータによって決まる．この点については，3.1.4-3.1.5で検討する．このように，Dirichlet分布とMultinomial分布(Dirichlet-Multinomial分布)にもとづいて潜在変数$z_{d,i}$のトピックを決めるモデルであることがLatent Dirichlet Allocation (LDA)モデルの名前の由来である．なお，サイコロのアナロジーは誤解を生みやすいものの，多項分布をイメージしやすい．以

降も，サイコロのアナロジーをしばしば使う．

以上のLDAの基本アイディアは難解ではない．「石油価格」，「制裁」，「輸入代替」，「資源依存」などの単語が頻繁にあらわれるトピックはロシア経済に関するトピックと予想でき，「世界の工場」，「EV」，「不動産不況」，「五カ年計画」などの単語が頻繁にあらわれるならば「中国経済」に関するトピックと予想できる．ただし，ロシア経済，中国経済のトピックで共通に使われる単語も多い．経済に関する文書であれば，「経済」という単語はどちらのトピックにも高頻度であらわれるだろう．実際上は，特定のトピックだけにしか使われない単語は稀であり，大部分の単語は多かれ少なかれどのトピックにも使われるだろう．ある文書$d^{\star}$が，ロシア経済と中国経済の両方のトピックをあつかっているが，文書の長さ(総単語数)の9割ほどをロシア経済に，1割ほどを中国経済にあてているとする．文書$d^{\star}$での単語種分布$P(v \mid d^{*})$は，ロシア経済トピックの単語種分布に近いものの，中国経済トピックにあらわれる単語種もあらわれる．ただし，中国経済トピック単語種の文書$d^{\star}$における出現頻度は中国経済だけを扱う文書での出現頻度と比べれば1/10程度になると予想できる．計算上問題となるのは，ロシア経済トピックと中国経済トピックの両方に現れ，出現頻度だけが違う単語種が相当数存在することである．単語種は観察可能であるが，同じ単語種の単語がロシア経済トピック由来か中国経済トピック由来なのか判別できない．LDAモデルは，全文書に共通なトピック別単語種分布と各文書別トピック分布とがそれぞれ独立に存在するとの前提のもとで，これら分布の情報を組み合わせることで，その単語がどのトピック由来であるのかを推定するモデルである．

LDAモデルを実際に計算するためには，まずトピックの総数Kを天下り的に設定する．トピック分析をおこなう前提は，通常は，その文書が何のトピックを扱っているかわからないことである．したがって，トピック総数Kを事前に決定することは困難である．Kがなんらかの方法で決まれば，全文書Dのうちの特定の文書dにおける文書別トピック分布はK項分布$k_d \sim Multinomial\left(\theta^{(K)}\right)$にしたがい，特定のトピック$k$における$V$個の単語種の分布は$V$項分布$v_k \sim Multinomial\left(\phi^{(V)}\right)$にしたがうと想定する．多項分布の各項目の出現確率である多項分布パラメータ$\theta^{(K)}$と$\phi^{(V)}$は，それぞれDirichlet分布$\theta^{(K)} \sim Dirichlet\left(\alpha^{(K)}\right)$と$\phi^{(V)} \sim Dirichlet\left(\beta^{(V)}\right)$にしたがって決まり，Dirichlet分布パラメータ以外にそれぞれの多項分布に影響する要因はないと想定する．

ここで記号(~)は，右辺の確率分布から左辺の値をサンプルすることを意味している．確率分布からサンプルするとは，サイコロのアナロジーを使うならば，K面サイコロをK回(あるいはK個のK面サイコロを1回)振ってある文書dのトピック分布を決めることを意味している．同様に，トピックkが与えられたときに，V面サイコロをV回(あるいはV個のV面サイコロを1回)振ってトピックkの語彙分布を決める．ここで，各文書用に異なるD個のK面サイコロを使うのか，各トピック用に異なるK個のV面サイコロを使うのか，という問題が生じる．この点については，3.1.3で検討する．とりあえず，1種類のK面サイコロ，1種類のV面サイコロを使うと想定する．1種類のサイコロでも，サイコロをふった時に出る面は異なる．

文書が扱うトピックには偏りがあり，各トピックが使う単語種にも偏りがあると予想することは自然である．つまり，K面サイコロとV面サイコロは，各面が同じ確率で出るのではなく，出やすい面と出にくい面のある歪んだイカサマサイコロである．Dirichlet分布がこの歪んだサイコロの面の出る確率を制御する．K面サイコロ用のDirichlet分布パラメータをα，V面サイコロ用のDirichlet分布パラメータをβとすると，α仕様のDirichletサイコロを振ってK面サイコロの多項分布パラメータθ，つ

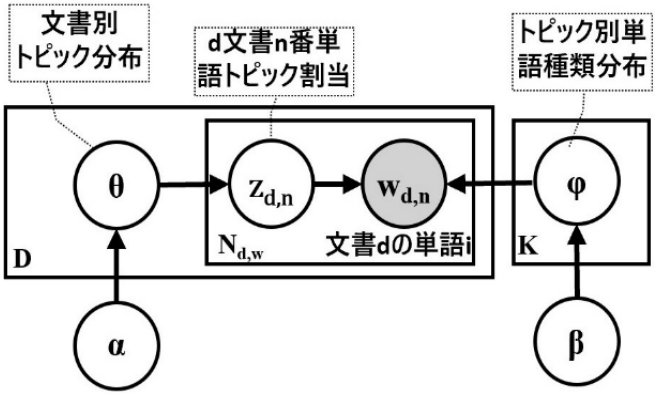
まりK面各面の出目確率を決め，β仕様のDirichletサイコロを振ってV面サイコロの多項分布パラメータϕ，つまりV面各面の出目確率を決める．K面サイコロ，V面サイコロは，Dirichlet分布と多項(multinomial)分布の2段構えで作られるので，Dirichlet-Multinomial分布という呼び方をすることもある．ただし，Dirichlet分布は目が出る確率(期待値)を与える．その総計は1であるが，個々の目の出る確率は0-1の間のどのような値でも取り得る．Dirichletサイコロは無限の面を持つサイコロということになる．無限の面を持つサイコロは作れないため，サイコロのアナロジーは不正確となる．

LDAの実際の計算については以下で検討するが，基本的には，高速に計算するためにK面サイコロ，V面サイコロを振ることをシミュレーションするような計算方法は使わない．この点でもサイコロのアナロジーは不正確である．ただし，具体的な計算方法にかかわらず，LDAモデルの本質は，文書d用にK面サイコロを振って単語のトピック$k^{\star}$を決め，その$k^{\star}$用にV面サイコロを振って単語種をきめることを全単語数N_W回繰り返すモデルである．K面サイコロ，V面サイコロがあれば，この方法で全文章Dを生成できるため，LDAモデルは生成(Generative)モデルの1つである．ここで注意すべき点は，LDAモデルが生成した文書は，まったくの偶然以外，自然言語的な意味を持たない文書であることである．Bag of wordsを仮定するLDAは，文書の単語の並び順には関心がなく，単語の分布だけに関心を寄せているためである．特定の単語位置(d,i)でトッピク k であり，そのトピックkにおいて異なる単語種，例えば”girl”, “boy”が同じ出現確率である場合，LDAはgirlとboyのどちらが選ばれるかについて無関心である．この文書が”I am a”であり，続く4番目の単語$(d,4)$がトピックk由来であるとされたとき，その文章が”I am a girl”でも”I am a boy”でもLDAモデルにとっては同じである．LDAモデルは，kトピック中の単語種”girl”と”boy”の総出現回数が全文書内で最終的に同じであることだけを注目している．

3.1.2 ベイズ確率モデルとしてのLDAモデル

LDAモデルをグラフィカルモデルであらわすとFigure 1のようになる．

Figure 1. LDAのグラフィカルモデル



Note: The rectangles indicate the multiple plates with the numbers of D, $N_{d,w}$ and K. The grey circle is observable.
Source: Authors' drawing.

Dirichletパラメータが，$\alpha_k = \text{const.}\ i \in \{1{:}K\}$，$\beta_v = \text{const.}\ v \in \{1{:}V\}$のようにすべて均一の値で固定であるとすれば，LDAの確率分布は，

$$p(w,z,\theta,\phi \mid \alpha,\beta) = \prod_{k=1}^{K} p(\phi_k \mid \beta) \prod_{d=1}^{D} \left[p(\theta_d \mid \alpha) \prod_{n=1}^{N_d} p(w_{d,n} \mid z_{d,n},\phi) p(z_{d,n} \mid \theta_d) \right] \quad (1)$$

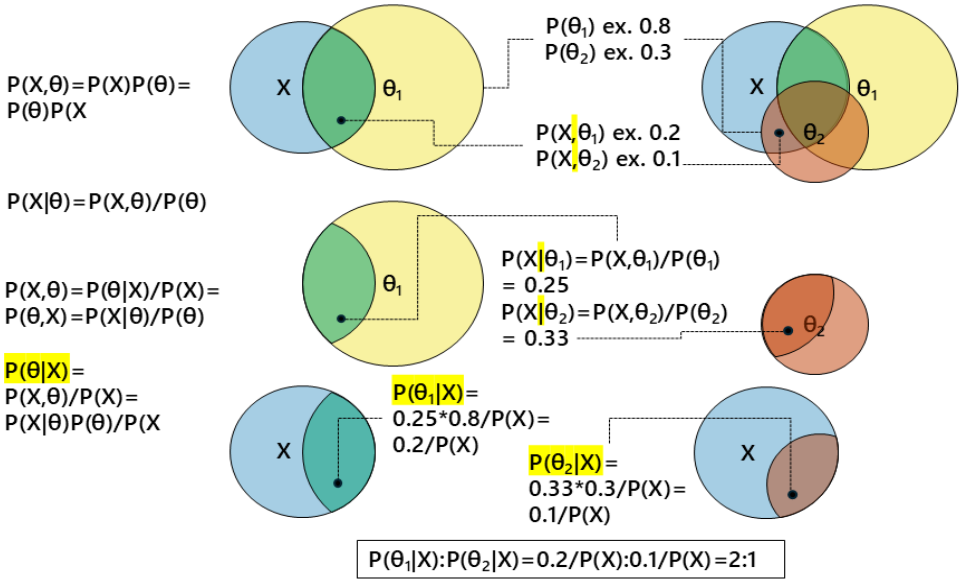
となる．なおDirichletパラメータの要素の値が均一に与えられている状態を対称(symmetric)と呼ぶことが一般的であるので，以下ではそれに従い「対称固定」とよぶ．

このLDAモデルは，多項分布パラメータの事前分布に多項分布の事前共役分布であるDirichlet分布を設定したベイズ確率モデルである．まずベイズ統計の基本概念について確認しておく．ベイズ統計の考え方の基本は，直感的，概念的には非常に単純なベイズ定理に要約される．ベイズ定理は，

$$\begin{aligned} p(\theta)p(x) &= p(x)p(\theta) \\ p(\theta \mid x)p(x) &= p(x \mid \theta)p(\theta) \\ p(\theta \mid x) &= \frac{p(x \mid \theta)p(\theta)}{p(x)} \end{aligned} \quad (2)$$

であらわせる．1行目は自明の等式である．θと同時にxが起きる確率は，xと同時にθが起きる確率と同じである．2行目は，条件付確率の記号$p(A \mid B)$で書き換えただけである．この記号は，Bが起きた前提でAが起きる確率を意味している．Figure 2のベン図をみれば直感的に理解できる．3行目は単に2行目左辺の$p(x)$項を右辺に移行しただけである．

Figure 2. 条件付確率とベイズ定理



Source: Authors' drawing.

ここで，θを確率モデルを構成する1個から複数個のパラメータ，$\boldsymbol{x}$を観察データとすると，式(2)の右辺を計算すれば左辺の確率がわかる．ベイズ統計では，$P(\theta)$を事前分布(prior distribution)，$P(\theta \mid x)$を尤度(likelihood)，$P(x)$を周辺確率(あるいはEvidence，正規化項)，左辺の$P(x \mid \theta)$を事後分布(posterior distribution)と呼ぶ．例えば，観察データ$\boldsymbol{x}$が正規分布に従って出現しているのではないかと

考えるならば，尤度を正規分布で計算する$p(x) \sim ND(\mu, \sigma^2)$．実際に尤度を計算するには，正規分布モデルのパラメータである平均μと分散σ^2を指定しなければならない．平均μと分散σ^2を正確に知らないことが，わざわざ計算をする前提であるから，$\mu \sim ND(0,1)$，$\sigma^2 \sim IG(0.01,0.01)$のように幅を持った値をとるようにそれぞれの事前分布(正規分布NDと逆ガンマ分布IG)を設定し，サンプリングにより平均μと分散σ^2を決める．つまり，事前分布は，根拠があるかないかにかかわらず，分析者の主観的判断を反映している．

概念的にはベイズの原理は自明であるが，実用上は(2)式の右辺が計算できるかが問題である．実際，右辺の計算は一般的には容易ではない．上の例の平均の事前分布を*ND*(0,1)に，分散を*IG*(0.01,0.01)とすれば，右辺の分子を手動で計算可能である．しかし，*ND*(0,1)と*IG*(0.01,0.01)からは無数の平均と分散の値があらわれるので，すべての場合で右辺の分子を手動計算することは現実的ではない．ベイズの原理自体は18世紀にトーマス・ベイズが示していたが，ハード，ソフトのコンピュータ技術の発展によりようやく実用レベルで計算できるようになったといえる．

事前分布$P(\theta)$は確率モデル作成者がどれだけそのモデルを信じているかであるから，基本的には適当に設定できる．ベイズ統計では，通常，θをある範囲(無限小から無限大の場合もある)で$P(\theta)$という確率分布をもつ事前分布として設定する．事前分布は，0%，100%のようにピンポイントで指定することもできる．ただし，0%しか自分を信じていないなら，右辺は何をしても0のままであるからモデルとして意味がない．逆に，自分の確率モデルを100%信じても良いが，それはθのある特定の値だけで$P(x \mid \theta)$を計算することを意味し，現実のデータにあてはまらなくなる可能性が高い．例えば，自分の買った1枚の宝くじが1等に100%当選すると信じても良いが，1等に当選する実際の確率は1千万分の1程度であるので，尤度1千万分の1×事前確率1で1/1000万である．右辺の分母は，この場合は，モデル内でのデータ出現確率(当選確率)=1であるから1である．結局，事後確率は1/1000万で，「自分が必ず当選するモデル」は現実説明能力が1千万分の1程度しかない，ほぼ意味のない確率モデルとなる．

右辺の分母である周辺確率$p(x)$は想定した確率モデル内のすべての状態(モデルのパラメータθがとりえるすべての値)でデータxが出現する確率である．上の宝くじの例では，モデル内では何が起きても常に当選しているので，当選状態が確率1で周辺確率は*p*(当選)=1になる．周辺確率は，通常は，計算が困難である場合が多い．しかし，(2)式およびFigure 2からわかるように，周辺確率はθの値に関係なく一定の値になるため，多くの場合，周辺確率の絶対的な値を知る必要はない．周辺確率$p(x)$が定数であれば，事後分布は右辺の分子の値(=尤度×事前確率)に比例する．事後分布の絶対的な値はわからなくても，事後分布の形状，つまりθの値のどこで$p(x)$が高いか低いかはわかる．

LDAに戻る．Dirichlet分布が多項分布の事前共役分布であるとは，次のことを意味している．

(1) 多項分布パラメータ(θ, ϕ)がDirichlet分布にしたがっていると想定する．この場合，Dirichlet分布が多項分布の事前分布である．

(2) 事前分布から多項分布のパラメータの値が与えられるので観測データの出現確率，つまり尤度を計算できる．

(3) 事後分布は*P*(観察データおよびモデル内で計算したパラメータ | 想定したモデルの固定パラメータ)の形になっているので，事後分布の確率の一番高いところが，観察データに一番あてはまるモデ

ルのパラメータになる．

(4) 事前分布と尤度の積である事後分布の関数形は通常はわからないが，Dirichlet分布を多項分布のパラメータの事前分布とした場合は，事後分布は再びDirichlet分布の形になる．このように事後分布の関数形を事前分布の関数形と同じにする事前分布を共役事前分布と呼ぶ．

(1)式の右辺はDirichlet分布で関数形がわかるが，$z_{1,10}$ =?のようにクローズドな形で表現することは変数の数を考えただけでほぼ不可能であることがわかる．MPGの場合，zの数だけで約20万になる．なお「クローズドな形」に明確な定義は無いようであるが，一般的な理解として，$z_{d,n} = f(x)g(y) + h(z)$のように，変数を単純な関数の組み合わせだけでだけで示すことが可能な状態を指している．(1)式をクローズドな形で解くことはできないため，数値計算によって解く．LDAモデルはベイズ確率モデルであるから，JAGS (Just Another Gibbs Sampler)やHMC (Hamiltonian Monte-Carlo)といったパッケージあるいは自作の汎用Malkov Chain Monte-Carlo (MCMC) サンプラーを使って数値計算することが可能である．LDAモデルをMCMCシミュレーション用のモデル記述言語であるBAGS風でスケルトン部分のみを書くと，

$$\begin{aligned} \phi[K][V] &\sim Dirichlet(\beta[V]) \\ \theta[D][K] &\sim Dirichlet(\alpha[K]) \\ z[D][N] &\sim Multinomial(\theta[D][K]) \\ w[D][N] &\sim Multinomial\big(\phi\big[z[D][N]\big][V]\big) \end{aligned} \tag{3}$$

とかける．(3)式の $X[Y][Z]$の Y, Zは，Xが $Y \times Z$のサイズの配列であることを意味している．BAGS記法の”~”は右辺の分布にしたがって左辺の値をサンプリングすることを意味している．このモデルは，直感的には，α仕様 Dirichlet サイコロ，β仕様 Dirichlet サイコロを振って K面サイコロと V面サイコロの仕様を決め，各文書用の K面サイコロを振って各文書の単語位置のトピックを決め，決まったトピック用の V面サイコロを振って単語位置の単語種を決めることをコンピュータ上でシミュレーションしている様子を示している．

(3)式にしたがって各分布から値をサンプルすることを繰り返すと，事後分布の平均，分散，分位数等の主要な統計量を計算できる．これは，(2)式の右辺をθに具体的数値を与えたいくつかの(通常は多数の)場合で実際に計算していることを意味している．汎用MCMCサンプラーによるLDAモデルの数値計算は何をおこなっているかわかりやすく，原理的にもコード的にも特に問題となる点はない．しかし，汎用MCMCサンプラーによるLDAモデル数値計算は，トピック分析のデータとしては小規模なMPGのデータサイズ(16文書，4トピック，20万単語，6000単語種)でも計算時間がかかりすぎ実用的ではない．LDAモデルを数値計算で解く場合は，LDAモデル用にカスタマイズしたサンプラーを使うか，変分ベイズ法による最適化を使った近似計算にとどめることで計算資源への要求を減らすことが一般的である．本研究ノートでも汎用MCMCサンプラーを使わないことから，汎用MCMCによるLDAモデル計算についてはこれ以上検討しない．3.2節と3.3節で本研究ノートで実際に使うLDA用計算アルゴリズムを検討する．アルゴリズムの検討に入る前に，Dirichlet分布と多項分布の関係についてみておく．

3.1.3 Dirichlet分布と多項分布の一般的な関係

Dirichlet分布，多項分布ともに，計算が相対的に容易な指数型分布族に属している．多項分布の事前分布としてDirichlet分布を想定するDirichlet-Multinomialモデルは，その取扱いが相対的に容易であ

るため，LDAモデル以外にも多くの分野で使われている(Minka 2003; Murphy 2023, pp. 137–141)．ここでは，一般的なDirichlet-Multinomialモデルを例として多項分布とDirichlet分布の関係を検討する．

記号を節約するため，LDAモデルの記号法から離れて，より簡易な記号を使う．$\omega^{(M)}$でM次多項分布のパラメータ，$\eta^{(M)}$をM次Dirichlet分布のパラメータ，xをN個の観察されたデータとする．Dirichlet分布が与える多項分布のパラメータベクトル$\omega^{(M)}$，つまり$\omega_m \ m \in M$がカテゴリm(M面サイコロの面m)が出る確率とすると，

$$
\begin{aligned}
p\left(\omega^{(M)} \mid \eta^{(M)}\right) &\sim \mathrm{Dirichlet}(\eta_1, \eta_2, \cdots, \eta_\mathrm{M}) \\
&= \frac{\Gamma(\sum_{m=1}^{M} \eta_m)}{\prod_{m=1}^{M} \Gamma\left(\eta_m\right)} \prod_{m=1}^{M} \omega_m^{\eta_m - 1} \\
&= \frac{1}{B(\eta)} \prod_{m=1}^{M} \omega_m^{\eta_m - 1}
\end{aligned}
\tag{4}
$$

である．(4)式のΓは階乗の一般形であるガンマ関数である．ガンマ関数は，$\Gamma(x) = \int_0^{infty} t^{x-1}\, e^{-t} dt$で定義される．この定義式から，

$$
\begin{aligned}
\Gamma(1) &= \int_0^{\infty} t^0\, e^{-t} dt = [-e^{-t}]_0^{\infty} = 1 \\
\Gamma(x+1) &= \int_0^{\infty} t^x\, e^{-t} dt \\
&= [-t^x e^{-t}]_0^{\infty} - \int_0^{\infty} -x t^{x-1} e^{-t} dt \\
&= 0 + x \int_0^{\infty} t^{x-1}\, e^{-t} dt \\
&= x\Gamma(x)
\end{aligned}
\tag{5}
$$

となり，任意の正の実数に対して$\Gamma(x+1) = x\Gamma(x)$が成り立ち，階乗の一般化になっていることがわかる．ここで，$x! = x \cdot (x-1)!$である整数の階乗と異なり，1ずれることに注意する．(4)式のBは，

$$
B(\eta) = \frac{\prod_{m=1}^{M} \Gamma\left(\eta_m\right)}{\Gamma(\sum_{m=1}^{M} \eta_m)}
$$

で定義されるベータ関数である．ベータ関数はDirichlet分布のガンマ関数を含む最初の項を書き換えているだけであるが，この部分がパラメータηだけで決まっていることがわかる．Dirichlet分布のパラメータが固定的に与えられていれば，$B(\eta)$は正規化定数項であり，分布のスケールを変えるだけで形状には関係しないため，$P(\omega \mid \eta) \propto \prod_{m=1}^{M} \omega_m^{\eta_m - 1}$となる．

N個の観測データ$\boldsymbol{x}$は，ωをパラメータとするM次多項分布に従ってM種類にラベル付けされたデータである．その尤度は，観察されたデータxの各要素をM次多項分布にあてはめた結果の積であるか

ら，

$$P(\mathbf{x} \mid \omega) = \frac{N!}{\prod_{m=1}^{N} x_m\,!} \prod_{m=1}^{M} \omega_m^{x_m} \propto \prod_{m=1}^{M} \omega_m^{x_m} \tag{6}$$

