トピックモデル
LSA/LSI と pLSA/pLSI と LDA

櫻井 彰人

ベクトル空間モデル

- 目的: 文書検索
- 文書を、高次元ベクトルで表現する
  - 単語＝次元という空間
- 各要素は、異なる単語に対応し、当該単語が当該文書に現れた回数を表す
  - 非常に次元が高くなる
  - 必要に応じて、単語を制限する
    - 機能語や共通に現れる内容語（stop word）は使わない
    - 語幹抽出を行う（複数形、過去形、派生形等は一つにして扱う
    - 実際に取り扱う単語の集合は、辞書と呼ばれる
- 文書間の類似性は、しばしば、ベクトルの内積で表現する

G. Salton, A. Wong, and C. S. Yang (1974), "A Vector Space Model for Automatic Indexing," *Computer Science Technical Reports*, 1974-07, Cornell University,

G. Salton, A. Wong, and C. S. Yang (1975), "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, nr. 11, pages 613–620

ベクトル空間モデルの問題点

- 単語の表現形（文字の連なりとして。ビットパターンで表現したとして）を用いることの問題点
  - コンピュータを用いるときの、共通の問題点
- 多義語
  - Java はコーヒーを意味するがあるコンピュータ言語も意味する
  - 意味は異なるが、表現は同じ
- 同義語
  - 計算機、コンピュータ、PC はほぼ同じ意味
  - 意味は（ほぼ）同じだが、表現異なる

問題設定

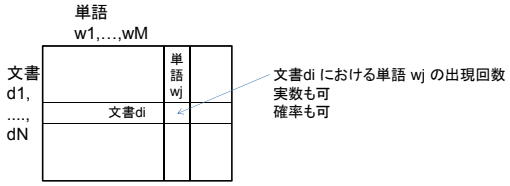
- 目的
  - 文書検索。意味的に類似な文書を検索したい
- 課題
  - （文書、単語の）意味は、コンピュータには分からない
  - 単語の表現そのものを用いては、意味は表せない
- 解決案
  - 文書AとBが別のものであっても、使っている単語集合が似ていれば、意味が似ているのでは？
  - 単語XとYが別のものであっても、同じような文書群に現れるなら、意味が似ているのでは？
  - 似ている文書（単語）に似ている文書（単語）は、似ているのでは？
  - この再帰的構造は一気に解けるのでは？

類似の問題

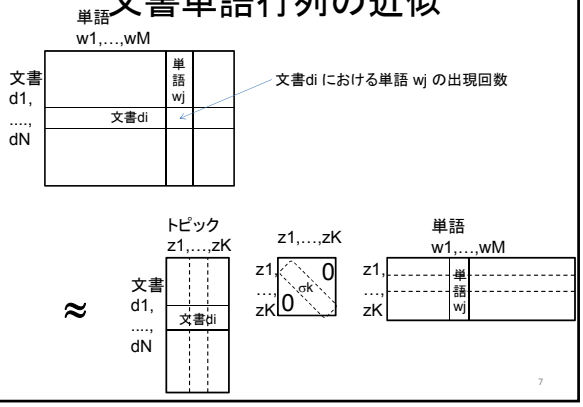
- 顧客 vs 購入商品
  - 類似した商品を購入している顧客は、類似した行動をとる（再び類似した商品を購入する）
  - 「類似」しているかどうかは、同じような顧客が購入していることで、判断したい
  - 顧客Aと顧客Bが類似しているが、商品Xを顧客Aが購入しているのに、顧客Bが購入していなければ、顧客Bは商品Xを購入する可能性が高い。では、これを推薦しよう
- ツイッターユーザ vs. フォロアー
- ユーザ vs お気に入り
- 画像 vs 画像のパッチ
- 企業（株価） vs 株価の動き（ある時間幅）
- 人 vs 筆跡の一部
- Audio scene vs 音のclip

トピック

- 意味が似ている単語の集合をトピックと呼ぶ
  - 集合は multi-set と呼ばれるものの更に拡張で、個数として実数がとれるものとする。さらに拡張して、確率事象でもよいことにする。
- 1文書は、1個以上のトピックからなる
- ある単語があるトピックに属する程度、あるトピックがある文書を構成する程度は、0/1ではなく程度を持つとする。
- そう仮定すると「文書単語行列」を考えることができる



文書単語行列の近似



7

実際の近似例

- 行列の「積による分解」による近似と考えることができる
- その一例として、SVDが知られている
  - Singular Value Decomposition 特異値分解

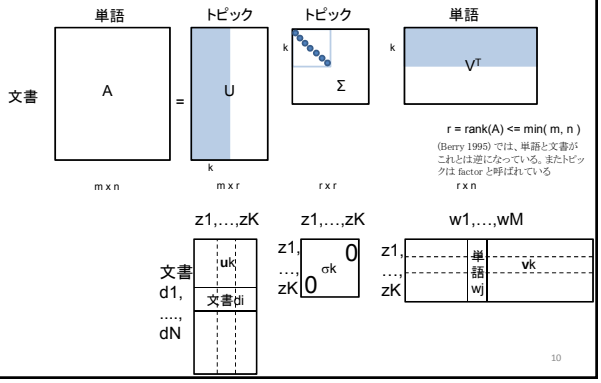
8

SVD

- 特異値分解: SVD (Singular Value Decomposition)
  - A{m×n 行列} = U{m×r 行列} Σ{r×r 対角行列} V*{r×n 行列}
    r = min(m, n)
  - U, V: (それぞれ)正規直交ベクトルの列
  - Σ: 正(または0)の特異値からなる対角行列
  - 近似行列を得る: k (≤r)個の特異値のみ使用
    - Ak{m×n 行列} = U{m×k 部分行列} Σ{k×k 部分行列} V*{k×n 部分行列}

9

SVDとの対応



10

蛇足: 固有値と特異値

固有値分解： $N' = \sum_{k=1}^{K} \lambda_k \mathbf{u}_k \mathbf{u}_k^{\mathrm{T}}$

N': 実対称行列, λ_k : 固有値, $\mathbf{u}_k$: 固有ベクトル

$N'\mathbf{u}_k = \lambda_k \mathbf{u}_k$

ベクトルは縦ベクトル

特異値分解： $N = \sum_{k=1}^{K} \sigma_k \mathbf{u}_k \mathbf{v}_k^{\mathrm{T}}$

σ_k : 特異値, $\mathbf{u}_k$: 左特異ベクトル, $\mathbf{v}_k$: 右特異ベクトル

$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}, \mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}$

$N\mathbf{v}_k = \sigma_k \mathbf{u}_k, \quad \mathbf{u}_k^T N = \sigma_k \mathbf{v}_k^T$

11

単なる近似を越えて
LSA: Latent Semantic Analysis

- Latent – "潜在", "隠れ" (観測できないということ)
- Semantic – “意味”

LSA を用いると単語の “隠れた意味”を、
単語が文書中に現れる様子から見出すことができたらいいなぁ/ことができる

12

潜在意味空間
Latent Semantic Space

- LSA は、単語と文書を潜在意味空間に写像する（べし）。そのとき、
- 潜在意味空間においては、同義語（類義語）は近くにくるべきである
- 実際に、SVDでそれが実現できる

LSA vs. LSI

- LSA と LSAI
  - LSA: Latent Semantic Analysis
  - LSI: Latent Semantic Indexing
- 両者の違いは？
  - LSI: 情報の indexing , i.e. 情報検索に用いる.
  - LSA: 解析（いろいろ）に用いる.
  - 同じ技術を異なる分野に適用しただけ.

