統計数理研究所 H24年度公開講座

「確率的トピックモデル」

持橋大地 (統計数理研究所)
石黒勝彦 (NTTコミュニケーション科学基礎研究所)

2013/1/15-16 統計数理研究所 会議室1

本講座の構成

- 1日目: トピックモデルの基礎
  - トピックモデルとは, Naïve Bayes, PLSI, LDA
  - EMアルゴリズム, VB-EMアルゴリズム, Gibbsサンプラー, 他のモデルとの関係
- 2日目: トピックモデルの応用
  - 複雑なトピックモデル、時系列モデル
  - 画像、音声、ネットワークデータ
  - 半教師あり学習、補助情報あり学習

無限モデル(ノンパラメトリックベイズ)は本講座では扱わない

講義予定

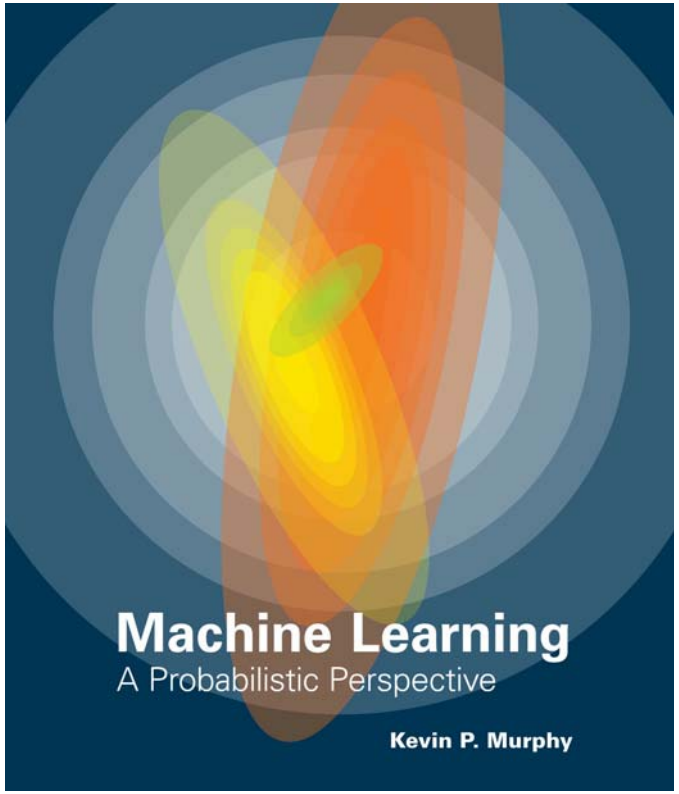
- 1日目
  - AM/ 導入, LSI, ナイーブベイズ, PLSI, EMアルゴリズム
  - PM/ LDA, Gibbs, VB-EMアルゴリズム, GaP, NMF, Boltzmann Machine
- 2日目
  - AM/ 時系列モデル, 共分散モデル, 半教師あり学習, 共変量学習
  - PM/ 画像、音声、ネットワーク等への応用例
- 両日とも、10:00-12:00, 13:00-16:00 が基本

統数研図書室

- 本公開講座の開講中、1Fの統数研図書室で、本講座に関係する参考図書および参考文献を提示していただいています

参考図書

C.M.Bishop, “パターン認識と機械学習” (上)(下), Springer/丸善, 2007-2008
http://ibisforest.org/index.php?PRML



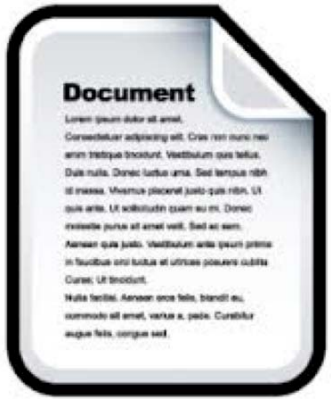
Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
http://www.cs.ubc.ca/~murphyk/MLbook/

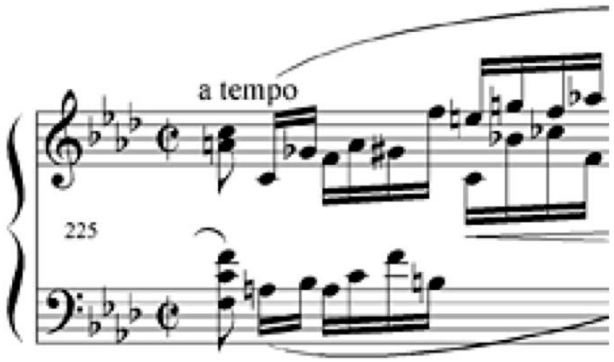
Lecture 1

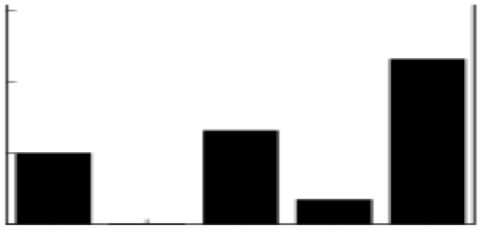
トピックモデルとは

トピックモデルとは

- 様々な(離散)データに隠れた潜在的なトピックを推定するモデル
  - トピック‥話題, 分野など, 大ざっぱな「意味」のようなもの







トピックモデルとは (2)

- このテキストは, 大体どういう内容?
- このテキスト群の中には、どんな「話題」がある?
- この単語は、大体どんな意味?
- ・・・・

を、人が一切教えることなく、データから自動的に学習したい

トピックモデルの学習例

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

(Blei+ 2003)より

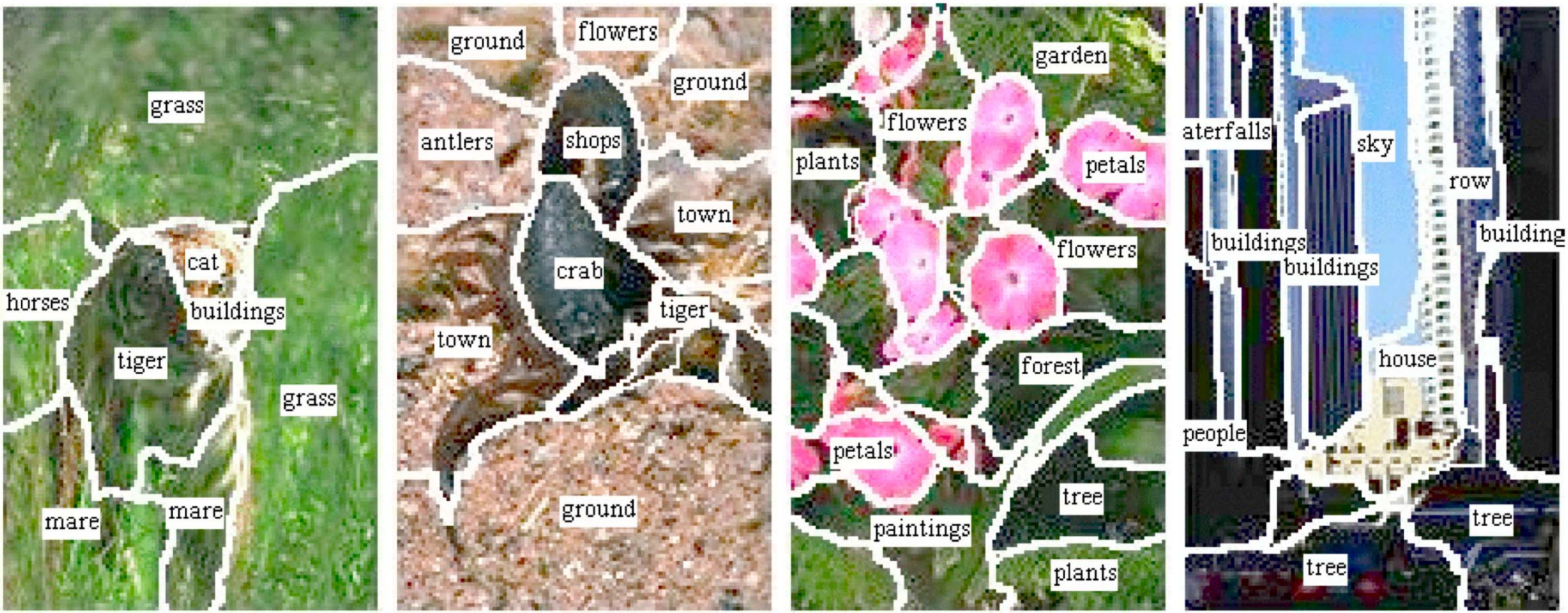
- コーパスから完全に自動的に関連語を抽出できる

トピックモデルの学習例 (2)　(Blei+ 2003)より

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

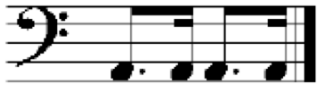
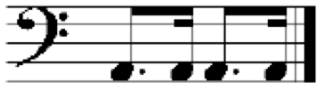
- それぞれの語がどんな「話題」(トピック)に属するのかが、大量のデータだけからわかる
  - 実際には、話題の確率がわかる
- 文書=人、単語=その人の買った商品、URL、・・・・

トピックモデルの学習例 (画像)



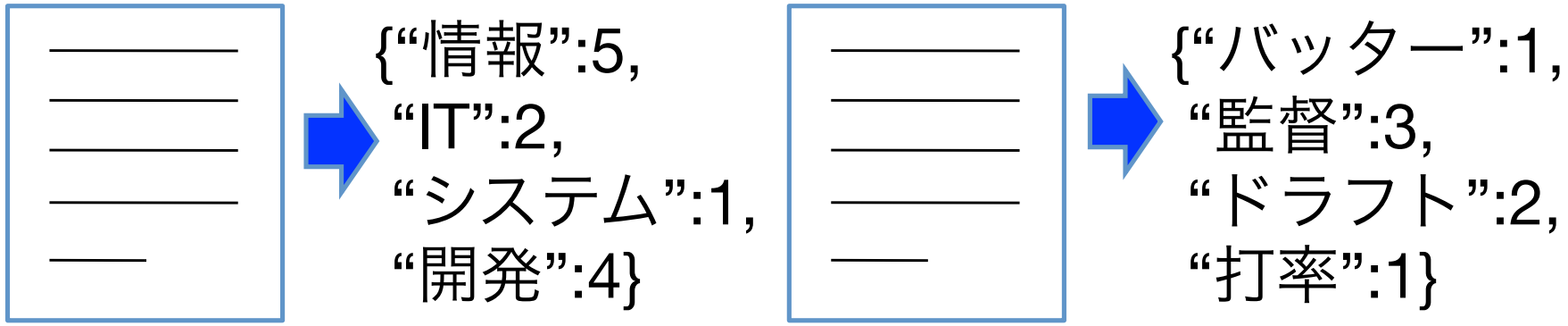
- 画像の領域が何を表しているかを統計的に学習 (Barnard+ 2003より)

トピックモデルの学習例 (音楽)

QUERY	RETRIEVED SONGS
come on, come on, get down	*Erksine Hawkins – Tuxedo Junction* *Moby – Bodyrock* *Nine Inch Nails – Last* *Sherwood Schwartz – 'The Brady Bunch' theme song*
	*The Beatles – Got to Get You Into My Life* *The Beatles – I'm Only Sleeping* *The Beatles – Yellow Submarine* *Moby – Bodyrock* *Moby – Porcelain* *Gary Portnoy – 'Cheers' theme song* *Rodgers & Hart – Blue Moon*
	*Moby – Bodyrock*
come on, come on, get down	

- 歌詞や楽譜の断片から、その潜在トピックを通じて音楽を検索 (Brochu+ 2003) より

“Bag of Words” 表現



- テキストを、含まれる単語とその頻度だけで表す
  - 単語の順番は無視
  - 厖大な語彙の中で、一文書に現れる単語はそのごく一部 (～数100種類程度)

Bag of Words の行列表現

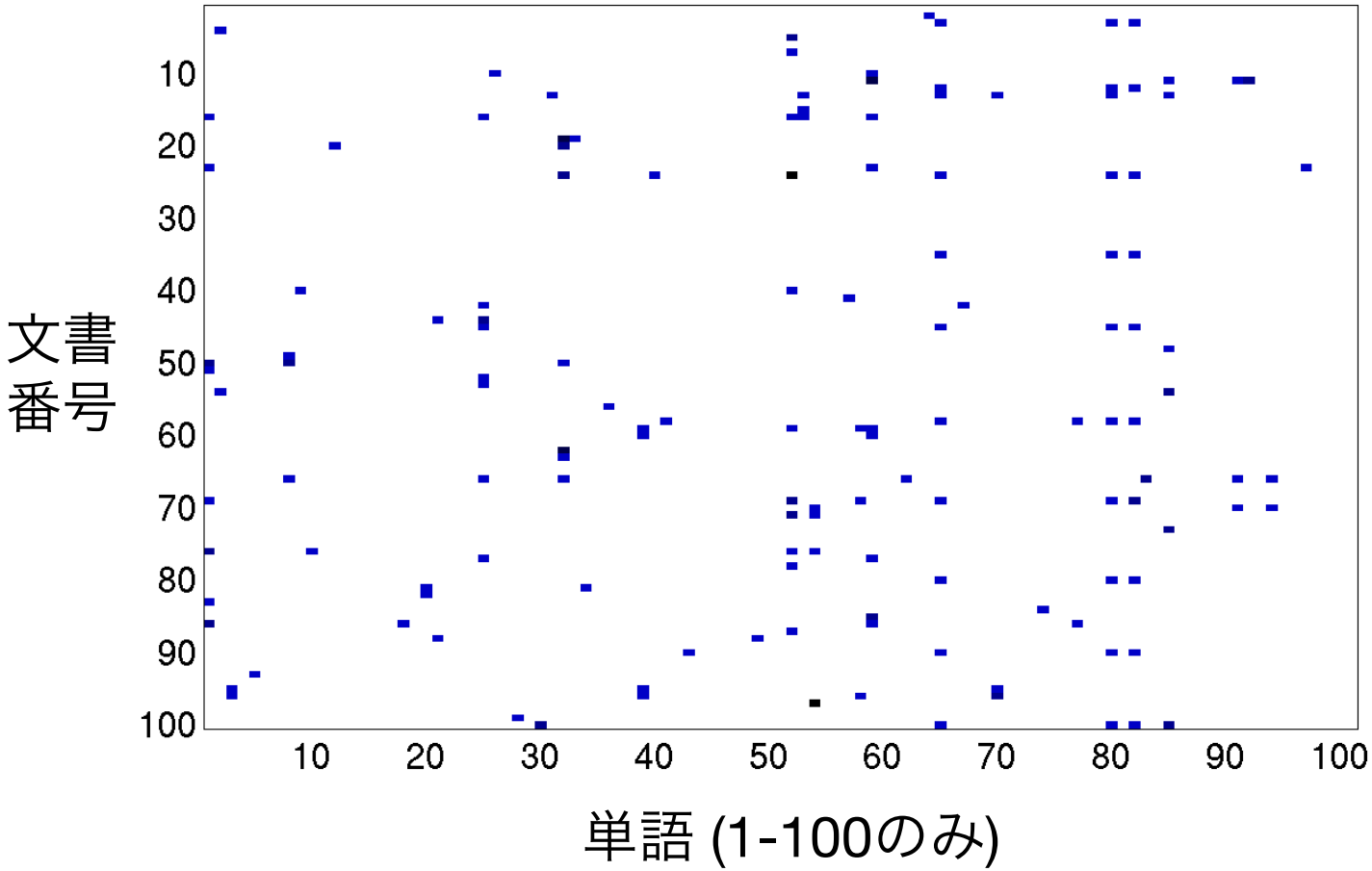
$$d_1 \to \{w_1 : 1, w_5 : 2, w_{27} : 1, w_{105} : 1\}$$
$$d_2 \to \{w_2 : 1, w_3 : 1, w_{15} : 4, w_{32} : 3\}$$