となる．ここでx_mは，mにラベル付けされたデータxの数を示している．

ベイズの定理により，事後分布$P(\omega \mid \mathbf{x})$，つまりデータを観測した後のパラメータ$\omega$の確率分布あるいは信頼性は，事前分布と尤度の積である．モデルのパラメータωに関係なく決まる正規化定数$\frac{N!}{\prod_{m=1}^{N} x_m!}$と周辺確率$P(\mathbf{x}, \eta)$を省略すると，事後分布$P(\omega \mid \eta, \mathbf{x})$は，

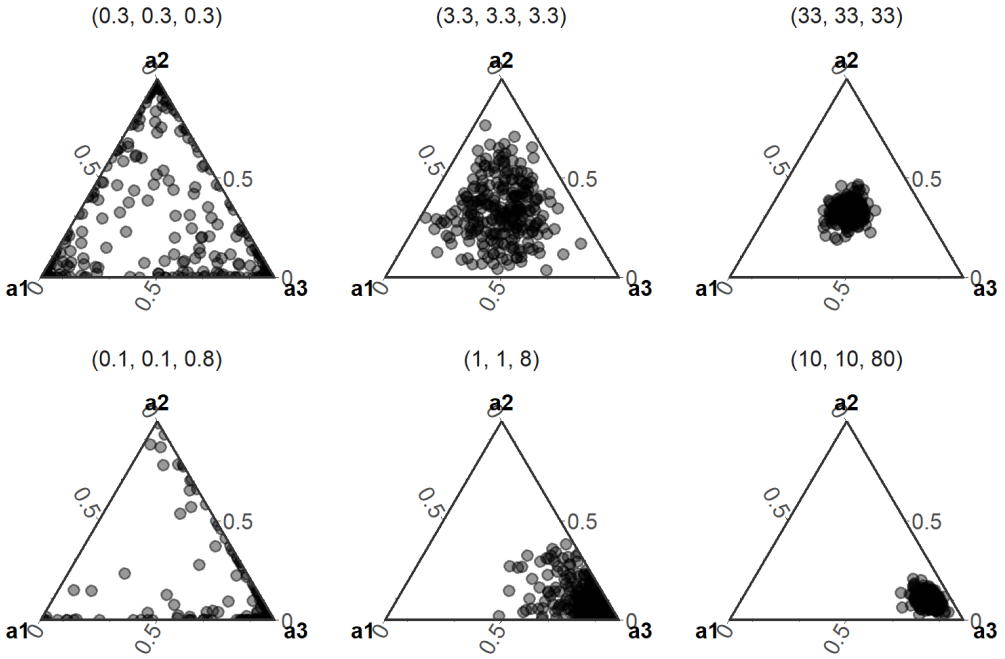
$$\begin{aligned} P(\omega \mid \mathbf{x}, \eta) &\propto P(\mathbf{x} \mid \omega, \eta) \cdot P(\omega \mid \eta) \\ &\propto \left(\prod_{m=1}^{M} \omega_m^{x_m} \right) \cdot \left(\prod_{m=1}^{M} \omega_m^{\eta_m - 1} \right) \\ &= \prod_{m=1}^{M} \omega_m^{(x_m + \eta_m) - 1} \end{aligned} \tag{7}$$

となり，事後分布が再びDirichlet分布$\mathrm{Dir}(\eta_1 + x_1, \ldots, \eta_M + x_M)$となることがわかる．事後分布がDirichlet分布であるから，事後分布の確率の期待値，つまり次のxがラベルmである確率(M面サイコロのm面が出る確率)の期待値は，Dirichlet分布の期待値の公式から$(\omega_k + x_m)/(\sum_k \omega_k + n_x)$となる．ここで，$n_x$は観察したデータの数(すでに何回サイコロを振ったか)で，分母全体は確率の総計を1にするための正規化定数である．T回の試行でラベルmのxが出現する回数(T回振ったときのm面の回数)の期待値は$T \cdot (\omega_k + x_m)/(\sum_k \omega_k + n_x)$である．

Dirichlet分布パラメータはConcentration parameter (集中度パラメータ) あるいはWeightとよばれるスカラーと，多項分布の各項の出現確率の期待値の積に分解できる．η_mであれば，$\eta_0 = \sum_m \eta_m$を集中度パラメータとして，$\eta_m = \eta_0 \cdot \eta_m^*$に分解できる．$\eta_0$は$\omega$の分散を制御し，$\eta_0$が大きいとDirichlet分布からサンプルされるωの分散は小さくなる．η_m/η_0はパラメータω_mの期待値に対応する．

Figure 3は，3項分布の場合のDirichlet分布パラメータと多項分布パラメータの関係を例示している．一般的には，サンプルされるM次元$\omega^{(M)}$ベクトルの要素の総計は常に1であるから，各サンプルは幾何学的にはM次元座標の各軸上の点1(任意のm軸の値が1，残りの軸の値がすべて0である座標)を結んでできる(M-1)次元平面(Simplex)上の1つの点であらわすことができる．η_m^*が各点のsimplex上の平均的位置を決め，η_0がその平均的な位置からどれだけ外れやすいかを決める．Figure 3の上段3つのplotは$\eta_1 = \eta_2 = \eta_3 = 1/3$のときに，$\eta_0$を1，10，100と変えている．下段は$\eta_1 = 0.1, \eta_2 = 0.1, \eta_3 = 0.8$で，$\eta_0$を1，10，100と変えている．Figure 3は，Dirichletパラメータが対称固定でも，η_0が小さい(分散が大きい)とサンプルは広範囲にちらばり，無情報事前分布に近くづく．つまり，多項分布のパラメータの値についてほぼ何も想定していないことになる．なお，Rの主なパッケージではDirichlet分布の引数としてη_m^*とη_0と別々にせず，η_mを与えると，$\sum_m \eta_m$で自動的にスケーリングする．

Figure 3. 3項分布パラメータのDirichlet分布からのサンプリング



Source: see the qmd file in Appendix for the codes.

LDAモデルに戻り，LDAモデルにおいて注意すべきDirichlet分布と多項分布の関係をみておく．

(1) LDAモデルでは，文書別トピック分布とトピック別単語種分布の多項分布パラメータは，相互に無関係に決まる．文書別トピック分布側Dirichlet分布パラメータ$\alpha^{(K)}$とトピック別単語種分布側Dirichletパラメータ$\beta^{(V)}$の間には何の関係もない．つまり，K面サイコロとV面サイコロは別々に作られ，K面サイコロの作り方がV面サイコロの作り方に影響する，あるいはその逆のようなV面サイコロ，K面サイコロ間の相互関係はないと想定している．

(2) Dirichlet分布のパラメータ$\alpha^{(K)}$と$\beta^{(V)}$は基本的に対称固定に設定する．つまり，複数の仕様の異なるK面サイコロ，V面サイコロではなく，同一の仕様のK面サイコロ，V面サイコロを使う．ここで注意すべき点は，Dirichlet分布パラメータが対称固定であってもサイコロの目の出方は2つの意味で変わる点である．第1に，Dirichlet分布パラメータが指定するのは各面が出る確率の期待値である．期待値が対称固定であっても，トピック分布は文書ごとに，単語種分布はトピックごとに変動できる．1から6までの数字が均等に出ることが期待される正常な6面サイコロを振ったとき，常に1から6までの数字が均等に出るわけではない．サイコロ各面の出目確率の期待値が固定されていても，サイコロを振った結果はその都度変わる． 1が10回出現するといった偏ったパターンが生じる確率は低くなるものの，ゼロではない．第2に，Dirichlet分布は多項分布パラメータの期待値を分散つきで指定するため，多項分布パラメータの値はゼロから無限大までのどのような値でも取ることができる．Dirichlet分布パラメータが指定する分散が大きければ(集中度パラメータα_0, β_0が小さければ)，Dirichlet分布からサンプルされる多項分布パラメータは大きくばらつく．対称固定パラメータのDirichlet分布であっても，個々の文書d，個々のトピックkに適した多項分布パラメータ(出目確率の期待値)を持ったK面サイコロ，V面サイコロを想定することができる．ただし，Dirichlet分布パラメータが与える期待値と分散から大きく外れる値を多項分布パラメータの値とすることは，そのDirichlet分布ではほとんど起こりえない事象ということになり，結果的にLDAモデル全体の観測値へ

のあてはまりが悪くなる．

(3) LDAモデルの実際の計算では，$\alpha^{(1)} = \alpha^{(2)} = \cdots = \alpha^{(K)} = 0.1$，$\beta^{(1)} = \beta^{(2)} = \cdots = \beta^{(V)} = 0.01$のように，Dirichlet分布パラメータの値を対称固定で与えることが多い．Blei, Ng and Jordan (2003)は，$\alpha = 50/K$，$\beta = 0.01$，Griffiths and Steyvers (2004)は$\alpha = 0.1$，$\beta = 0.01$を推奨している．Blei, Ng and Jordan (2003)とGriffiths and Steyvers (2004)では文書別トピック分布側Dirichletパラメータαの設定値にかなり大きな違いが出る可能性がある．これは両者が想定している分析対象文書の性格の違いを反映している．Blei, Ng and Jordan (2003)は文書総数Dが相対的に小さく，1文書の単語数$N_{d,w}$が相対的に大きく，トピックの数Kが相対的に小さな文書を，Griffiths and Steyvers (2004)はSNS投稿文のような，Dが大きく，$N_{d,w}$が相対的に小さく，多数の雑多なトピックの存在が予想される文書を分析対象として想定している．

一方，対称固定Dirichletパラメータの想定が現実の文書の状態を正確に反映しているとは考えにくい．各文書でトピック分布の期待値，分散が変わり，トピックごとに単語種の期待値，分散が変わると考える方が現実的である．Wallach, Mimno and McCallum (2009)は，Dirichletパラメータα，βを非対称，可変にすることでトピック分析の精度が大幅に向上すると報告している．ただし，α非対称化が精度改善にもっぱら貢献し，β非対称化の貢献は小さいか，ほぼ無いとしている．またαについても，対称のままで，α_0だけを可変にしても，つまり分散だけを可変としても，かなりの精度改善が期待できるとしている．文書別トピック分布が分析対象文書の性格によっても，また分析対象文書D中の個別文書d間でも変わるであろうことは容易に想像できる．

対称固定Dirichlet分布パラメータを使う理由は，第1に，α，βの各要素の値を指定するための情報は通常存在しないためである．それだけの情報があるならば，トピック分析をおこなう必要はないだろう．第2に，計算コストの節約のためである．第1の理由により，パラメータ値が自動推定できなければDirichlet分布パラメータを非対称化は困難である．α，βの推定は，原理的には，ベイズ確率モデルをさらに階層化し，α，βをなんらかの確率分布からサンプルする設定にすることが考えられる．ただし，この階層化LDAモデルを汎用MCMCサンプラーで数値計算することは，計算時間がかかりすぎるため，ほぼ実行不可能である．第4章で使うRのパッケージが，α，βをどのように扱うかは，以下でパッケージごとに検討する．

3.2 LDAモデルのRパッケージ

Table 2は，Rの代表的なトピックモデル用パッケージをまとめている．Table 2の説明欄が示すように，LDAモデルの計算方法には，大きくは，Malkov Chain Monte Carlo (MCMC)によるサンプリング法(CGS: Collapsed Gibbs Sampling)と変分ベイズ(Variational Bayes: VB)による最適化法(VEM: Variational Expectation Maximization)の2つの方法がある(Abdelrazek et al. 2022; Stevers 2007)．いずれの場合も，モデルそのものの改善，近似計算による計算省略，アルゴリズムの細部の改善，C言語化，並列化，メモリ利用改善など様々なレベルで高速化のための工夫がおこなわれ，数多くのバリエーションがある (Blei, Kucukelbir, and McAuliffe 2017; Chen et al. 2016; Mimno, Hoffman and Blei 2012; Phan, Nguyen and Horiguchi 2008; Porteous et al. 2008)．以下では，第4章で利用するパッケージが実装しているCGSとVEMのみを検討する．

Table 2. LDA用各種Rパッケージ

Package名	アルゴリズム	説明	利用
topicmodels	Mean-field VEM	Blei, Ng and Jordan (2003)．変分ベイズ最適化．alpha_0推定に対応.	Y
topicmodels	Collapsed Gibbs (CGS)	Blei, Ng and Jordan (2003)．Dirichletパラメータは対称固定のみ.	Y
lda	Collapsed Gibbs (FastCGS)	Griffiths and Steyvers (2004)．C実装で高速化．Dirichletパラメータは対称固定のみ.	Y
mallet	Collapsed Gibbs (SparseLDA)	McCallum (2002), Graham, Weingart and Milligan (2026)．スパースデータ対応により 高速化．alpha推定に対応．beta_0のみ推定に対応.	Y
text2vec	Collapsed Gibbs (WarpLDA)	Hoffman et al. (2013), Chen et al. (2016)．独自のサンプリングアルゴリズムでCGSを高速化．巨大データのストリーミング処理に対応.	N
stm	VEM	Roberts, Stewart and Tingley (2019)．LDAの拡張とみなされるがDirichlet - Multinomialモデルを使わない.	N

Source: Authors' compilation.

CGSは，汎用のMCMC法であるGibbsサンプリング法のLDA用カスタマイズ版である．CGSは，LDAモデルの構造を利用し，多項分布のパラメータ(θ, ϕ)を積分消去(周辺化)した結果を使うことで，θ, ϕのサンプリングを不要にする．θ, ϕの数はテキスト分析データとしては小規模なMPGでも3万を超えるため，θ, ϕのサンプリングをおこなわないことは計算高速化に大きく貢献する．

topicmodelsパッケージのCGSとldaパッケージのFast CGSの違いは，Fast CGSがC実装，メモリ利用最適化等で高速化をおこなっているためR実装のtopicmodelsパッケージのCGSより速いということである．ただし，ldaパッケージはFast CGSのコア部分のみを提供し，分析補助機能がほとんどないため使いにくい．

MalletのSparse LDAもCGSである．Sparse LDAでは，多項分布パラメータθ, ϕを積分消去した後の式を3つの部分に分解し，計算順序を工夫することで高速化を達成している(Graham, Weingart and Milligan 2026; Magnusson 2026; McCallum 2002)．すべての文書がすべてのトピックスを扱うことはなく，すべてのトピックがすべての単語種を使うことはないとの妥当な想定のもとで，扱われていないトピック，単語種の計算をしないことで高速化している．MalletはJAVA実装であり，R経由での利用は必ずしも使いやすくはないが，いくつかの点に注意すればRから十分に使うことができる．Malletはテキスト分析の古典的ともいうべきパッケージであり，LDAモデルの計算においても評価が高く，Dirichletパラメータ設定も柔軟におこなえる．

text2vecパッケージは，WarpLDAとよばれる高速化CGSのC++実装，自動並列実行により高速に動く(Chen et al. 2016)．加えて，分析対象データをメモリ上にすべて置く必要が無く，したがってメモリ容量の制約を受けることが基本的にない．Dirichletパラメータは対称固定のみである．text2vecのLDAは，基本的に継続的に流入する(streaming)大量の文書データの分析を目的としている．相対的に少ないデータ量の専門文献に対するテキスト分析の速度と精度を検証するという本研究ノートの目的とは離れるため，第4章ではtext2vecは使っていない．pythonで良く使われるscikit-learnやgensimのLDA用パッケージも大規模データ，ストリーミングデータに対応することを目的として，VB法の一種であるOnline Variational Bayes を使っている (Blei, Kucukelbir, and McAuliffe 2017; Hoffman et al. 2013; Yao, Mimno and McCallum 2009)．これらのパッケージについても，本研究ノートの検証の目的とは離れるため検証対象としていない．

stmパッケージはLDAモデルに対応しているが，主目的をstructural topic modelingの実行におくパッケージである(Roberts, Stewart and Tingley 2019)．Structural topic modelingは，その文書の立場，見解によって文書のトピック分布は変わり，トピックが使う単語種分布も変わることに対応するためのモデルである．例えば，同じ「温暖化ガス排出規制」というトピックでも，規制を支持する側の文書と規制に反対する側の文書では，使っている単語種は異なるだろう．Structural topic modelingは，各文書に付帯情報(メタデータ)を加えることで，同一トピックに対する立場の違いによる単語種パターンのスイッチを考慮できる．アイディアとしてはLDAモデルの拡張とみなされるが，stmはメタデータ情報を取り込みやすくするために，Dirichlet-Multinomial分布ではなくLogistic-Normal分布を使う．Logistic-Normal分布の共役事後分布は存在しないため，事後分布はラプラス近似，つまり多次元ガウス分布によって近似している．Structural topic modelingはテキスト分析と社会科学分析を融合する強力なツールであるが，モデルの構造がLDAモデルとは異なることと，文書データ以外にメタデータが必要になることから，本研究ノートではstmパッケージは利用せず，Structural topic modelingの検討もしない．Structural topic modelingについては，本研究ノートの第3部で扱う予定である．

3.3 Collapsed Gibbs Sampling (CGS)

3.3.1 Gibbs Sampling

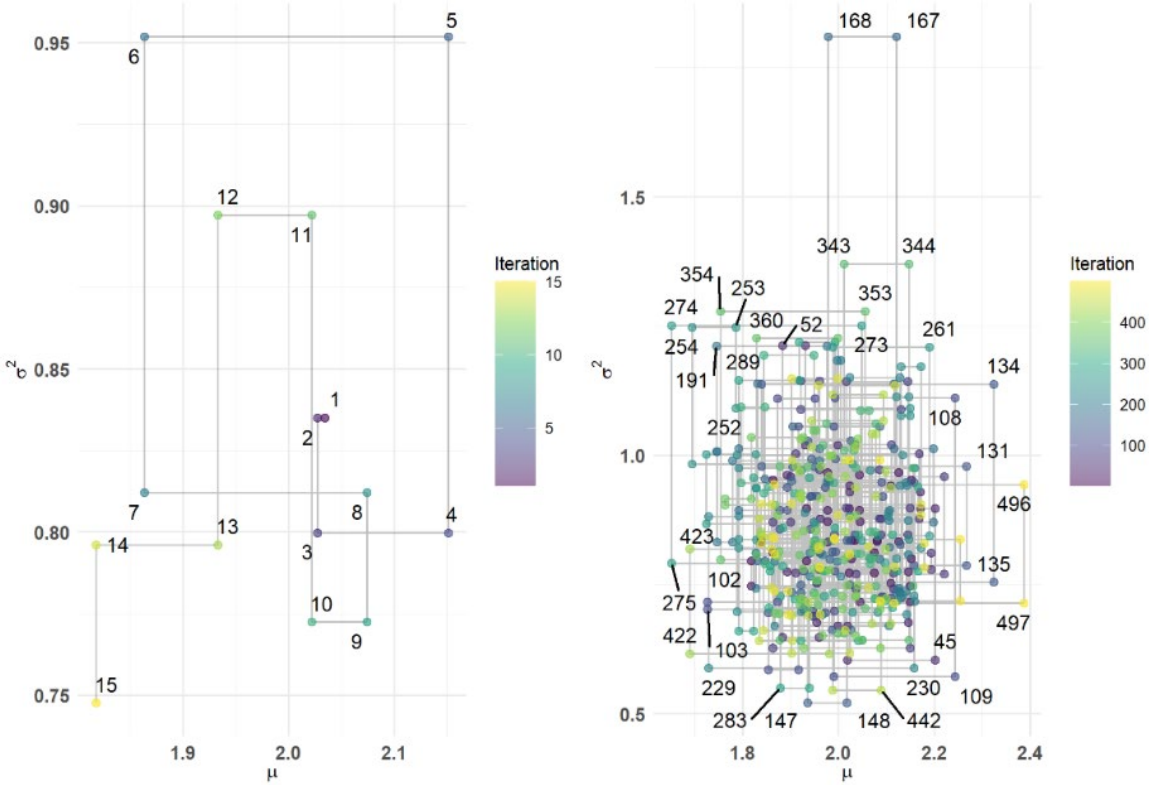
Gibbsサンプリングは，ベイズ確率モデルを数値的に計算する一般的な方法の一つである．一般的なGibbsサンプリングとCGSとの間にサンプリング方法自体に違いはない．CGSは，計算時間を短縮するため，(1)式そのものではなく，ある単語へのトピック割当，つまりLDAの潜在変数の確率分布をGibbsサンプリングの対象とする．ある単語へのトピック割当は，多項分布パラメータθ, ϕを積分消去した式として得ることができる．ここでは，まず一般的なGibbsサンプリングについて簡単にみた後，CGSについて検討する．ただし，Gibbsサンプリングについては多くの解説があるので，ここではCGSアルゴリズムを検討するために必要な概念的な説明のみにとどめる．

Gibbsサンプリングをおこなうための前提は，事前分布と尤度の積である事後分布の関数形がわかり，実際に計算できることである．この条件が満たされることは通常はほとんどないが，Dirichlet-Multinomial分布を用いるLDAの場合は事後分布の関数形がわかるため，Gibbsサンプリングが可能である．

事後分布が低次元の関数であれば解析的に解ける可能性が高いから，Gibbsサンプリングの対象となる事後分布は通常はM個の複数のパラメータ$m^{(M)}$を持つ高次元の関数である．Gibbsサンプリングは，事後分布を構成するM個のパラメータのうちの1つ$m^{(t)}$の事後分布は，そのパラメータの完全条件付き確率分布から計算できることにもとづいている．完全条件付き確率分布とは，M個のパラメータのうち1個を除く他のすべてのパラメータの値を固定して計算した確率分布である．概念的には，M+1次元空間(パラメータ数M次元+確率の値1次元)に描かれる事後確率分布において，M-1個のパラメータを固定すると，未知の計算対象の1個のパラメータ$m^{(t)}$とそれに対応する確率値$p\left(m^{(t)} \mid m^{(\neg t)} = m^{*(\neg t)}\right)$の2次元確率分布となる．ここで$\neg m$記号は，インデックス集合Mの$m$以外のすべての要素を意味している．$m^*$を計算中のパラメータとすれば，$m^{*(\neg t)}$の値は既に計算に使った値，つまり既知の値である．この2次元確率分布は，M+1次元確率分布を$m^{*(\neg t)}$)超平面で切った切り口であるから，もとのM+1次元確率分布の一部である．つまり，限られた範囲(切り口)であるが全体像を求めるべきM+1次元確率分布からm^*をサンプルしている．