簡単な例

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
ANTHONY	1	1	0	0	0	1
BRUTUS	1	1	0	1	0	0
CAESAR	1	1	0	1	1	1
CALPURNIA	0	1	0	0	0	0
CLEOPATRA	1	0	0	0	0	0
MERCY	1	0	1	1	1	1
WORSER	1	0	1	1	1	0

または

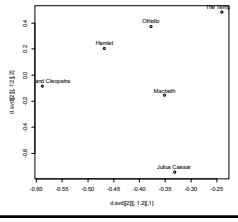
	ANTHONY	BRUTUS	CAESAR	CALPURNI	CLEOPATI	MERCY	WORSER
Anthony and Cleopatra	1	1	1	0	1	1	1
Julius Caesar	1	1	1	1	0	0	0
The Tempest	0	0	0	0	0	1	1
Hamlet	0	1	1	0	0	1	1
Othello	0	0	1	0	0	1	1
Macbeth	1	0	1	0	0	1	0

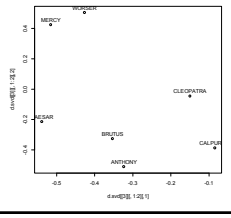
Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.

簡単な例

```
setwd("E:/R/")
d <- read.csv("08LSI-Shakespeare.csv", header=T, row.names=1)

d.svd <- svd(d)

d - d.svd[[2]] %*% diag(d.svd[[1]]) %*% t(d.svd[[3]])

plot(d.svd[[2]][,1:2])
text(d.svd[[2]][,1:2],labels=rownames(d),pos=3)

plot(d.svd[[3]][,1:2])
text(d.svd[[3]][,1:2],labels=colnames(d),pos=3)
```

	ANTHONY	BRUTUS	CAESAR	CALPURNI	CLEOPATI	MERCY	WORSER
Anthony and Cleopatra	1	1	1	0	1	1	1
Julius Caesar	1	1	1	1	0	0	0
The Tempest	0	0	0	0	0	1	1
Hamlet	0	1	1	0	0	1	1
Othello	0	0	1	0	0	1	1
Macbeth	1	0	1	0	0	1	0





簡単な例：近似

```
> d
                    ANTHONY BRUTUS CAESAR CALPURNIA CLEOPATRA MERCY WORSER
Anthony and Cleopatra       1      1      1         0         1     1      1
Julius Caesar               1      1      1         1         0     0      0
The Tempest                 0      0      0         0         0     1      1
Hamlet                      0      1      1         0         0     1      1
Othello                     0      0      1         0         0     1      1
Macbeth                     1      0      1         0         0     1      0

> d.svd[[2]][,1:2] %*% diag(d.svd[[1]][1:2]) %*% t(d.svd[[3]][,1:2])
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,]  0.8327142 0.87075536 1.2819192  0.258569439 0.3538872 1.12392852  0.9036710
[2,]  1.1568480 0.92786465 1.0082634  0.665723465 0.2585694 0.06248386 -0.1700712
[3,] -0.1699857 0.03122636 0.3112280 -0.280056074 0.1005788 0.88202181  0.8725302
[4,]  0.3948208 0.52230162 0.9078465  0.000357143 0.2582987 1.11497839  0.9820280
[5,]  0.1137595 0.29134444 0.6477284 -0.152156330 0.1909063 1.07039479  0.9953852
[6,]  0.6006586 0.58576299 0.8086186  0.232555085 0.2202575 0.58555509  0.4377091
> d.svd[[2]][,1:4] %*% diag(d.svd[[1]][1:4]) %*% t(d.svd[[3]][,1:4])
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,]  1.02285340  1.02912701 0.9457880  0.009006517  0.965970538  1.0385090  0.99063266
[2,]  0.92900279  1.06869323 1.1062884  0.840727430  0.009006517 -0.0134736 -0.09274244
[3,] -0.17268628  0.03570139 0.3098034 -0.279034563  0.101727564  0.8796082  0.87498743
[4,]  0.06021168  0.84004232 0.9495523  0.218959286  0.054149957  0.9436014  1.15649897
[5,]  0.05649234  0.14951847 0.8357352 -0.041673681 -0.131215398  1.1468902  0.91750879
[6,]  1.00590863 -0.08575238 1.0223925  0.079268842  0.047876347  0.9477444  0.06898092
> d.svd[[2]][,1:6] %*% diag(d.svd[[1]][1:6]) %*% t(d.svd[[3]][,1:6])
          [,1]          [,2]          [,3]          [,4]          [,5]          [,6]          [,7]
[1,]  1.000000e+00  1.000000e+00  1.000000e+00 -2.949030e-16  1.000000e+00  1.000000e+00  1.000000e+00
[2,]  1.000000e+00  1.000000e+00  1.000000e+00  1.000000e+00 -1.491862e-16 -8.222589e-16 -6.002143e-16
[3,] -1.734723e-17 -1.089406e-15 -6.591949e-16  2.151057e-16 -6.245005e-17  1.000000e+00  1.000000e+00
[4,] -3.851086e-16  1.000000e+00  1.000000e+00 -4.857226e-17 -4.857226e-16  1.000000e+00  1.000000e+00
[5,] -3.261280e-16 -8.049117e-16  1.000000e+00  1.665335e-16 -9.714451e-17  1.000000e+00  1.000000e+00
[6,]  1.000000e+00 -9.020562e-16  1.000000e+00 -5.412337e-16 -1.318390e-16  1.000000e+00 -9.228729e-16
```

共通（かつ大きな）データ例

```
library(slam)
library(irlba)
library(topicmodels)
data(AssociatedPress)

AP <- Matrix(AssociatedPress, nrow=AssociatedPress$nrow)
AP.svd <- irlba(AP, nv = 5)

for ( i in 1:5 ) print(AssociatedPress$dimnames$Terms[sort(sort.int(AP.svd$v[,i],decreasing=T, index.return=T)$ix[1:10])])
```

government	bush	east	bush	cent
i	gorbachev	german	germany	cents
last	i	government	gorbachev	dollar
million	people	officials	party	futures
new	police	police	president	lower
people	president	soviet	soviet	market
percent	soviet	two	states	million
president	state	union	trade	new
two	told	united	union	stock
year	two	west	united	york

データの説明

- Blei がLDAを提案する論文で用いたものと類似したもの（小規模）
- Associated Press data: the First Text Retrieval Conference (TREC-1) 1992.

```
<<DocumentTermMatrix (documents: 2246, terms: 10473)>>
Non-/sparse entries: 302031/23220327    計 23,522,358
Sparsity           : 99%
Maximal term length: 18
Weighting          : term frequency (tf)
```

D. Harman (1992) Overview of the first text retrieval conference (TREC-1). In Proceedings of the First Text Retrieval Conference (TREC-1), 1–20.

19

疎行列（sparse matrix）

- 要素がほとんど0の、大きな行列
- 実応用に良く出てくるので、疎行列のための、効率のよい記憶方法とそれを操作する関数との組が用意されることが多い
- 今回は、R のパッケージ slam で使用されている "simple triplet matrix" 形式を用いる
  - 疎行列用のRパッケージとしては、Matrix が著名

20

pLSA: 確率的トピックモデル

21

LSA/LSIの問題点

- トピックの意味づけがなされていない
  - 確かに、うまく行っている
  - しかし、特異ベクトルは、数学的には意味付けされているが、文の意味や単語の意味に関連しての意味づけはされていない
  - なぜうまく行くか、なぜうまくいかないかが説明できていない。

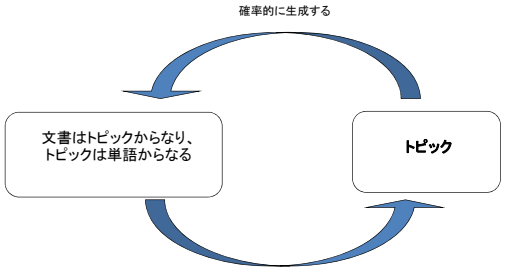
22

確率的トピックモデル

- 確率で意味づけしよう
  - 各文書は、トピックの上のある確率分布
  - 各トピックは、単語上のある確率分布
- LSA/LSI の確率モデル版pLSA/pLSIが最初.