のような情報を行列で表してみる

$$N \overset{W}{\begin{pmatrix} & 1 & 1 & & & & \\ 1 & & & & & 1 & 1 \\ 2 & 4 & & & & & \\ & 2 & 1 & 1 & & & 1 \\ & & & & 1 & & \end{pmatrix}} = D$$

Bag of Words (実際のデータ)

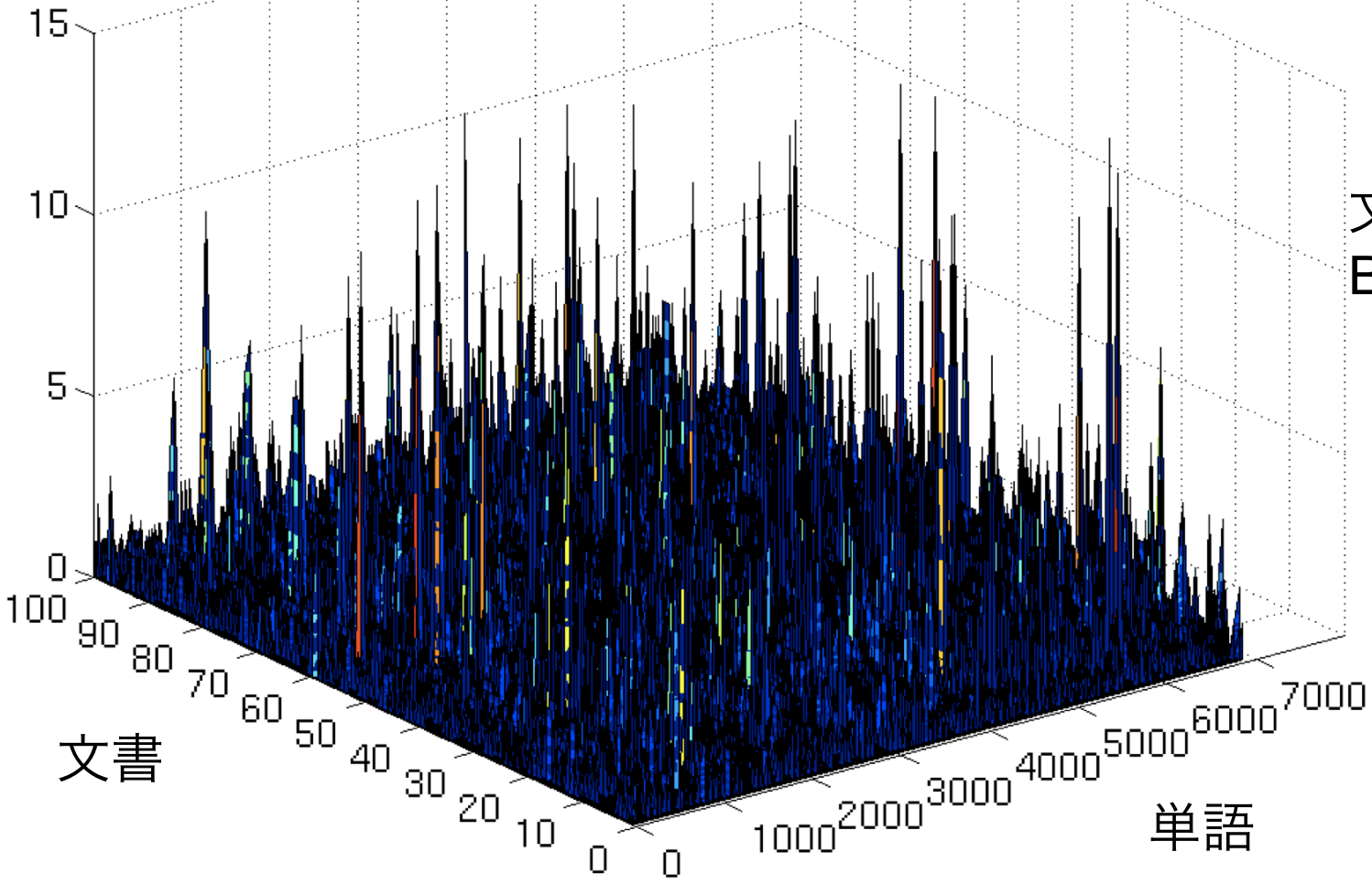
- DailyKOS (新聞)のよく使われるデータセット (一部)



- ほとんどの値は0
- 非負の値も1か、非常に小さい値

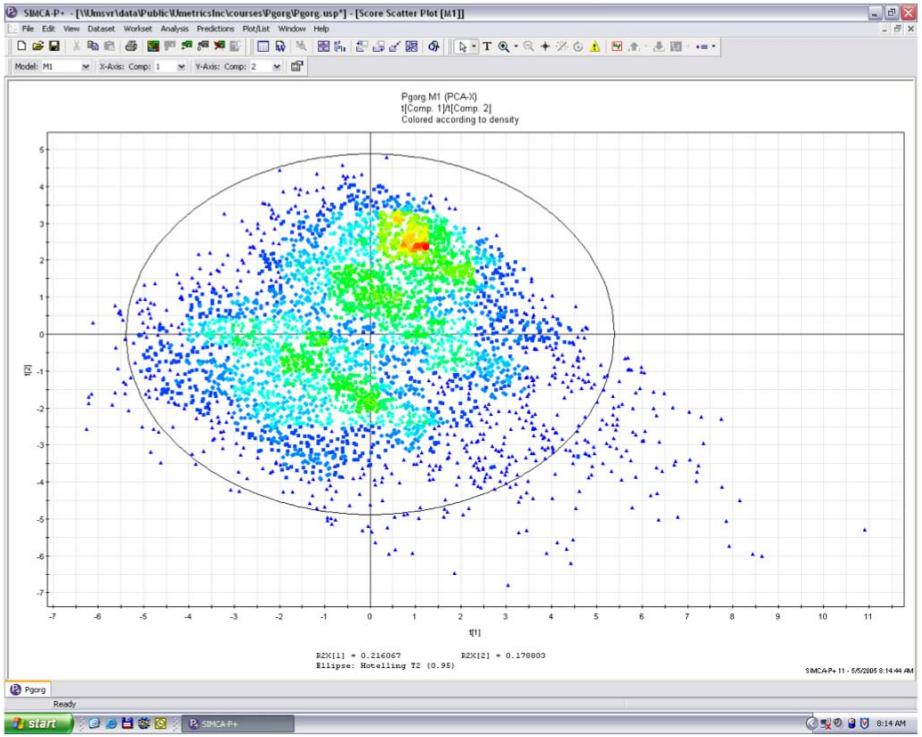
Bag of Words (実際のデータ)

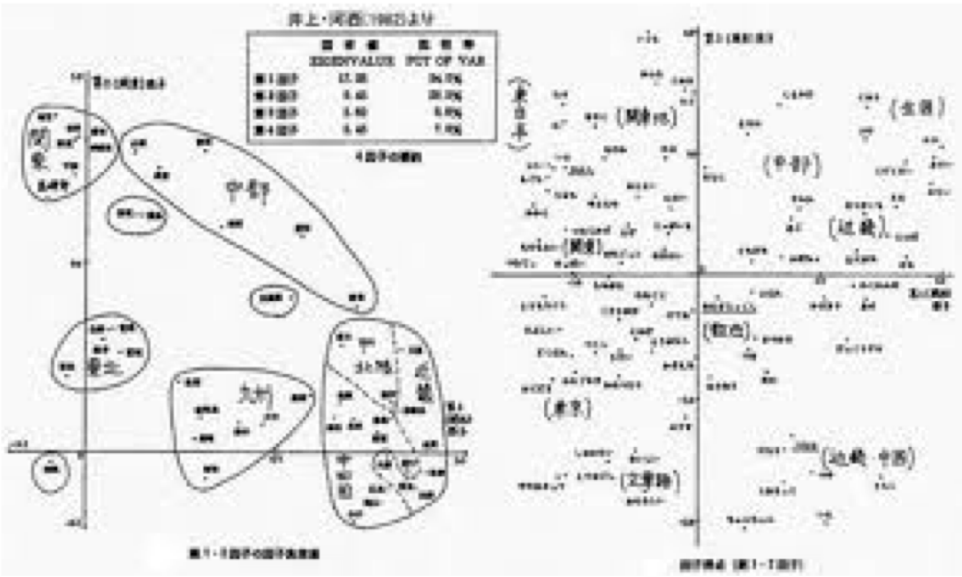
- DailyKOS (新聞)のよく使われるデータセット (一部)



文書1-100の
Bag of Words

古典的な分析法: 多変量解析



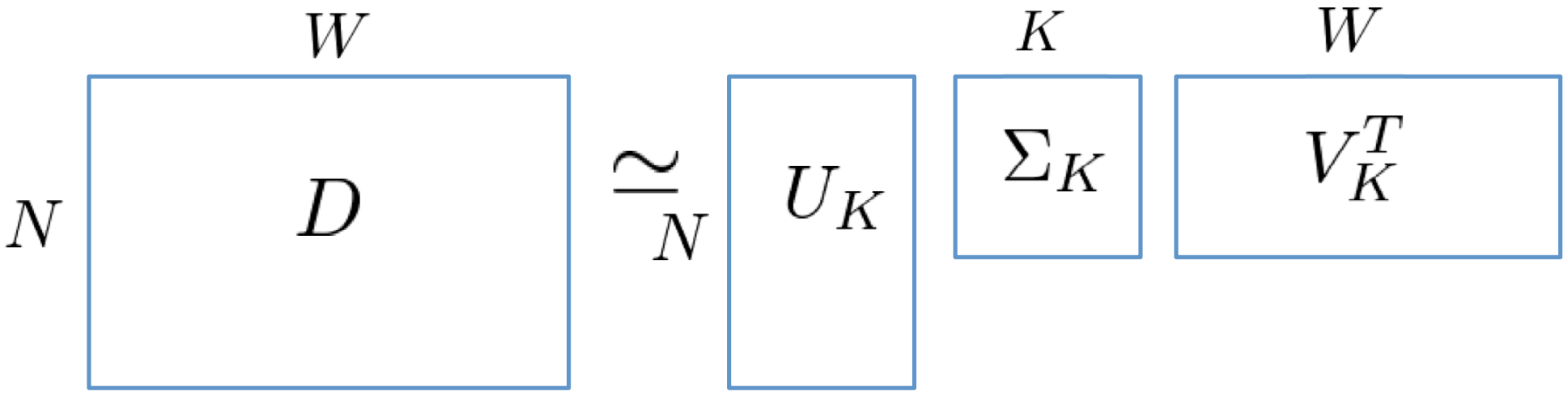


- 相関分析、回帰分析、判別分析、・・・・
  - 一部の変量を抽出して分析
  - 解析に漏れがある可能性が高い

古典的な分析法: 多変量解析 (2)

- ある人が、“開腹” (頻度1)、または “恢復” (頻度1) の語を使っていたとする
  - ものすごい情報！
    - この人は医者／文学者／老人／‥‥
  - 多変量解析では、こうした低頻度の情報を扱うのは難しい (隠れたガウス分布の仮定、変数の選択)
- トピックモデルは、ある意味で高次元離散データに適した多変量解析の方法 (午後)

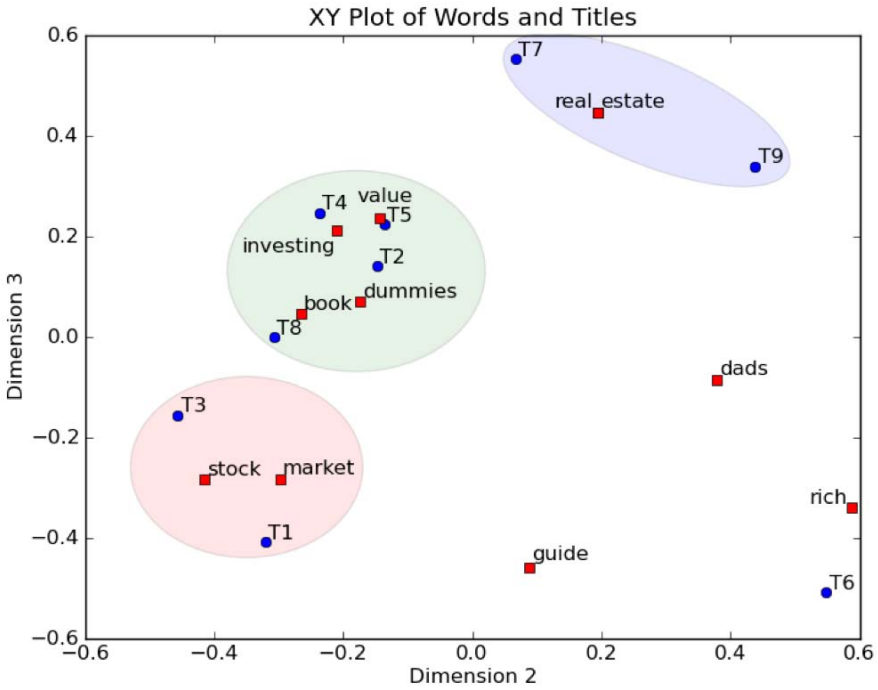
古典的な方法 (2)：主成分分析



- *D*をスペクトル分解‥*DD*^{T}の固有値と固有ベクトルで表す
  - 上位K個だけ使うことで、*D*を「Denoise」
  - 固有ベクトルが言葉の「意味」に対応?
    → LSI (Latent Semantic Indexing)

LSI (Latent Semantic Indexing)

- 情報検索の分野で、(Deerwester+ 1990)により提案
  - LSA (Latent Semantic Analysis)ともよばれる
  - 上位K個の固有値/固有ベクトルで、頻度行列を低次元に圧縮



LSI (2)

- 一見、素晴らしいように見えるが‥
- LSIの欠点
  - Kの設定がアドホック、拡張性がない
  - 負の値の意味付けがない
  - Implicitな、頻度のガウス分布の仮定
    (頻度は常に正で離散なのに!)

どういうこと??