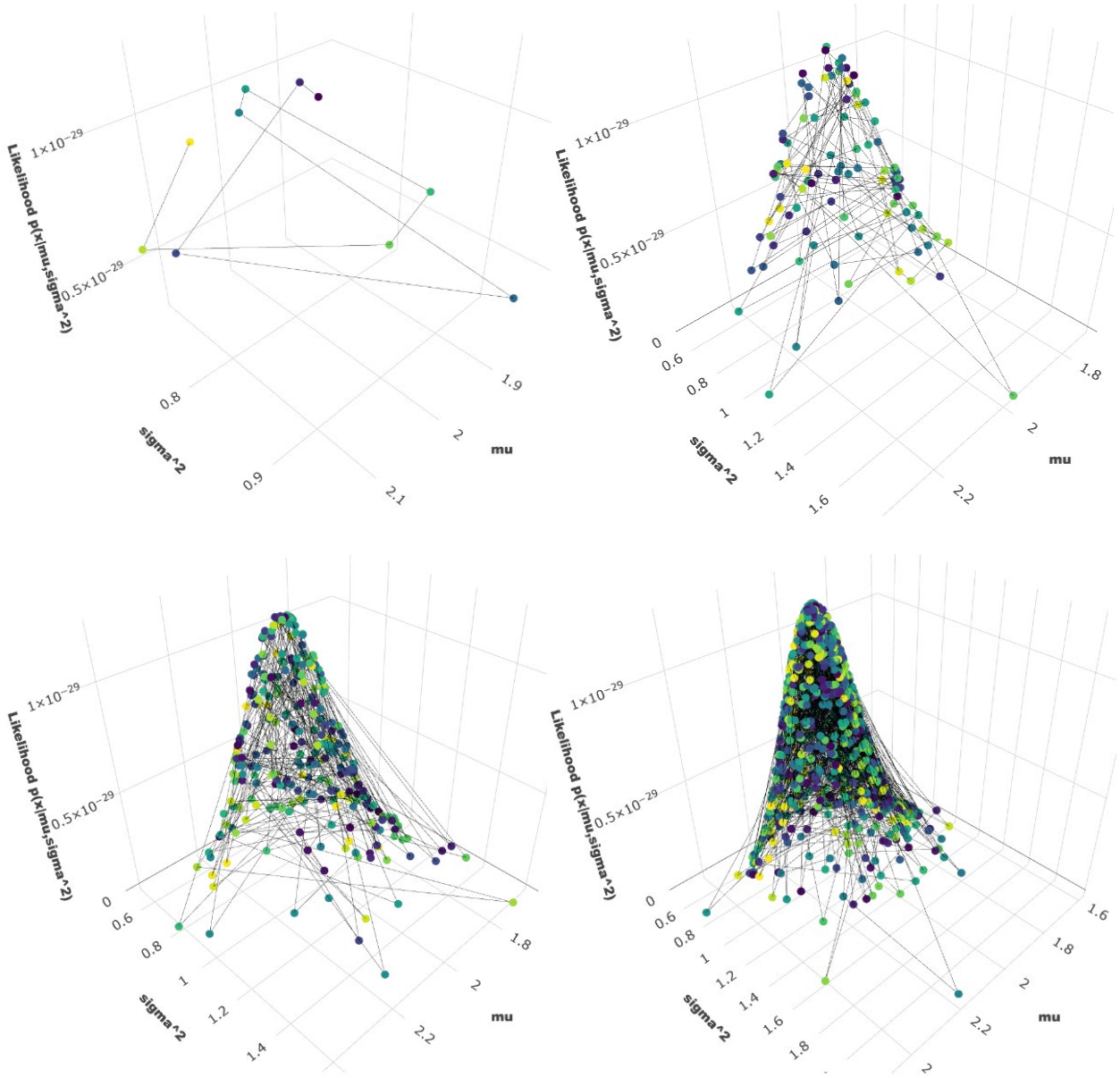
Figures 4,5は，Gibbsサンプリングのアルゴリズムを概念的に示している．ここでは，50標本分の観察データを正規分布から生成し，分散パラメータσ^2の事前分布を逆ガンマ関数，平均パラメータμの事前分布を正規分布，尤度を正規分布とするモデルで説明する．この場合，それぞれの事前分布は共役事前分布であるため事後分布は正規分布になるため，事後分布の関数形がわかっている．Figure 4は，σ^2とμを交互に固定しながら，次の計算点に移動して，事後分布の値を実際に計算する様子をみている．一方のパラメータは固定されているので，次の計算点への移動が常に縦軸か横軸に平行であることをみることだけが目的である．Figure 5は，各計算点で計算された事後確率を垂直方向にプロットしてFigure 4を3次元でみている．Gibbsサンプリングを続けると事後分布の全体像が次第にあきらかになってくる様子を示している．Figure 5では，σ^2側では$\sigma^2 = 1$で，μ側では$\mu = 2$で事後確率がもっとも大きくなっている．つまり，50個の観察データがあるとき，このモデルでは，$\sigma^2 = 1$，$\mu = 2$がもっとも出やすい値だということになる．実際，50個の観察データは，$\sigma^2 = 1$，$\mu = 2$の正規分布から生成したデータである．Figure 4とFigure 5に分けている理由は3Dプロットでは計算点の移動が見にくいことだけである．ただし，Figure 5ではコード上でのindex操作単純化のために1イテレーションでσ^2, μの両方を更新しているので計算点は軸に沿わずに移動している．

Figure 4. Gibbs サンプリング: 2Dトレース



Source: See the qmd file in Appendix for the code to draw the plot.

Figure 5. Gibbsサンプリング: 3Dトレース



Source: See the qmd file in Appendix for the code to draw the plot.

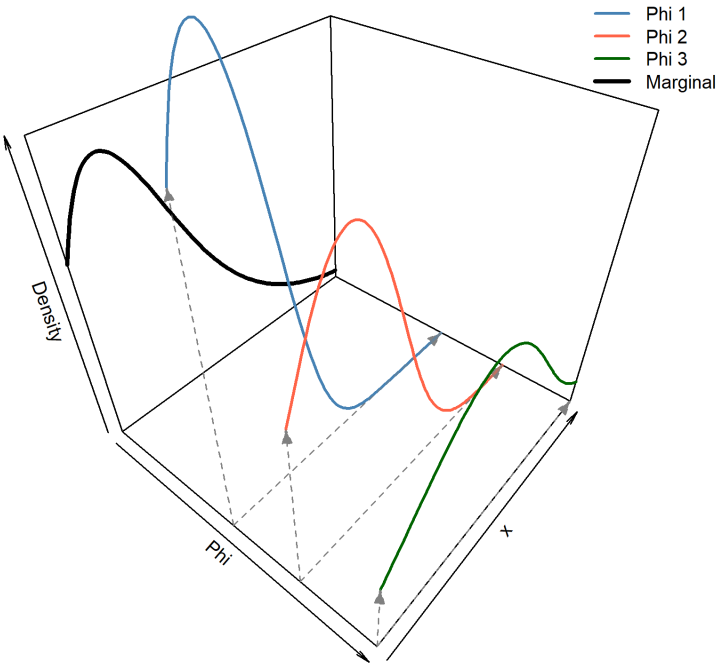
3.3.2 周辺化とCGS

CGSの基本的アイディアは，ある単語，あるいは「単語の場所」へのトピック割当分布をGibbsサンプリングで数値計算することである(Heinrich 2008; Marecek 2025)．ある単語へのトピック割当の分布は，(1)式から文書別トピック分布パラメータ$\theta_{d,k}$，トピック別単語種分布パラメータ$\phi_{k,v}$を積分消去した結果として得られる．積分によるパラメータ消去は「周辺化」と呼ばれる手順である．

周辺化のための計算を実際におこなうことがどの程度困難かは別として，周辺化自体は単純な概念である．Figure 6は，パラメータphiを積分消去する例を示している．xの確率分布が$p(phi_i)$のように決まるとする．Marginalラベルの曲線は$\sum_i[p(phi_i)\cdot f_i(x)]$であり，$phi_i$が取り得るすべての状態で

のxの確率分布である．Marginalラベルの曲線の確率分布はすでにphi_i要因を考慮しているため，その意味でxの確率分布へのphi_i要因の影響は考慮する必要はない．Figure 6では3次元から2次元への移行を1枚の2次元図で表しているためわかりにくいが，$\sum_i[p(phi_i)\cdot f_i(x)]$の計算後には$phi$軸は消え，$x$軸とその確率$p(x)$軸の2次元の図になっているとみていただきたい．

Figure 6. 周辺化の概念図



結局，周辺化は，ある確率分布に影響を与える要因があるとき，その要因がとりえるすべての状況で確率分布にどのように影響を与えるかを考慮した後の確率分布がわかるのであれば，確率分布の形状を考える際にその要因はもはや考慮しなくてもよいという，あたりまえのことをいっているだけである．ただし，周辺化後の確率分布は，周辺化した分，情報量が少なくなっている点に注意しなければならない．Figure 6では，Marginal曲線が示すxの確率分布だけをみて，その確率分布がxの全域で1/3ずつのweightで合成された3つの確率分布から構成されているという情報を復元することは容易ではない．

周辺化によるパラメータ消去は概念的には簡単であるが，周辺化した後の関数形がわかり，かつ計算可能であることは一般的ではない．LDAモデルは，Dirichlet-Multinomial分布を使っているため，積分が可能であり，その積分後の関数形もわかる．観察文書データ$\boldsymbol{w}$，Dirichlet分布パラメータα,βが与えられ，位置(d,i)以外の単語へのトピック割当が固定された状態で$z_{d,i}$にトピックkが割当てられる確率$p\left(z_{d,i}=k \mid \mathbf{w}, z_{\neg(d,i)}, \alpha, \beta\right)$は，

$$
\begin{aligned}
p\left(z_{d,i}=k \mid \boldsymbol{w}, z_{\neg(d,i)}, \alpha, \beta\right) &= \frac{p\left(z_{d,i}=k \mid z_{\neg(d,i)}, \alpha, \beta\right)\cdot p\left(\boldsymbol{w} \mid z_{d,i}=k, z_{\neg(d,i)}, \alpha, \beta\right)}{p\left(\boldsymbol{w} \mid z_{\neg(d,i)}, \alpha, \beta\right)} \\
&\propto p\left(z_{d,i}=k \mid z_{\neg(d,i)}, \alpha\right)\cdot p\left(\boldsymbol{w} \mid z_{d,i}=k, z_{\neg(d,i)}, \beta\right) \\
&\propto p\left(z_{d,i}=k \mid z_{\neg(d,i)}, \alpha\right)\cdot p\left(w_{d,i} \mid w_{\neg(d,i)}, z_{d,i}=k, z_{\neg(d,i)}, \beta\right)
\end{aligned}
\tag{8}
$$

と書ける．ここで，(8)式の1行目の右辺は，条件付き確率の定義$p(A \mid B)p(B) = p(B \mid A)p(A)$を使って書き換えただけである．2行目は，単語(*d*,*i*)へのトピック割当の確率分布全体$p\left(z_{d,i} = k \mid \cdot\right)$は，右辺の分母は計算しなくてもよいことを示している．単語(*d*,*i*)へのどのトピックkへ割当確率の計算でも1行目右辺の分母は共通であるため，$P\left(z_{d,i} = k \mid \cdot\right) / \sum_{k=1}^{K} P\left(z_{d,i} = k \mid \cdot\right)$を計算するときに分母は相殺されて消えるからである．ただし，分母を無視するため，1行目と2行目は等号関係ではなく比例関係でつながる．さらにDirichletパラメータβはトピック割当に無関係で，αは単語種割当に無関係であるから，2行目の式では不要なDirichletパラメータを書いていない．3行目は，まず，各単語位置への単語種割当はBag of words仮定から互いに独立で無関係であるので，条件付き確率の定義を使って$p(\mathbf{w})$を$p\left(w_{d,i}\right) \cdot p\left(w_{\neg(d,i)}\right) = p\left(w_{d,i} \mid p\left(w_{\neg(d,i)}\right)\right) \cdot p\left(w_{\neg(d,i)}\right)$と書き換える．Gibbsサンプリングで単語(*d*,*i*)の単語種割当分布を計算する場合には書き換え後の$p\left(w_{\neg(d,i)}\right)$は固定された既知の値であるため，単語(*d*,*i*)のトピック割当分布の計算では相殺される．つまり，LDAモデルでは$p\left(w_{d,i} \mid p\left(w_{\neg(d,i)}\right)\right)$を計算するために$p\left(w_{\neg(d,i)}\right)$の値を知る必要はないので，この項を式から除くことができる．ただし，に$p\left(w_{\neg(d,i)}\right)$項を除いてしまうため，2行目と3行目は，再び，等号関係ではなく比例関係でつながる．なお，すでに検討したように(*d*,*i*)は単語のある位置のラベルにすぎず，単語(*d*,*i*)という表現はやや誤解を招く．しかし，LDAの実際の計算では位置(*d*,*i*)の観察データ(単語)の情報を使っているので，以下では単語(*d*,*i*)で統一する．

(8)式3行目は，ドットの前半がある単語(*d*,*i*)でのLDAによるトピック割当，後半が単語(*d*,*i*)へのLDAによる単語種割当である．トピック分布は文書dごとに決まることとBag of words仮定から，ある文書d^*へのトピック割当の期待値と文書d^*内のある単語(d^*, i)へのトピック割当の期待値は一致する．ドットの後半の単語種割当はトピックのみに依存する．再びBag of words仮定から，あるトピックk^*のときに，ある単語種v^*があらわれる期待値は，全文書*D*でも，ある文書*d*でも，ある単語(*d*,*i*)でも，同じトピックk^*であれば同じである．

以上から多項分布パラメータの積分消去は容易である．(8)式の最終行は，文書別トピック分布の期待値とトピック別単語種分布の期待値の積を計算することと同等であるから，

$$
\begin{aligned}
& p\left(z_{d,i} = k \mid z_{\neg(d,i)}, \alpha\right) \cdot p\left(w_{d,i} \mid w_{\neg(d,i)}, z_{d,i} = k, z_{\neg(d,i)}, \beta\right) \\
\propto & \int p\left(z_{d,i} = k \mid \theta_d\right) p\left(\theta_d \mid z_{\neg(d,i)}, \alpha\right) d\theta_d \cdot \int p\left(w_{d,i} \mid \phi_k\right) p\left(\phi_k \mid w_{\neg(d,i)}, z_{\neg(d,i)}, \beta\right) d\phi_k \\
& = \frac{\alpha + N_{d,w,k}}{K\alpha + N_{d,w} - 1} \cdot \frac{\beta + N_{w,v}}{V\beta + \sum_{v=1}^{V} N_{k,w,v}} \\
& \propto \left(\alpha + N_{d,w,k}\right) \cdot \frac{\beta + N_{w,v}}{V\beta + \sum_{v=1}^{V} N_{k,w,v}}
\end{aligned}
\tag{9}
$$

とかける．(9)式1行目の右辺の積分記号内はそれぞれ，(文書内の単語ごとにみた)文書別トピック分布の事後分布とトピック別単語種分布の事後分布である．それぞれが事前分布にDirichlet分布，そのDirichlet分布からサンプルしたパラメータ値をもつ多項分布を尤度とするDirichlet-Multinomial分布の事後分布となっている．したがって，各事後分布をそれぞれの多項分布のパラメータで積分した結果は，パラメータの値ごとの事後確率値をそのパラメータ値に対応する確率をウェイトとして加算した結果になり，結局，各事後分布の期待値になる．この式では，単語(*d*,*i*)以外のトピック$(z_{\neg(d,i)})$

を固定し，Dirichlet分布パラメータα, β，単語(d,i)以外の観察された単語種($w_{\neg(d,i)}$)が与えられたときの$z_{d,i} = k$の事後分布の期待値となる．LDAモデルの設定から，ドット記号前後間に，つまり(文書内の単語ごとにみた)文書別トピック分布とトピック別単語種分布間に依存関係はないので，別々に積分できる．また，単語(d,i)以外のトピック，Dirichlet分布パラメータ，単語(d,i)以外の単語種は単語(d,i)の計算の際には固定されているから積分には関係しないことに注意する．

2行目右辺は，Dirichlet-Multinomial分布の事後分布は再びDirichlet分布であることを使っている．つまり，各積分の結果はDirichlet分布の期待値になるのであるから，Dirichlet分布のパラメータでDirichlet分布の期待値をあらわす公式をそのまま使うことができる．Dirichlet-Multinomial分布の式を積分することで同じ結果が得られるが，Dirichlet分布の期待値の公式を確認する以上の意味はないので，ここでは期待値の公式をそのまま使う．ただし，左辺と右辺は等号関係ではなく比例関係で結ばれている．(8)式で左辺を得る際に途中で定数項を相殺，省略したため，(9)式の左辺は既に期待値の真の値ではない．加えて，Gibbsサンプリングの多数回の結果の平均は期待値に収束するが，Gibbsサンプリングの1回の計算結果が期待値と一致するわけではないためである．定数である分母$K\alpha + N_{d,w} + 1$を削除したので3行目は2行目と比例関係になる．

3.3.3 CGSアルゴリズム

CGSアルゴリズムを実際に計算する擬似コードの骨格部分は，次のようになる．

```
Set initial k to all Z[d][i]
W[d][i] stores v at [d][i]
for (r in 1:R){
   for (d in 1:D){
     for (i in 1:NW[d]){
       zdi<-Z[d][i]
       wdi<-w[d][i]
       NDK[d][zdi]--; NKW[zdi][wdi]--; NKWT[zdi]--;
       for (k in 1:K){
         p[t]<-(alpha+NDK[d][k])/(K*alpha+NW[d]-1)·(beta+NKW[k][v])(V*beta+NKWT[k])
       }
      sample k from p[k]
      Z[d][i]<-k
      NDK[d][k]++; NKW[k][wdi]++; NKWT[k]++;
      }
   }
}
```

擬似コードが端的に示すように，CGSでは，Z[d][i+1]のトピック割当計算のために，Z[d][i]のトピックを具体的にk個のトピックの中のどれかに決めるときに$p[k]$からサンプリングをおこなうだけで，アルゴリズムの基幹部分は単純な更新式で足りる．これにより，CGSの計算速度は，(3)式のナイーブなベイズモデルの汎用MCMCでの計算より圧倒的に速くなる．

CGSアルゴリズムで注意すべき点は次の点である．第1に，初期値$z_{d,i}^{(0)}$の情報は通常は事前にはないから，すべての$z_{d,i}^{(0)}$についてどのトピックも同一の確率で現れるとみなして1/Kを与える，あるいはどれか1つのkを1とし他をゼロとするといった主観的方法で与える．したがって，$z_{d,i}^{(1)}$を$z_{\neg(d,i)}^{(0)}$の状態から計算を開始し，(d,i)の次は(d,i+1)のように1つずつ順番に計算していくため，CGS開始からしばらくの間，計算結果はまったく的外れ，つまり事後分布の非常に確率の低いところしかみていない可能性が高い(Figures 4, 5参照).

第2に，$z_{d,i}$の計算における標準的なループの構造は，一番内側にある1つのd,iにおける$p(z_{d,i})$，つまりd文書のi番目の単語(の場所)への各トピックの割当確率を計算するKループ，その外側に同一個別文書d^*内で$z_{d^*,i}$を計算する単語の$N_{d,w}$ループ，その外側に全文書内の個別文書のDループがあり，さらにその外側に(d,i,k)ループを繰り返すループがある．なお，$z_{d,i}^{(t,j)}$の添え字tで一番外側のループのt回目，jで(d,i)ループの何回目かを示している．jは$j \in \{1,\ldots,N_{w,d}\}$である．

第3に，$z_{d_1,i_1}^{(t,j)}$から，次にどの$z_{d_x,i_y}^{(t,j+1)}$，あるいは一番外側のループをまたがる場合は$z_{d_x,i_y}^{(t+1,j+1)}$の計算の順番を決める規則は特にない．次の計算点にランダムに移動しても，indexを体系的に移動しても，本質的な違いはない．ただし，移動順は計算速度に大きく影響する．LDAの場合は，通常は，体系的に移動する．LDAで体系的に移動する理由は次の点に関連する．

第4に，$z_{d_1,i_1}^{(t,j)}$は，ある単語(の場所)にあるトピックkが割当てられる確率であるから，離散確率分布である．そのため，一番内側のKループのように各(d^*,i^*)で$z_{d^*,i^*}^{(t,j)}$の分布全体を計算できる．つまり，

$$p\left(z_{d^*,i^*}^{(t,j)}=1 \mid z_{\neg d^*,i^*}^{(t,j-1)}, w_{\neg d^*,i^*}^{(t,j-1)}\right),\cdots,p\left(z_{d^*,i^*}^{(t,j-1)}=K \mid z_{\neg d^*,i^*}^{(t,j-1)}, w_{\neg d^*,i^*}^{(t,j-1)}\right) \tag{10}$$

を計算する．次の計算点$z_{d_x,i_y}^{(t,j+1)}$の計算には，どれかのトピックの確率値$z_{d^*,i^*}^{(t,j)}=k^1(k^1 \in K)$を$z_{\neg d_x,\neg i_y}^{(t,j+1)}$の1つの値として使う．次の計算点でどのトピック$k$に対応する $p\left(z_{d^*,i^*}^{(t,j)}=k\right)$ の値を使うかは，計算した$P\left(z_{d^*,i^*}^{(t,j)}\right)$からサンプルして決める．このサンプリング自体は簡単である．$P\left(z_{d^*,i^*}^{(t,j)}\right)$の各トピックの出現確率に応じて0-1区間を区分し，各区間にトピックのラベルを付けたスケールを用意しておく．例えば，AとBの2つのトピックで，それぞれの割当確率が0.7，0.3であれば，0-0.7をラベルA，0.7-1.0をラベルBとする．その後，0-1区間でランダムな値を生成し，その値とスケールを比べて，トピックAかBかを決めればよい．

第5に，$z_{d,i}$を計算する際の右辺から$w_{d,i}$，つまり計算対象の単語(d,i)の観察された単語種は計算対象から除く必要がある．概念的には，単語(d,i)計算時には，$z_{d,i}$は確率分布しか決まっていないのだから，$w_{d,i}$はLDAモデル上ではまだ存在していないためと理解できる．逆にみると，3.1.2でみたように，$w_{d,i}$つまり単語(d,i)の単語種vを観察された$w_{d,i}$と関係なくサンプリングで決めることも原理的には可能であり，Bag of wordsを仮定するLDAではそのように，つまり生成モデルとして考えるべきである．観察された$w_{d,i}$と関係なく文書全体を生成する計算方法でも結果は変わらないが，一方でコードはあきらかに複雑になり，実践的ではない．

第6に，(9)式が示すとおり，Dirichletパラメータの値の影響は消滅しないが，実際の観測データが増えるとその影響は小さくなる．Dirichletパラメータが現実のデータからはかけ離れた設定であっても，その条件のもとで，データにもっとも適合するような単語へのトピック割当分布が計算される．つまり，データをもっとも良く説明するためには，与えられたDirichlet分布からの出現確率が非常に低くなるような多項分布パラメータの値が必要な場合もあり得る．ここで注意すべき点は，確率は低くても，連続確率分布であるDirichlet分布はそのような値が出現することを排除しない点である．この場合，与えられたLDAモデルの枠組みの中でもっともよくデータを説明するパラメータが選ばれるが，LDAモデル全体としてのデータ説明能力は低くなる．