23

生成モデル

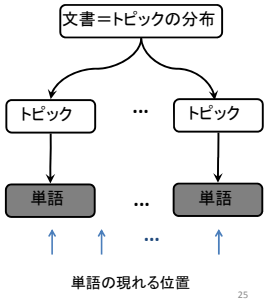


なお、pLSI は生成モデルとはいえない一面がある。未知文書のトピック分布が生成されないからである。[Blei et al. 2003]

24

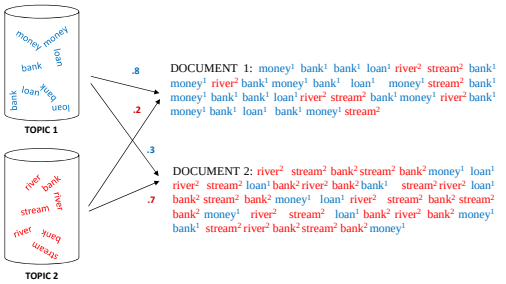
文書の生成過程

1. 各文書につき、トピックの、ある分布を定める
2. 各トピックにつき、単語の、ある分布を定める
3. （各文書の各単語位置で）トピックをサンプルし、
4. そのトピックから単語をサンプルする



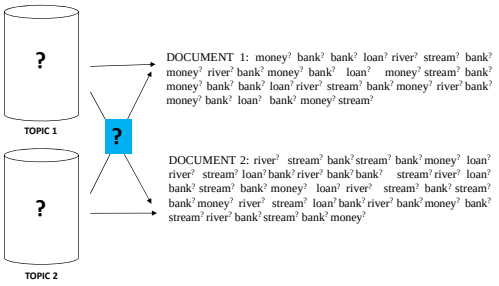
25

文書の生成例



http://helper.ipam.ucla.edu/publications/cog2005/cog2005_5282.ppt

モデルの学習



27

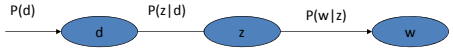
Aspect Model

- pLSA の一つの方法として
- アスペクトモデル
  - 文書は、基盤である（潜在的な）K 個のアスペクトの混合である
  - 個々のアスペクトは単語分布 p(w|z) で表される
- 学習には Tempered EM を使用

28

アスペクトモデル

- Hofmann 1999 の提案
- 共起するデータに対する潜在変数モデル
  - 個々の観測データ (w,d) にクラス変数 z ∈ Z={z_1,…,z_K} を付随させる
- 生成モデル
  - 確率 P(d) で文書を選ぶ
  - 確率 P(z|d) で、潜在クラス z を選ぶ
  - 確率 p(w|z) で、単語 w を選ぶ



29

等価なモデル

$$P(d,w) = P(d)P(w|d), \quad \text{where}$$

$$P(w|d) = \sum_{k=1}^{K} P(w|z_k)P(z_k|d)$$

$$P(d,w) = \sum_{k=1}^{K} P(z_k)P(d|z_k)P(w|z_k)$$

30

文書クラスタリングとの比較

- 文書は、クラスタ（アスペクト）一つだけに関連付けられるものではない
  - 文書ごと P(z|d) はアスペクトのある混合を定める
  - より柔軟性が高く、より有効なモデルができよう

ただ, P(z), P(z|d), P(w|z) を計算しないといけない. あるのは文書(d) と単語(w)のみなのに.

31

モデルの学習

- アスペクトモデルに従い、対数尤度を記述することができる。それを最大化すればよい

$$\mathcal{L} = \sum_{d \in \mathcal{D}} \sum_{w \in W} n(d,w) \log P(d,w)$$

- EM (Expectation Maximization) 法が使える
  - 過学習を避けるため tempered EM を用いる

32

まずは EM

- E-ステップ（指数型分布の場合）
  - 現在のパラメータ値に従って、潜在変数の期待値を求める
    - 潜在変数の分布を求める、でもある
    - 混合正規分布の場合、各観測点が「どの正規分布から生成されたか」ではなく「各正規分布からどのくらいの確率で生成されたか」を表す
- M-ステップ
  - 上記「対数尤度関数の期待値」を最大化するようパラメータを定める
    - 混合正規分布の場合、各正規分布の平均と分散共分散行列

33

多項分布（multinomial distribution）

- 通常、多項分布を用いる

$$p(y_1,...,y_k) = P(Y_1 = y_1,...,Y_k = y_k) = \frac{n!}{y_1!...y_k!} p_1^{y_1} ... p_k^{y_k}$$

但し、$\sum_{i=1}^{k} y_i = n, \sum_{i=1}^{k} p_i = 1, y_i \geq 0, p_i \geq 0$

$$p_i(y_i) = \frac{n!}{y_i!(n-y_i)!} p_i^{y_i} (1-p_i)^{n-y_i} \quad y_i = 0,1,..,n$$

(Y_i の周辺分布は2項分布となる)

$$\Rightarrow E(Y_i) = np_i \qquad V(Y_i) = np_i(1-p_i)$$

34

E ステップ

- 文書 d 中に現れる単語 w が所属する、潜在変数 z の分布（多項分布である）

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z \in Z} P(z)P(d|z)P(w|z)}$$

混合正規分布のときと同じで、各サンプルが複数個のトピックに属する。

各分布のパラメータを用いて、z の分布を得ている
右辺の P(z), P(d|z), P(w|z) はそれぞれの分布のパラメータと見る
左辺の P(z|d,w) は所属確率の期待値とみる

35

M ステップ

- 下記のパラメータは、E ステップで求めた p(z|d,w) を用いて表現できる（多項分布のパラメータ）

$$P(w|z) = \frac{\sum_d n(d,w)P(z|d,w)}{\sum_{d,w'} n(d,w')P(z|d,w')} \qquad P(z,d,w) \propto n(d,w)P(z|d,w)$$

$$P(d|z) = \frac{\sum_w n(d,w)P(z|d,w)}{\sum_{d',w} n(d',w)P(z|d',w)} \qquad P(z) \propto \sum_{d,w} n(d,w)P(z|d,w)$$

$P(d,w|z)$ を推定し、和を求めている

$$P(z) = \frac{\sum_{d,w} n(d,w)P(z|d,w)}{\sum_{d,w} n(d,w)} \qquad P(z) \propto \sum_{d,w} n(d,w)P(z|d,w)$$

- 尤度関数の局所最大値に収束する

36

pLSA 更新式まとめ

- pLSA の対数尤度

$$\mathcal{L} = \sum_{d\in\mathcal{D}}\sum_{w\in\mathcal{W}} n(d,w)\log P(d,w) \qquad P(d,w) = \sum_{z\in Z} P(z)P(d\mid z)P(w\mid z)$$

- EM アルゴリズム

- E - Step $P(z\mid d,w) = \frac{P(z)P(d\mid z)P(w\mid z)}{\sum_{z\in Z} P(z)P(d\mid z)P(w\mid z)}$