LSIに隠れたガウス仮定

- Papadimitoriou+ (1998)による解析
  - 文書が1つのトピックのみなっている時、
  - LSIは同じトピックに属する文書ベクトルの
    コサイン距離を保存しつつ、ノイズを除ける

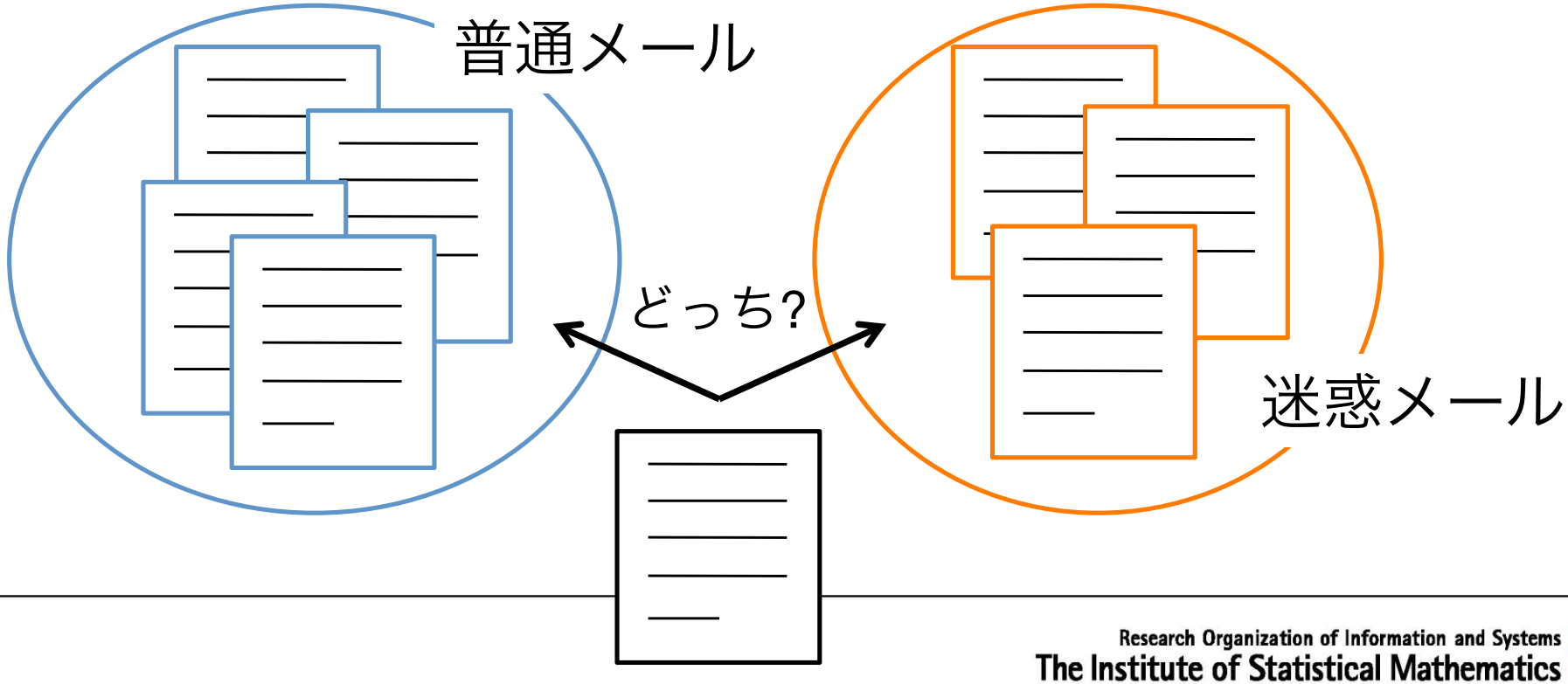
Implicitな仮定:
(1) 文書にトピックは1つしかない
(2) 文書ベクトルやノイズはガウス分布に従う
本当??

何がだめなのか?

- 観測データDの「由来」を説明していない
  - 回帰分析、PCA等は「後付け」の解析手法
  - 手法に発展性がない ← 性能も低い
- 離散、正の観測データを生み出すモデルを先に考えよう
  - モデルに従って、カウントが確率的に生成
  - モデルのパラメータθを推定
  - 複雑なモデルが作れる

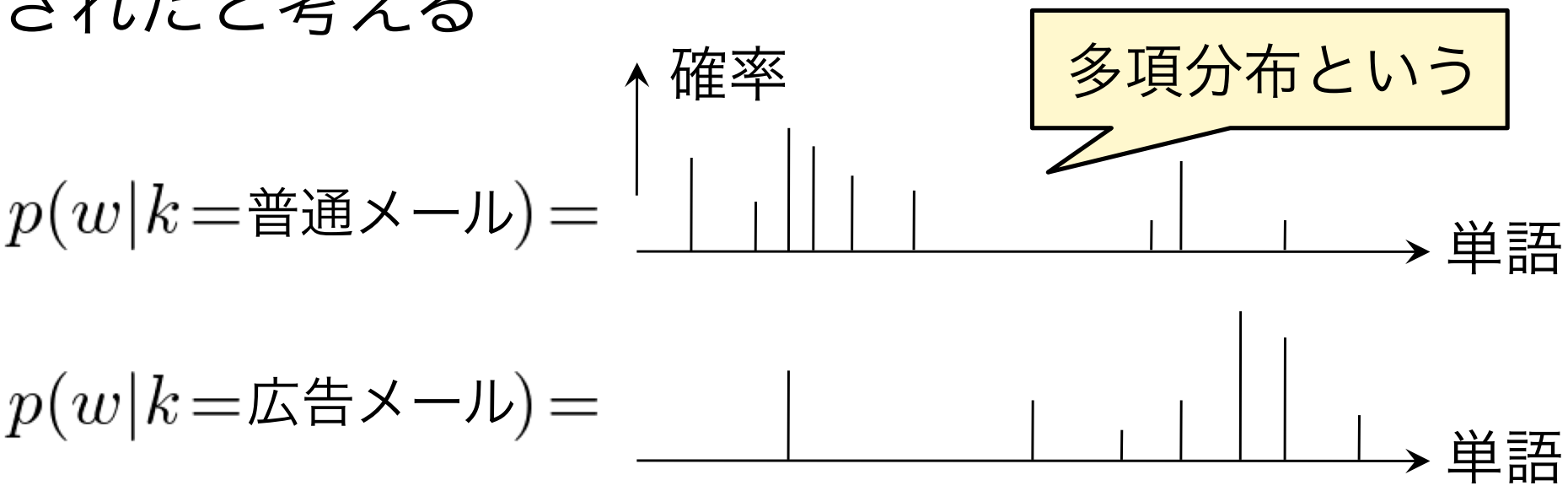
簡単な例: ナイーブベイズ

- ナイーブベイズ (Naïve Bayes)
  ‥テキストの非常に簡単な生成モデル
- いま、普通のメールと迷惑メールを分類する問題を考えよう



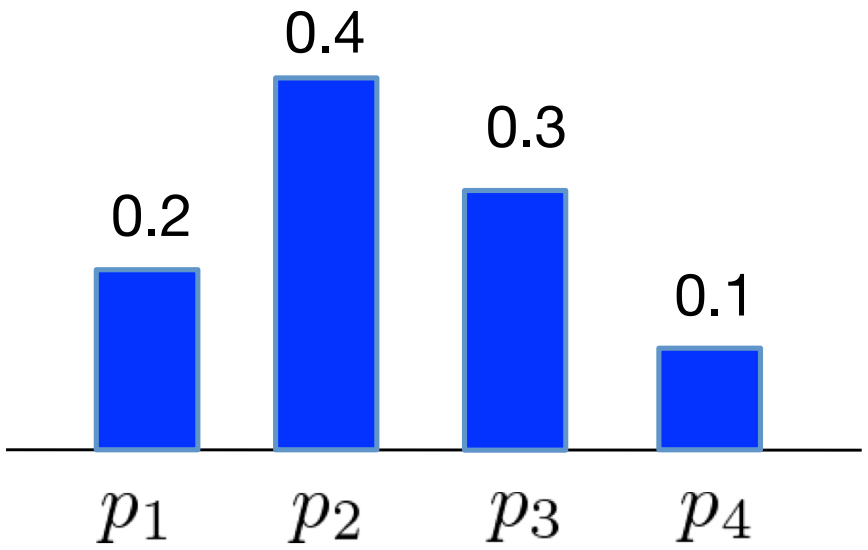
ナイーブベイズ (2)

- いま、カテゴリ$k \in$ {普通メール, 広告メール}毎にメールの単語が確率分布$p(w|k)$ から生成されたと考える



$$p(\mathbf{w}|k) = \prod_{n=1}^{N} p(w_n|k) \quad \text{(BOWの仮定)}$$

多項分布



$$\boldsymbol{p} = (p_1, p_2, \cdots, p_K)$$

$$p_k \geq 0, \ \sum_k p_k = 1$$

- アイテムiが確率p_iで出る離散分布
- 多項分布は本来,全部でn回のうちkがn_k回出る分布

$$\mathrm{Mult}(\boldsymbol{p}, n) = \frac{n!}{n_1! n_2! \cdots n_K!} \prod_k p_k^{n_k}$$

だが、$\mathrm{Mult}(\boldsymbol{p}, 1)$ のことを $\mathrm{Mult}(\boldsymbol{p})$ ともいう

ナイーブベイズ (3)

- 生成モデルとしては、
  (1) $k \sim p(k)$ からクラス $k \in \{0, 1\}$ を選択
    - たとえば、$p(k) = [0.9, 0.1]$
  (2) for *n* = 1 … *N*,
  *n*番目の単語 $w_n \sim p(w|k)$ を生成.

$$p(d, k) = p(k)p(d|k) = p(k) \prod_{n=1}^{N} p(w_n|k)$$

- パラメータ $p(k), p(w|k)$ は簡単に推定できる.

ナイーブベイズ (例)

$$D = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \end{array} \\ \begin{pmatrix} 1 & 2 & 1 & & 1 & & \\ & 2 & & & 1 & 1 & 1 \\ 1 & & 1 & 1 & & 2 & \end{pmatrix} \end{array}$$

普通メール

広告メール

- これから、
  - $p(k=0) = 2/3,\ p(k=1) = 1/3$
  - $p(w|k=0) = [0.1\ 0.4\ 0.1\ 0\ 0.2\ 0.1\ 0.1]$
    $p(w|k=1) = [0.2\ 0\ 0.2\ 0.2\ 0\ 0.4\ 0]$
- ex. $p(d_1, k=0) = \frac{2}{3}\left(\frac{1}{10}\cdot\frac{4}{10}\cdot\frac{4}{10}\cdot\frac{1}{10}\cdot\frac{2}{10}\right) = 2.13\times10^{-4}$

ナイーブベイズ (5)

- 新しいメール d をどちらに分類?
  → 確率 $p(k|d)$ が高い方に分類すればよい。
- $p(k|d)$ の計算?

確率の復習

	y_1	y_2	y_3	y_4
x_1	0.1	0.05	0	0.2
x_2	0	0.25	0.05	0.15
x_3	0	0.1	0.05	0.05

商品			
2	1	0	4
0	5	1	3
0	2	1	1

客

$$p(x_1) = 0.1 + 0.05 + 0 + 0.2 = 0.35$$

$$= p(x_1, y_1) + p(x_1, y_2) + p(x_1, y_3) + p(x_1, y_4)$$

$$= \sum_y p(x_1, y)$$

$$p(x) = \sum_y p(x, y)$$

(周辺化)

確率の復習 (2)

	y_1	y_2	y_3	y_4
x_1	0.1	0.05	0	0.2
x_2	0	0.25	0.05	0.15
x_3	0	0.1	0.05	0.05

$$p(x|y) = \frac{p(x,y)}{\sum_x p(x,y)}$$

(ベイズの定理)

$$p(x_2|y_2) = \frac{0.25}{0.05 + 0.25 + 0.1} = 0.625$$

$$= \frac{p(x_2, y_2)}{p(x_1, y_2) + p(x_2, y_2) + p(x_3, y_2)}$$

$$= \frac{p(x_2, y_2)}{\sum_x p(x, y_2)}$$

ベイズの定理、連鎖則

$$p(x|y) = \frac{p(x,y)}{\sum_x p(x,y)} = \frac{p(x,y)}{p(y)}$$

- これから、$p(x,y) = p(x|y)p(y)$ (確率の連鎖則)
- 「ベイズの定理」を覚える必要はない!
  (連鎖則から自明)

確率の復習 (3)

	y_1	y_2	y_3	y_4
x_1	0.1	0.05	0	0.2
x_2	0	0.25	0.05	0.15
x_3	0	0.1	0.05	0.05

$$p(x, y) = p(x|y)p(y)$$

(確率の連鎖則)

$$p(y_2) = 0.05 + 0.25 + 0.1 = 0.4$$

$$p(x_2, y_2) = p(x_2|y_2)\, p(y_2)$$

$$= 0.625 \times 0.4 = 0.25$$

ベイズの定理 (2)

$$p(x|y) = \frac{p(x,y)}{p(y)} \propto p(x,y)$$

	y_1	y_2	y_3	y_4
x_1	0.1	0.05	0	0.2
x_2	0	0.25	0.05	0.15
x_3	0	0.1	0.05	0.05

- xの条件つき確率 p(x|y) の計算では、p(y)は定数
  → p(x|y) は p(x,y) に比例
  - xに関して正規化すればよい

ベイズの定理 (3)

- $p(x|y) \propto p(x,y) = p(y|x)p(x)$
  - p(x|y) を p(y|x) で引っくり返せる！

	y_1	y_2	y_3	y_4
x_1	0.1	0.05	0	0.2
x_2	0	0.25	0.05	0.15
x_3	0	0.1	0.05	0.05

$$p(x_2|y_2) = 0.625$$
$$p(x_2) = 0.4$$
$$p(y_2|x_2) \propto 0.625 \times 0.4$$
$$= 0.25$$

y_2について正規化して、

$$p(y_2|x_2) = 0.5555$$

確率の復習 (まとめ)

- 確率の連鎖則、Chain rule

$$p(A, B) = p(A|B)p(B)$$

  - AとBが起こることは、まずBが起き、次にその下でAが起きることと同じ
- 確率の周辺化、Marginalization

$$p(A) = \sum_B p(A, B)$$

  - Aの確率は、同時に起こるBの可能性について和をとったもの

ベイズの定理

- $p(A,B) = p(A|B)p(B) = p(B|A)p(A)$ より、

$$p(A|B) = \frac{p(A,B)}{p(B)}$$

$$\propto p(A,B) = p(B|A)p(A)$$

$p(A|B)$ を $p(B|A)$ で書ける！！

  - $p(B)$は、Aについての和を1にする正規化定数
  - 条件つき確率を引っくり返せる!!

ナイーブベイズ (5)

- 新しいメール d をどちらに分類?
  → 確率 $p(k|d)$ が高い方に分類すればよい。
- $p(k|d)$ の計算?

$$p(k|d) \propto p(d|k)p(k)$$

  - $p(k)$ も $p(d|k) = \prod_{w \in d} p(w|k)$ もわかっている!

ナイーブベイズ (6)

- 前の例で、d=$\{w_1, w_6\}$ のとき、これは何メール?

$$p(k|d) \propto p(d|k)p(k)$$

$$= \begin{cases} \frac{2}{3} \cdot \left(\frac{1}{10} \cdot \frac{1}{10}\right) = \frac{2}{3} \times 10^{-2} \\ \frac{1}{3} \cdot \left(\frac{2}{10} \cdot \frac{4}{10}\right) = \frac{8}{3} \times 10^{-2} \end{cases} \propto \begin{cases} 0.2 \\ 0.8 \end{cases}$$

- よって、$p(k|d) = [0.2, 0.8]$
  → dは広告メール.

事前分布と事後分布

- 今の例で、

$$p(k) = [0.67,\ 0.33]$$

← データの確率 $p(d|k)$

$$p(k|d) \propto p(d|k)p(k) = [0.2,\ 0.8]$$

  - 事前分布 *p(k)* が、データの確率(尤度) *p(d|k)* により、事後分布 *p(k|d)*∝*p(d|k)p(k)* になった!