第7に，CGSのサンプリング結果の扱い方には注意が必要である．3.3.1，Figures 4, 5で検討したように，$z_{d,i}$の計算を(d,i)について一巡するだけで，確率分布$P(z_{d,i})$を正確に把握できる可能性は低い．一般的なMCMCサンプリングでは，まとはずれな値かもしれない初期値の影響がなくなるまで一定数MCMCサンプリングをおこない(burn-in)，その後何巡(T回)かサンプリングを繰り返し確率分布が収束したことを確認し，収束した確率分布から何回かサンプリング(t回)し，T回から$T+t$回の結果の平均をとって，最終結果とする．最終結果用のt回のサンプリングにおいてサンプル間の相関を避けるため，t回のうち，s回に1回しか最終結果用のサンプルとせず，残りは使わないとすることも多い(thinning)．LDAでも，$z_{d,i}$の収束した事後分布が得られたとしても，事後分布からの1回のサンプリングでその事後分布の期待値と一致する$z_{d,i}$が出現するわけではない．各面が1/6の確率で出るサイコロを正しくモデル化しても，そのサイコロをふった結果として1の目が10回連続で出ることがあり得ることと同じである．したがって，CGSの結果からパラメータθ,ϕの値を計算する場合も，T回サンプリング後の事後分布からさらにt回サンプリングし，t回のサンプリング平均を使うことが考えられる．ただし，実装上は，サンプリング結果から平均をとることはおこなわず，最後のイテレーション1回の結果を最終結果として使うことが一般的である．この点はLDAモデルの構造そのものにかかわっているため，実際にLDAモデルの計算をおこなう4.2節で再び検討する．

3.3.4 MalletのSparse LDA

MalletのSparse LDAアルゴリズムもCGSであるが，(1) サンプリング順序の工夫による高速化，(2) 文書別トピック分布側Dirichletパラメータの最適化，(3) サンプリング結果平均の出力可能という計算方法の違いがある．ここでは，この3点を簡単にみておく．

Sparse LDAの第1の特徴は，サンプリング順序の違いである．ある文書で出現頻度がゼロのトピック，あるトピックで出現頻度がゼロの単語種のサンプリングを避けるため，(9)式の最終行から得られるCGS更新式を次のように3つの部分に分解する．

$$
\begin{aligned}
p\left(z_{d,i} \mid z_{\neg(d,i)}, \alpha, \beta\right) &\propto \left(\alpha + N_{d,w,k}^{\neg(d,i,k)}\right) \cdot \frac{\beta + N_{v,w,k}^{\neg(d,i,k)}}{V\beta + \sum_{v=1}^{V} N_{w,v,k}^{\neg(d,i,k)}} \\
&= \frac{\alpha\beta}{V\beta + \sum_{v=1}^{V} N_{w,v,k}^{\neg(d,i,k)}} + \frac{\alpha N_{w,v,k}^{\neg(d,i,k)}}{V\beta + \sum_{v=1}^{V} N_{w,v,k}^{\neg(d,i,k)}} + \frac{N_{d,w,k}^{\neg(d,i,k)} N_{w,v,k}^{\neg(d,i,k)}}{V\beta + \sum_{v=1}^{V} N_{w,v,k}^{\neg(d,i,k)}}
\end{aligned}
\tag{11}
$$

ここで，(11)式2行目への分解は単純に1行目のカッコをはずしただけである．ただし，2行目には分子が$N_{d,w,k}^{\neg(d,i,k)}\beta$の交差項は存在しない．LDAモデルはDirichletパラメータβと文書別トピック分布とは無関係と設定しているため，この交差項は単語へのトピック割当分布計算式にもともと存在しない．なお，擬似コードと(11)式からわかるように，右辺の単語カウントからは単語へのトピック割当を計算している単語(d,i)の単語種の分に相当する1を引くことに注意する．記号法がややこしくなるが，$N_{d,w,k}^{\neg(d,i,k)}$は，(d,w,k)に該当する単語数から(d,i,k)分を引いたカウント数を意味している．

この分解はA/B/C分解と呼ばれ(McCallum 2002)，(11)式2行目の右辺第1項から順にA項，B項，C項となる．Malletは，初期値として文書別トピック分布はある1つのトピックだけが1で残りのトピックはゼロ，トピック別単語種分布ではある単語種だけが1で残りの単語種にはゼロを割当てる．したが

って，初期状態では，$D\times K$，$K\times V$のそれぞれの確率分布表のそれぞれD個のセル，V個のセルが1で，残りのセルはゼロである．サンプリングした結果，ある文書に新たなトピック，あるトピックに新たな単語種があらわれたときに確率分布表の対応するセルがゼロでなくなる．

あるトピックkについてのα_k，$\beta_k=\sum_v \beta_{k,v}$は非ゼロであるため，A項は文書，単語にかかわらず常に非ゼロの値である．一方，α_kとβ_kの積が分子であるためA項の値は一般的に小さくなる．ここから，A項は背景項とよばれる．特に，Dirichletパラメータが対称固定の場合は，A項の分子は定数になり，A項の計算は単純になる．B項の値は単語種vがトピックkにどれだけ出現するかに依存し，トピックkでvが使われていなければ値はゼロである．C項の値は文書dがトピックkの単語をどれだけ使っているかに依存し，トピックkが文書に現れなければC項の値はゼロである．

Malletの単語(d,i)へのトピック割当では，まずA，B，Cのどの項を使ってkをサンプルするかを，$\sum_k A_k/T:\sum_k B_k/T:\sum_k C_k/T \quad (T=\sum_k A_k+\sum_k B_k+\sum_k C_k)$の比に従って選択する．例えばサンプリングする項がB項に決まると，トピック割当を$B_1:B_2:\cdots:B_K$の比に従って選択する．A項の値が相対的に小さいため，結果的に，A項, B項, C項の中でA項はサンプリングにはあまり使われない．B項，C項はサンプリングに使われることが多いが，B項，C項はゼロ値を多く含むためイテレーションは高速におこなわれる．これがMalletのCGSをSparse LDAと呼ぶ理由である．一方，A項がサンプリングに使われる可能性はゼロではないため，ある単語(d,i)とトピックのすべての組合せで計算がおこなわれることが原理的に保証されている．

Sparse LDAの2番目の特徴は，文書別トピック分布側のDirichletパラメータα_kを要素ごとに最適化する機能を持っていることである．Malletは，サンプリングでは無く，対数尤度の最大化を不動点反復法で解くことでタα_kの最適値を計算している．つまり，

$$\frac{\partial}{\partial \alpha_k}\log p(z\mid \alpha_k)=0 \tag{12}$$

とするα_kをみつける．長々とした式の変形となるため途中を省略しているが，この式は次のクローズドな形であらわすことができる．

$$\begin{aligned}\frac{\partial}{\partial \alpha_k}\log p\,(z\mid \alpha) &= \sum_{d=1}^{D}\left[\psi\left(\alpha_k+N_{w,d,k}\right)-\psi(\alpha_k)\right]-\sum_{d=1}^{D}\left[\psi\left(\sum_k \alpha_k+N_{w,d,k}\right)-\psi\left(\sum_k \alpha_k\right)\right]\\ &= G_k(\alpha_k)-H_\alpha(\alpha_k)\\ &= 0\end{aligned} \tag{13}$$

ここで，$\psi(\ \ )$はΓ関数の微分であるdigamma関数 $\psi(x)=\frac{d}{dx}\log\Gamma\,(x)=\frac{\Gamma\prime(x)}{\Gamma(x)}$ である．もとの尤度関数がΓ関数を含んでいるため，尤度関数の微分にdigamma関数があらわれる．G_k, H_αはそれぞれの項を単に置き換えただけである．

不動点反復法は，一般的には，$F(x^*)=0$とするx^*を求めるとき，$F(x)=f(x)-x$のように書き換えることができれば，$f(x^*)=x^*$とするx^*が求める答えになることを使う．直感的には，x-y平面上に単調な$y=f(x)$を描き，さらにx=yである45度線を描くと，$y=f(x)$と45度線の交点はy=x+0=$f(x)$+0

になる．45度線と$f(x)$が一定条件にあるとき，単純な更新規則で交点に収束することは作図するとわかる．ただし，α_kの最適化には，$G_k - H = 0$ならば$G_k/H = 1$であることを利用し，差分ではなく，比を使った更新式

$$\alpha_k^{(t+1)} = \alpha_k^{(t)} \frac{G_k\left(\alpha_k^{(t)}\right)}{H_\alpha\left(\alpha_k^{(t)}\right)} = f(\alpha_k) \tag{14}$$

を使って反復計算する．(14)更新式は，直感的には，ある文書dで実際に使われているトピックの総量$H_\alpha\left(\alpha_k^{(t)}\right)$に対するトピック$k$の使用量$G_\alpha\left(\alpha_k^{(t)}\right)$にの比を使って，現在のトピック$k$の出現確率の期待値である$\alpha_k^{(t)}$を修正していると解釈できる．つまり，$t$回目イテレーションで$k$がよく使われたならば$\alpha_k^{(t+1)}$を増やし，使われていないならば減らす．ここで，$H_\alpha\left(\alpha_k^{(t)}\right)$は基本的に$\alpha_k$の総和を意味しているものの，各$\alpha_k$は更新途中であり，かつ$H$で置き換える前の式からわかる通り$\alpha$の非線形項であるので，定数にならない点に注意する．

不動点反復法は，3.4で検討するldaパッケージのVEMが使うNewton-Raphson法と比べると2次の偏微分(ヘッシアン)を計算する必要がないため計算量が少なく，収束するとすればNewton-Raphson法より安定して収束する．ただし，更新式$f(x)$には，$x \in \{x_1{:}x_2\}$がとる範囲より$f(x \in \{x_1{:}x_2\})$の範囲が狭くなければならないという一般的な制限がある．これはLDAのα最適化の場合は問題にならない．

トピック別単語種分布側のDirichletパラメータβも同様な方法で最適化できるが，Malletはβについてはβ_0の最適化だけを実装している．β_iの最適化は計算コストに対してトピック分析精度改善の効果が小さいか，そもそも改善効果がほぼないためである(Wallach, Mimno and McCallum 2009).

Sparse LDAの3番目の特徴はサンプリング結果の計算方法である．詳しくは4.2.1で検討するが，topicmodelsパッケージのGibbsおよびldaパッケージは，サンプリングの最後の1回の情報だけを用いて文書別トピック分布θ，トピック別単語種分布ϕの事後期待値の推定値として返す．Malletもデフォルトでは，Gibbs，ldaと同様に最後の1サンプルだけを使うが，オプションでいくつかのサンプル平均を推定値として出力させることが可能である(Magnusson 2026)．これは，MalletはLDAモデルをJAVAのオブジェクトとして作成し，そのオブジェクトにtrainメソッドとestimateメソッドをもたせているからである(McCallum 2002)．つまり，モデルをオブジェクトとして作成することで，モデルの再利用を容易にしている．その結果，同じオブジェクトに対して，trainメソッドでパラメータを推定した後にestimateメソッドでサンプルを集めるということが容易におこなえる．estimateメソッドではサンプル間の相関を下げるためにサンプルの間引きをおこなうthinも指定できる．topicmodels，ldaでも，若干のコード上の工夫により，Malletと同様の処理は可能である．ただし，4.2.1で検討するLDAモデルをCGSで数値計算する場合の構造的問題により，本研究ノートでは複数サンプルによる推定機能は使わない．

3.4 Variational Expected Method (VEM)

3.4.1 真の分布と近似分布

VEMはVariational Bayes (VB)法の一つの手法である．VB法の基本的アイディアは，真の確率分布

$P(x)$を近似し，かつ計算しやすい近似確率分布$Q(x)$を設定したうえで，$Q(x)$を最適化によって調整することで$P(x)$になるべく近づけることである(Blei, Kucukelbir and McAuliffe 2017; Murphy 2023, pp. 439-482; Tran, Nguyen and Dao 2021)．したがって，最終的に得られる最適化後の$Q^*(x)$は真の確率分布$P(x)$には一致せず，どれだけ$P(x)$に近くなるかも場合によって変わる．$P(x)$と$Q(x)$の差は，通常はKallback-Leibler (KL) divergenceで定義する．真の確率分布を$P(x)$，近似分布を$Q(x)$とすると，確率分布間の違い(距離)の定義の1つであるKL divergence (KL)は，

$$\begin{aligned}\mathrm{KL}\big(Q(x) \parallel P(x)\big) &= -\int q\,(x)\log\frac{p(x)}{q(x)}dx \\ &= -\int q\,(x)\log p(x)\,dx - \left(-\int q\,(x)\log q\,(x)dx\right) \\ &= \int q\,(x)\log\frac{q(x)}{p(x)}dx \\ &= \mathbb{E}_p\left[\log\frac{q(x)}{p(x)}\right]\end{aligned} \tag{15}$$

となる．つまり，KL divrgenceは確率分布のxの各点での真の確率と近似確率の対数差を近似分布の確率をウェイトとして加重平均したものである．(15)式2行目から$P(x)$と$Q(x)$の相対エントロピーともよばれる．3行目の式は$\log\big(q(x)/p(x)\big)$の期待値の計算式になっているため，4行目の式に書き換えることができる．(15)式からわかるように，2つの確率分布$P(x)$と$Q(x)$のKL divergenceとして$\mathrm{KL}\big(Q(x) \parallel P(x)\big)$と$\mathrm{KL}\big(P(x) \parallel Q(x)\big)$の2通りを考えることができ，それらの値は異なる．通常は，$P(x)$が計算不能あるいは未知のため$P(x)$をウェイトとして使うことはできないから，$\mathrm{KL}\big(Q(x) \parallel P(x)\big)$を使う．

ここであたらしい確率変数Yを$Y:=p(x)/q(x)$と定義する．Yの期待値は，

$$\mathbb{E}_p[Y] = \mathbb{E}_p\left[\frac{p(x)}{q(x)}\right] = \int q\,(x)\frac{p(x)}{q(x)}dx = \int p\,(x)dx = 1 \tag{16}$$

となる．Jensenの不等式(Bishop 2006, p. 56)より，$f(t)$が凸関数のとき$f(E[t]) \le E[f(t)]$である．$-\log(t)$は凸関数であるから，

$$-\log\left(\mathbb{E}_p\left[\frac{p(x)}{q(x)}\right]\right) = -log(1) = 0 \le \mathbb{E}_p\left[-\log\frac{p(x)}{q(x)}\right] = \mathbb{E}_p\left[\log\frac{q(x)}{p(x)}\right] = \mathrm{KL}(Q \parallel P) \tag{17}$$

となり，KL divergenceは非負である．

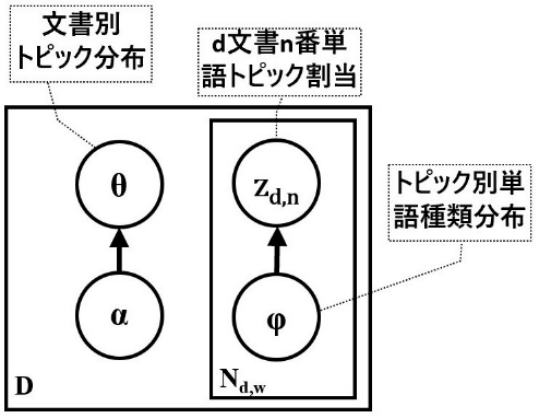
VB法でLDAモデルを計算するために，まずLDAモデルの近似分布を考える．topicmodelsパッケージのVEMでは，トピック別単語種分布は多項分布パラメータϕを直接推定し，ϕをサンプルするDirichlet分布は使わない．なお，Blei, Ng and Jordan (2003)のオリジナルペーパーではトピック別単語種多項分布パラメータの記号にβを使っているので注意されたい．(1)式を簡潔にするため，LDAモデルをある1つの文書dについてのみ考えると，

$$p(w_d, z_d, \theta \mid \alpha, \phi_k) = p(\theta_d \mid \alpha) \prod_{n=1}^{N_d} p\left(w_{d,n} \mid z_{d,n}, \phi_k\right) p\left(z_{d,n} \mid \theta_d\right) \prod_{k=1}^{K} p\left(\phi_k\right) \tag{18}$$

とかける．この確率分布$(w_d, z_d, \theta \mid \alpha, \phi_k)$をVEMで推定するために，近似分布として(19)式のθ_dとz_dの同時分布を設定する(Blei, Ng and Jordan 2003; Gruen and Hornik 2011).

$$q\left(\theta_d, z_d \mid \gamma(\boldsymbol{w}_d), \eta(\boldsymbol{w}_d)\right) = q_1\left(\theta_d \mid \gamma_d(\boldsymbol{w}_d)\right) \cdot \prod_{i=1}^{N_{d,w}} q_2\left(z_{d,i} \mid \eta_i(\boldsymbol{w}_d)\right) \tag{19}$$
$$\text{where } \theta_d \sim Dirichlet_k\left(\gamma_d(\boldsymbol{w}_d)\right),\ z_{d,i} \sim Multinomial_k\left(\eta_i(\boldsymbol{w}_d)\right)$$

Figure 7. VEMのGraphical Model



近似分布$q(\cdot)$は，(19)式とFigure 7のGraphical Modelが示すように，文書別トピック分布q_1とトピック別単語種類分布q_2の積であり，q_1とq_2に共通するパラメータはないから，q_1とq_2は完全に分離できると想定している．確率分布$Q(A, B)$を相互に独立した因子の積$Q(A) \cdot Q(B)$にすることを因子化といい，$Q(A) \cdot Q(B)$を因子化された分布(factorized distribution)という．q_1はγ_kを近似調整パラメータとするDirichlet分布，q_2は$\eta_{i,k}$を近似調整パラメータとする多項分布を想定する．これら調整パラメータの調整は観察データである文書dの全単語W_dに依存している．なお，以下では，ややまぎらわしいが，本来のLDAモデルを「真のモデル」とよぶこととする．

変分近似の調整パラメータであるγとηを調整して近似分布$q(x)$を真の分布$p(x)$に近づける．しかし，真の分布$p(x)$がわからないため，$p(x)$と$q(x)$のKL divergenceを直接に最小化することは困難である．そこで，近似分布のELBO (Evidence Lower Bound) を最大化すれば，近似分布が真の分布に近づくという性質を利用する．まず真のモデルの周辺化確率(evidence，正規化項ともいう)である$P(w \mid \alpha, \phi)$と近似分布が与えるELBOの関係は(20)式のようになる．

$$
\begin{aligned}
\log p(w \mid \alpha, \phi) &= \log \int_\theta \sum_z p\,(\theta, z, w \mid \alpha, \phi) d\theta \\
&= \log \int_\theta \sum_z p\,(\theta, z, w \mid \alpha, \phi) \frac{q(z, \theta \mid \gamma, \eta)}{q(z, \theta \mid \gamma, \eta)} d\theta \\
&\geq \int_\theta \sum_z q\,(\theta, z) \log p\,(\theta, z, w \mid \alpha, \phi) d\theta - \int_\theta \sum_z q\,(\theta, z) \log q\,(\theta, z, w \mid \gamma, \eta) d\theta \\
&= \mathbb{E}_q[\log p(\theta, z, w \mid \alpha, \phi)] - \mathbb{E}_q[\log q(\theta, z \mid \gamma, \eta)]
\end{aligned}
\tag{20}
$$

ここで，周辺化確率$p(w \mid \alpha, \phi)$とは，形式的には，$p(w \mid \alpha, \phi)$を与える確率モデルの全パラメータα, ϕについて積分(周辺化)した結果である．$p(w \mid \alpha, \phi)$の周辺化確率の定義は，3.3.2で検討した周辺化と同じである．つまり，周辺化確率は，ある確率モデルのすべてのパラメータがとり得るすべての値におけるデータwの出現確率であり，すべてのモデルのパラメータを積分消去した後の確率である．したがって，周辺化確率は，特定の確率モデル世界の中でのデータ出現確率とみることができる．周辺化確率を実際に計算することは一般的に困難であるが，概念的には難しいものではない．周辺化確率のこのような理解はベイズ確率モデルの核心にある考え方である．モデルAの周辺確率$p(x)_A$とモデルBの周辺確率$p(w)_B$を比べて，$p(w)_A > p(w)_B$であるとき，現実世界ではデータxは観察されていて$p(w) = 1$であるから，それにより近いモデルAの方がより現実説明力の高いモデルという考え方である．なお，(20)式では，式を簡潔にするために添え字は可能な限り省略した．

(20)式で，文書別トピック分布パラメータθは連続値，単語へのトピック割当zは離散値であるので，積分と集計を使い分けているが本質的ではない．どちらもパラメータのとり得る値で周辺化している．2行目は，1行目に$q(\cdot)/q(\cdot)$というθの全域で1となる項を加えただけである．2行目と3行目の関係は関数logについてJensenの不等式を適用することで不等号の関係であることがわかる．3行目は，第1項が$\backslash logp(\cdot)$を$q(\cdot)$ウェイトで，第2項が$\backslash logq(\cdot)$を$q(\cdot)$ウェイトで積分しているので，符号に注意すれば，それぞれの情報量の期待値である．最後の行が近似分布$q(\cdot)$に関するELBOの定義になっている．(20)式の左辺と最終行，KL divergenceの(15)定義式から，対数周辺化確率とELBOの差がKL divergenceになることはあきらかである．あるいは，ELBOとKL divergenceの和が対数周辺確率になる．(20)式を書き直すと，

$$
\begin{aligned}
&\log p\,(\boldsymbol{w} \mid \alpha, \phi) = \mathrm{ELBO}_q + \mathrm{KL}\big(q(\cdot) \parallel p(\cdot)\big) \\
&\mathrm{KL}\big(q(\cdot) \parallel p(\cdot)\big) = -\mathrm{ELBO}_q + \mathrm{const.}
\end{aligned}
\tag{21}
$$

と書ける．周辺確率$\log p\,(w \mid \alpha, \phi)$は，このモデルについては定数であるから，KL divergenceを最小にするにはELBO_qを最大にすればよいことがわかる．

3.4.2 ELBOの最大化

ELBOが最大のときが近似モデルが真のモデルにもっとも近いときであるから，ELBOの最大化を考える．ELBO を$\mathbb{L}(\gamma, \eta; \alpha, \beta)$ とおくと，次のように分解できる．

$$
\begin{aligned}
\mathbb{L}(\gamma, \eta; \alpha, \phi) &= \mathbb{E}_q[\log p(\theta, z, w \mid \alpha, \phi)] - \mathbb{E}_q[\log q(\theta, z \mid \gamma, \eta)] \\
&= \mathbb{E}_q[\log p(\theta \mid \alpha)] \\
&\quad + \mathbb{E}_q[\log p(z \mid \theta)] \\
&\quad + \mathbb{E}_q[\log p(w \mid z, \phi)] \\
&\quad - \mathbb{E}_q[\log q(\theta \mid \gamma)] \\
&\quad - \mathbb{E}_q[\log q(z \mid \eta)]
\end{aligned}
\tag{22}
$$

(22)式の分解は，同時確率と条件付確率の定義に従って$p(\cdot), q(\cdot)$を分解しているだけである．LDAモデルの設定から，真のモデルの文書別トピック分布パラメータθはDirichletパラメータαのみに依存し(第2行右辺第1項)，単語へのトピック割当zはθのみに依存し(第2行右辺第2項)，単語への単語(種)割当wはzとϕのみに依存し(第2行右辺第3項)，近似分布qは因子化の設定から調整パラメータはγはθのみ(第2行右辺第4項)，ηはzのみ(第2行右辺第5項)に影響する．分解の結果，想定した多項分布，Dirichlet分布を分解後の各項に単純にあてはめることができる．多項分布，Dirichlet分布の期待値の公式を使うと，ELBOの期待値を(23)式の計算可能な関数形であらわすことができる．