- M- Step $P(w\mid z) = \frac{\sum_{d} n(d,w)P(z\mid d,w)}{\sum_{d,w'} n(d,w')P(z\mid d,w')}$

$$P(d\mid z) = \frac{\sum_{w} n(d,w)P(z\mid d,w)}{\sum_{d',w} n(d',w)P(z\mid d',w)} \qquad P(z) = \frac{\sum_{d,w} n(d,w)P(z\mid d,w)}{\sum_{d,w} n(d,w)}$$

37

過学習

- ところが、pLSA ではパラメータ数が多いため、過学習（学習データはよく説明するが、未知データ上での性能は悪い）が起こってしまう
- フィットしすぎないようにする
- E ステップを少し修正する

38

TEM (Tempered EM)

- 学習量を制御するパラメータ β を導入する

$$P_\beta(z\mid d,w) = \frac{P(z)\left[P(d\mid z)P(w\mid z)\right]^\beta}{\sum_{z'} P(z')\left[P(d\mid z')P(w\mid z')\right]^\beta}$$

- β (> 0)は1から開始し、次第に減少させていく

39

Simulated Annealing

- 焼きなまし（annealing）: 金属を加工するにあたって、加工硬化による内部のひずみを取り除き、組織を軟化させ、展延性を向上させるため、一定温度に熱したのち、ゆっくりと冷却する方法 – 初期状態よりさらに内部エネルギーが低い状態にもっていく
- 疑似焼きなまし: 最小値解の候補解を繰り返し求めるにあたり、パラメータ β が大きければ大きく動き、小さければ小さく動くようにし、繰り返すに従って、徐々に β を小さくしていく方法。β が温度の働きをする。

40

β の選び方

- 適切な β はどう選べばよいか?
- β は 学習不足と学習過多を分ける
- Validation データセットを用いる簡便な方法は
  - 学習データを対象とした学習を β =1 から開始する
  - Validation dataset を用いて学習モデルをテストする
  - 前回より改善しているなら、同じ β で継続する
  - 改善がなければ、β <- ηβ where η<1

41

例: Perplexity の比較

- Perplexity – Log-averaged inverse probability （対未知データ）
- 確率が高ければ（よく予測できていれば） perplexity は下がる

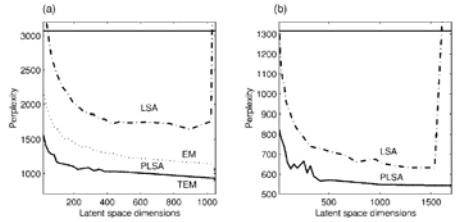


Figure 6. Perplexity results as a function of the latent space dimensionality for (a) the MED data (rank 1033) and (b) the LOB data (rank 1674). Plotted results are for LSA (dashed-dotted curve) and PLSA (trained by TEM = solid curve, trained by early stopping EM = dotted curve). The upper baseline is the unigram model corresponding to marginal independence. The star at the right end of the PLSA denotes the perplexity of the largest trained aspect models (K = 2048).

(Hofmann 2001)

例: トピック分解

- 1568 文書のアブストラクト
- 128 潜在クラスに分解

- "power" と名付けたいトピックに属する単語の語幹, i.e., p(w|z) が大きい語幹

Power1 – 宇宙関連
Power2 - 電気関連

"power 1"	power 2"
POWER	load
spectrum	memori
omega	vlsi
mpc	POWER
hsup	systolic
larg	input
redshift	complex
galaxi	arrai
standard	present
model	implement

Thomas Hofmann. Probabilistic Latent Semantic Analysis. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99)

例: 多義語

- 二つの異なる文脈(一文書一文脈)に出現する "segment" を検出している

Document 1, $P\{z_k|d_1, w_j = \text{'segment'}\} = (0.951, 0.0001, \ldots)$
$P\{w_j = \text{'segment'}|d_1\} = 0.06$

SEGMENT medic imag challeng problem field imag analysi diagnost base proper **SEGMENT** digit applic involv estim boundari object classif tissu abnorm shape analysi contour detec textur **SEGMEN** specif medic imag remain crucial problem [...]

Document 2, $P\{z_k|d_2, w_j = \text{'segment'}\} = (0.025, 0.867, \ldots)$
$P\{w_j = \text{'segment'}|d_2\} = 0.010$

consid signal origin sequenc sourc specif problem **SEGMENT** signal relat **SEGMENT** sourc addres resolu method ergod hidden markov model hmm hmm state correspond signal sourc signal sourc seq algorithm forward algorithm observ sequenc baumwelch train estim hmm paramet train materi appli experi perform unknown speaker identif [...]

Thomas Hofmann. Probabilistic Latent Semantic Analysis. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99)

共通(かつ大きな)データ例

```
library(topicmodels)     # for Associated Press data
data(AssociatedPress)

set.seed(1)
res <- plsa(AssociatedPress, K=5, eps=0.995, max_itr=30 )

for ( i in 1:5 ) print(
AssociatedPress$dimnames$Terms[sort.int(res$pw_z[i,],decreasing=T,index.return=T)$ix[1:10]] )
```

i	i	percent	soviet	president
people	court	million	governmen	bush
dukakis	years	year	police	states
two	two	new	people	new
new	people	market	party	house
campaign	case	billion	two	united
state	state	prices	president	year
bush	judge	company	military	committee
air	school	stock	united	governmen
officials	attorney	last	war	congress

pLSA

government	bush	east	bush	cent
i	gorbachev	german	germany	cents
last	i	government	gorbachev	dollar
million	people	officials	party	futures
new	police	police	president	lower
people	president	soviet	soviet	market
percent	soviet	two	states	million
president	state	union	trade	new
two	told	united	union	stock
year	two	west	united	york

LSA

手作りです

```
# R-code for pLSA/pLSI
# Reference: http://wg-stein.blogspot.jp/2009/11/
# probabilistic-latent-semantic-analysis.html

library(gtools)  # for rdirichlet
library(slam)

plsa <- function(x, K=10, eps=0.995, max_itr=100 ){
 if ( "simple_triplet_matrix" %in% class(x) ) { }
 else x <- as.simple_triplet_matrix(x)

 D <- x$nrow
 W <- x$ncol
 B <- 1      # Beta
 llhprev <- 0
 total_occurrences <- sum(x$v)

 pz_dw <- rep(0,K)
 pz <- rep(1/K, length=K)
 pd_z <- matrix(0, K, D)
 pw_z <- matrix(0, K, W)

 for(k in 1:K){
  pd_z[k,] <- rdirichlet(1,rep(1, length=D))
  pw_z[k,] <- rdirichlet(1,rep(1, length=W))
 }
 cz  <- rep(0, length=K)
 cd_z <- matrix(0, K, D)
 cw_z <- matrix(0, K, W)

 for(t in 1:max_itr){

  cat("Iteration: ",t," Beta: ",B,"  ")

 #E-step
  cz[]   <- 0
  cd_z[,] <- 0
  cw_z[,] <- 0

  for(dw in 1:length(x$i))  {
   d <- x$i[dw]
   w <- x$j[dw]
   xdw <- x$v[dw]

   pz_dw <- (10^100 * pz * pd_z[,d] * pw_z[,w])^B
   pz_dw <- pz_dw / sum(pz_dw)

   tmp <- xdw * pz_dw
   cz <- cz + tmp
   cd_z[,d] <- cd_z[,d] + tmp
   cw_z[,w] <- cw_z[,w] + tmp
  }

 #M-step
  pz  <- cz / sum(cz)
  pd_z <- cd_z / rowSums(cd_z)  # sum per k
  pw_z <- cw_z / rowSums(cw_z) # sum per k
```

```
 #converged?  (very costly)
  # for perplexity, see Hofmann, Unsupervised Learning by Probabilistic
  # Latent Semantic Analysis, Machine Learning, 42, 177-196, 2001
   llh <- 0
   lpp <- 0

   for(dw in 1:length(x$i))  {
    d <- x$i[dw]
    w <- x$j[dw]
    xdw <- x$v[dw]

    tmp <- sum( pz[] * pd_z[,d] * pw_z[,w] )
    llh <- llh + xdw * log( tmp )
    lpp <- lpp + xdw * log( tmp /sum( pz[] * pd_z[,d] ) )   }

   cat(" log-likelihood : ",llh, " Perplexity : ", exp(-1/total_occurrences * lpp),"¥n")

   if(t > 1){
    if( abs((llh - llhprev) / llh) < 1e-5 || llhprev > llh ){
     cat("Converged.¥n")
     break
    }
   }

   llhprev <- llh
   B <- eps * B
  }

  return(list("pz_dw"=pz_dw, "pz"=pz, "pd_z"=pd_z, "pw_z"=pw_z))
}
```