- ベイズ統計の基本:

$$p(\theta|D) \propto p(D|\theta)\, p(\theta)$$

  - (事後分布) ∝ (尤度)×(事前分布)

さて,

- 今までは、テキスト(メール)に {普通,広告} のようなカテゴリが与えられている場合を考えてきた

  ⇩

  実際にはそんなことはほとんどない!

Unigram Mixtures (Nigam+ 2000)

- ナイーブベイズの式

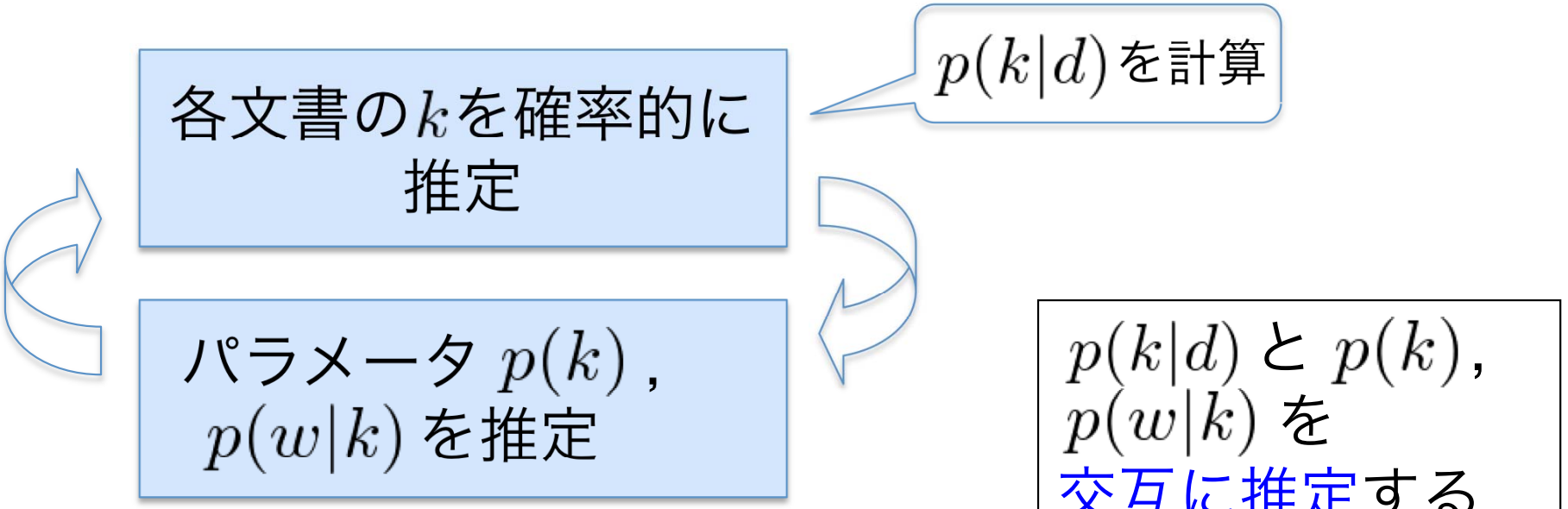
$$p(d,k) = p(k) \prod_{n=1}^{N} p(w_n|k)$$

において、kを推定すべき変数(潜在変数)とする

$$p(d) = \sum_{k} p(k) \prod_{n=1}^{N} p(w_n|k)$$

  - kに関して和をとって周辺化

Unigram Mixtures (2)



- 手順:
  (0) p(k|d) を乱数で設定.
  (1) p(k|d) から、p(k), p(w|k) を推定.
  (2) p(k) p(w|k) から、p(k|d) を再推定.
  (3) 収束していなければ、goto (1).

Unigram Mixtures (3)

$$\begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \end{array} \\ \begin{pmatrix} 1 & & 1 & & 2 & & 1 \\ & 1 & 2 & & 2 & & \\ 1 & 1 & & 1 & & 2 & \end{pmatrix} \end{array}$$

(0) $p(k|d_1) = [0.4, 0.6], p(k|d_2) = [0.6, 0.4], p(k|d_3) = [0.4, 0.6]$
から始める.

(1) $p(k) \propto \sum_d p(k|d)$

直感的には、各文書がその
トピックについて持つ確率の和

(導出は後で)

$$= \begin{cases} 0.4 + 0.6 + 0.4 = 1.4 \\ 0.6 + 0.4 + 0.6 = 1.6 \end{cases} \propto \begin{cases} 0.467 \\ 0.533 \end{cases}$$

Unigram Mixtures (4)

(2) $p(w|k) \propto \sum_d p(z|d)n(d,w)$

文書*d*と単語*w*が共起した頻度 n(d,w) を、*d*がもつトピック確率 *p(z|d)* で重みづけしたものの総和

より、

(導出は後で)

$$
\begin{aligned}
p(w_1|k=0) &\propto 0.4 + 0.4 &&= 0.8 \\
p(w_2|k=0) &\propto 0.6 + 0.4 &&= 1.0 \\
p(w_3|k=0) &\propto 0.4 + 0.6 \times 2 &&= 1.6 \\
&\vdots \\
p(w_7|k=0) &\propto 0.4 &&= 0.4
\end{aligned}
$$

Unigram Mixtures (5)

- 正規化して、

$$p(w|k=0) = \begin{pmatrix} 0.114 \\ 0.143 \\ 0.229 \\ 0.057 \\ 0.286 \\ 0.114 \\ 0.057 \end{pmatrix}$$

同様に、

$$p(w|k=1) = \begin{pmatrix} 0.150 \\ 0.125 \\ 0.175 \\ 0.075 \\ 0.250 \\ 0.150 \\ 0.075 \end{pmatrix}$$

Unigram Mixtures (6)

- $p(k), p(w|k)$ から、$p(k|d)$ が計算できる

$$p(k|d) \propto p(k,d)$$
$$= p(d|k)p(k) = p(k)\prod_{w \in d} p(w|k)$$

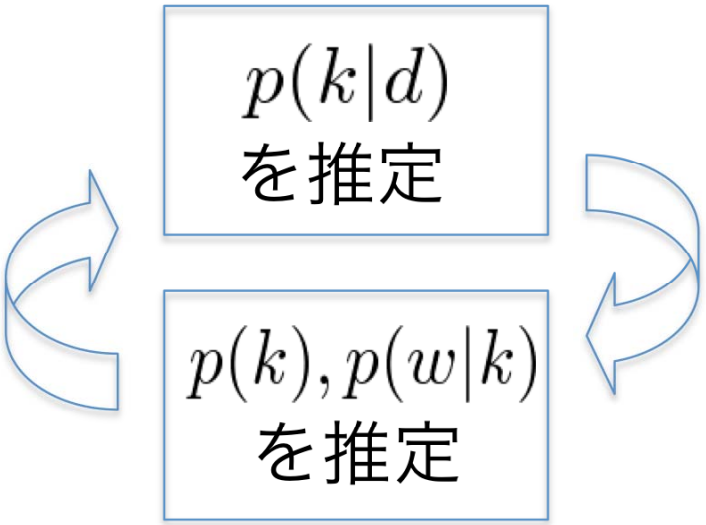
- 具体的に計算する

$$p(k=0|d_1) \propto 0.467 \times 0.114 \times 0.229 \times 0.286^2 \times 0.057$$
$$= 5.684 \times 10^{-5}$$
$$p(k=1|d_1) \propto 0.533 \times 0.150 \times 0.175 \times 0.250^2 \times 0.075$$
$$= 6.558 \times 10^{-5}$$
$$\therefore \ p(k|d_1) = (0.464, 0.536)$$

  - *p(k|d1)*が(0.4,0.6)→(0.464,0.536)に更新された!

Unigram Mixtures (7)

- *p(k|d)*が更新されたので、また *p(k)*, *p(w|k)* が求められる
- 以下繰り返し
- 実際にやってみると、



一種のクラスタリング

Unigram Mixtures (8)

	d_1		d_2		d_3	
Step	k=0	k=1	k=0	k=1	k=0	k=1
1	0.4000	0.6000	0.6000	0.4000	0.4000	0.6000
2	0.4642	0.5358	0.6902	0.3098	0.2520	0.7480
3	0.7034	0.2966	0.8746	0.1254	0.0304	0.9696
4	0.9797	0.0203	0.9924	0.0076	0.0000	1.0000
5	1.0000	0.0000	1.0000	0.0000	0.0000	1.0000

- 計算を続けると、
  - ステップ t=5 程度で収束
  - ほぼ何も教えていないのに分類できた!!

Unigram Mixtures (9)

- Unigram Mixtures‥1つの文書に潜在トピックが1つある、最も簡単なトピックモデル
  - 離散データのクラスタリング
- パラメータ:
  - *p(k)*：トピックkの事前分布
  - *p(w|k)*：トピックkから出る単語の分布

Unigram Mixtures (例)

- 毎日新聞2001年度のテキスト(一部)から計算したUMのトピック別単語分布p(w|k)の上位特徴語

Topic 2
の,円,億,する,を,は,
生産,に,など,年度,
兆,万,約,#,削減,事業,
予算,や,化,計画,販売,
いる,費,旅行,国内,工場,
なる,減,グループ,から,
機,月,USJ,向け,会社,
同社,開業,年間,発表,
赤字,統合

Topic 3
社長,を,た,さ,月,
発泡,酒,容疑,年,者,
れ,相,藤,氏,首相,会,
は,化,秋山,検出,
市原,石川,辞任,社,
取締役,出身,就任,
から,灯油,アサヒ

Topic 4
の,を,米,テロ,米国,
する,パキスタン,同時,
インド,アフガニスタン,
タリバン,し,支援,へ,
多発,政府,アフガン,
国,いる,ドル,政権,
経済,組織,国際,金融,
資金,攻撃,IMF,など,
協議

Unigram Mixtures (例)

- 毎日新聞2001年度のテキスト(一部)から計算したUMのトピック別単語分布p(w|k)の上位特徴語

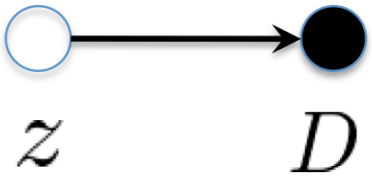
Topic 5
の,を,に,する,細胞,
など,船,レーザー,
ブロック,し,こと,
靴,から,や,な,型,
銀河,融合,核,足,
研究,状,宇宙,評価,
が,方法,サイズ,不審,
物質,高速,なる,ず,
意見,建造,グループ,星

Topic 10
た,し,に,と,が,て,
者,い,処分,こと,
は,生徒,れ,人,さ,
を,教委,問題,府,
女子,男性,被害,
保護,県,生活,保険,
など,ない,浪人,
よる,あっ,教職員

Topic 100
た,さん,て,で,容疑,
い,調べ,ごろ,と,捜査,
署,市,れ,時,者,事件,
が,いる,し,逮捕,午後,
男,み,県,県警,分,
男性,本部,殺人,いう,
から,午前,町,車,
同署,人,員,死亡,疑い,
乗用車,女性,府警

Unigram Mixtures: EMアルゴリズム

- 今の計算はどうして収束するのか?
  → EMアルゴリズム.



- パラメータ θ の下で、隠れ変数zを持つモデルについて、データの確率

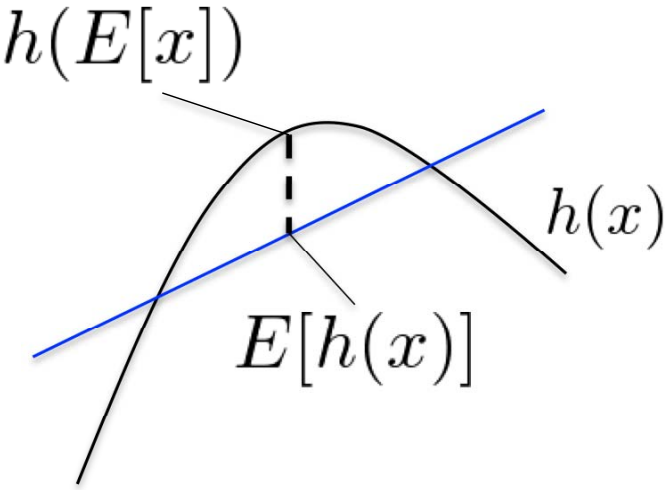
$$p(D|\theta) = \int p(D, z|\theta) dz$$

を最大化したい.

準備

- Jensenの不等式
  - 上に凸な関数 h(x) について,

$$h(E[x]) \geq E[h(x)]$$



log(x)は上に凸なので,

$$\log \int p(x) f(x) dx \geq \int p(x) \log f(x) dx$$

- KLダイバージェンス

$$D(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx \geq 0 \,.$$

  - p=qのときに等号成立

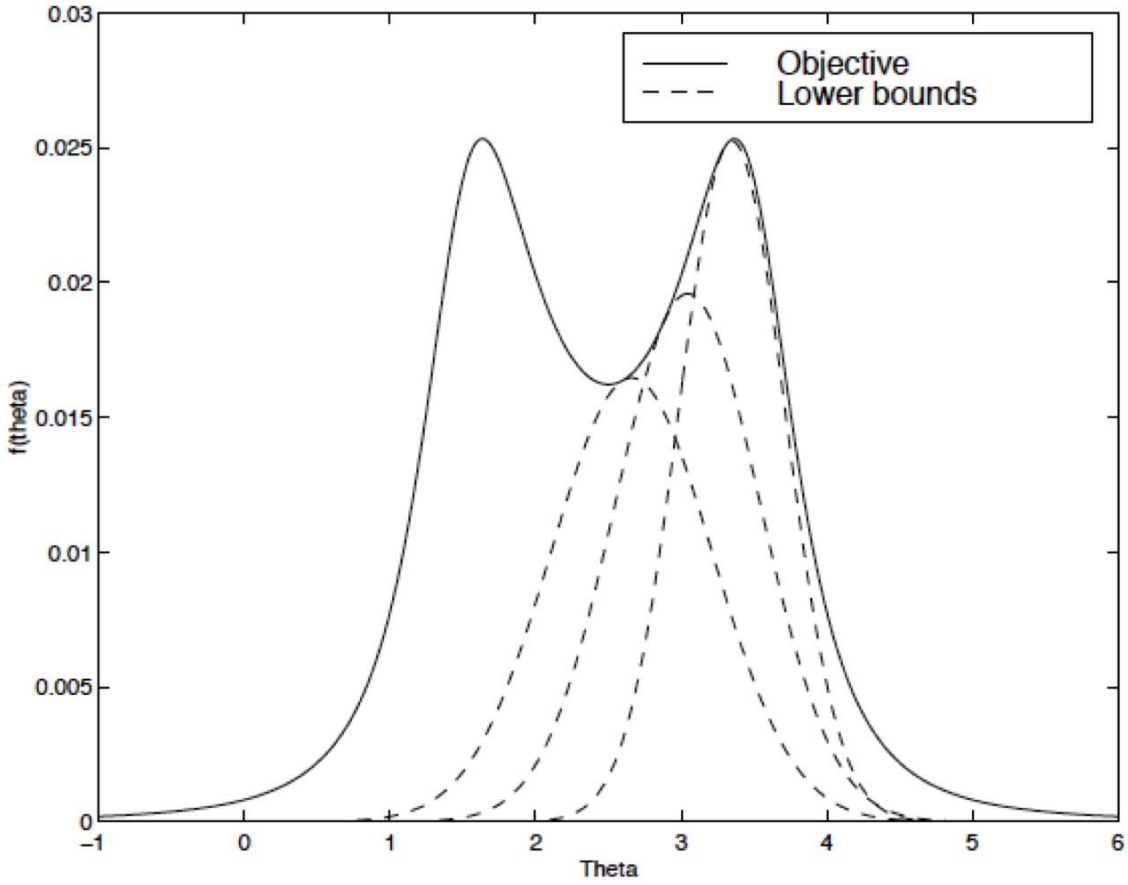
EMアルゴリズム

- $\log p(D|\theta) = \log \int p(D, z|\theta) dz$ を最大化したい
- Jensenの不等式から、

$$\begin{aligned}
\log \int p(D, z|\theta) dz &= \log \int q(z|D) \frac{p(D, z|\theta)}{q(z|D)} dz \\
&\geq \int q(z|D) \log \frac{p(D, z|\theta)}{q(z|D)} dz \\
&\equiv F(q(z|D), \theta)
\end{aligned}$$

  - よって、下限 $F(q(z|D), \theta)$ を $q(z|D), \theta$ についてそれぞれ最大化すればよい.