(23)式の5つの項が(22)式の5つの項に順番通りに対応している．$\Psi(x)$関数は$\Gamma(x)$関数の対数微分であるDigamma関数である．(22)式の各項の(23)式の対応する項への書き換えは，基本的に多次元のDirichlet分布とMultinomial分布の平均の公式を使って書き換える．対数Dirichlet分布$\backslash logp(\theta \mid \alpha)$の期待値への書き換えは，まず対数Dirichlet分布をパラメータα_iと十分統計量$\log \theta_i$を使って指数族分布の一般形に書き換える．十分統計量とは，その情報を使って計算すれば元の分布を再現するのに十分な情報のことである．つまり，多項分布の各項の出現数が全部の項について存在すれば(1からの6のサイコロの目が何回ずつ出たがすべてわかっていれば)，各項の出現確率の期待値(Dirichlet分布パラメータ)を計算できるので，各項の出現数は十分統計量である．続いて，指数型分布族の期待値の公式により$\theta_i = \psi(\alpha_i) - \psi(\alpha_0)$であることを使う．これは，結果だけをみれば，3.1.3でみたDirichlet分布$P(\theta \mid \alpha)$の期待値$\mathbb{E}[\theta_i] = \alpha_i / \alpha_0$の対数表示である．これらの書き換えは，変分ベイズで一般的に使われる手順であるため，ここでは書き換え手順の詳細は検討はしない．

$$
\begin{aligned}
&\mathbb{L}(\gamma,\eta;\alpha,\phi)\\
&= log\Gamma\left(\sum_{j=1}^{K}\alpha_j\right)-\sum_{i=1}^{K}\log\Gamma\left(\alpha_i\right)+\sum_{i=1}^{K}(\alpha_i-1)\left(\Psi(\gamma_i)-\Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right)\\
&+\sum_{n=1}^{N_w}\sum_{k-1}^{K}\eta_{k,n}\left(\Psi(\gamma_i)-\Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right)\\
&+\sum_{n=1}^{N_w}\sum_{k=1}^{K}\sum_{j=1}^{V}\eta_{k,n}\,w_n^j\log\phi_{i,j}\\
&-\log\Gamma\left(\sum_{j=1}^{K}\gamma_j\right)+\sum_{i=1}^{K}\log\Gamma(\gamma_i)-\sum_{i=1}^{K}(\gamma_i-1)\left(\Psi(\gamma_i)-\Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right)\\
&+\sum_{n=1}^{N_w}\sum_{i=1}^{K}\eta_{i,n}\log\eta_{i,j}
\end{aligned}
\tag{23}
$$

ELBOの最大化アルゴリズムは次に検討するが，そこではELBOの2階導関数が必要となり，ガンマ関数の2階微分であるtrigamma関数$\Psi'(x)$が必要となる．trigamma関数は，

$$\Psi'(x)=\frac{d}{dx}\log\Psi\left(x\right)=\frac{d^2}{dn^2}\log\Gamma\left(x\right)$$

である．trigamma関数は直接には計算できないが，Rは級数表示と漸近展開を用いてGamma関数のn階導関数を近似計算するpolygamma関数を実装している．polygamma関数の詳細は，stat.ethz.ch/R-manual/R-devel/library/base/html/Special.html#reference+Special.Rd+R+3AAmos +3A1983にある．

3.4.3 ELBOの最大化アルゴリズム

第4章で使うパッケージでは，topicmodelsパッケージのみがVEMを実装している．topicmodelsのVEMは，ELBOの最大化にExpectation-Maximization (EM) 法を使っている．EM法は，潜在変数を持つ確率モデルにおいて，最適なモデルのパラメータを用いるために一般的に使われるアルゴリズムである．アルゴリズムの概略は，(1)未知のパラメータに初期値をセットする，(2)現在のパラメータのもとで観測データから潜在変数の事後分布の期待値を計算する(Eステップ)，(3)Eステップで得られた期待値を最大化するようパラメータを更新する(Mステップ)，(4)期待値の変化が一定範囲内に収束するまで(2)と(3)を繰り返す，というものである．VEMの場合は，最大化の対象がELBOであるため，一般的なEM法での潜在変数やパラメータを適切に読み替えなければならないが，アルゴリズムの基本構造は同じである．topicmodelsのVEMによるLDAモデルでのELBO最大化のアルゴリズムは次のようになる．

(1)初期値設定

真の分布Pのパラメータα,βはCGSの初期値設定と同様に設定する． 近似分布Qで文書別トピック分布を制御する調整パラメータ$\gamma_{d,k}$の初期値は，デフォルトでは$\alpha_k+N_{d,w}/K$を設定する．この初期

値は，文書dの単語数$N_{w,d}$のときにトピックkに属する単語の出現数を事前の信念α_kと総単語数をトピック数で割った数で調整して予想した数である．予想される最適値に近いとおもわれる値を初期値として与えることで最適値への計算を節約する工夫といえる．近似分布の調整パラメータη_iの初期値は，単語種vが各トピックで同等に出現すると予想し，1/Kとする．

(2)E-step

E-Stepでは，γ,ηを初期値からスタートして，ELBOを最大化する方向に調整する．最適化の一般的な考え方どおり，ELBOをγ,ηで偏微分し，偏微分をゼロにするγ,ηを求める．前節でみたとおりELBOの関数形はわかっている．そこから，ELBO偏微分をゼロにするγ,ηを与える式を次のようなクローズドな形で得ることができる．まず，ηの更新式は，

$$\eta_{d,i,k}^{(t+1)\prime} = \phi_{d,i,k}^{(t)} \exp\left(\Psi\left(\gamma_{d,k}^{(t)}\right) - \Psi\left(\sum_{k=1}^{K}\gamma_{d,k}^{(t)}\right)\right) \quad \left(d \in \{1{:}D\}, i \in \left\{1{:}N_{w,d}\right\}, k \in \{1{:}K\}\right)$$
$$\eta_{d,i,k}^{(t+1)} = \frac{\eta_{d,i,k}^{(t+1)\prime}}{\sum_{j=1}^{K}\eta_{d,i,j}^{(t+1)\prime}} \quad (d \in \{1{:}D\}) \tag{24}$$

となる．ここで，tは計算の順番であり，t=0を初期値とする．ηでは更新後に正規化するイテレーションの中間ステップ(t+1)’が入る．

続いて，γの更新は，

$$\gamma_{d,k}^{(t+1)} = \alpha^{(t)} + \sum_{n=1}^{D}\eta_{d,n,i}^{(t+1)} \tag{25}$$

となる．η,γともに，基本的に，トピック$k^{\star}$に割当てられることになった単語数で調整している．

一般のEM法のE-stepでは期待値を計算するだけであるが，VEMのE-stepではE-step内で反復計算をおこなって，与えられた$\alpha^{(t)},\phi^{(t)}$のもとで，つまりt番からt+1番のイテレーションの間に，γとηを更新をサブループで反復計算して更新する．γとηが相互に影響しているため，$\alpha^{(t)},\phi^{(t)}$のもとでのγとηの最適値を同時に求めることはできないからである．E-step内のtからt+1へ移るためのサブループでγとηの両方が最適値γ^{*}とη^{*}に収束すると，γ^{*}とη^{*}をEM全体の第t+1番ループのγとηの値としてセットしM-stepに渡す．この構造から，ELBO最大化のアルゴリズでは，E-stepの収束判定基準，次に説明するM-stepを合わせたEM全体の収束判定，さらにELBO最大化の収束基準の3つの収束基準が使われる．この点については，EM法の説明の後で検討する．

(3) M-step

M-stepでは，与えられたγとηのもとで，ELBOを最大化するϕを求める．ϕの更新式は，γ,ηと同様に，ϕについてのELBOの1階偏導関数を0とする条件式を使えばよいので，

$$\phi_{k,i}^{(t+1)} \propto \sum_{d=1}^{D} \sum_{i=1}^{N_{w,d}} \eta_{d,n,i}^{(t+1)} w_{d,i}^{j} \tag{26}$$

ここで$w_{d,i}^{j}$は文書dのi番目の単語$w_{d,i}$が単語種jのときに1となるフラグ変数である．つまり，E-stepで更新した$w_{d,i}$へのトピック割当$\eta_{d,n,i}^{(t+1)}$をウェイトとして単語種jの単語数を全文書Dの全単語Wについてトピック別単語種類別にクロス集計している．

3.4.4 Dirichletパラメータの最適化

topicmodelsのVEMは，文書別トピック分布パラメータθを制御するDirichlet分布パラメータのα_0を可変に設定して，最適化することができる．ELBOのα_0についての1階偏導関数をゼロにする式はクローズドな解にならないため，topicmodelsのVEMでは，ELBO最大化の枠外で，Newton-Raphson法を使ってELBOの変化をゼロとする$\alpha_0^{\star}$をみつける．Newton-Raphson法は，関数$f(x)$を2次の項までテイラー展開した関数で近似し，$f(x^{\star}) = 0$とする$x^{\star}$を探す一般的なアルゴリズムである．関数$f(x)$において，x点から$x+\delta$点までわずかに移動した場合の$f(x+\delta)$は，

$$f(x+\delta) \approx f(x) + f'(x)\delta + \frac{1}{2} f''(x)\delta^2 \tag{27}$$

と近似できる．$f(x)$の極値は，計算点をわずかに移動しても$f(x)$の値が変わらない点であるので，

$$\begin{gathered} \frac{d}{d\delta} f(x+\delta) = f'(x) + f''(x)\delta = 0 \\ \delta = -\frac{f'(x)}{f''(x)} \end{gathered} \tag{28}$$

となる．ここで，数値計算上注意する点は，$\delta = -f'(x)/f''(x)$で与えられる計算点の移動は，現在の計算点xでの$f(x)$の近似式から計算したものであるため，δ分移動した点$x+\delta$で$f(x+\delta) = 0$になるとは限らないことである．しかし，(33)式による計算点の移動は常に$f(x) = 0$とする点に向かうため，反復計算により$f(x) = 0$となる点に到着する可能性は高い．「可能性が高い」という表現にとどまるのは，Newton-Raphson法は，テイラー展開の打ち切りによる近似誤差の存在が避けられない，$f(x)$の関数形によっては計算点の移動によって振動，発散が起き収束しない，$f''(x)$があまりに微小になると数値計算できなくなるといった原理的，数値計算的な問題を持っているからである．

数値計算上，特に問題になるのは，(28)式は実際にどれだけ移動すべきかという情報を与えないことである．つまり，テイラー展開が真の値の近似であることはδが”微小量”であることで成り立っているが，δが実際の計算上どれだけのスケールなのかは(28)式からはわからない．Newton-Raphson法では，実際にどれだけ移動するのか，つまりステップ幅を別途なんらかの方法で計算する場合も多い．topicmodelsのVEMでは，(29)式を使って計算点を移動する．

$$\boldsymbol{\alpha}_k^{(t+1)} =: \boldsymbol{\alpha}_k^{(t)} - \boldsymbol{H}^{-1} g\left(\boldsymbol{\alpha}_k^{(t)}\right) \tag{29}$$

ここで，gは$\alpha^{(t)}$での勾配，つまり$\left(\partial ELBO/\partial \alpha_1^{(t)}, \cdots, \partial ELBO/\partial \alpha_K^{(t)}\right)$である．$H^{-1}$はELBOのヘッセ行列$H$の逆行列である．逆行列の計算量は$O(N^3)$であり，D個の文書それぞれについて，$\alpha$の1回の更新ごとに，$K \times K$サイズの逆行列計算をおこなう必要があるため，計算量が大きくなる．しかし，ELBOのヘッセ行列Hは，$i \neq j$のとき，

$$\frac{\partial^2}{\partial \alpha_i \partial \alpha_j} ELBO = \frac{\partial^2}{\partial \alpha_j \partial \alpha_i} ELBO = z \tag{30}$$

となり，非対角成分がすべて同一になる．ここから，$H = \mathrm{diag}(\mathbf{h}) + r\mathbf{1}\mathbf{1}^T$と書くことができる．$\mathbf{1}$はH行列のサイズと同じ長さの単位ベクトルである．行列がこの形式の場合，Sherman-Morrisonの公式を用いることで、通常の逆行列の計算をすることなく，各要素 h_i と r を使った簡潔な形で表せる．ヘッセ行列Hが3×3であれば，H^{-1}の各成分i, jは(31)式のようになる．

$$\begin{aligned} H_{i,i}^{-1} &= \frac{1}{h_i} - \frac{C}{h_i^2} \\ H_{i,j}^{-1} &= -\frac{C}{h_i h_j}, \text{ if } i \neq j \end{aligned} \tag{31}$$

Dを文書数，h_iをi番目の対角成分とすると，

$$\begin{aligned} S &= \sum_{i=1}^{3} \frac{1}{h_i} = \frac{1}{h_1} + \frac{1}{h_2} + \frac{1}{h_3} \\ C &= \frac{z}{1 + zS} \\ z &= D \cdot \psi'\left(\sum_{k=1}^{K} \alpha_k\right) \end{aligned} \tag{32}$$

となる．なお，zは$\partial^2 ELBO/\partial \alpha_i \partial \alpha_j = \partial^2 ELBO/\partial \alpha_j \partial \alpha_i = z$の$z$である．

3×3行列の例では次のようになる．

$$H^{-1} = \begin{pmatrix} \frac{1}{h_1} & 0 & 0 \\ 0 & \frac{1}{h_2} & 0 \\ 0 & 0 & \frac{1}{h_3} \end{pmatrix} - C \begin{pmatrix} \frac{1}{h_1^2} & \frac{1}{h_1 h_2} & \frac{1}{h_1 h_3} \\ \frac{1}{h_2 h_1} & \frac{1}{h_2^2} & \frac{1}{h_2 h_3} \\ \frac{1}{h_3 h_1} & \frac{1}{h_3 h_2} & \frac{1}{h_3^2} \end{pmatrix} \tag{33}$$

以上のように，ELBOのヘッセ行列の逆行列計算の計算量は$O(K)$のオーダーになり，通常の逆行列計算の$O(K^3)$とくらべて圧倒的に少なくなり，高速な計算を可能にしている．

3.4.5 反復と収束

topicmodelsのVEMは，α_0最適化オプションをTrueにした場合，E-step，M-step，Newton-Raphson法による$alpha_0$最適化を1イテレーションとして，ELBO最大化が収束するまで反復計算する．E-step，M-stepおよびα_0最適化のどのステップでもELBOを増大させているため，反復を繰り返すとELBOを最大化できる．ただし，ELBOを構成するパラメータを2群(EとM)に分けて，各群内での最適化を別々におこなうことを繰り返すVEMは局所解，つまりローカルな極小値に陥りやすいとされている．一方，topicmodelsのVEMは，局所解に陥ったと思われる場合に，再初期化，ステップサイズ調整，収束判定基準の自動調整等で対処することを実装していない．

topicmodelsのVEMは，E-stepの収束基準(var_tol)，EM-stepの収束基準(em_tol)，ELBO最大化の収束基準(tolerance)の3つの収束基準を持っている(Gruen and Hornik 2011)．カッコ内はオプション名で，デフォルト値はそれぞれ10^{-6}，10^{-4}，10^{-4}である．topicmodelsはEM-Step収束条件が満たされれば計算を終了するため，tolerance設定の調整は局所解対策にはならない．局所解に対する絶対的な対策は一般的に存在しないが，scikit-learn等のパッケージが採用する全パラメータの1次，2次導関数によって最適化をおこなう勾配下降法とくらべると，topicmodelsのVEMが局所解に停留する可能性は相対的に大きい．勾配下降法は，パラメータを2群に分けることなく，パラメータ全体をみて最適化点を探す．勾配下降法等のアルゴリズムでも局所解に陥る可能性を完全に排除することはできない一方，勾配下降法では計算量が増える．topicmodelsのVEMがscikit-learn等と比べて絶対的に劣るということではない．しかし，topicmodelsのVEMは局所解に陥りやすいという点には注意する必要がある(Asuncion et al. 2009; Hanker, Kasri and Beni-Hssane 2025).

4. MPGのトピック分析トピック分析

ここでは，専門分野の相対的に少ない文書データを分析することを前提として，RのLDA用パッケージでMPGのトピック分析をおこなう．トピック分析の目的は，代表的LDA用パッケージの処理速度と出力結果特性の比較であり，ロシア金融経済の分析ではない．パッケージ，データ前処理，アルゴリズム，Dirichletパラメータ設定はTable 3のとおりである．

R用パッケージを使う理由は，本研究ノートはRのqmdとして執筆しているため，Rでの一貫処理が都合が良いというだけで，トピック分析上の理由はない．実際，以下では，Stanzaではpythonを，MalletではjavaをRから使っている．Rからpython用のgensimやscikit-learnパッケージを使うことに絶対的な困難があるわけではなく，pythonそのものの上で動かすことは難しくない．topicmodelsのVEM以外のVB法を用いるパッケージを使っていない理由は，それらのパッケージは基本的にオンライン更新，大量データの高速処理を目的としているためである．本研究ノートは，オンライン更新を必要としない相対的に少量の文書データのトピック分析を目的としているため，目的がやや異なるVB法のパッケージは割愛した．topicmodelsのVEMは同じtopicmodelsのGibbs(CGS)と比較が容易であるために試行した．トピック分析の目的，対象は多様であるので，目的に合わせて必要なパッケージとアルゴリズムを選択する必要がある(Asuncion et al. 2009; Blei 2012).

分析対象の文書データは，tidytextによる通常処理データ(研究ノート第1部参照)とStanza処理デー

タで分析の両方を分析する．stop words処理，語形統一，固有名詞処理の問題および多言語間のテキスト分析結果との比較可能性を考えると，実際にトピック分析をおこなう場合はStanza処理データを使うことになるだろう．ここでは，通常処理データ(Tidy)とStanza処理データ(Stanza)がトピック分析の結果に与える影響をみることが目的である．

Table 3. 利用するパッケージとその設定

Package	Method	Data	α	β	
Topicmodels	CGS	Tidy (OHV)	Sym-Fix	Sym-Fix	(XX)
Topicmodels	VEM	Tidy (OHV)	Sym-Fix	-	(XX)
Topicmodels	VEM	Stanza (STZ)	Sym-Fix	-	(XX)
Topicmodels	VEM	Stanza(STZ)	α_0	-	(0X)
lda	Fast CGS	Tidy (OHV)	Sym-Fix	Sym-Fix	(XX)
lda	Fast CGS	Stanza (STZ)	Sym-Fix	Sym-Fix	(XX)
mallet	Sparse LDA	Stanza (STZ)	Sym-Fix	Sym-Fix	(XX)
mallet	Sparse LDA	Stanza (STZ)	α_i	β_0	(E0)

Notes: Data: Tidy: 通常処理データ, Stanza: Stanza処理データ, ()内はデータ処理略号．α, β: Sym-Fix: 対称固定，α_0: 文書別トピック分布側Dirichletパラメータの集中度パラメータのみ最適化, α_i: 文書別トピック分布側Dirichletパラメータの文書別最適化, β_0: トピック別単語種分布側Dirichletパラメータの集中度パラメータのみ最適化, -: not applicable; ()内はDirichletパラメータの設定略号.

4.1 データ形式の変換

第4章で使うパッケージは，それぞれ入力用のデータの形式が異なるため，ここで簡単に検討する．なお，tidytext通常処理後のデータはMPG16年分で総トークン数209,320，総単語種数6,233，Stanza処理後のデータは総トークン数201,385，総単語種数5,072である．Stanza処理データの単語数，単語種数は，通常処理データのそれぞれの約95%と約80%になっている．

4.1.1 topicmodels用およびLDA用データ

Rのtopicmodelsパッケージは，Document Term Matrix (DTM)形式のデータを入力データとして使う．DTMの主たる内容は各文書が含む単語の出現数を文書D行×単語W列形式で並べた行列である．topicmodelsは，Rのオブジェクト指向拡張パッケージであるS3オブジェクト(simple-triplex-matrix形式)としてのDTM，S4オブジェクトとしてのDTM(dgCMatrix 形式)のどちらにも対応しているGruen and Hornik 2011)．tidytext::cast_DTM()関数によってデータフレームからdgCMatrix 形式のDTMを作成できる． DTMのデフォルトの列名がissueとtermであるので，qmdファイル内では本研究ノートで作成したtidy_ohv_tblの対応する列名issue，wordをデフォルト名に付け替える．

Stanza処理データについては，テキスト分析の対象とならないような単語を排除するstop words処理をおこなわない．その代わり，Stanza処理データをLDAでトピック分析する場合の標準的方法に従って，内容語のみを分析対象とする．内容語は，upos属性がNOUN, PROPN, VERB, ADJである単語，つまり名詞，代名詞，動詞，形容詞に分類された単語である．これによって，一般的なstop words処理とほぼ同等のstop wordsが排除される．

4.1.2 ldaパッケージ用データ

ldaパッケージのlda()関数は，高速化のため，整数ベクトル化した0-based word IDのリスト形式のデータしか受け取らない(Chang 2025)．このデータ形式は，単語を0から始まる整数の単語IDで置き換え，1つの文書(各年MPG)を次のような単語IDとそれに対応する出現回数という2行行列で表現する．

```
      [ , 1] [ , 2] [ , 3] …
[1, ] 2630   3431   1759   …  単語ID (0-based)
[2, ] 1      1      3      …  出現回数
```

これを文書数まとめたリストが全体データ(ldaではdocumentsと呼ぶ)となる．

単語IDは正確には単語種IDで，0から始まる整数IDである．Rでは配列indexを通常は1から始めるが，ldaパッケージでは，C言語で高速に計算をおこなうため，C言語の仕様に合わせて0から始める．まず，DTM形式データの単語を単語IDに変換した単語IDリストを作成する．続いて，単語IDリストから単語種別に集計し，文書単位で1つの単語IDリストとし，この単語IDリストをリストにした文書リスト(documents)を作成する．ldaパッケージはdocumentsしか分析に使わないため，各文書の文書名情報(例えばtidy_ohv_tblのissue列)のような追加情報は別に保存する．documentsはもとのDTM形式データからの文書(d)の読み込み順をそのまま保持してリスト化している．ldaパッケージの出力結果document_sumsも，読み込み順の情報を維持している．文書名等の情報を回復するとすれば，document_sumsから取り出した列をtibble化し，そのtibbleにあらためて文書名情報を付加するという方法になる．DTM形式データのdocumentsへの変換過程，document_sumsからのデータ抽出過程において，文書の読み込み順，取り出し順が想定通りになっているか注意する必要がある．本研究ノートの分析では，MPGの文書データを1文書1ファイルにする際に，読み込み順序が自然にソートされるようにファイル名の命名規則を統一している．

ldaパッケージは高速なGibbsサンプラーの提供のみを目的としているため，DTM形式のデータを0-base word id形式データに変換するための補助関数のようなものを用意していない．DTMから0-base word id形式データへの変換にはtopicmodelsパッケージのdtm2ldaformat()関数が使えるが，dtm2ldaformat()関数は単語IDをR用の1-basedで出力することに注意する必要がある．dtm2ldaformat()関数を使っても，結局，1-basedから0-basedへの書き換えが必要になるため，本研究ノートではカスタム変換関数を作成して使っている(Appendixのqmdファイル参照).