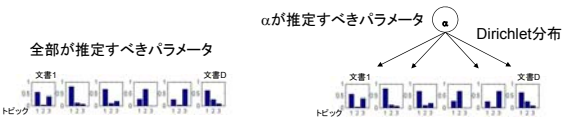

Latent Dirichlet Allocation

pLSA の欠点

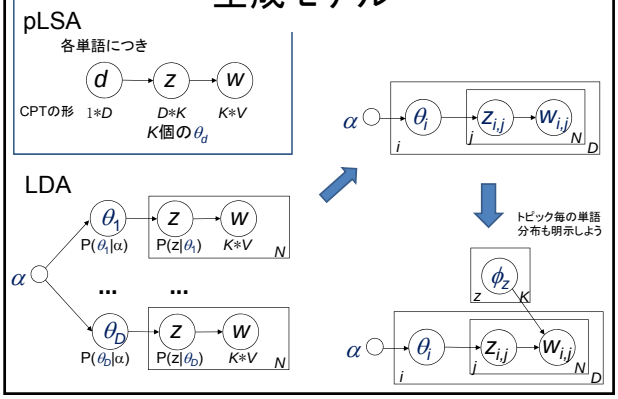
- pLSA においては, 観測可能変数 d はある学習データで用いる索引番号である。従って、未知文書を扱う自然な方法がない.
- pLSA のパラメータ数は、学習データ中の文書数にも比例する部分がある. トピック数が増えると過学習しがちである（そこで、T-EM を用いている）
- トピック混合にもベイズ的にできないか.

過学習の押さえ方

- 正規化項を導入する
  - 複雑さに対するペナルティ項
    - 例えば、パラメータ値が中心付近から離れすぎることに対するペナルティ
    - 最小化すべき関数に、ペナルティに比例する項を加算する
- 今回は、文書毎のトピック分布に制約を加えよう
  - トピック分布は、多項分布であった。そのパラメータ P(z|d) が d 毎に大きく異なると大きなペナルティとなる項を考えよう



生成モデル



LDAの特徴

- Latent Dirichlet Allocation
  - PLSA の問題を解決
    - （生成モデルとして）任意のランダム文書が生成できる
    - 新規文書に対応できる
  - パラメータ学習:
    - 変分 EM（Variational EM）
    - Gibbs サンプリング
      - 統計的なシミュレーション
      - 解にバイアスはない
      - 統計的な収束

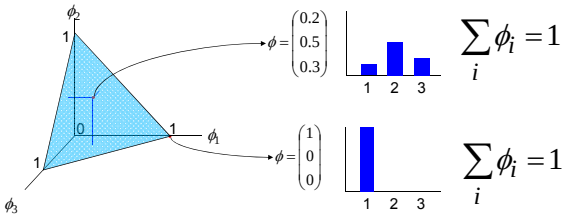
Dirichlet 分布

$$p(\theta \mid \alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \propto \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

- 有用な性質:
  - この分布は (k-1)-単体の上で定義される. すなわち, k 個の非負の引数を持ち、その総和は1であるという制約がある. 従って、これは多項分布に対して用いるのに極めて自然な分布である.
  - 事実, Dirichlet 分布は多項分布の双対分布である（これは、用いる尤度が、Dirichlet 分布を事前分布とする多項分布であれば、事後分布もDirichlet 分布となることを意味する）
  - Dirichlet 分布のパラメータ α_i は、i 番目のクラスの「事前」発現回数と考えることができる.

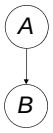
Dirichlet 分布

•k 次元単体上の各点は、一つの多項分布に対応する:



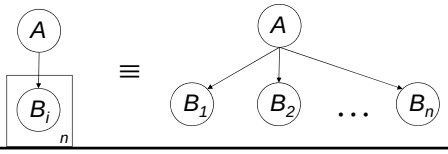
グラフィカルモデル

Bayesian Network の表現方法



≡ p(A,B) = p(B|A)p(A)

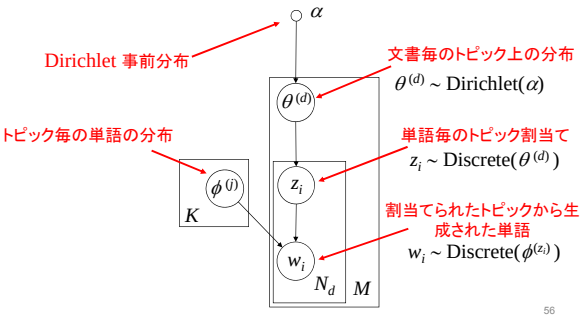
Plate記法



55

Latent Dirichlet Allocation

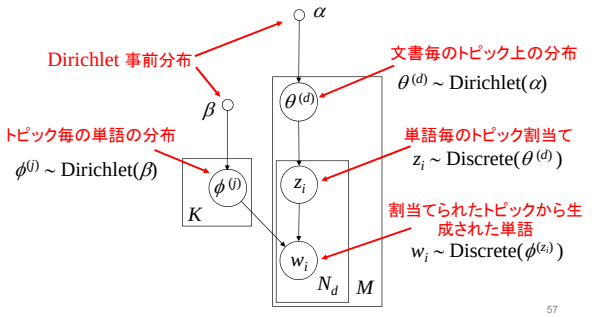
(Blei, Ng, & Jordan, 2001)



56

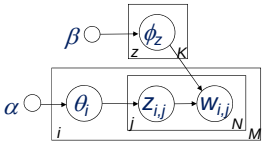
Smoothed Latent Dirichlet Allocation

(Blei, Ng, & Jordan 2003)



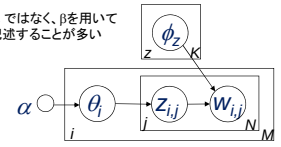
57

LDA モデル（比較. 念のため）



smoothed LDA

φz ではなく、βを用いて記述することが多い

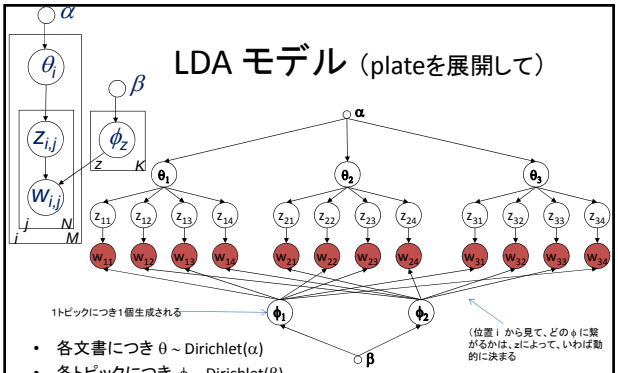


LDA

- 各文書につき θ ~ Dirichlet(α)
- 各トピックにつき φ ~ Dirichlet(β)
- 全N 単語 w_n のそれぞれにつき:
  - トピック z_n ~ Multinomial(θ)
  - 単語 w_n ~ $p(w_n|z_n,\phi)$, ただし、p は多項分布, i.e., $w_n \sim Multinomial(\phi_{z_n})$.