EMアルゴリズムのイメージ



- Minka (1998): “Expectation-Maximization as lower bound maximization” より

EMアルゴリズム (2)

- Eステップ: $q(z|D)$ について最大化

$$
\begin{aligned}
F(q(z|D),\theta) &= \int q(z|D)\log\frac{p(D,z|\theta)}{q(z|D)}dz \\
&= \int q(z|D)\log\frac{p(z|D,\theta)p(D|\theta)}{q(z|D)}dz \\
&= -D(q(z|D)||p(z|D,\theta)) + \log p(D|\theta)
\end{aligned}
$$

  - はKLダイバージェンスの性質から、
    $q(z|D) = p(z|D,\theta)$ のとき最大

EMアルゴリズム (3)

- Mステップ: θ について最大化

$$F(q(z|D),\theta) = \int q(z|D)\log\frac{p(D,z|\theta)}{q(z|D)}dz$$
$$= \langle \log p(D,z|\theta)\rangle_{q(z|D)} + H(q(z|D))$$

$\langle\cdots\rangle$ は $E[\cdots]$ の省略形

  - θ に依存するのは第1項だけなので、

    $Q(\theta) = \langle \log p(D,z|\theta)\rangle_{q(z|D)}$ (Q関数) について

    $\frac{\delta Q}{\delta\theta} = 0$ を解いた θ を新しい θ とする.

EM: Unigram Mixtures の場合

$D = \{d_1, d_2, \cdots, d_D\}$ について、

$$p(D|\theta) = \prod_d p(d, z|\theta)$$

$$= \prod_d \sum_z p(z) \prod_n p(w_n|z) \quad \text{を最大化.}$$

- Eステップ:

$$p(z|d) \propto p(d|z)p(z) = p(z) \prod_n p(w_n|z)$$

EM: Unigram Mixtures の場合 (2)

- Mステップ:

$n(d,w)$ は文書dに単語wが現れた回数

$$p(D, z|\theta) = \prod_d p(z) \prod_w p(w|z)^{n(d,w)}$$ と書けるから、

$$\log p(D, z|\theta) = \sum_d \left[\log p(z) + \sum_w n(d,w) \log p(w|z) \right]$$

よって、

$$Q(\theta) = \left\langle \log p(D, z|\theta) \right\rangle_{q(z|D)}$$

$$= \sum_d \sum_z p(z|d) \left[\log p(z) + \sum_w n(d,w) \log p(w|z) \right]$$

EM: UMの場合 (2)

- Mステップの続き
  - $p(z)$ に関して最大化
    - $\sum_z p(z) = 1$ の制約があるので、ラグランジュの未定乗数法により、

$$\frac{\partial}{\partial p(z)}\left[Q + \lambda\left(\sum_z p(z) - 1\right)\right] = 0$$

$$\Longleftrightarrow \sum_d \frac{p(z|d)}{p(z)} + \lambda = 0 \quad \therefore\ p(z) \propto \sum_d p(z|d)\,.$$

  - $p(w|z)$ についても同様にして、
    - $p(w|z) \propto \sum_d p(z|d)n(d,w)\,.$

EMアルゴリズムのまとめ

- データDに潜在変数zがある場合の推定法
- 尤度の下限を、zとパラメータ θ について逐次最大化
  - Eステップ: 各データのzの確率分布 p(z|D)を計算
  - Mステップ: p(z|D)を使って、Q関数を最大化する θ を求める

Lecture 1のまとめ

- ナイーブベイズ: 文書のトピックが与えられている時の、簡単な生成モデル
  - 未知の文書のトピックを予測できる
- Unigram Mixtures (UM): ナイーブベイズでトピック自体未知の場合、トピックを潜在変数と見なしてEMアルゴリズムで推定
- 基本知識
  - Bag of words の仮定、多項分布
  - ベイズの定理、事前分布・事後分布
  - EMアルゴリズム=対数尤度の下限最大化

最尤推定とベイズ推定

$$D(3,:) = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 2 \quad 0]$$

広告メール中の単語生起回数

- 頻度を単純に割り算した

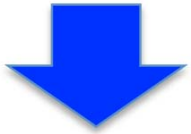
$$\hat{p}(w) = \frac{n(w)}{N}$$

は、データの確率を最大にするので、最尤推定量とよばれる (n(w)=0なら確率は0)

- 実際には、単語の次元は数万次元‥ほとんどのn(w)は0
  - それらの確率は本当に0か?

最尤推定とベイズ推定 (2)

- NB/UMでは、
  $p_0 = \{p(w|k=0)\}, p_1 = \{p(w|k=1)\}$
  などの複数の多項分布を考える
  - ある分布で頻度が0でも、他では現れていることが多い (共通性がある)



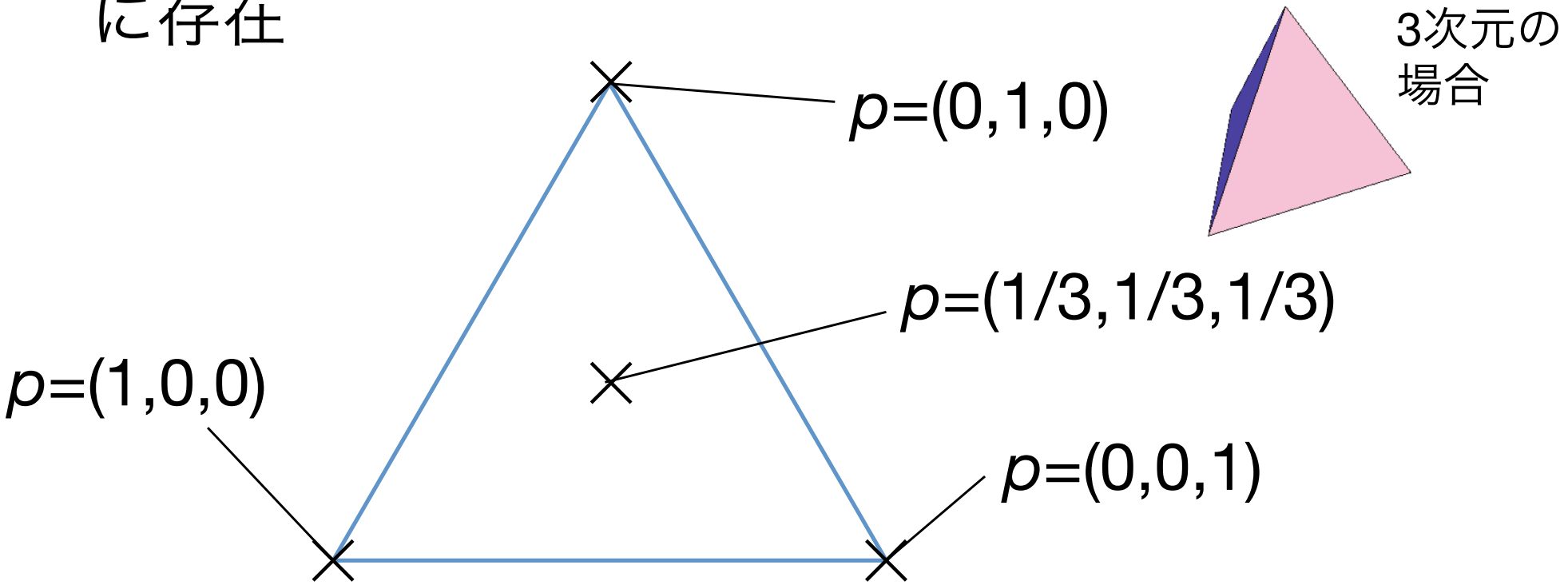
- $p_0, p_1, \cdots$ の 多項分布を生成する分布 を考えるべきなのでは?
  - 最も簡単な分布: ディリクレ(Dirichlet)分布.

多項分布と単体

- K次元の多項分布

$$\mathbf{p} = (p_1, p_2, \cdots, p_K) \ (p_k \geq 0, \ \sum_k p_k = 1)$$

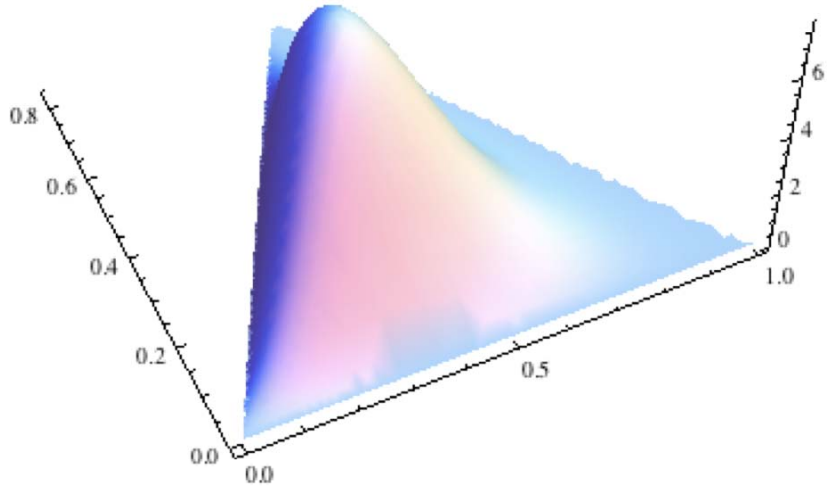
は、単体(Simplex)とよばれるK-1次元の図形の中に存在



ディリクレ分布

- $\mathbf{p}=(p_1,p_2,\cdots,p_K)$ $(p_k \geq 0, \sum_k p_k = 1)$ のとき、ディリクレ分布

$$p(\mathbf{p}|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} p_k^{\alpha_k - 1}$$
$$= \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k - 1}$$

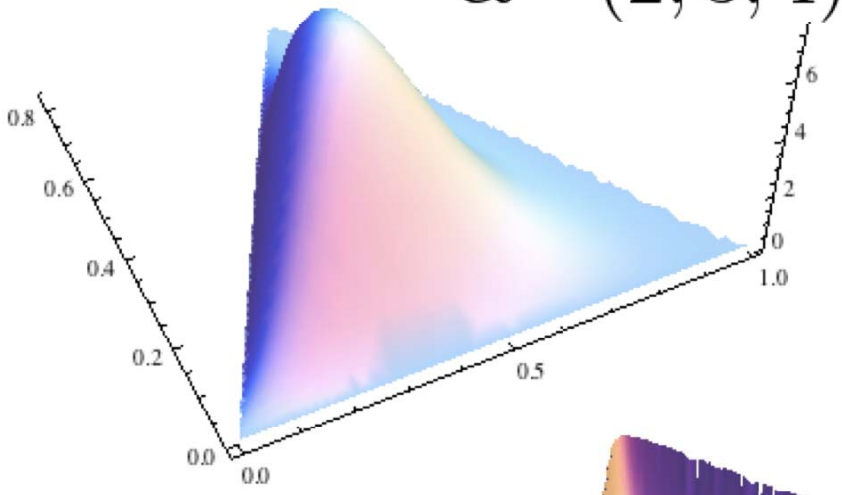


  - $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_K)$：パラメータ $(\alpha_k > 0)$
  - 期待値：$E[p_k|\boldsymbol{\alpha}] = \dfrac{\alpha_k}{\sum_k \alpha_k}$

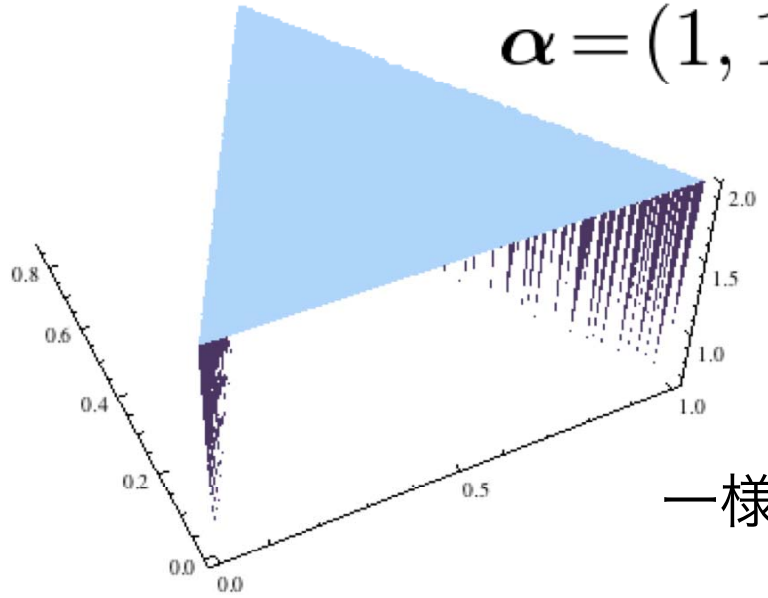
ディリクレ分布 (2)