4.1.3 Mallet用データ

本研究ノートでは，RのMalletパッケージを使い，Malletパッケージ内にあるJAVAでMallet(以下，Rmalletと表記する)を動かす．MalletはLDAだけでなく多様な自然言語処理用の機能を備えているが，Rmalletではその機能を完全に使うことはできない(Magnusson 2026; McCallum 2002)．RのMalletパッケージを使ってR経由でJAVA上のMallet(以下，Jmalletと表記する)を動かすことも可能で，この場合はMalletの全機能を利用できる．ただし，Rmalletでも本研究ノートのLDAトピックモデル分析に利用する範囲では特に支障はない．

Malletは，基本的には，doc_id(あるいはname)，class(あるいはlabel)，tokenの3つの要素をデータとして要求する．doc_idは，1つの文書を特定するためのidである．classは，通常は，学習用の正解ラ

ベルを格納する．例えば，文書のトピック別分類であれば，politics, culture, sports等の正解分類が格納される．LDAトピック分析のように正解がない場合は，何も入力しないままでよい．tokenは名前のとおりtokenであり，単語種ではなく単語(token)を格納する．tokenの格納形式は，1つの文書中の全tokenをスペースで区切って格納するpipe形式と各1行のtoken列に1つのtokenのみを格納するtidy形式の両方が使える．pipe形式の場合は，1文書が1行になるため，高速な読み込みが可能になる．tidy形式の場合，1文書のtoken数の行数を使って1つの文書データになる．

Jmalletの場合は，pipe形式データをTSVファイルでJAVAに渡す必要がある．Rmalletの場合は，Rからmallet.import()関数を使って，tidy形式doc_idとtidy形式tokenをそれぞれlistに変換し，listをRmalletに渡すことができる．Rmalletは新規学習機能を使えないためclassデータは不要であり，classデータは受け取らない．qmdファイルのコードでは，df_mpg_stz2dtmのissueをmal_docid_lstリスト，wordをmal_token_lstとして出力し，これらのlistをmallet.import()関数でRmalletに渡している．なお，qmdファイルのコードでは，リスト作成のためにデータフレームdf_mpg_malを作成しているが，これはclassの作成手順を示すためである．df_mpg_stz2dtmからlistを直接作成すればよい．

Rmallet側では，mallet_instances()関数によってRから引き渡されたデータ(mal_docid_lst, mal_token_lst)からInstanceを作成する．Rmalletの分析対象はInstanceオブジェクトであり，Rから引き渡したリストそのものではない点に注意する．mallet_instances()関数でInstanceを作成する際に，stoplistオプションで，stop wordsを格納したファイルを指定できる．qmdファイル中でコメントアウトしている”stoplists/en.txt”はMalletが標準で持つ英語用stop wordsリストである．stop wordsオプションそのものを削除しても，デフォルトのstoplists/en.txtでstop words処理をするため，ここでは他の分析結果との比較のためstop wordsオプションにcharacter(0)を明示的に指定してmallet_instances()関数のstop words機能をオフにしている．なお，stop wordsオプションはデフォルトでstop wordsファイルを探す．Rmalletの場合は，stop words=NULLとすると，ファイルがみつからずにエラーとなる場合がある．中身のないstop wordsファイルを意味するcharacter(0)を指定する方が確実である．なお，stoplists/en.txt以外のstop wordファイルあるいはカスタムstop wordsを指定することもできる．stop wordsファイルのフォーマットは，1行にstop wordを1つ記入ずつ記入したUTF-8テキストファイルであればよい．

mallet_instances()関数はトークン化機能ももっている．mallet_instances()関数のtoken.regexpオプションで指定した正規表現を使ったrule-basedなトークン化処理である．英語については，デフォルトの正規表現をMalletが持っている．処理が単純で高速であり，英語の場合はrule-basedなトークン化で十分であることも多いため，mallet_instances()関数のtoken.regexpオプションが使われることも多い．本研究ノートではStanzaでトークン化されたデータを使うので，token化オプションは設定はしていない．英語以外の多くの言語では，token.regexpオプションで指定するのではなく，データ前処理段階で処理した方がよい．

4.2 LDAモデル推定

ここでは各パッケージを実際に使う際の注意点と問題点をまとめて検討する．

4.2.1 トピック設定とトピックの自然言語的意味

トピック設定と抽出されたトピックの自然言語的意味の問題は，Bag of wordsを仮定するトピック

分析のもっとも重要な問題である(Asuncion et al. 2009; Blei 2012; Chang et al. 2009; Hemmatian et al. 2019; Laureate, Buntine and Ling 2023)．ここでは，実際の推定におけるトピックの設定と解釈について注意すべき点だけを検討しておく．

第1に，トピック数Kは，先験的に決めるか，LDA分析結果をみて試行錯誤的に決めるしかない．トピックが不明な状態でトピック分析を開始する以上，トピック数Kの決定が困難であることはあきらかである．自然言語的には，「文の数だけトピックがある」あるいは「全文書でトピックは1つ」とすることが誤りだとはいえない．2024年の為替レートについての文と2025年の為替レートについての文書を，為替レートという1つのトピックとみることができるかもしれないし，2024年についてと2025年についての2つのトピックだとみることもできる．すべてのMPG文書は「ロシア中央銀行の通貨金融政策」という1つのトピックを扱っていると考えることも奇妙ではない．以下のMPGの分析はトピック数を4としておこなう．トピック数4は研究ノート第1部のワードネットワーク分析が実物経済・成長・資金供給，インフレーション・物価，金融システム・金融監督，金利・為替にかかわるとみられる4つの単語クラスターを同定したことに依拠している．ただし，この分析結果はbigram tokenのカウントベースの分析によるものであり，強い根拠ではない．トピック数4は一般的なトピック分析と比べるとかなり少ない．しかし，MPGのデータ量および基本的に通貨金融政策にかかわる定型的内容を扱うというMPGの性格から，まったく的外れな数ということはないだろう．

トピック数4の妥当性はMPGの経済的内容にかかわる問題であるため，本研究ノートではこれ以上の検討はしない．ここで問題とすべきことは，トピック数をトピック分析とは別におこなった分析で決めていることである．単語(bigram)カウントベースの分析あるいはNNモデルによる文章分類からトピック数を設定し，それにしたがってトピック分析をおこなうという分析フロー自体は選択の1つである．その場合は，それらの分析の妥当性を検討しなければならない．現状では，トピック数を決めるための決定的方法はない(Chang et al. 2009, Hanker, Kasri and Beni-Hssane 2025, Hemmatian et al. 2019)．繰り返し指摘しているように，トピックが何かがわかっているならば，トピック分析をおこなう必要はほとんどない．実際的な方法としては，トピック数を変えてLDAモデルを推定し，モデル全体のあてはまりの度合いを比べる方法がある．

トピック分析においてモデル全体のあてはまりの指標としては，単語あたりの平均尤度であるPerplexityがよくつかわれる．

$$\text{Perplexity}(\boldsymbol{w}) = exp\left(-\frac{1}{N_W}\sum_{i=1}^{D}\sum_{j=1}^{N_{w,d}}\log p\left(w_{d,i}\right)\right) \tag{34}$$

ここで$p(\cdot)$はLDAモデルの事後分布である．Perplexityが小さいほど観測データWのモデル上の出現確率が高くなるため，モデルとして良いことになる．ただし，Perplexityは，データ量(単語種数)，データ前処理，実装，アルゴリズムにより偏りが出るため，これらの分析の枠組み自体が異なっている場合は，モデル間の比較は困難である．以下のTable 4は参考情報としてのみ各試行結果のPerplexityを報告している．Perplexityの相違の検討はMPGの経済的内容と密接にかかわるため，本研究ノートではおこなわない．Perplexityの最大の問題は，Perplexityが単語の出現確率をみていることからわかる通り，Perplexityはトピックの自然言語的な意味の判定とは直接には関係しないことである．同じ

分析フレームワークの下で，例えば，Dirichletパラメータの設定が違うモデルのPerplexityの比較であれば，Perplexityが低い方が良いモデルだろうといえるだけである．

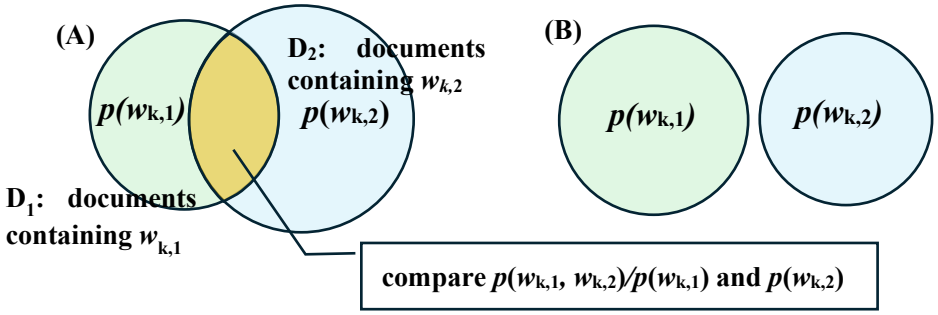
第2に，抽出されたトピックがどれだけ自然言語的意味のあるトピックであるかを評価する決定的方法は存在しない．トピックの質を検討する方法としては，Topic Coherenceがよく使われる．Topic Coherenceの基本的アイディアは，トピックの中で使われている単語が同じ文書あるいは文書のセグメントの中でともに出現する確率が高ければ，そのトピックは意味のあるトピックである可能性が高いというアイディアである．つまり，「石油」「制裁」「権威主義」を出現頻度上位語として持つトピックが抽出されたとき，多くの文書で「石油」「制裁」「権威主義」が同時に出現するようであれば，このトピックはまとまった意味を持つトピック(おそらくロシアについてのトピック)であろうということになる．逆に，分析対象以外，「石油」「制裁」「権威主義」が同時に出現するような文書がなければ(Figure 8 (B))，このトピックは単なる偶然で抽出されたトピックである可能性が高い．なお，ここでLDAモデルでは，トピック別単語種分布は文書全体Dについての分布であるから，トピックの出現頻度上位語が同時に出現する文書がゼロあるいはゼロに近いことはある点に注意する．Figure 8 (B)の状態が稀な状態というわけではない．

Topic Coherenceの基本的な計算方法は，あるトピックkの出現確率上位M個の$(w_{k,1}, w_{k,2}, \cdots, w_{k,M})$のすべてのペアについて，1つのペアの2つの単語があらわれた文書数(文書セグメント数)とそれらの総数の比で単語の出現確率とする．RのtextmineRパッケージ，tidyldaパッケージの場合であれば，kトピックのTopic Coherence (TC_k)は，

$$TC_k = \frac{1}{m}\sum_{m \in M}\left[p\left(w_{k,m} \mid w_{k,m'}\right) - p\left(w_{k,m}\right)\right] \tag{35}$$

である．ここで，mはM中のm以外の1つの単語である．(34)式の基本的アイディアは，Figure 8のオレンジ色部分(両方の単語がともにあらわれた文書数)と水色部分(w_2があらわれた文書数)を比べることにある．共起関係をみる文書として，分析対象文書，外部コーパス，あるいはそのミックスを使うかどうかや，具体的計算方法によってTopic Coherenceには多くのバリエーションがある (Roeder, Both and Hinneburg 2015; Stevens et al. 2012).

Figure 8. Topic Coherence



Topic Coherenceは，トピックの質について一定の情報を与えてくれるものの，あるトピックに自

然言語的意味があるかという問いに直接答えるものではない．Topic Coherenceとトピックの自然言語的意味の間にどのような関係があるかは知られていない．加えて，Topic Coherenceの計算方法からあきらかなように，Topic Coherenceの大小は，単語種数，単語数，文書数，共起計算をするcorpusを何にするかという要因で変わるため，異なる文書(corpus)間のTopic Coherenceの比較に意味を与えることは困難である．Topic CoherenceにはNNモデルベースの指標を使う場合もある．その場合も教師なし学習(分類)であるから，最終的な結果が正しいかは結局のところ分析者が判断することになり，共起確率ベースのTopic Coherenceと本質的に違うとはいえない．教師ありで学習した場合は，トピックに自然言語的意味があるかという問題にある程度答えられる可能性がある．しかし，教師あり学習でNNモデルを訓練するということは，自然言語的に意味のあるトピックがどのようなものなのかを事前に正解として与えることを意味する．そのような正解セットを作ることは，通常，困難であろう．さらに，例えば，MPGについて正解セットを作るのであれば，もはやMPGに対してトピック分析をおこなう必要はない．数多くの中央銀行の政策文書について自然言語的内容のあるトピックの正解例をつくれば，MPGの自然言語的意味のあるトピックの正解例として使えるかもしれない．しかし，ロシアの特殊な状況を考えると，その可能性は低いように思われる．

topicmodelsのGibbs以外ではTopic Coherenceの計算は容易であるが，以下のMPG分析ではTopic Coherenceを報告していない．MPGでは，MPG文書全体を共起関係計算用の文書(corpus)とすると， 4トピックのTopic Coherenceはすべてほぼゼロでほとんど差が出ない．MPGが単語種6,000以上で文書数が16しかないため，単語間の共起は必然的に低くなる．外部corpusを共起計算の基準とすればこの問題は避けられるが，外部corpusとして何を使うかでTopic Coherenceの値は変わる．外部corpusとして何を使うかは，MPGの経済的内容にかかわる問題となる．文書数が少ないことが予想される専門文献のトピック分析においてTopic Coherenceをどのように扱うかは今後検討すべき問題である．

トピックにかかわる第3の問題は，トピック分析の結果出力にかかわる．Bag of wordsを仮定するLDAモデルは，分析対象文書中の単語の順番も個々の文書の順番も考慮しない．トピック別単語種分布，文書別トピック分布は，どちらも文書の順番にも単語の順番にも関係なく決まる．これらは，LDAモデルのinterchangeabilityとよばれる特徴であり(Blei, Ng and Jordan 2003)，LDAモデルの統計処理の妥当性の基礎になっている．

LDAモデルではトピックを識別するための情報は確率分布の違いのみである．例えば，文書がd_1とd_2で，トピックがk_1とk_2の2つラベルで表示されるとき，文書別トピック分布がd_1は(k_1=0.1，k_2=0.9)，d_2は(k_1=0.6，d_2=0.4)という結果と，文書別トピック分布がd_1は(k_1=0.9, k_2=0.1)，d_2が(k_1=0.4, d_2=0.6)という結果とは，LDAモデル内では区別できない．LDAモデルがつけたk_1，k_2というトピックのラベルにはトピックの自然言語的意味とは関係が無い．LDAモデルが区別するトピックの違いとは，トピックを構成する単語の分布の違いだけである．

特に，Dirichletパラメータを対称固定にしたCGSサンプリングの場合，トピックが区別できなくなることはほぼ避けられない．直感的には，すべてのα_iに同じ値を設定しているのだから，k番目のトピックをどのα_iに割り当てて計算をスタートしても違いはでない．加えて，CGSのサンプリング対象は多項分布パラメータを積分消去した後の「interchangeableな単語」(単語位置)へのトピック割当であるため，特定のトピックと結びつく$\theta_{d,k}$，$\eta_{k,v}$パラメータ自体が存在せず，したがって特定のトピックの情報を保持する必要も，場所もない．単語(位置)に確率的にkを割り当てて，初めて単語(位置)のトピックk*が決まるのであって，単語(位置)がどれかのトピックk*と先に結びついていること

はない．対称固定Dirichletパラメータならば，ある1つの単語(位置)は，初期状態ではすべてのトピックと対称に結びついているため，対称固定Dirichletパラメータの情報だけではトピックの識別はできない．トピックを識別する情報は，はk=1とは異なるk=2を計算するといった計算の便宜につけたラベルの意味しかない．結果として，あるk=1というIDのついた確率分布のパターンA，k=2というIDのついた確率分布のパターンBが違う，あるいは同じということが便宜的にわかる．ここで，パターンAをk=2とし，パターンBをk=1としても，呼び名が変わるだけで内容(=確率分布)は変わらない．LDAでは，トピックの内容とは確率分布であって，自然言語的意味ではない．

CGSの場合は以上の問題に加えて，アルゴリズムそのものによってトピックの区別は基本的にできなくなる．CGSでは確率分布のすべての場所からサンプリングすることが基本であるため，計算途中で，k_1に分類していた全単語がk_2に移動し，k_2に分類していた全単語がk_1に移動する可能性がある．単語の移動というよりトピックのラベルの付け替えである．ラベルを付け替えても確率分布は変わらず，一方，トピックのラベルはトピックの自然言語的意味のような固定的な意味を持っていない． Dirichletパラメータを非対称としてパラメータ間に大きな違いを設定する場合やVEMを使う場合は，前者では非対称Dirichletパラメータの添え字が特定の確率分布と結びつくことでトピックを識別できる可能性が高まること，後者では最適化であるため計算の初期状態を同一に保てば同じ結果になる可能性が高いことから，CGSよりは同一のトピック識別IDが特定の確率分布と結びつく可能性が高まる．ただし，LDAモデルが，Bag of words仮定のもとで，単語，文書の順序情報に依拠しない以上，非対称DirichletパラメータでもVEMでもトピック識別IDとトピックの確率分布パターンとの組合せが計算ごとに変わる可能性を完全に排除することはできない．完全に防ぐとすれば，3.1.2の多項分布パラメータを消去しないベイズ確率モデルを使うか，計算過程において何らかの方法でトピックID情報を保持する工夫をすることになる．前者は計算時間の問題から実用的とは言い難い．後者は基本的に計算過程のログをとっておき，ラバルの付け替えを追跡することになる．特定のトピックにマーカを付ける(キーワードを指定する)という対処も考えられるが，この方法は，非対称Dirichletパラメータを設定する対処方法と同様，トピックについて何らかの事前情報がありキーワードが付けられることを前提にしている．

トピックの自然言語的意味を考慮しないことは，LDAモデルの欠点でもあり，利点でもある．LDAモデルは文書を単語の分布とみることで，トピックの自然言語的意味の情報源(単語間の関係)を捨ててしまうが，他方で，それによって文書別トピック構造，トピック別単語種構造を確率分布の違いとして定量的に示すことができる．実際，分析対象文書がどのような自然言語的意味のあるトピックをもつのかわからないことを前提とするなら，計算上のトピック識別IDが一定のトピックと結びつかないという問題はさほど重要な問題ではない．「一定のトピック」とは何のことかがそもそも不明だからである．ここでの実際的な問題は，トピック識別IDと「一定の自然言語的意味のトリングを使うtopicmodels，lda，Malletのいずれのパッケージでも，サンプリング結果はサンプル平均ではなくサンプリングの最後の1回の結果しか使わない．Malletではサンプリング平均をとることが可能であるが，デフォルトは最終1回のサンプリング結果を出力する．トピック分析では経験的ピック」の関係の問題ではなく，CGSの計算結果をどのように扱うかという問題である．3.3.3でみたように，通常のMCMCサンプリングでは，サンプリングを何回かおこないそのサンプル平均(期待値の推定値)を推定結果とする．一方，トピック識別IDとトピック内容（自然言語的あるいは特定の確率分布)との対応が一定しないCGSでは，サンプリング平均をとると異なる内容のトピックの平均をとる可能性が高い．したがって，以下のGibbsサンプにンプリングの最後の1回の結果だけで十分安定

的な結果が得られるとされているが(Asuncion et al. 2009)，通常のMCMCサンプリングとは大きく異なる点であるので注意が必要である．

4.2.2 topicmodelsパッケージによる分析

topicmodels::LDA()関数の必須の引数は，DTM形式のデータの指定(第1引数)とトピックの数(k)のみである．method引数は，VEMでVEM，GibbsでCGSを選択する．topicmodelsはCGSを実験用と位置づけているため，method引数を指定しなければデフォルトのVEMで計算する．control引数で計算過程を制御するためのパラメータを与える．なお，topicmodelsパッケージでは，文書別トピック分布側Dirichletパラメータ名はalpha，トピック別単語種分布側Dirichletパラメータ名はdeltaである．トピック別単語種分布の多項分布パラメータの名称にはbetaを使っている．topicmodelsパッケージの説明においては，そのままbetaをトピック別単語種分布の多項分布パラメータを指すものとして使う．本研究ノートではトピック別単語種分布側Dirichletパラメータをβとしているので注意していただきたい．

topicmodelsのGibbsでは，Dirichletパラメータは対称固定である．VEMでは，estimate.alphaオプションをTRUEにすれば，文書別トピック分布側のDirichletパラメータの集中度パラメータ(α_0)を推定する．3.4でみたとおり，VEMはトピック別単語種分布の多項分布パラメータを直接更新するため，トピック別単語種分布側Dirichletパラメータの推定はしない．Dirichletパラメータの初期値は，alphaオプション，deltaオプションで指定できる．他のパッケージで推定する場合を含むすべての場合でalpha=50/K，delta=0.01を共通に設定する．なお，Gibbs，VEMともに，estimate.betaオプションを持っているが，このbetaはトピック別単語種多項分布パラメータである．estimate.betaオプションの意味は，計算済みのモデルであらたな文書データを分析する際に，トピック別単語種多項分布パラメータを計算済みのまま固定するか，新たな文書データで更新するかを指定するオプションである．seedオプションは乱数発生方法のオプションである．他のパッケージで推定する場合を含むすべての場合で，再計算の際に結果の比較を可能にするため同一のseedを指定し，同じ乱数列を発生させている．method引数中のseedは乱数列のseedである．ただし，計算環境により計算結果を完全に正確には再現できない可能性がある．

Gibbsのcontrol引数中には，初期値の影響がなくなり事後分布が安定収束するまでのサンプル回数burn-in，安定収束後のサンプル回数iter, iterでサンプルしたうち事後分布のパラメータ計算のために実際に使うサンプルの間隔指定thinを指定するオプションがあり，デフォルトでiter=2000，burn-in=0, thin=2000である．4.2.1でみたとおり，CGSではサンプル平均は使えないため，Gibbsのburn-in，thinオプションは機能しておらず，オプションの指定にかかわらずサンプリングの最後の1回の結果が最終出力となる．一般のMCMCサンプリングにくらべてiterの数が小さいが，LDAモデルのCGSの場合は，burn-inが50回程度，全体で2000回程度のイテレーションで十分であるとされている(Asuncion et al. 2009; Gruen and Hornik 2011)．その理由は，1つは，サンプリングの対象が多項分布パラメータを積分消去した後の式であり，構造が単純化されているためである．2つには，トピック分析の目的は，トピックとみなされる単語のクラスター構造を識別することであり，トピックを厳密に規定することではないからである．4.2.1でみたとおり，文書別トピック分布，トピック別単語種分布が厳密にわかったとしても，それで「自然言語的に何のトピックか」が厳密にわかる可能0性は低い．LDAモデルで何らかのトピックを同定するというより，LDAモデルで抽出された単語のクラスターをトピックと呼ぶ，といった方がよいだろう．ここでは，デフォルトのサンプル回数を使う．その他のパッケージで分析する場合も，イテレーション回数は基本的に同じ設定とする．

topicmodels::LDA()関数は，実行結果をRのオブジェクト指向パッケージS4に従ったオブジェクトとして返す．tidytextパッケージのtidy()関数を使うことで，このS4オブジェクトから標準的な情報を簡単に取り出すことができる．tidy()関数のmatrix引数にgammaを指定すると文書別トピックK項分布パラメータを，betaを指定するとトピック別単語種V項分布パラメータを抽出できる．本文中ではgammaは$\theta_{d,k}$，betaは$\phi_{k,v}$あるが，パッケージの名称を使う．注意すべき点は，3.3.3で検討した通り，LDAモデルでは，topic番号はトピックを識別するが，番号が特定トピックを示すことはない点である．