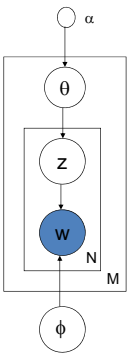
- 各文書につき, θ ~ Dirichlet(α)
- N 単語 w_n のそれぞれにつき:
  - トピック z_n ~ Multinomial(θ)
  - 単語 w_n ~ $p(w_n|z_n,\phi)$, ただし、p は多項分布, i.e., $w_n \sim Multinomial(\phi_{z_n})$.

LDA モデル（plateを展開して）



- 各文書につき θ ~ Dirichlet(α)
- 各トピックにつき φ ~ Dirichlet(β)
- 全N単語のそれぞれ（のindex n where n=1,...,N）につき:
  - トピック z_n ~ Multinomial(θ)
  - 単語 w_n ~ $p(w_n|z_n,\phi)$, ただし、p は多項分布, i.e., $w_n \sim Multinomial(\phi_{z_n})$.

LDA モデルと尤度



- For 文書 d = 1,⋯,M
  - 生成 θ_d ~ Dir(· | α)
  - For（文書中）位置 n = 1,⋯, N_d
    - 生成 z_n ~ Mult(· | θ_d)
    - 生成 w_n ~ Mult(· | ϕ_{z_n})

$$\prod_{d=1}^{M} p(w_1,\ldots,w_{N_d} \mid \alpha,\beta)$$

$$= \prod_{d=1}^{M} \int_{\theta_d} p(\theta_d \mid \alpha) \left\{ \prod_{n=1}^{N_d} \left(\sum_k p(z_n = k \mid \theta_d)\, p(w_n \mid \phi_k) \right) \right\} d\theta_d$$

$$= \prod_{d=1}^{M} \int_{\theta_d} p(\theta_d \mid \alpha) \left\{ \prod_{n=1}^{N_d} \left(\sum_k \theta_{dk} \phi_{kw_n} \right) \right\} d\theta_d$$

60

Gibbsサンプリング

- Gibbs サンプリング
  - 結合分布の評価は難しいが、条件付き確率なら容易な時
  - マルコフ連鎖を生成するようなサンプルの列を作る
  - この連鎖の定常分布が、求める結合分布になる

1. $x_{1:n}^{(0)}$ の初期化
2. for $i = 0$ to $N-1$
   - サンプルする: $x_1^{(i+1)} \sim p(x_1 | x_2^{(i)}, x_3^{(i)}, \ldots, x_n^{(i)})$
   - サンプルする: $x_2^{(i+1)} \sim p(x_2 | x_1^{(i+1)}, x_3^{(i)}, \ldots, x_n^{(i)})$
   - サンプルする: $x_j^{(i+1)} \sim p(x_j | x_1^{(i+1)}, \ldots, x_{j-1}^{(i+1)}, x_{j+1}^{(i)}, \ldots, x_n^{(i)})$
   - サンプルする: $x_n^{(i+1)} \sim p(x_n | x_1^{(i+1)}, \ldots, x_{n-1}^{(i+1)})$

61

Collapsed Gibbs サンプリング

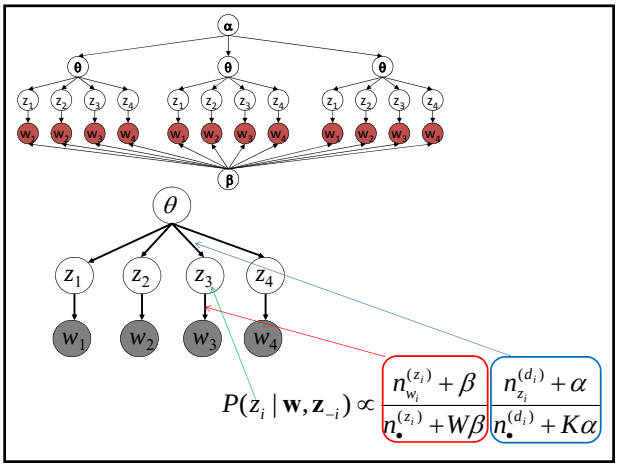
- パラメータを積分消去する
- 各 z_i を、次の $\mathbf{z}_{-i}$ で条件づけた分布でサンプルする

$$P(z_i \mid \mathbf{w}, \mathbf{z}_{-i}) \propto \frac{n_{z_i}^{(d_i)} + \alpha}{n_{\bullet}^{(d_i)} + K\alpha} \frac{n_{w_i}^{(z_i)} + \beta}{n_{\bullet}^{(z_i)} + W\beta} \propto (n_{z_i}^{(d_i)} + \alpha) \frac{n_{w_i}^{(z_i)} + \beta}{n_{\bullet}^{(z_i)} + W\beta}$$

$n_k^{(d)}$ は文書 d 中のトピック k の出現回数
$n_w^{(k)}$ は単語wのトピック k としての出現回数
d_i は文書中の i-th 単語が属する文書のID
w_i は文書中の i-th 単語の単語ID
z_i は文書中の i-th 単語のトピックID

- 容易に実行可能:
  - メモリ: 数え上げは、2個の疎行列で行える
  - 最適化: 特殊な関数はいらない, 単純な算術
  - **z** と **w** が与えられれば Φ と Θ の分布は求めることができる

M は文書数
W は異なり単語数
K はトピック数



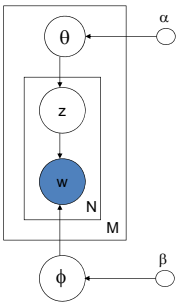
パラメータ推定

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_{\bullet}^{(d)} + K\alpha} \qquad \hat{\theta}_{j,k} = \frac{n_{j,(\cdot)}^{k} + \alpha}{n_{(\cdot),(\cdot)}^{(\cdot)} + K\alpha}$$

$$\hat{\phi}_j^{(w)} = \frac{n_w^{(j)} + \beta}{n_{\bullet}^{(j)} + W\beta} \qquad \hat{\varphi}_{k,i} = \frac{n_{(\cdot),i}^{k} + \beta}{n_{(\cdot),(\cdot)}^{k} + W\beta}$$

$n_j^{(d)}$ は文書 d 中のトピック j の出現回数
$n_w^{(j)}$ は単語wのトピック j としての出現回数
W は異なり単語数
K はトピック数

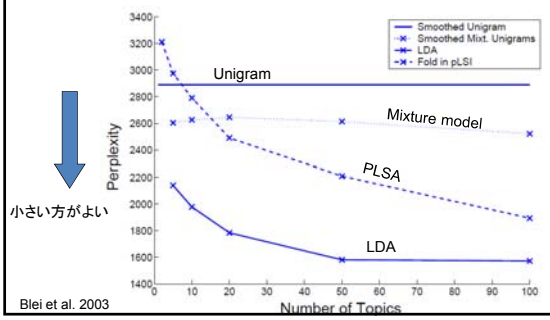
$n_{j,r}^{i}$: i-th トピック中の（辞書中）r-th 単語が j-th 文書に現れた回数