- ディリクレ分布のパラメータ$\boldsymbol{\alpha}$と分布の形

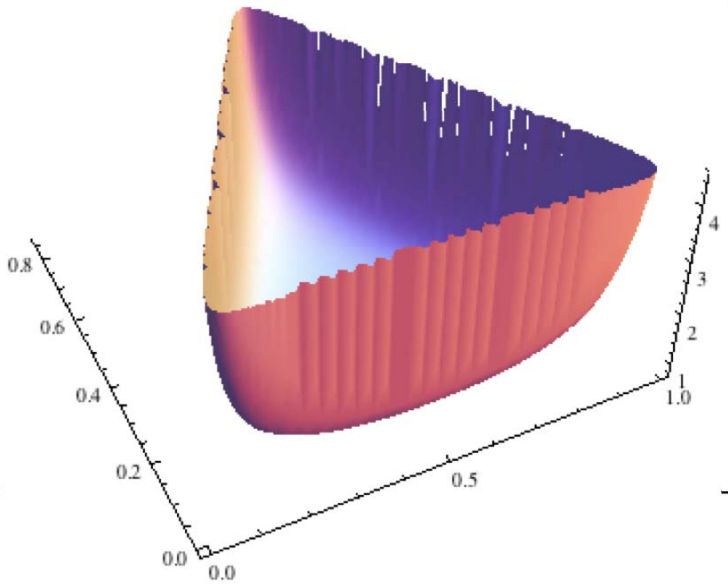
$\boldsymbol{\alpha} = (2, 3, 4)$



$\boldsymbol{\alpha} = (1, 1, 1)$



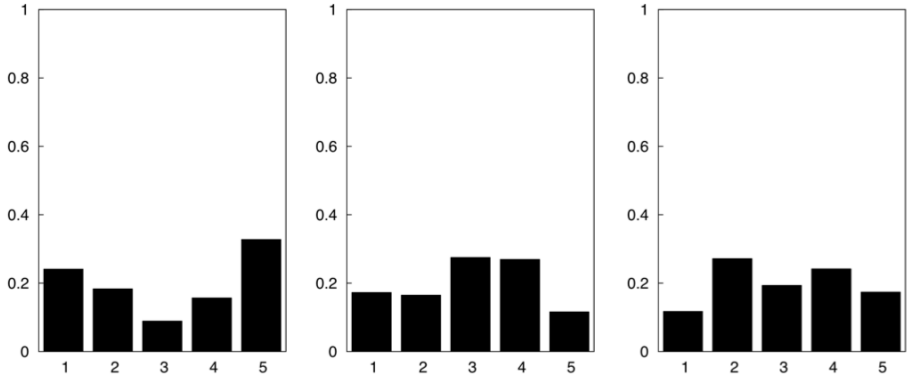
一様分布



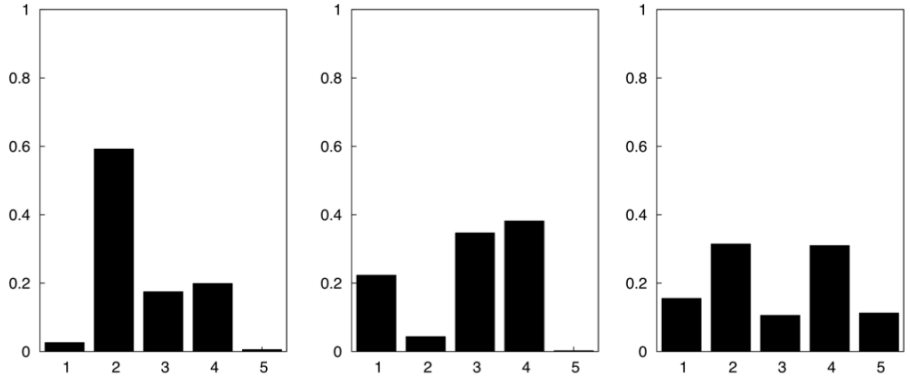
$\boldsymbol{\alpha} = (0.5, 0.5, 0.5)$

ディリクレ分布 (3)

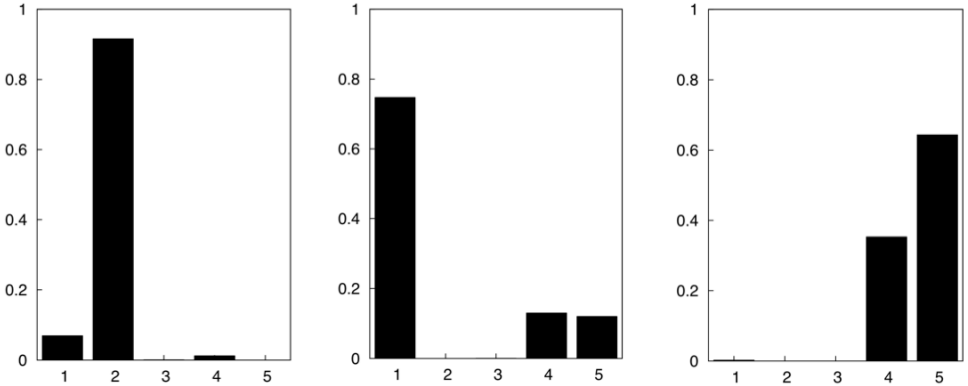
- ディリクレ分布からのサンプル $\mathbf{p}$



$$\boldsymbol{\alpha} = (10, 10, \cdots, 10)$$



$$\boldsymbol{\alpha} = (1, 1, \cdots, 1)$$



$$\boldsymbol{\alpha} = (0.1, 0.1, \cdots, 0.1)$$

最尤推定とベイズ推定 (3)

- $\mathbf{p}$ がディリクレ事前分布から生まれたとき、観測頻度 $X = n_1, n_2, \cdots, n_K$ による事後分布?
- ベイズの定理によれば、

$$p(\mathbf{p}|X) \propto p(X|\mathbf{p})\,p(\mathbf{p})$$
$$\propto \prod_k p_k^{n_k} \cdot \left(\prod_k p_k^{\alpha_k - 1} \right) = \prod_k p_k^{n_k + \alpha_k - 1}$$

  - これは $\mathrm{Dir}(\boldsymbol{\alpha} + \mathbf{n})$ なので、期待値は

$$E[p_k|X] = \frac{n_k + \alpha_k}{\sum_k (n_k + \alpha_k)}$$

最尤推定とベイズ推定 (4)

ディリクレスムージング
という

- 最尤推定　$\hat{p}_k|X = \frac{n_k}{N}$
- ベイズ推定　$E[p_k|X] = \frac{n_k+\alpha_k}{\sum_k (n_k+\alpha_k)} = \frac{n_k+\alpha_k}{N+\sum_k \alpha_k}$
  - 頻度にα_kを足して正規化することは、ディリクレ事前分布$\mathrm{Dir}(\boldsymbol{\alpha})$を考えていることに相当する
  - $\alpha_k \equiv 1$: 事前分布に一様分布を仮定
    - ラプラススムージングとよばれる (が、これが最良なわけではない)

UMの実装

- p(k)もp(w|k)もディリクレスムージング
  - EMが過学習しにくくなる

```
mondrian:~/work/um/src% ./um –h
um, Unigram Mixtures.
Copyright (C) 2012 Daichi Mochihashi, all rights reserved.
$Id: um.c,v 1.4 2013/01/05 06:33:55 daichi Exp $
usage : um -M mixtures [-e eta] [-g gamma] [-d epsilon] [-I emmax]
train model
eta       = Dirichlet prior for beta (default 0.01)
gamma = Dirichlet prior for lambda (default 0)
epsilon  = relative difference for convergence (default 0.0001)
```

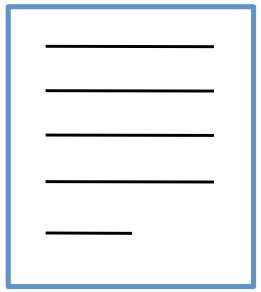
  - http://www.ism.ac.jp/~daichi/dist/um/um-0.1.tar.gz

Lecture 2

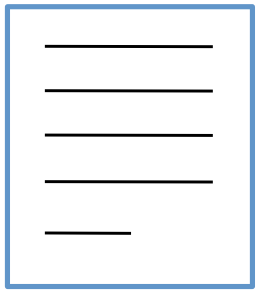
(本格的な)トピックモデル

UMとBag of wordsの復習

- Bag of words: テキストを単語の集合と頻度で表現



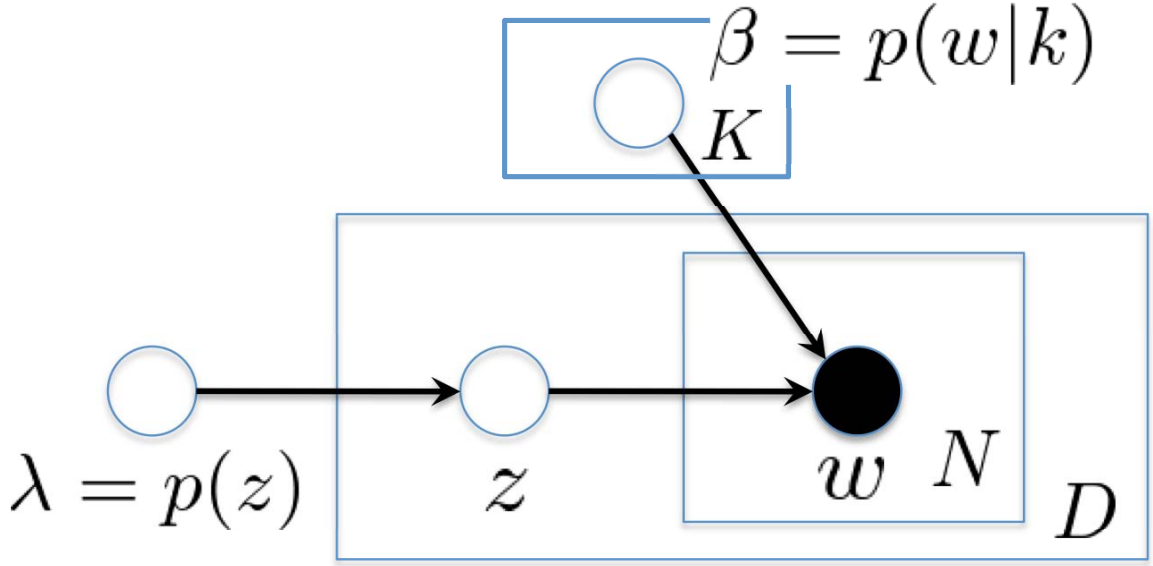
{“情報”:5,
“IT”:2,
“システム”:1,
“開発”:4}



{“バッター”:1,
“監督”:3,
“ドラフト”:2,
“打率”:1}

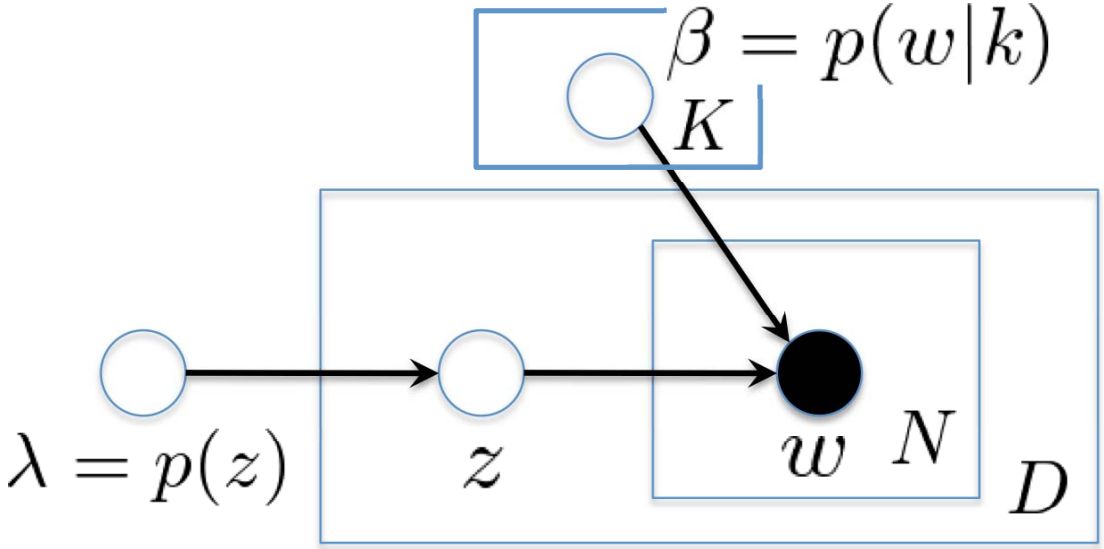
- Unigram Mixturesの生成モデル
  - For d = 1 … D,
    (1) $z \sim p(z)$ でテキストdのトピック z を選択
    (2) For n = 1 .. N_d,
    $w_n \sim p(w|z)$ でn番目の単語 w_n を生成.

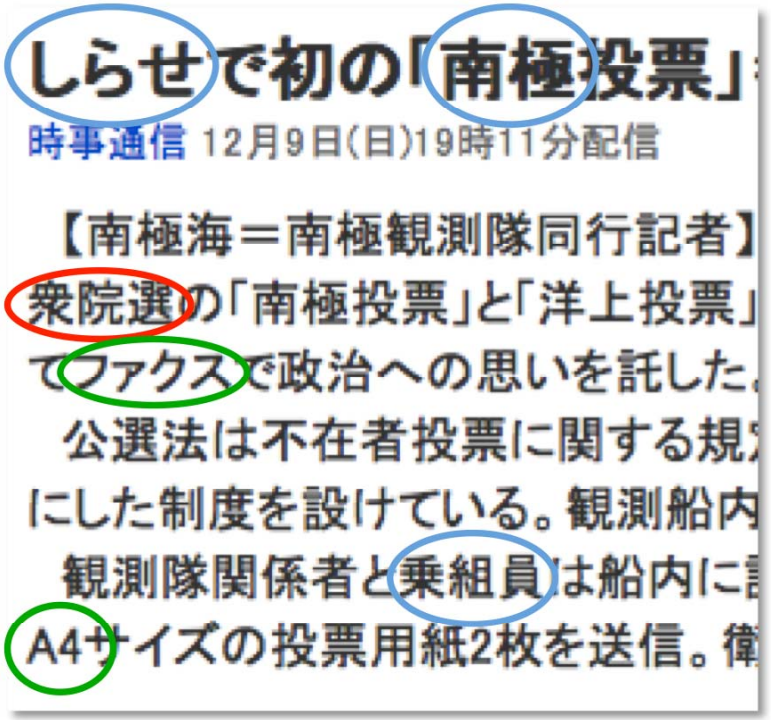
グラフィカルモデル表現



- UMのモデルは、上のようなグラフィカルモデル(プレート表現)で表せる [重要]