4.2.3 ldaパッケージによる分析

ldaパッケージのlda.collapsed.gibbs.sampler()関数を使って分析する．ldaパッケージでは，文書別トピック分布用Dirichlet分布パラメータalpha，トピック別単語種分布用Dirichlet分布パラメータetaはデフォルト値が設定されていないので手動で設定する．なお，ldaパッケージではトピック別単語種分布用Dirichlet分布パラメータ(これまでの説明ではβ)をetaとしている．わかりにくいが，パッケージの説明には，各パッケージの名称を使う．

lda.collapsed.gibbs.sampler()関数は，Dirichlet分布パラメータを最適化する機能，非対称固定で手動で与える機能はない．文書別トピック分布側Dirichletパラメータalpha，トピック別単語種分布側Dirichletパラメータetaともにスカラーで与え，その値で対称固定となる(Chang 2025)． alpha，etaの初期値は，本研究ノートでの共通の設定であるalpha=50/K，eta=0.01を使う．

lda.collapsed.gibbs.sampler()関数は，安定収束までのバーンイン回数を指定するburn-in，安定収束後のイテレーション回数を指定するnum.iterationsオプションが用意されている．デフォルトでburn-in=0, num.iterations=2000である．4.2.1での説明のとおり，出力はイテレーションの最後のサンプル結果のみであるので，thinオプションは用意されていない．burn-inとnum.iterationsを指定できるが，安定収束をみる指標等を自動で計算する機能はないため，burn-inとnum.iterationsを分けて設定する意味はあまりない．ただし，各イテレーションの対数尤度を計算し保持する等の収束判定をサポートするためのオプションは用意されている．ここでは，共通設定に合わせて，burn-in=0，num.iterations=2000のままとしている．

lda.collapsed.gibbs.sampler()関数は推定結果を巨大なリストとして返す．このリストは，文書別トピック分布，トピック別単語種分布のパラメータをそのまま格納していない．Appendixのqmdファイル中のコードでは，それぞれのパラメータを計算し，topicmodelsと同形式のデータフレームに格納している．リストからの必要情報の抽出，パラメータ計算処理は単純であるので，qmdファイルを参照していただきたい．

4.2.4 Malletパッケージによる分析

Rmalletでは，R側から推定に必要な情報を一括して引き渡す．引き渡す内容に特に変わった点はないが，整数の取り扱いという形式的な点で注意を要する．JAVAとRは整数の取り扱い方が内部的に異なっているため，Rのmalletパッケージも整数を想定通りに変換してRmallet，Jmalletに渡さない場合がある．特にR側でas.integer()を使った場合に，JAVA側の整数(int)に変換しない場合がある．この場合は，引き渡した整数をMallet側で整数と認識しないためエラーになる．この問題は，RとJAVAの技術的相違の問題であり，LDAやMalletの問題ではない．具体的には，Appendixのqmdファ

イルのコード中の整数引き渡しの例を参照していただきたい.

topic.modelsetOptimizInterval(N)オプションで，Nを0以上の整数に設定すると，文書別トピック分布側Dirichlet分布パラメータ$alpha_i$を推定する．集中度パラメータ$alpha_0$だけを推定する機能はないので，$alpha_i$の推定結果から$alpha_0$を計算する．Nは，3.3.4のとおり，CGSのNイテレーションごとにDirichletパラメータの不動点反復法による最適化をおこなうことを意味している．マニュアルにははっきりと書かれていないが，Nを0以上に設定すると，トピック別単語種類分布側Dirichlet分布パラメータ$beta$も対称のまま最適化をおこなう．$alpha_i$最適化と対称固定$beta$最適化は分離できず，Nを0以上に設定すると両方が推定される．MalletではDirichlet分布パラメータはalpha.sum = 50.0, beta=0.01のようにスカラーで与える．alpha.sumは集中度パラメータ$alpha_0$を意味し，初期値は50である．したがって，$alpha_i$の初期値は$50/K$である．alpha.sum=50/Kとすると，$alpha_i = 50/K^2$を指定することになるので注意が必要である．beta＝0.01は対称固定$beta_i$=0.01の意味で，戻り値も集中度パラメータ$beta_0$ではなく対称固定の$beta_i$の値であるので注意が必要である．ここでは，他のパッケージに合わせて，alpha.sum=50，beta=0.01のデフォルト初期値のままとする．Malletによる分析結果をldaと比較するために，($alpha$対称固定，$beta$対称固定)と($alpha$非対称推定，$beta$対称推定)の2つの場合でStanza処理データをMalletで分析する．

Malletでは，trainオプションでburun-in部分を分けることなくイテレーション回数を指定する．setBurn-inオプションが存在するが，LDAでは機能しない．3.3.4で説明した通り，trainオプションによるイテレーションをburn-inとみなし，estimationオプションによるイテレーションを収束した確率分布からのサンプリングとみなすことができる．ここでは，他のパッケージとの比較のため，trainオプションで2000回イテレーションをおこない，最終イテレーション1回の結果のみで出力とする．

推定結果はMalletのmodelオブジェクトが持っている．RmalletではRからmallet.doc.topics(model)およびmallet.topic.words(model)によって文書別トピック分布パラメータおよびトピック別語彙分布パラメータを，引数modelalphaおよびmodelbetaで，それぞれ文書別トピック分布側とトピック別単語種分布側のDirichletパラメータを抽出する．ただし，列名，表形式がtidytextの標準形式と異なるため，コード中では抽出と同時にtidy形式のデータフレームへ変換をおこなっている．変換の内容は複雑ではないのでqmdファイルのコードを参照していただきたい.

4.3 推定結果の比較

MPGの経済的内容の分析は本研究ノートの目的ではないので，各パッケージの各推定方法における(1)計算機資源要求量，(2)文書別トピック構成の相違，(3)トピック内容(トピック別語種分布)の相違を簡単にみるにとどめる．(3)のトピック内容の検討は，文書内容の分析，つまり形式的なトピック分析の結果と文書の自然言語的意味とを結びつける上で必要不可欠な手順である．その検討方法をすべての推定方法の結果について網羅的に記述することは，紙幅の制約により困難であり，また本研究ノートの目的外である．ここでは，topicmodelsの2つの推定結果でのトピック内容の検討手順をみるにとどめる．

4.3.1 計算機資源要求量

Table 4は，トピック分析の中核部分の処理の開始から終了までのSystem timeで測った処理時間である．topicmodelsのGibbsとVEMでは，やはり最適化計算のみのVEMがGibbsより速いが，極端な違

いがあるわけではない．これは良く知られた特徴である(Gruen and Hornik 2011)．GibbsのCGSは1イテレーションごとにサンプリングするというものの，大部分の処理はサンプリングではなく更新式の計算である．一方，VEMは最適化計算は高速におこなえるが，1イテレーションごとに次の計算点を計算して求める．次の計算点への移動は，逆行列計算を使わずに更新式で計算できるが，1イテレーションごとに計算することは必要である．結局，GibbsとVEMの間で計算時間にあまり大きな違いが出ることはない．MPGの場合は，トピック数を4として設定しているため，トピック数の少なさがCGS処理の高速化に貢献している可能性も高い．Malletは処理の高速さで知られているが(McCallum 2002)，今回の結果では，30秒以上ともっとも遅い．ただし，MalletでDirichletパラメータ固定と最適化との場合で処理速度にほぼ差が無いことから，Malletそのものが遅いのではなく，R経由でRmalletを動かしていることが原因であるかもしれない．Dirichletパラメータ最適化は，おおざっぱには，固定の場合の計算量×パラメータ最適化のためのイテレーション数で計算量を増やすため，通常は，計算時間も相当程度増大する．topicmodelsのVEMでα_0最適化だけで固定の場合の約1.6倍の計算時間になっている．α_iとβ_0の最適化をおこなっていることをかんがえると，Malletは高速であるといえるかもしれない．

Table 4はPerplexityも示しているが，既に述べたように，本研究ノートではPerplexityの値は参考情報以上のものではない．Perplexityに適切な値というものはもともと無いが，パッケージごとのPerplexityの相違は，おおむね経験的な値と一致している(Asuncion et al. 2009; Newman et al. 2007; Stevers 2007; Wallach, Mimno and McCallum 2009)．MalletのPerplexityが低い(あてはまりが良い)が，MalletではPerplexityの値がこの程度になることは良く知られている．

Table 4には示していないが，推定中のRプロセスのメモリ利用量(Resident Set Size: RSS)も計測している．これもtopicmodelsのVEMでDirichletパラメータ対称固定の場合が最小で1MB程度，MalletのDirichletパラメータ推定の場合が35MB程度である．MPG程度のデータ量では，現在のPCの一般的メモリ容量からみれば，35MB程度でも問題にならないであろう．RSSはOSのアプリケーションへのメモリ割当量であり，各パッケージが利用する物理メモリ量を正確に反映するわけではないうえに，実際の利用環境に依存するのでTable 4に掲載していない．

Table 4. 計算時間

Packages	α	β	Data	Time (Sec.)	Perplexity
textmodels_Gibbs	Sym-Fix	Sym-Fix	OHV	11.9	2,839.0
textmodels_Gibbs	Sym-Fix	Sym-Fix	STZ	16.9	2,370.2
textmodels_VEM	Sym-Fix	Sym-Fix	OHV	9.4	3,912.4
textmodels_VEM	Sym-Fix	Sym-Fix	STZ	12.2	2,805.7
textmodels_VEM	α_0	Sym-Fix	STZ	19.3	4,393.4
lda	Sym-Fix	Sym-Fix	OHV	2.4	2,538.7
lda	Sym-Fix	Sym-Fix	STZ	10.8	2,023.5
Mallet	Sym-Fix	Sym-Fix	STZ	32.6	647.2
Mallet	α_i	β	STZ	34.3	638.8

***Note**:* See Table 3 for the notation.
***Source**:* Authors' measurements.

StanzaによるUD化対応データ処理が数GBのメモリと70分以上の処理時間を必要とすることを考えると，今回検証した各種パッケージでのトピック分析処理時間，メモリ使用量は問題にならないといえる．計算資源量について対応すべきであるとすれば，Stanzaによるデータ前処理になる．ただし，データ量とトピック数の増大によりトピック分析の処理時間は加速的に増える可能性があることは

注意しなければならない.

4.3.2 トピック構造の同定: 文書別トピック分布

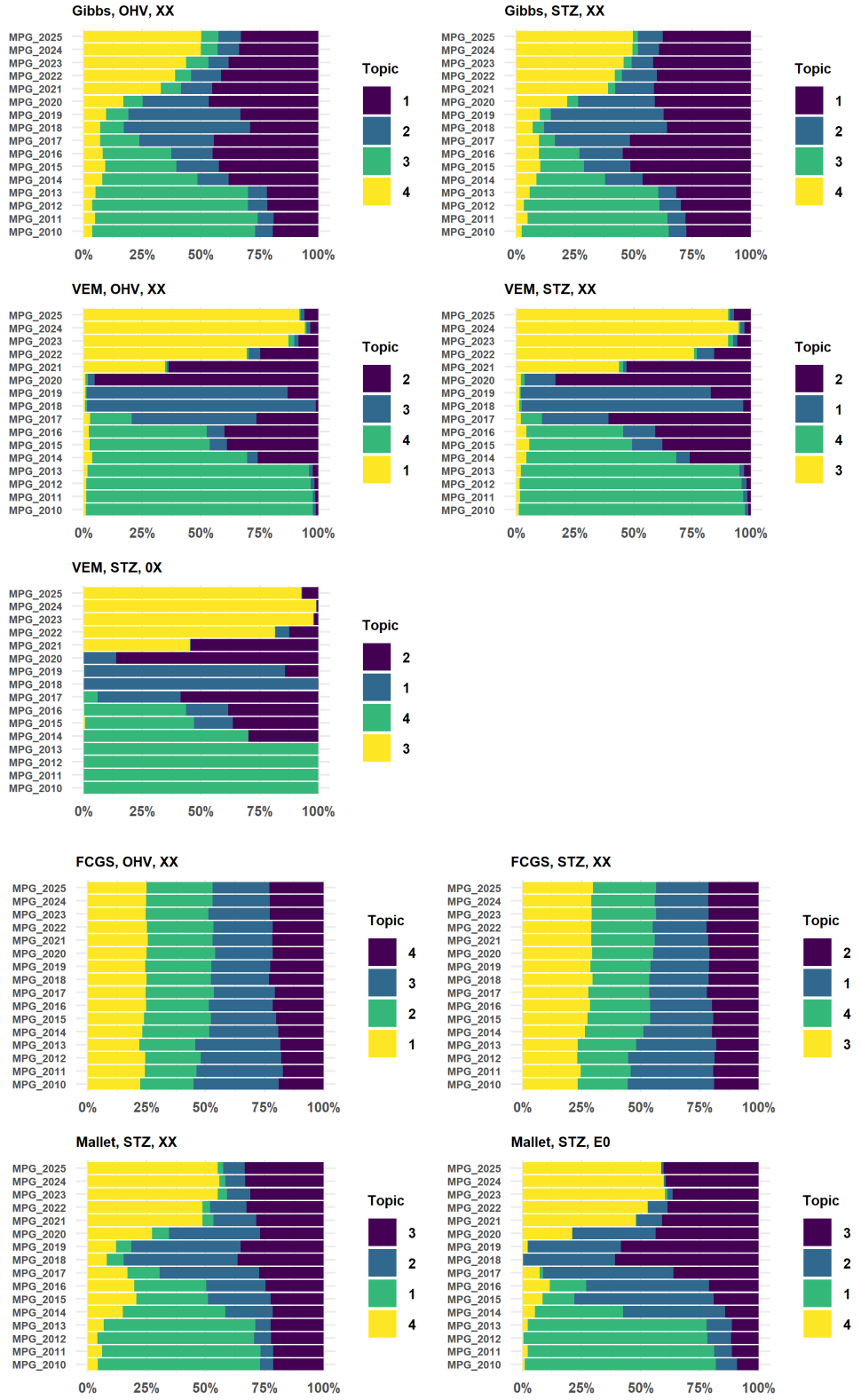
ここでは，各推定方式による文書別トピック分布の推定結果をグラフィカルにのみ表示する．4.2.1で検討したとおり，各推定結果のトピック識別ID番号(プロットの凡例の番号)は識別ラベル以上の意味はない．同じ識別番号でも，それが指しているトピックの確率分布あるいは自然言語的意味がプロット間で共通であることは偶然以外にはない.

文書別トピック分布の推定結果を識別番号だけで表示することも，自然言語的内容をある程度考慮して表示することも，どちらも誤解を招きやすい．ここでは，単に比較の便宜という観点から，各トピックの自然言語的内容をある程度考慮した図としている．つまり各トピックの色(モノクロの場合，グレートーン)は各プロット間で自然言語的意味が近いと思われるトピックに同じ色を割り当てている．各トピックの自然言語的内容の識別は，岡山・中村 (2025)がword network分析で同定した4つの単語クラスターと次の4.3.3の分析結果に依拠している．逆に，Figure 7(1)，7(2)の凡例のトピック識別番号1-4は自然言語的に同一のトピックを示していない点に注意されたい．岡山・中村 (2025)は通常処理データについてしか分析していないため，Stanza処理データが抽出したトピックの自然言語的意味は通常処理データの検討結果と照合することで同定している.

Figure 9は，ldaパッケージのFast CGS (FCGS)以外では，文書別トピック分布の時間変遷パターンは共通していることを示している．2010年に支配的だったトピックが2025年にはほぼ目立たなくなる一方，2025年に支配的なトピックは2010年にはほとんど目立たないトピックだったというパターンである．濃いグレーと黒のトピックは，CGSを使うパッケージとVEMとではやや異なるパターンとして捉えられているが，それでも濃いグレーのトピックは2018-2020年頃の比較的短期間支配的なトピックで，それ以外の期間ではあまり目立たないトピックであるというパターンを共通にみることができる．Gibbs，Malletで，文書別トピック分布側，トピック別単語種分布側両方のDirichletパラメータ対称固定(xx)の場合は，文書別トピック分布のパターンはよく似ている．topicmodelsのVEMの3つの結果の間でも，やはり文書別トピック分布の時間変遷パターンはよく似ている.

CGSとVEMの間のパターンの違いは，2018-2020年頃をみるとよくわかる．VEMでは，この時期は濃いグレー，黒のトピックが完全に支配的となり，他のトピックが消えてしまう．特にVEMの文書別トピック分布側Dirichletパラメータの集中度パラメータ最適化の場合(0x)は，2010-13年期についても明るいグレーのトピックが完全に支配している．一般に， VEMは最適化であるため，トピックがシャープになる，つまり少数のトピックだけが選ばれる傾向になるといわれる．Dirichlet分布パラメータの最適化も，同様にトピックのシャープさを引き上げる(Gruen and Hornik 2011)．一方，CGSは事後分布からのサンプリングであるため，あるトピックの出現確率が完全にゼロになることは基本的にない．MPGの場合，トピック数4は通常のトピック分析と比べると非常に少ない設定であるため(Asuncion et al. 2009; Griffith and Steyvers 2004; Phan, Nguyen and Horiguchi 2008)，VEMやDirichletパラメータ最適化はオーバーフィッティングをもたらした可能性が高い．トピックのシャープ化は，通常は，多数のトピックの存在が予想されるときに効果を発揮する．MPGでトピック数4を想定しているときに，あるトピックがある時期に完全に現れなくなるという結果は，自然言語的には解釈が困難な結果といえる．ただし，2018-2020年頃に支配的なトピックはコロナ・パンデミックと結びついている可能性が高く，この限られた時期だけ重要なトピックだった可能性は高い.

Figure 9 Document Topic Distribution



Note: see Table 3 for the notation.
Source: Authors' calculation.

Figure 9からもう一つ明確にわかることは，同じ推定方法であれば，通常処理データ(OHV)を用いても，Stanza処理データ(STZ)を用いても，結果として得られるトピック構造にほとんど違いがみられないことである．他とは異なる文書別トピック分布パターンとなっているldaパッケージのFast CGSでも，通常処理データとStanza処理データではほとんど結果が変わらないという点は同じである．この結果は，Bag of words仮定のもとでトピック構造を同定するLDAの頑健性を示すものとみてよい．一般に，LDAでは分析対象文書に10%程度エラー，つまり，もともとのスペリングミス，読み取りエラー，ノイズ等で自然言語的に意味をなさない単語が入っていてもトピック構造を分析できるとされている(Su et al. 2015, Zosa, 2021)．通常処理しても，Stanza処理しても，同じ文書から同じトピック構造を同定できたことになり，トピック構造の同定という点ではLDAモデルは有効といえる．なお，データのStanza処理は，Stop words除去，同義語の統一，固有名詞排除といった前処理，多言語間での比較可能性，トピックの自然言語的内容の同定の相対的容易さといった点で通常処理データより優れている．トピック構造の分析結果に大きな違いが出ないとしても，Stanza処理データを使った方がよいことは変わらない．

ldaのFast CGSが他の推定方法とは異なる文書別トピック分布パターンを与えた理由ははっきりしない．topicmodelsのGibbsはBlei, Ng and Jordan (2003)のCGSアルゴリズムをほぼ忠実にコード化している一方(Gruen and Hornik 2011)，Fast CGSはCGSアルゴリズムの根幹は変えていないものの，高速化のために多くの調整，独自の処理をおこなっている(Chang 2025)．Fast CGSの分析結果が必ずしも安定的でない，つまり，状況に合わせた調整が必要であることは一般的に知られている．本研究ノートでも，イテレーション回数，Dirichletパラメータの初期設定等を変えた推定もおこなったが，それらの結果も，Figure 9の文書別トピック分布パターンと大きく異なるものではなかった．Fast CGSの結果は，MPGに4つのトピックがほぼ均等に出現しているというより，トピックを4種類に分離することができなかったとみるべきであろう．Perplexityは妥当な値で，自然言語的意味のあるトピックが存在するのか不明である以上，Fast CGSの結果の方がより正しい結果である可能性はある．これらを精査するためには，次節でみるトピック別単語種分布の分析が重要になる．ただし，トピック内容の精査はMPGの経済的内容にかかわるため，本研究ノートではFast CGSについての検討はこれ以上はおこなわない．ldaパッケージのFast CGSのアルゴリズムや実装を分析することが目的でないのであれば，相対的にデータ量が少なく，トピック数も少ないと予想される文書では，Fast CGSの利用は避けるべきかもしれない．

4.3.3 トピックの自然言語的意味の同定: トピック別単語種分布

トピック数が大きい場合の各トピックの自然言語的意味の検討では，まずクラスター分析，コサイン類似度による分類，各種分類器による分類といった方法で抽出されたトピックをさらに分類，グループ化する必要がある場合も多い．トピック数を4に設定したMPGの場合は，4つのトピックをさらにグループ化する必要はないため，まず単純に，トピック別単語種分布から出現確率の高い単語を取り出して比較する．すべての推定結果の上位単語を表示することは紙幅の制約によりできないため，topicmodelsの(Gibbs, OHV, xx)，(VEM, OHV, xx), (Gibbs, STZ, xx)およびMallet (STZ, xx)の4つの結果の上位10単語だけをみる．なお，Figure 10ではtopicmodelsパッケージの用語法にしたがって単語種を指す用語としてtermを使っている．