64

結果の例

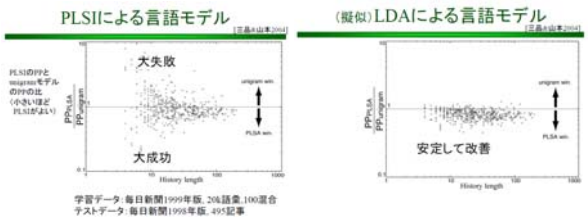
- pLSA等の比較



Blei et al. 2003

結果の例

- 過学習をしないという意味において、LDAの方がすぐれているという実験結果が多い。



http://chasen.org/~daiti-m/paper/topic2006.pdf

共通（かつ大きな）データ例

```
library(topicmodels)
data(AssociatedPress)

AP.ldaGibbs <- LDA(AssociatedPress, 5, method="Gibbs")

terms(AP.ldaGibbs,10)
```

percent	i	i	governmen	people
million	president	court	soviet	officials
year	bush	years	united	two
billion	house	police	police	air
new	new	case	military	city
company	committee	two	two	miles
last	congress	attorney	union	three
market	national	drug	party	area
prices	dukakis	school	people	fire
stock	campaign	children	states	day

LDA

i	i	percent	soviet	president
people	court	million	governmen	bush
dukakis	years	year	police	states
two	two	new	people	new
new	people	market	party	house
campaign	case	billion	two	united
state	state	prices	president	year
bush	judge	company	military	committee
air	school	stock	united	governmen
officials	attorney	last	war	congress

pLSA

パラメータ数による比較

手法	パラメータ数	効率的な解法	
LSA	(KW+KD)	SVD (Lanczos法)	
PLSA	KW+KD	EM	Dが入っているのでoverfitしやすい
LDA	KW+K	変分ベイズ/ Gibbs sampling	問題のDを消した

K: topicの数
W: 語彙数
D: 文書数

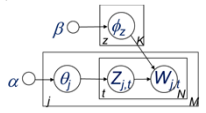
http://www.r.dl.itc.u-tokyo.ac.jp/study_ml/pukiwiki/index.php?openfile=PLSV.pptx&plugin=attach&refer=schedule%2F2008-11-06

付録

導出 ポイントだけ

$$p(\phi,\theta,\mathbf{z},\mathbf{w}\mid\alpha,\beta)=p(\phi\mid\beta)p(\theta\mid\alpha)\prod_{n=1}^{N}p(z_n\mid\theta)p(w_n\mid z_n,\phi)$$

$$p(\boldsymbol{\varphi},\boldsymbol{\theta},\mathbf{z},\mathbf{w};\alpha,\beta)=\prod_{i=1}^{K}p(\varphi_i;\beta)\prod_{j=1}^{M}p(\theta_j;\alpha)\prod_{t=1}^{N}p(z_{j,t}\mid\theta_j)p(w_{j,t}\mid\varphi_{z_{j,t}})$$



$$\begin{aligned}
&p(\mathbf{z},\mathbf{w};\alpha,\beta)\\
&=\int_{\boldsymbol{\theta}}\int_{\boldsymbol{\varphi}}p(\boldsymbol{\varphi},\boldsymbol{\theta},\mathbf{z},\mathbf{w};\alpha,\beta)d\boldsymbol{\varphi}d\boldsymbol{\theta}\\
&=\int_{\boldsymbol{\theta}}\prod_{j=1}^{M}p(\theta_j;\alpha)\prod_{t=1}^{N}p(z_{j,t}\mid\theta_j)d\boldsymbol{\theta}\int_{\boldsymbol{\varphi}}\prod_{i=1}^{K}p(\varphi_i;\beta)\prod_{j=1}^{M}\prod_{t=1}^{N}p(w_{j,t}\mid\varphi_{z_{j,t}})d\boldsymbol{\varphi}\\
&=\prod_{j=1}^{M}\int_{\theta_j}p(\theta_j;\alpha)\prod_{t=1}^{N}p(z_{j,t}\mid\theta_j)d\theta_j\prod_{i=1}^{K}\int_{\varphi_i}p(\varphi_i;\beta)\prod_{j=1}^{M}\prod_{t=1}^{N}p(w_{j,t}\mid\varphi_{z_{j,t}})d\varphi_i
\end{aligned}$$

$$\begin{aligned}
&\prod_{j=1}^{M}\int_{\theta_j}p(\theta_j;\alpha)\prod_{t=1}^{N}p(z_{j,t}\mid\theta_j)d\theta_j\prod_{i=1}^{K}\int_{\varphi_i}p(\varphi_i;\beta)\prod_{j=1}^{M}\prod_{t=1}^{N}p(w_{j,t}\mid\varphi_{z_{j,t}})d\varphi_i\\
&=\prod_{j=1}^{M}\int_{\theta_j}\frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K}\theta_{j,i}^{\alpha_i-1+n_{j,(\cdot)}^{i}}d\theta_j\prod_{i=1}^{K}\int_{\varphi_i}\frac{\Gamma\left(\sum_{r=1}^{V}\beta_r\right)}{\prod_{r=1}^{V}\Gamma(\beta_r)}\prod_{r=1}^{V}\varphi_{i,r}^{\beta_r-1+n_{(\cdot),r}^{i}}d\varphi_i\\
&=\prod_{j=1}^{M}\frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\frac{\prod_{i=1}^{K}\Gamma(n_{j,(\cdot)}^{i}+\alpha_i)}{\Gamma\left(\sum_{i=1}^{K}(n_{j,(\cdot)}^{i}+\alpha_i)\right)}\prod_{i=1}^{K}\frac{\Gamma\left(\sum_{r=1}^{V}\beta_r\right)}{\prod_{r=1}^{V}\Gamma(\beta_r)}\frac{\prod_{r=1}^{V}\Gamma(n_{(\cdot),r}^{i}+\beta_r)}{\Gamma\left(\sum_{r=1}^{V}(n_{(\cdot),r}^{i}+\beta_r)\right)}
\end{aligned}$$

$n_{j,r}^{i}$: i-th トピック中の（辞書中）r-th 単語が j-th 文書に現れた回数

$$\int_{\theta_j}\frac{\Gamma\left(\sum_{i=1}^{K}(n_{j,(\cdot)}^{i}+\alpha_i)\right)}{\prod_{i=1}^{K}\Gamma(n_{j,(\cdot)}^{i}+\alpha_i)}\prod_{i=1}^{K}\theta_{j,i}^{\alpha_i-1+n_{j,(\cdot)}^{i}}d\theta_j=1 \quad\Longleftarrow\quad \int\frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K}x_i^{\alpha_i-1}dx=1$$