UM/NBの問題

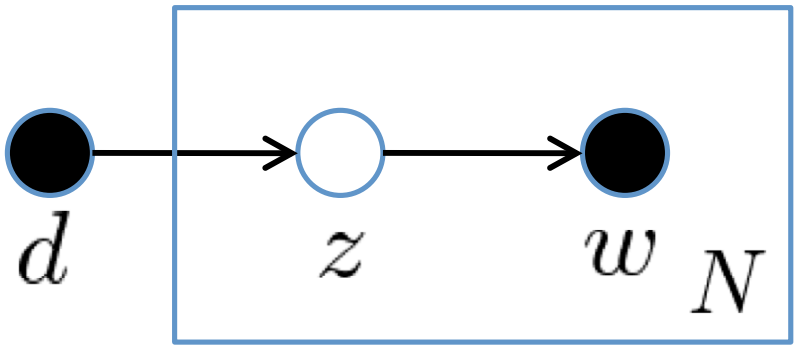




- 1つの文書は、すべて同じトピックに属する
  - 実際の文書は、複数のトピックが入り混じっている!
- UM/NBは制限の強い、単純すぎるモデル

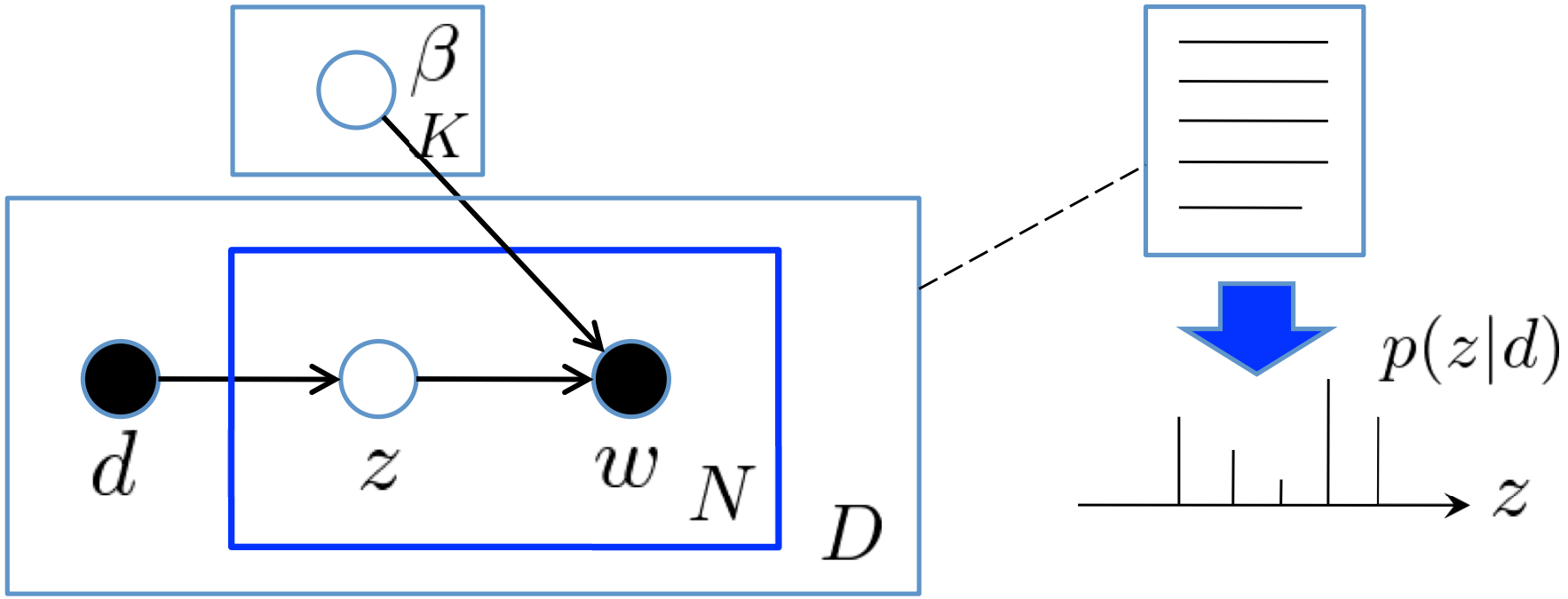
PLSI (Hofmann 1999): 「トピックモデル」

- Probabilistic Latent Semantic Indexing:
  複数のトピック、LSIの確率化
  - PLSA (Probabilistic Latent Semantic Analysis)
    ともよばれるが、以下PLSI



- 1単語ごとに、違うトピックz
  - 文書には1つのzではなく, “トピック分布” $p(z|d)$

PLSIのグラフィカルモデル



- 文書dには、トピック分布 $p(z|d)$ が存在
  - 一単語ごとに、$p(z|d)$ からトピックzを生成
    → z から単語を生成する

PLSIの学習結果 (1)

- トピック別単語分布 $p(w|z)$ の上位語

Topic 1	Topic 2	Topic 3	Topic 4
先,後,#,歩,銀,四, 五,六,同,二,飛, 八,成,玉,七,三, 金,九,桂,角,と, 谷川,が,た,手,は, 丸山,一,香,の,で, 局,図,戦,段	の,号,事故,機,が, た,に,安全,#,部分, を,原発,原因,は, 基,水,運転,装置, 爆発,器,原子力, 炉,作業,し,燃料, で,漏れ,発生,と, 配管,原子,ガス	#,勝,敗,戦, イチロー,日,回, リーグ,大リーグ, マリナーズ,新庄, 試合,安打,点,ス, で,手,共同,メッツ, 外野,は,大,投手, 第,米,の,打席, ソックス, ヤンキース,記録, バックス,打率, ニューヨーク	研究,細胞, 遺伝子,移植, の,治療,物質, 教授,を,患者, 科学,脳,医療, 病院,ローン, ヒト,実験,薬, グループ,遺伝, が,臓器,体,ク, 病,する,に,学会, さ,DNA,開発, 臨床,人間,神経

PLSIの学習結果 (2)

- トピック別単語分布 $p(w|z)$ の上位語

Topic 5	Topic 10	Topic 50	Topic 100
イスラエル,	#,回,た,を,勝,	た,#,を,容疑	の,を,#,に,
パレスチナ,	の,で,監督,は,	し,の,者,に,	は,環境,化,
自治,議長,	が,敗,点,に,	で,は,て,逮捕,	が,で,する,
和平,中東,派,	投手,登板,試合,	と,月,い,同,	し,など,量,
攻撃,の,	巨人,一,近鉄,	など,捜査,れ,	と,書,や,
エルサレム,	本塁打,安打,	疑い,元,さ,	削減,開発,
と,延期,人,	打,死,球,戦,	県警,が,違反,	た,年,て,いる,
政府,を,過激,	ヤクルト,	事件,日,万,	地球,も,生産,
日,アラファト,	先発,ダイエー,	ら,処分,人,	国,議定,温暖,
ユダヤ,イラク,	阪神,西武,初,	として,課,	ガス,エネルギー,
軍,テロ,小倉,	チーム,今季,番	地検,調べ	効果,技術,さ,
は,シャロン			国内,計画

PLSIの生成モデル

- PLSIでは、次のようにして文書群が生成されたと仮定する
  - For i = 1 … D,
    (1) 文書 $d \sim p(d)$ を選択.
    (2) For n = 1 … N,
      (a) トピック $z \sim p(z|d)$ を選択.
      (b) 単語 $w \sim p(w|z)$ を生成.
- 確率で書くと、

$$p(d,w) = \sum_z p(d,w,z)$$
$$= \sum_z p(d)p(w,z|d) = p(d)\sum_z p(w|z)p(z|d)\,.$$

PLSIの生成モデル (2)

- PLSIによる文書とコーパス全体の確率
  - PLSIでは、文書のインデックスdを観測値として考える

$$p(d, \mathbf{w}) = \prod_n p(d, w_n) = \prod_n p(d) \sum_z p(w_n|z_n) p(z_n|d)$$

$$p(D, W) = \prod_i p(d_i, \mathbf{w}_i)$$

PLSIの生成モデル (3)

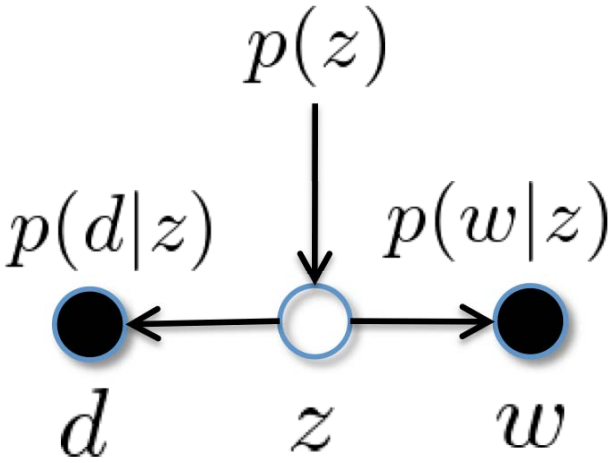
- ベイズの定理を使って計算すると、単語の確率は

$$p(d,w) = p(d)\sum_z p(w|z)p(z|d)$$

意味が謎

$$= \cancel{p(d)}\sum_z p(w|z)\frac{p(d|z)p(z)}{\cancel{p(d)}}$$

$$= \sum_z p(z)p(d|z)p(w|z)\,.$$



  - 文書dと単語wの共起にトピックzが存在
- 新しいパラメータ: $p(z), p(w|z), p(d|z)$

PLSIの学習

- 文書-単語行列Wと文書インデックスD、各単語の潜在トピックZについて、

$$
\begin{aligned}
p(D,W,Z) &= \prod_d p(d, W_d, Z_d) \\
&= \prod_d \prod_n p(d, w_{dn}, z_{dn}) \\
&= \prod_d \prod_n p(z_{dn}) p(d|z_{dn}) p(w_{dn}|z_{dn})
\end{aligned}
$$

- よって、対数尤度は

$$
\log p(D,W,Z) = \sum_d \sum_n \left[\log p(z_{dn}) + \log p(d|z_{dn}) + \log p(w_{dn}|z_{dn}) \right]
$$

PLSIの学習 (2) : EMアルゴリズム

- Eステップ:

$$p(Z|W,\theta) = p(z|d,w)$$
$$\propto p(z,d,w_n) = p(z)p(d|z)p(w|z) \text{ を計算.}$$

  - 文書dの単語wが、どの潜在トピックに属するかの確率分布

- Mステップ: Q関数 $Q(\theta) = \langle \log p(W,Z|\theta) \rangle_{p(Z|W,\theta)}$ をθについて最大化.

$$Q(\theta) = \sum_d \sum_n \sum_z p(z|d,w_{dn}) \Big[\log p(z_{dn}) + \log p(d|z_{dn}) + \log p(w_{dn}|z_{dn}) \Big]$$

PLSIの学習 (3)

- $\delta Q/\delta\theta=0$ を計算すると、

$$\frac{\delta Q}{\delta p(z)} = \frac{\sum_d \sum_n p(z|d,w_{dn})}{p(z)} + \lambda = 0$$

よって

$$p(z) \propto \sum_d \sum_n p(z|d,w_{dn}) \propto \sum_d \sum_n p(z,d,w_{dn})$$
$$\propto \sum_d \sum_n p(z|d,w_{dn})p(d,w_{dn}) \propto \sum_d \sum_w p(z|d,w)n(d,w)$$

- 同様にして、

$$p(w|z) \propto \sum_d p(z|d,w)n(d,w), \quad p(d|z) \propto \sum_w p(z|d,w)n(d,w)$$

PLSIの学習 (4)

- まとめ: PLSIのEMアルゴリズム
  - 初期化: $p(z), p(w|z), p(d|z)$ を乱数で設定.
  - Eステップ:
    各文書dの各単語wについて、トピック分布
    $$p(z|d,w) \propto p(z)p(d|z)p(w|z)$$
    を計算.
  - Mステップ:
    今計算した $p(z|d,w)$ を用いて、パラメータ $p(z), p(w|z), p(d|z)$ を更新. Eステップに戻る.
- 実装: http://chasen.org/~taku/software/plsi/plsi-0.03.tar.gz

PLSI: retrospective

- 単語ごとに潜在トピックを考えることで、UMよりずっと良いトピック分布
- UMは、普通の混合モデル
  - 混合比からトピックzを選ぶ→zから文書を生成
- PLSIは、文書ごとの混合モデル
  - 文書ごとに混合比を選ぶ
  - 混合比からトピックzを選ぶ→zから単語を生成

PLSIは、混合モデルの混合モデル

モデルの評価法

- モデルの学習‥データの確率 $p(D|\theta)$ を最大にするパラメータθを求めること
  - テストデータ D' について、$p(D'|\theta)$を最大にするのが良いモデル
- $p(D'|\theta)$ はデータ数Nに依存→ $p(D'|\theta)^{1/N}$ を考える
  - わかりやすく、確率の逆数(=分岐数)をとった

$$\mathrm{PPL} = p(D'|\theta)^{-\frac{1}{N}} = \exp\left(-\frac{1}{N}\log p(D'|\theta)\right)$$

を、パープレキシティ(平均分岐数)という

確率の「平均」について

- N 個のデータから得られる確率 $p_1, p_2, \cdots, p_N$

$$\times \quad \frac{1}{N}(p_1 + p_2 + \cdots + p_N)$$

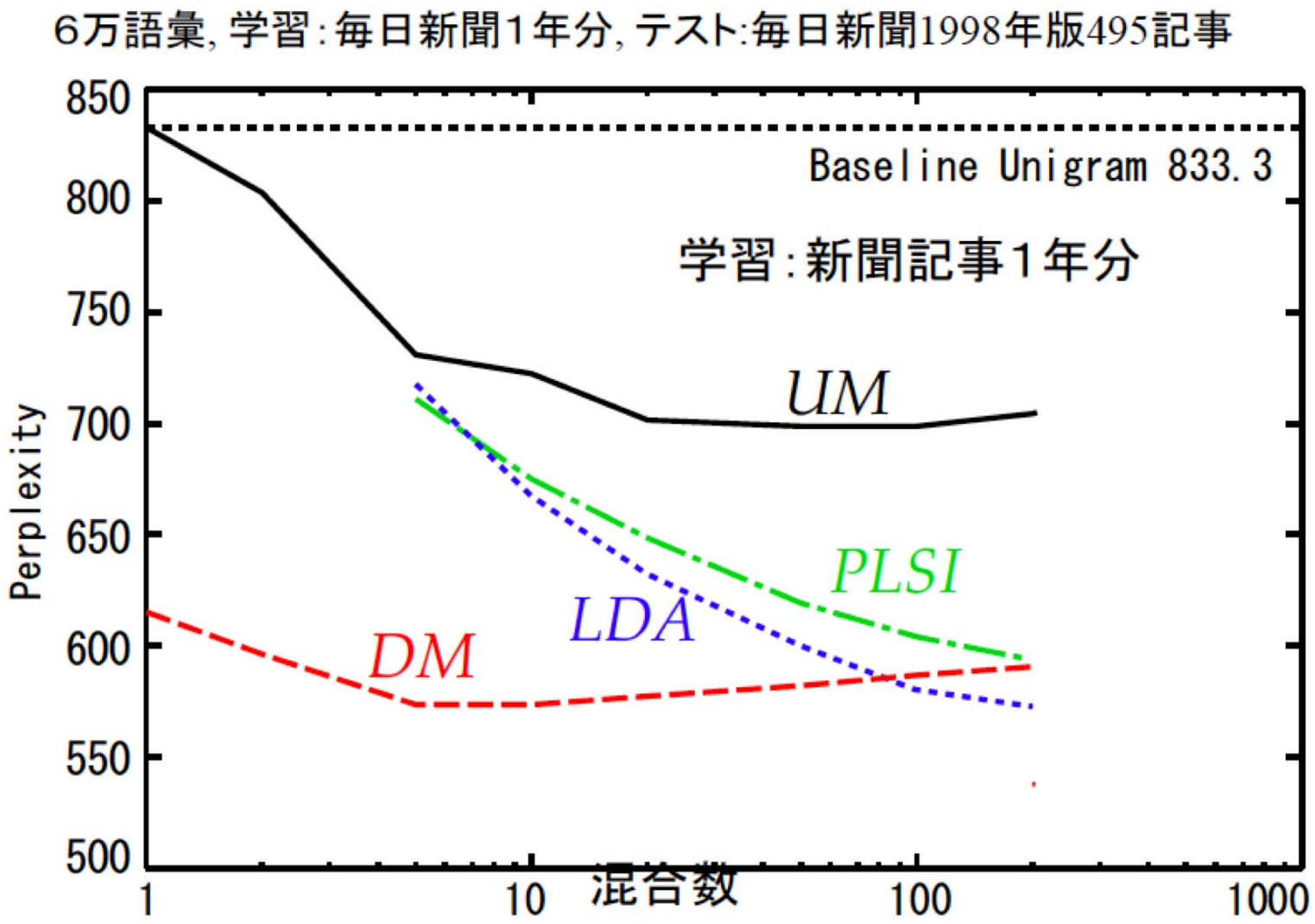
$$\bigcirc \quad \sqrt[N]{\prod_i p_i} = \left(\prod_i p_i\right)^{\frac{1}{N}}$$

  - 本来は、同時確率が独立な場合

$$p(x_1, x_2, \cdots, x_N) = p(x_1)p(x_2)\cdots p(x_N)$$

を考えているため.