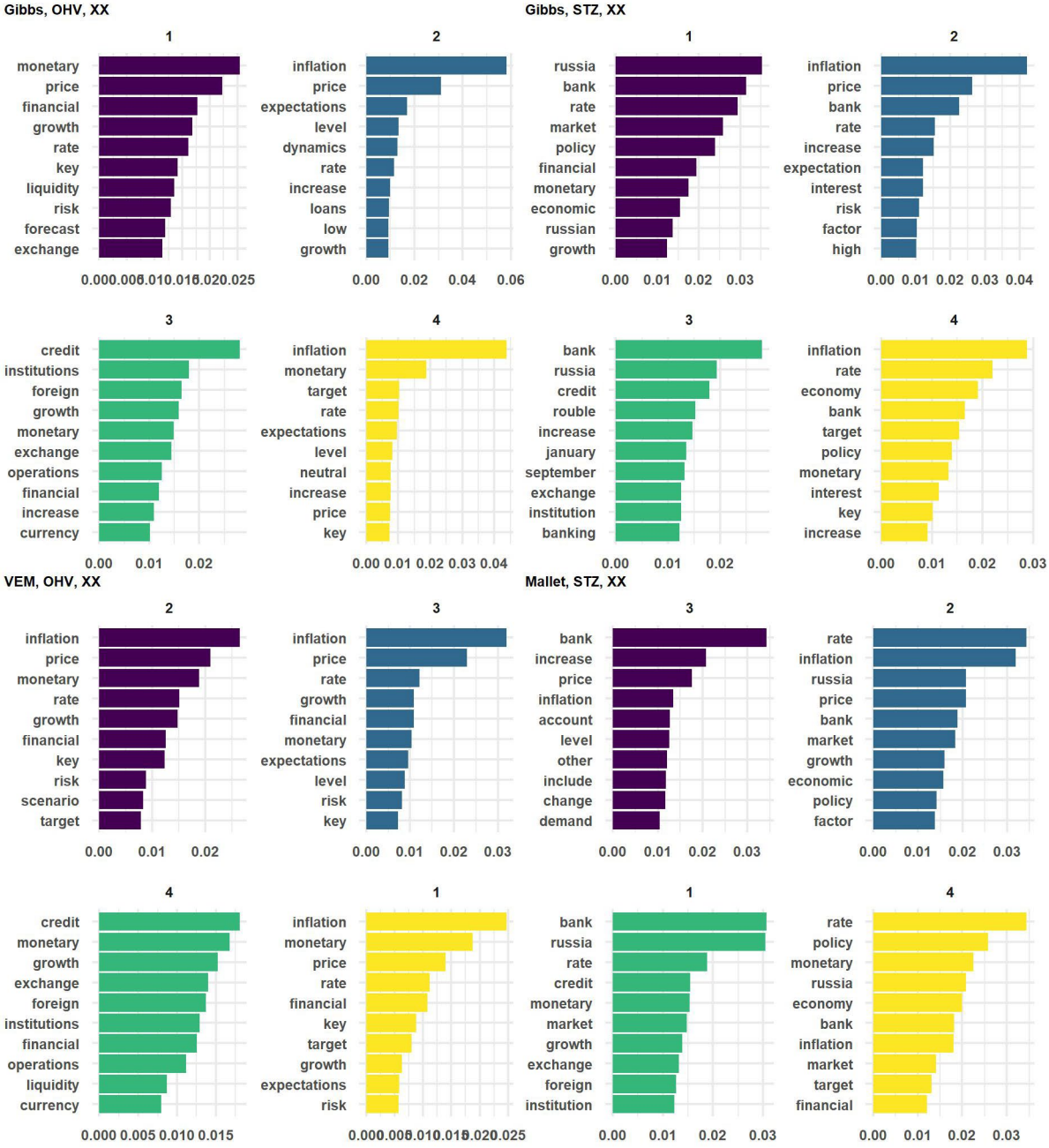
Figure 10は，Figure 9と同様に，1から4の番号は，各推定手法中でトピックを識別するためのラベルであり，トピックの自然言語的内容とは関係しない．一方，各パネル中のBar plotの色と表示順は

Figure 9と同じになるように，つまり自然言語的な内容が類似していると思われるトピックを同じ色としている．ただし，AppendixのqmdファイルでFigure 10を再現した場合，乱数発生器の実装等によってもトピック識別番号は変わるため，Figure 10を再現できるとは限らない点に注意していただきたい．分析対象が通常処理データ(OHV)とStanza処理データ(STZ)の場合では，分析対象となっている単語種セットが違うので頻出単語も変わり，同じデータでも推定方法によって単語種の出現確率は変わる．Gibbsの場合は，モデルのパラメータの確率分布を得ているだけであるから，計算を繰り返すと結果は微妙に変わる．VEMの場合も，計算過程においてランダムに決まる要素を含んでいるため，常に同じ結果になるとは限らない．

頻出語上位10語のパターンを見ると，Gibbsの1,2,3,4にVEMの2,3,4,1がほぼ対応するようにみえる．Gibbsの1とVEMの2は通貨政策，金利政策にかかわるトピック，Gibbsの3とVEMの4は資金供給，為替レート，実物経済にかかわるトピック，Gibbsの4とVEMの1はインフレーションと物価にかかわるtopicではないかと思われる．Gibbsの2，VEMの3は上位語からだけではそれぞれGibbsの4，VEMの1との違いがはっきりしないが，Gibbsの2，VEMの3は金融システム管理にかかわるトピックではないかと思われる．Stanza処理データでは，上位語からトピックの自然言語的内容を識別することは通常処理データより困難であるように思われる．各推定手法で通常処理データでもStanza処理データでもトピック構造の類似性が高いことを考慮して，上位10語の5-10番程度に着目すると，通常処理データとStanza処理データの間にFigure 11のトピック番号のような対応関係があると考えられる．

Stanza処理データでは，MPG文書の自然言語的内容を分析する上ではあまり意味のないRussia，Bank，月名等が頻出語の上位にある．通常処理データでは，MPG文書の性格を考慮して，Russia, Bank, Central，月名のような分析上ノイズとなる可能性の高い単語を手動で取り除いている．一方，Stanza処理データでは，おそらくStanzaのlenmmaやnerが一般的な文書で訓練されているため，MPG文書の分析にとってはノイズになるような単語が残ったと考えられる．つまり，MPGのトピック分析では，Stanzaの標準的処理だけでは不十分であることがわかる．加えて，通常処理データ，Stanza処理データともに，rateが頻出語の上位に入っている．MPGの場合，rateはinflation rate，growth rate，key rate，exchange rate等として多くの複合語で使われている．本研究ノートの第1部で問題点として指摘した複合語の専門用語をどう取り扱うかという問題がここにも現れている．場合によっては，rateそのものを削除することも考えられる．トピックの識別は単語種全体の分布の違いでおこなわれるため，rateがトピックにかかわらず多数使われていることはトピック識別にとっては重要ではないかもしれない．しかし，トピックの自然言語的な内容の識別においてはrateのような単語の存在はノイズになる．この場合，前処理においてrateを削除するかどうかは決め難い．

Figure 10. トピック別出現頻度上位10単語



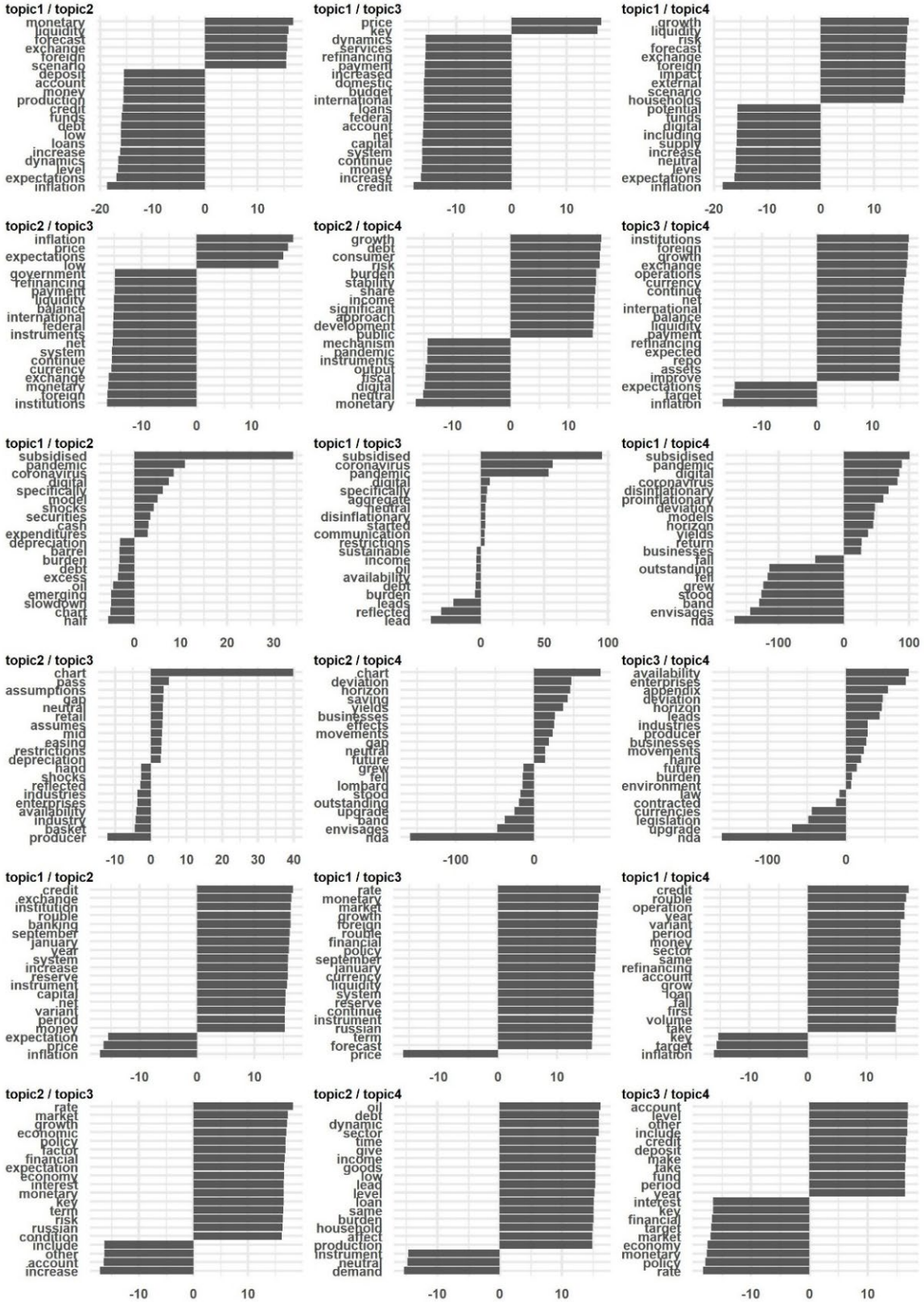
Note: see Table 3 for the notations.
Source: Authors' calculation.

トピックの自然言語的内容の識別には，トピック間で出現確率が大きく異なる単語をみることが有益である．Figure 11 は，紙幅の制約からTable 3中の3つの設定で得られた4トピックのすべての組合せの単語(term)の出現確率の比を示している．Figure 11の上部2行の6つのプロットがGibbs, OHV, XX設定，中央2行の6つのプロットがVEM, OHV, XX設定，下部2行の6つのプロトがMallet, STZ, XX設定である．異なる推計方法によって得られたトピック間で比較することもできるが，ここではおこなっていない．

まず，出現確率そのものが低いtermはそもそも重要性が低いと想定し，2つのトピックのいずれかにおいて出現確率が0.001以下であるtermを除いた上で，出現確率比の大きい上位20 termを選ぶ．閾値0.001は恣意的に設定した値であるため検討の余地がある．出現確率比の値は範囲が広くなるため対数(log2)に変換して表示する．対数化した出現確率比の絶対値が大きい20 termを取り出しているが，Figure 11ではオリジナルの値の符号を回復し，符号付の値で表示している．符号は，分母のトピックで出現確率が大きければ負，小さければ正としている．

Figure 11の詳細な検討は，MPGの経済的内容の検討と不可分であるため，ここではおこなわない．おおざっぱには，通常処理データ(OHV)については，Figure 10でおこなったトピックの自然言語的内容の検討結果をほぼ支持するものとなっている．Stanza処理データ(STZ)については，やはりrussia, january，bank，policy，other等のMPGの場合は分析上の意味を特定することが困難なtermが上位に入っている．Stanza処理データの場合は，Figure 11だけから抽出されたトピックの自然言語的内容を同定することは困難に思われる．

Figure 11. 単語出現確率比上位20単語(log2)



Source: Authors' calculation.

5. むすび

ロシア中央銀行の16年分の政策文書MPGを対象にRのtopicmodels，lda，Malletのパッケージを使ってLDAによるトピック分析を試行した．分析対象テキストは，研究ノート第1部でおこなったtidytextを使った通常処理データとStanzaによるUD準拠処理をおこなったStanza処理データの2種類を使った．この試行分析が明らかにしたことは次の点である．

(1) 文書別トピック構造の把握については，通常処理データかStanza処理データかで大きな違いは出ないことがわかった．一方，訓練済みのStanzaをそのまま使う場合は，定型的な構成，表現をもちい，複合語を含む専門用語を多用する専門分野の文書に対して十分対応できない可能性がある．MPGの場合，Stanza処理後も，MPGに特有であるが，トピック分析には不要と思われる単語が相当程度残った．トピック構造の同定にはほとんど影響していないが，抽出されたトピックの自然言語的意味の解釈を困難にした．MPGの場合は，通常処理とStanza処理の両方をおこなうことがより適切な前処理となるだろう．前処理をどのようにおこなうべきかを一律に判断することは困難であり，分析対象の専門分野知識に依拠して，どのような処理をすべきか判断するしかないであろう．

分析結果の比較可能性，多言語対応，より高度な(自然言語的内容を考えた)トピック分析への接続を考えると，Stanzaによるデータ処理をおこなうことは望ましい．さらにStanza処理によるPOS情報とLDAを組み合わせることで，トピック分析精度の向上と抽出されたトピックの自然言語的意味の識別可能性の向上が可能になると思われる．この点については，研究ノート第3部で試行，検討する予定である．

(2) LDAモデルは，トピック構造の抽出については，相対的に高速に頑健な結果を与える．LDAモデルを計算するための異なる分析アルゴリズム間でも，ほぼ同じようなトピック構造となったことは，驚きというべきかもしれない．ただし，ldaパッケージが異なる結果を与えたように，その結果が常に頑健であるとは言えない．LDAモデルでは抽出されたトピックの自然言語的意味は原理的に確実にはわかることは原理的にない．LDAモデルが異なる結果を与えたとき，異なる結果となった原因を探ることは困難である．ここからわかることは，1つの分析アルゴリズム，パッケージだけに頼ってLDAトピック分析をおこなうことは危険である可能性が高いということである．

(3) Bag of words仮定は，文書別トピック分布，トピック別単語種分布というトピック構造の同定を可能にする．このようなトピック構造の同定は，文の分類を基本とするNNモデルベースのトピック分析では困難である．一方，トピックを単語の確率分布とみるのか，自然言語的意味のあるまとまりとみるのかという根本的問題が残る．この問題は解釈の問題だけではなく，トピック数の設定というトピック分析開始段階で直面する実践的問題である．MPGの場合，研究ノート第1部においてword network分析により4つの単語クラスターを識別していたことがトピックの自然言語的内容の同定に一定程度貢献した．

Bag of wordsを仮定とするトピック分析は，多数の雑多なトピックからなる大量の文書からトピック構造を抽出するという目的には適しているであろう．一方，本研究ノートでおこなったMPGのトピック分析の限りでは，Stanza処理データだけを分析対象データとしていたとすれば，LDAで抽出したトピックに自然言語的意味を対応させることはほぼできなかったであろう．相対的に少ないトピックからなり，日常用語と同じ用語が専門用語として別の意味に使われるような，相対的に少量のデータしかない専門文献を分析対象とする場合は，word network分析により単語クラスターを同定し，

トピック数，トピック内容について一定の分析をおこない，トピック数を先に決めておくというワークフローがよりよい分析ワークフローかもしれない．つまり，専門分野のドメイン知識を用いたword network分析用のような事前の追加的分析がないと抽出されたトピックの自然言語的な意味の解釈が困難になる可能性がある．以上のような「ただし書き」にかかわらず，専門分野の文書の分析においても，LDAモデルはトピック構造とその変化を識別し，さらにその自然言語的意味の解釈を支援する強力なツールであることは間違いない．

6. Reference

Asahara, M., Kanayama, H., Tanaka, T., Miyao, Y., Uematsu, S., Mori, S., Matsumoto, Y., Omura, M., & Murawaki, Y. (2018). Universal Dependencies Version 2 for Japanese. https://www.lsta.media.kyoto-u.ac.jp/mori/research/public/asahara-LREC18.pdf.

Asuncion, A., Welling, M, Smyth, P., Teh, Y.W. 2009. On Smoothing and Inference for Topic Models, *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, https://arxiv.org/pdf/1205.2662.

Bishop, C.M. 2006. *Pattern Recognition and Machine Learning*, Springer.

Blei, D.M. 2012. Probabilistic Topic Models, *Communications of the ACN*, 55(4): 77-84, doi:10.1145/2133806.2133826.

Blei, D.M., Kucukelbir, A., McAuliffe, J.D. 2017. Variational Inference: A Review for Statisticians, Journal of the American Statistical Association, 112:518, 859-877, DOI:10.1080/01621459.2017.1285773.

Blei, D.M, Ng, A.Y., Jordan, M.I. 2003. Latent Dirichlet Allocation, *Journal of Machine Learning Research*, 3: 993-1022.

Chang, J. et al. 2009. Reading tea leaves: how humans interpret topic models. *Proceedings of the 23rd annual conference on neural information processing systems*. IEEE, 288–296.

Chang, J. 2025. *Package* '*lda*', https://cran.r-project.org/web/packages/lda/lda.pdf.

Chen, J. et al. 2016. *WarpLDA: a Cache Efficient O(1) Algorithm for Latent Dirichlet Allocation*. https://doi.org/10.48550/arXiv.1510.08628.

Dozat T., Manning, C.D. 2017. Deep Biaffine Attention for Neural Dependency Parsing, *Conference paper at ICLR 2017*, https://nlp.stanford.edu/pubs/dozat2017deep.pdf.

Graham, S., Weingart, S., Milligan, I. 2026. *Getting Started with Topic Modelling and MALLET*, https://doi.org/10.46430/phen0017.

Griffith, T.L., Steyvers, M. 2004. Finding scientific topics, *PNAS*, 101, suppl.1, 5228-5235. www.pnas.org/cgi/doi/10.1073/pnas.0307752101.

Gruen, B., Hornik, K. 2011. topicmodels: An R Package for Fitting Topic Models, *Journal of Statistical Software*, 40(13): 1–30. https://doi.org/10.18637/jss.v040.i13.

Heinrich, G. 2008. Parameter estimation for text analysis, *Technical Note*, vsonix GmbH+University of Leipzig. https://www2.cs.uh.edu/~arjun/courses/advnlp/text-est_heinrich.pdf.

Hanker, M., Kasri, M., Beni-Hssane, A. 2025. A comprehensive overview of topic modeling: Techniques, applications and challenges, *Neurocomputing*. https://doi.org/10.1016/j.neurocom.2025.129638.

Hemmatian, B. et al. 2019. Think of the consequences: A decade of discourse about same-sex marriage,

Behaviour Research Methods, 51:1565–1585. https://doi.org/10.3758/s13428-019-01215-3.

Hoffman, M.D. et al. 2013. Stochastic Variational Inference, *Journal of Machine Learning Research*, 14: 1303-1347.

Hornik, K., Schwendinger, F., Amon, J. 2025. *Stanza: 'Stanza' - A 'R' NLP Package for Many Human Languages*. R package version 1.0-3. https://cran.r-project.org/web/packages/Stanza/.

Laureate, C.D.P., Buntine, W., Ling, A. 2023. A systematic review of the use of topic models for short text social media analysis, *Artificial Intelligence Review*, https://doi.org/10.1007/s10462-023-10471-x.

Magnusson, M. 2026. *Package* '*mallet*', https://cran.r-project.org/web/packages/mallet/mallet.pdf.

Marecek, D. 2025. *Gibbs Sampling for LDA*, NPFL097 Unsupervised Machine Learning in NLP, Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, https://ufal.mff.cuni.cz/~marecek/npfl097/06_lda.pdf, last accessed on June 25, 2026.

de Marneffe, M.-C., Manning, C.D., Nivre, J. and Zeman, D. 2021. Universal Dependencies. *Computational Linguistics*, 47(2): 255–308.

McCallum, A. K. 2002. *MALLET*: *A Machine Learning for Language Toolkit*. http://mallet.cs.umass.edu.

Minka, T.P. 2003. *Estimating a Dirichlet distribution*, https://people.csail.mit.edu/brussell/research/words/ICCV05/Min03.pdf.

Mimno, D., Hoffman, M.D., Blei, D.M. 2012. *Sparse Stochastic Inference for Latent Dirichlet Allocation*, https://doi.org/10.48550/arXiv.1206.6425.

Newman, D. et al. 2007. Distributed Inference for Latent Dirichlet Allocation, *NeurIPS Proceedings*, https://proceedings.neurips.cc/paper_files/paper/2007/file/2dea61eed4bceec564a00115c4d21334-Paper.pdf.

Nivre J. et al. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection, *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pp. 4034–4043.

Murphy, K.P. 2023. *Probabilistic Machine Learning, Advanced Topics*, The MIT Press, Cambridge and London. https://github.com/probml/.

Phan, X.-H., Nguyen, L.-M., Horiguchi, S. 2008. *Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections,* WWW 2008, April 21–25, 2008, Beijing, China. DOI: 10.1145/1367497.1367510.

Porteous, I. et al. 2008. Fast Collapsed Gibbs Sampling For Latent Dirichlet Allocation, *KDD '08*: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 569-577. https://doi.org/10.1145/1401890.1401960.

Qi, P. et al. 2020. *Stanza*: *A Python Natural Language Processing Toolkit for Many Human Langu*ages. In Association for Computational Linguistics (ACL) System Demonstrations.

R Core Team, 2025. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Robert, M.E., Stewart, B.M., Tringley, D. 2019. stm: An R Package for Structural Topic Models, *Journal of Statistical Software*, 91(2), doi:: 10.18637/jss.v091.i02.

Roeder, M., Both, A., Hinneburg, A. 2015. Exploring the Space of Topic Coherence Measures, *WSDM'15*, February 2–6, 2015, Shanghai, China, .doi:: 10.1145/2684822.2685324.

Schmid, H. 2019. *Deep Learning-Based Morphological Taggers and Lemmatizers for Historical Texts*. DATeCH, May 2019, Brussels, Belgium, https://www.cis.uni-muenchen.de/~schmid/papers/Datech2019.pdf.

Stevens, K. et al. 2012. Exploring Topic Coherence over many models and many topics, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 952–961.

Stevers, M. 2007. Probabilistic Topic Models. In: Landauer, T.K. McNamara, D.S., Dennis, S. and Kintsch W. (eds) *Handbook of Latent Semantic Analysis*, Routledge: New York and London, 427-448.

Tran, M.-N., Nguyen, T.-N., and Dao, V.-H. 2021. *A practical tutorial on Variational Bayes*, https://doi.org/10.48550/arXiv.2103.01327.

Wallach, H. M., Mimno, D., McCallum, A. 2009. Rethinking LDA: Why priors matter. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1973–1981.

Yao, M., Mimno, D., McCallum, A. 2009. *Efficient Methods for Topic Model Inference on Streaming Document Collections*, KDD'09, June 28–July 1, 2009, Paris, France. https://mimno.infosci.cornell.edu/papers/fast-topic-model.pdf.

岡山香・中村靖「社会経済研究のためのテキスト分析: ロシア中央銀行通貨政策ガイドラインを例として」『エコノミア』76(1･2): 65–100, jstage.jst.go.jp/article/economia/76/1-2/76_65/_pdf/-char/ja, 2025年.

Appendix

本研究ノートのテキスト内容とRコードが入っているR Quarto qmdファイルは，MPGデータともに以下にある: https://osf.io/fvrhw/.