$$\begin{aligned}
&p(z_{(m,n)}=k\mid\mathbf{z}_{-(m,n)},\mathbf{w};\alpha,\beta)\\
&\propto p(z_{(m,n)}=k,\mathbf{z}_{-(m,n)},\mathbf{w};\alpha,\beta)\\
&=\prod_{j=1}^{M}\frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\frac{\prod_{i=1}^{K}\Gamma(n_{j,(\cdot)}^{i}+\alpha_i)}{\Gamma\left(\sum_{i=1}^{K}(n_{j,(\cdot)}^{i}+\alpha_i)\right)}\prod_{i=1}^{K}\frac{\Gamma\left(\sum_{r=1}^{V}\beta_r\right)}{\prod_{r=1}^{V}\Gamma(\beta_r)}\frac{\prod_{r=1}^{V}\Gamma(n_{(\cdot),r}^{i}+\beta_r)}{\Gamma\left(\sum_{r=1}^{V}(n_{(\cdot),r}^{i}+\beta_r)\right)}\\
&=\left(\frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\right)^{M}\prod_{j=1,j\neq m}^{M}\frac{\prod_{i=1}^{K}\Gamma(n_{j,(\cdot)}^{i}+\alpha_i)}{\Gamma\left(\sum_{i=1}^{K}(n_{j,(\cdot)}^{i}+\alpha_i)\right)}\\
&\quad\times\left(\frac{\Gamma\left(\sum_{r=1}^{V}\beta_r\right)}{\prod_{r=1}^{V}\Gamma(\beta_r)}\right)^{K}\prod_{i=1}^{K}\prod_{r=1,r\neq v}^{V}\Gamma(n_{(\cdot),r}^{i}+\beta_r)\\
&\quad\times\frac{\prod_{i=1}^{K}\Gamma(n_{m,(\cdot)}^{i}+\alpha_i)}{\Gamma\left(\sum_{i=1}^{K}(n_{m,(\cdot)}^{i}+\alpha_i)\right)}\prod_{i=1}^{K}\frac{\Gamma(n_{(\cdot),v}^{i}+\beta_v)}{\Gamma\left(\sum_{r=1}^{V}(n_{(\cdot),r}^{i}+\beta_r)\right)}\\
&\propto\prod_{i=1}^{K}\Gamma(n_{m,(\cdot)}^{i}+\alpha_i)\prod_{i=1}^{K}\frac{\Gamma(n_{(\cdot),v}^{i}+\beta_v)}{\Gamma\left(\sum_{r=1}^{V}(n_{(\cdot),r}^{i}+\beta_r)\right)}
\end{aligned}$$

$$p(z_{(m,n)} = k \mid \mathbf{z}_{-(m,n)}, \mathbf{w}; \alpha, \beta)$$

$$\propto \prod_{i=1}^{K} \Gamma(n^{i}_{m,(\cdot)} + \alpha_i) \prod_{i=1}^{K} \frac{\Gamma(n^{i}_{(\cdot),v} + \beta_v)}{\Gamma\left(\sum_{r=1}^{V} (n^{i}_{(\cdot),r} + \beta_r)\right)}$$

$$\propto \prod_{i=1}^{K} \Gamma(n^{i,-(m,n)}_{m,(\cdot)} + \alpha_i) \prod_{i=1}^{K} \frac{\Gamma(n^{i,-(m,n)}_{(\cdot),v} + \beta_v)}{\Gamma\left(\sum_{r=1}^{V} (n^{i,-(m,n)}_{(\cdot),r} + \beta_r)\right)}$$

$$\times (n^{k,-(m,n)}_{m,(\cdot)} + \alpha_k) \frac{n^{k,-(m,n)}_{(\cdot),v} + \beta_v}{\sum_{r=1}^{V} (n^{k,-(m,n)}_{(\cdot),r} + \beta_r)}$$

$$\propto (n^{k,-(m,n)}_{m,(\cdot)} + \alpha_k) \frac{n^{k,-(m,n)}_{(\cdot),v} + \beta_v}{\sum_{r=1}^{V} (n^{k,-(m,n)}_{(\cdot),r} + \beta_r)}$$

Collapsed Gibbs サンプリング

- Dirichlet 分布と多項分布の双対性を用い，連続値パラメータを積分消去する

$$P(\mathbf{z}) = \int P(\mathbf{z} \mid \Theta) p(\Theta) d\Theta = \prod_{d=1}^{M} \frac{\prod_k \Gamma(n^{(d)}_k + \alpha)}{\Gamma(\alpha)^T} \frac{\Gamma(T\alpha)}{\Gamma(\sum_k n^{(d)}_k + \alpha)}$$

$$P(\mathbf{w} \mid \mathbf{z}) = \int P(\mathbf{w} \mid \mathbf{z}, \Phi) p(\Phi) d\Phi = \prod_{k=1}^{K} \frac{\prod_w \Gamma(n^{(k)}_w + \beta)}{\Gamma(\beta)^W} \frac{\Gamma(W\beta)}{\Gamma(\sum_w n^{(k)}_w + \beta)}$$

$$P(\mathbf{z} \mid \mathbf{w}) = \frac{P(\mathbf{w} \mid \mathbf{z}) P(\mathbf{z})}{\sum_{\mathbf{z}} P(\mathbf{w} \mid \mathbf{z}) P(\mathbf{z})}$$

$$\propto P(\mathbf{w} \mid \mathbf{z}) P(\mathbf{z})$$

$n^{(d)}_k$ は文書 d 中のトピック k の出現回数
$n^{(k)}_w$ は単語wのトピック k としての出現回数
M は文書数
W は異なり単語数
K はトピック数

- zの更新式をこれから求める

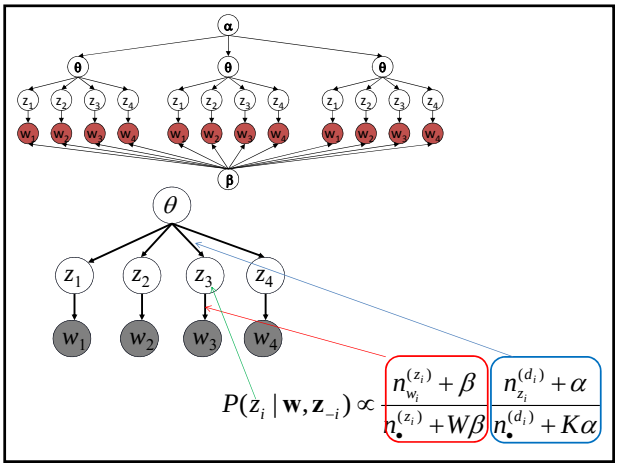
Collapsed Gibbs サンプリング

- 各 z_i を、次の $\mathbf{z}_{-i}$ で条件づけた分布でサンプルする

$$P(z_i \mid \mathbf{w}, \mathbf{z}_{-i}) \propto \frac{n^{(d_i)}_{z_i} + \alpha}{n^{(d_i)}_{\bullet} + K\alpha} \frac{n^{(z_i)}_{w_i} + \beta}{n^{(z_i)}_{\bullet} + W\beta} \propto (n^{(d_i)}_{z_i} + \alpha) \frac{n^{(z_i)}_{w_i} + \beta}{n^{(z_i)}_{\bullet} + W\beta}$$

$n^{(d)}_k$ は文書 d 中のトピック k の出現回数
$n^{(k)}_w$ は単語wのトピック k としての出現回数
d_i は文書中の i-th 単語が属する文書のID
w_i は文書中の i-th 単語の単語ID
z_i は文書中の i-th 単語のトピックID
M は文書数
W は異なり単語数
K はトピック数

- 容易に実行可能:
  - メモリ: 数え上げは、2個の疎行列で行える
  - 最適化: 特殊な関数はいらない，単純な算術
  - **z** と **w** が与えられれば Φ と Θ の分布は求めることができる



パラメータ推定

$$\hat{\theta}^{(d)}_j = \frac{n^{(d)}_j + \alpha}{n^{(d)}_{\bullet} + K\alpha} \qquad \hat{\theta}_{j,k} = \frac{n^{k}_{j,(\cdot)} + \alpha}{n^{(\cdot)}_{j,(\cdot)} + K\alpha}$$

$$\hat{\phi}^{(w)}_j = \frac{n^{(j)}_w + \beta}{n^{(j)}_{\bullet} + W\beta} \qquad \hat{\varphi}_{k,i} = \frac{n^{k}_{(\cdot),i} + \beta}{n^{k}_{(\cdot),(\cdot)} + W\beta}$$

$n^{(d)}_j$ は文書 d 中のトピック j の出現回数
$n^{(j)}_w$ は単語wのトピック j としての出現回数
W は異なり単語数
K はトピック数

$n^{i}_{j,r}$: i-th トピック中の（辞書中）r-th 単語が j-th 文書に現れた回数