UMとPLSIの評価



- 山本&持橋(言語処理学会2006)より

DM= Dirichlet Mixtures (触れません)

PLSIの欠点 (1): 新規文書

- PLSIでの新しい文書 d^{new} の確率は、
  $p(z|d^{new})$ は未定義なので

$$p(d^{new}) = \underbrace{\sum_d p(d) \prod_{n \in d^{new}} \underbrace{\sum_z p(w_n|z)p(z|d)}_{w_n \in d^{new} \text{がdから現れる確率}}}_{\text{の学習文書dについての期待値}}$$

  - アドホック?
  - 計算量も莫大 (学習文書dの数は数万~数百万以上のことも)

PLSIの欠点 (2)

- PLSIのパラメータ: $p(z), p(w|z), p(d|z)$

学習データに比例して増大!
(D×K=数万～数千万×数百)

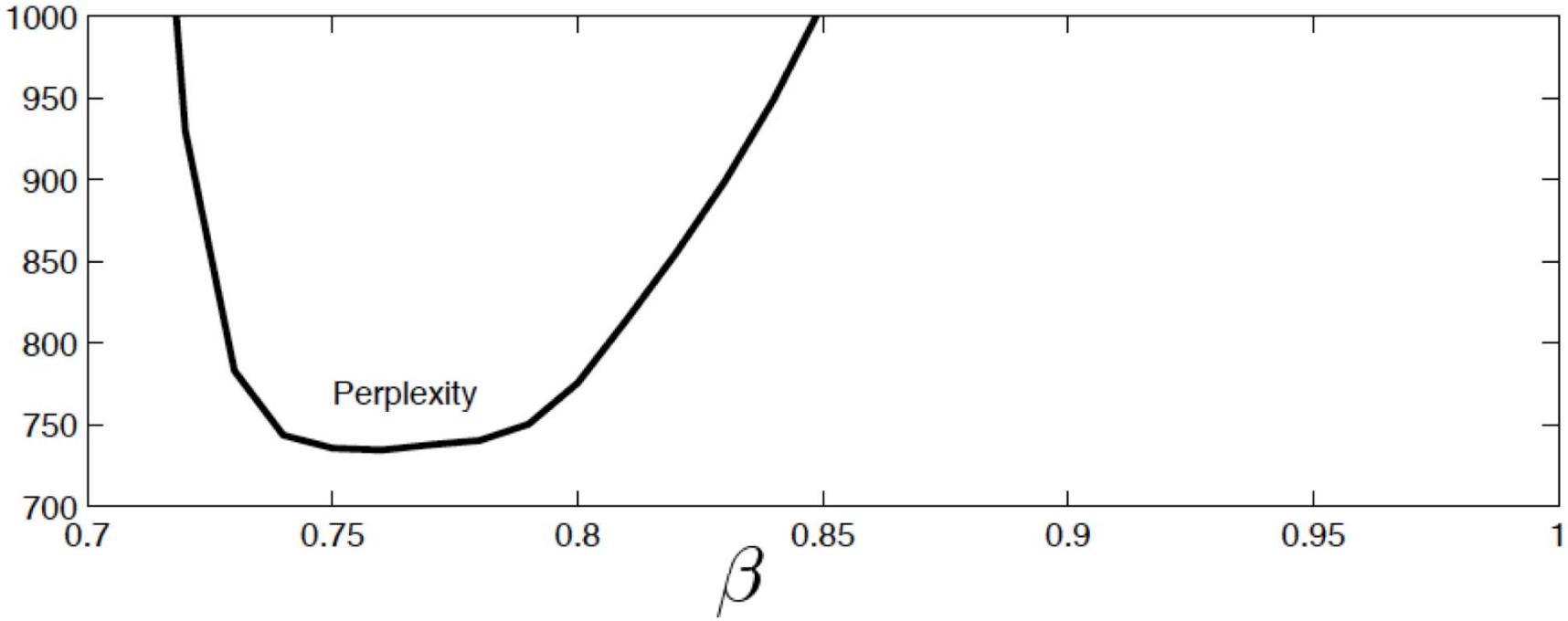
簡単に学習データにオーバーフィット.

- アドホックな解決: Tempered EM (焼きなまし)

$$p(z|w,d) \propto \{ p(z)p(w|z)p(d|z) \}^{\beta}$$

  - $0 < \beta \leq 1$ ととって、確率分布p(z|w,d)を「なめらか」にする

Tempered EMの効果 (Hofmann99)



- $\beta = 0.75$程度で最良 ($\beta = 1$では大幅に悪化)
- アドホック。。。
  - そもそも、なぜオーバーフィットするか?

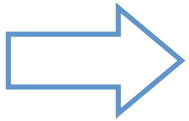
PLSIからLDAへ

- PLSIの問題点: $p(z|d)$ が学習データについてだけ定義されている
(由来がない)

真の生成モデルではない！

  – 学習データについての最尤推定

- $\theta = p(z|d)$ 自体を確率的に生成するモデルを考えるべきなのでは?



LDA (Latent Dirichlet Allocation)
(Blei+ 2001,2003)

午前のまとめ

- トピックモデルとその背景
- Naïve Bayes→Unigram Mixtures
  - EMアルゴリズム
- PLSIとその学習法
- PLSIの欠点→LDAによるベイズ化

Lecture 3

トピックモデル: LDA

LDAとは

- Blei+ (NIPS 2001, JMLR 2003)で提案
- トピックモデルの最も基本となるモデル
  - PLSIの完全なベイズ化
- 基本的な考え方は、PLSIと同じ
  - 単語ごとに潜在トピックがある

トピック分布の由来

- PLSIでは、文書dにトピック分布 $\theta = p(z|d)$ があった

これを確率変数(×パラメータ)とみて、生成したい

- ディリクレ分布を使えばいい!
  - $\theta = (\theta_1, \theta_2, \cdots, \theta_K)$ を生成する、K次元のディリクレ分布 $\mathrm{Dir}(\theta|\boldsymbol{\alpha})$

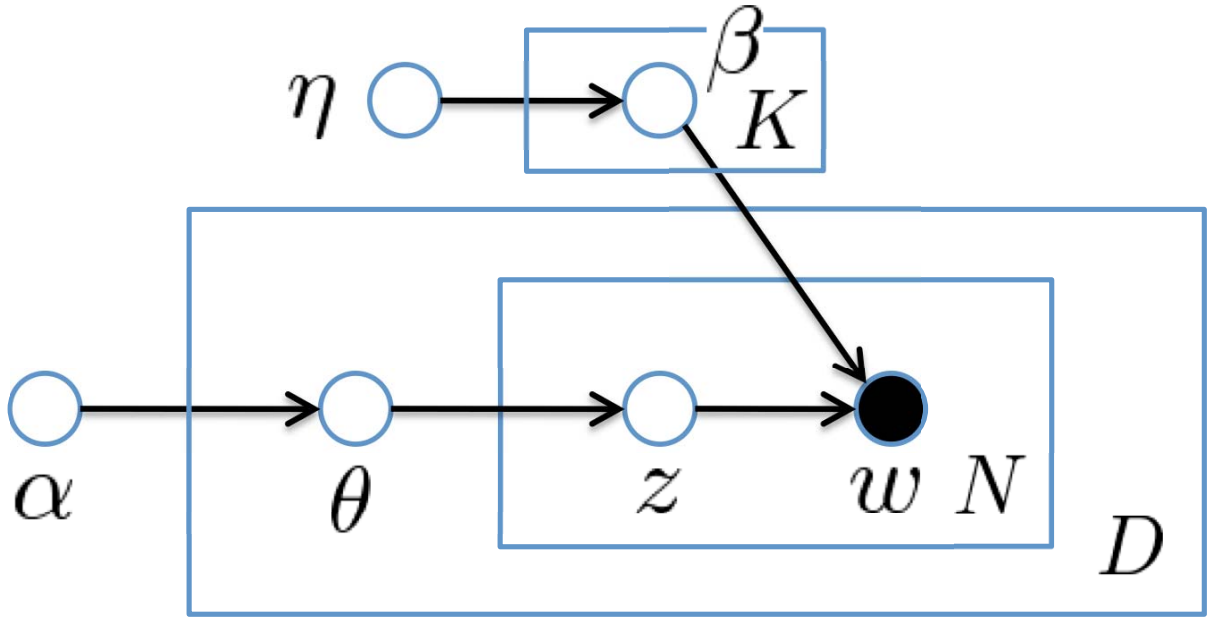
$$p(\theta|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

二つの生成モデル

- PLSIの生成モデル
  (1) $\theta = p(z|d)$ (固定)からトピックzを生成
  (2) $p(w|z)$ から単語wを生成

- LDAの生成モデル
  (0) トピック分布 $\theta \sim p(\theta|\boldsymbol{\alpha})$ を生成
  (1) $\theta = p(z|d)$ からトピックzを生成
  (2) $p(w|z)$ から単語wを生成
  - グラフィカルモデルで書くと?

LDAの生成モデル



- $\theta \rightarrow z \rightarrow w$ の順で単語 w を生成
- θ は D 回(文書数)、z は N 回(単語数) 生成
  - $\beta = (\beta_1, \cdots, \beta_K)$ はトピック別単語分布 $p(w|k)$

LDAの生成モデル (2)

$$p(w, z, \theta) = p(w|z)p(z|\theta)p(\theta|\alpha)$$

観測単語　トピック　トピック分布

よって、文書 $\mathbf{w} = w_1 w_2 \cdots w_N$ について

$$p(\mathbf{w}, z, \theta) = p(\theta|\alpha) \prod_n p(w_n|z_n)p(z_n|\theta)$$

$$p(\mathbf{w}) = \int \sum_z p(\mathbf{w}, z, \theta) d\theta$$

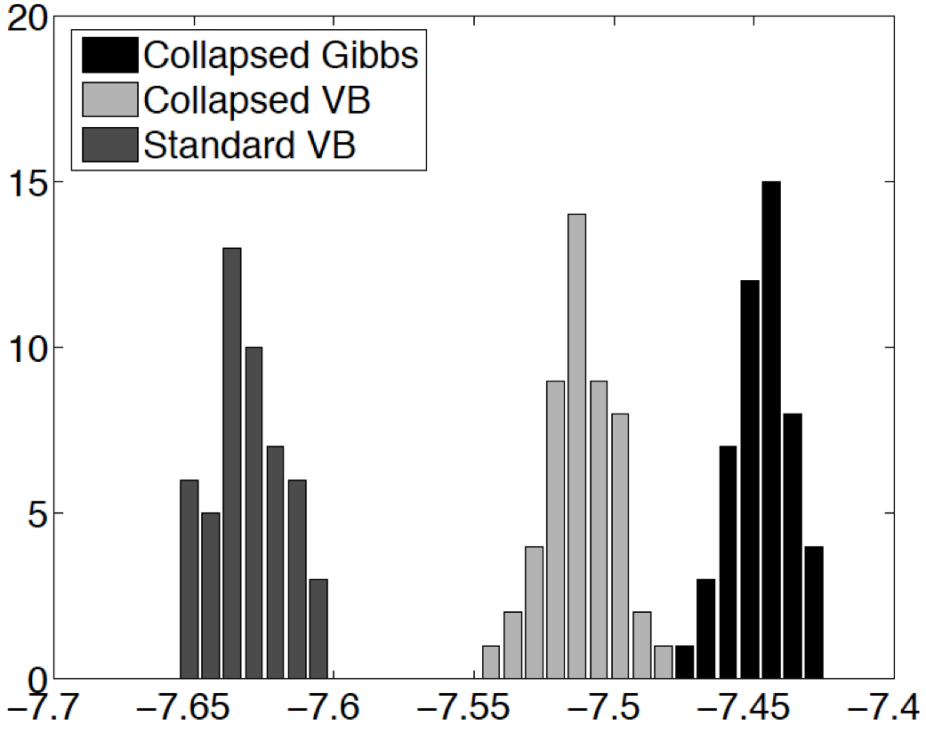
$$= \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \int \left(\prod_k \theta_k^{\alpha_k - 1} \right) \prod_n \sum_k p(w_n|k)\theta_k d\theta$$

- パラメータは α と $\beta = \{ p(w|k) \}$

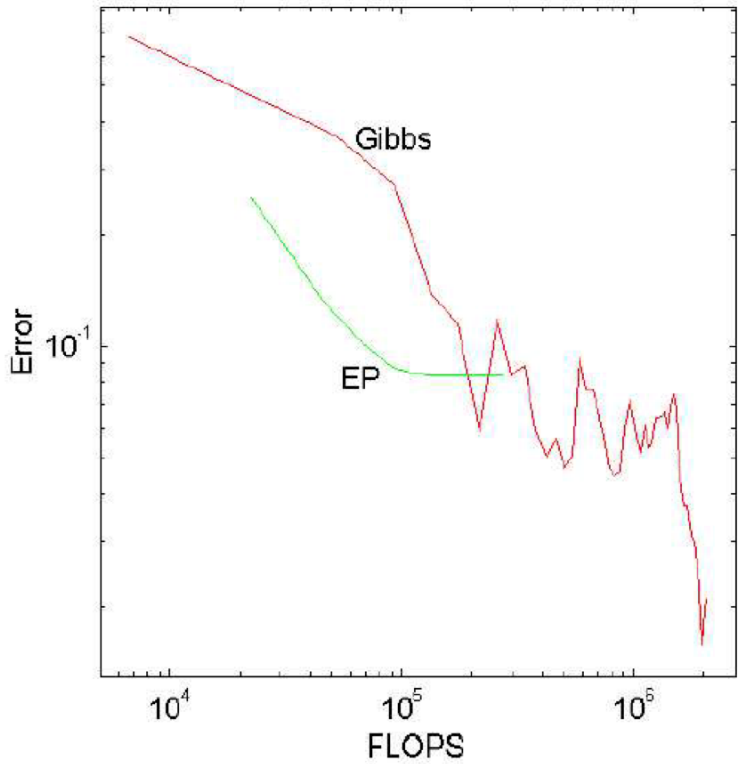
LDAの解法

- どうやって解くか?
  - 変分ベイズ法 (Blei+ 2001,2003)　(オリジナル)
  - Gibbs サンプリング (Griffiths&Steyvers 2004)
  - 期待値伝播法 (Minka&Lafferty 2002)
  - Collapsed 変分ベイズ法 (Teh+ 2006)
  - 固有値計算 (!!) (Anandkumar+, arXiv 2012)

各学習法の比較



Teh+(2006)より (KOSコーパス)



EPとGibbsの比較 (無限混合モデル)

- 性能はGibbs>CVB>VB (Gibbsは局所解に陥り難い)
  - 計算の速さではVB>CVB>Gibbs、数倍程度

LDAの学習: Gibbs Sampler

- 導出や実装が簡単で、高性能
  - 最近は並列化も研究されている (Ihler+09など)
- Gibbs Samplerとは
  ‥マルコフ連鎖モンテカルロ法 (MCMC)の
  最も簡単な場合
  - 潜在変数を、分布ではなく条件つき分布から
    実際にサンプリング
    ＝単語の潜在トピックを次々とサンプリング
  - EMと違い、原理的に無限回繰り返せば、
    真の分布からのサンプル

Gibbs Sampler

- 潜在変数 $z_1, z_2, \cdots, z_N$ を持つ確率モデル

$$p(X, z_1, z_2, \cdots, z_N)$$

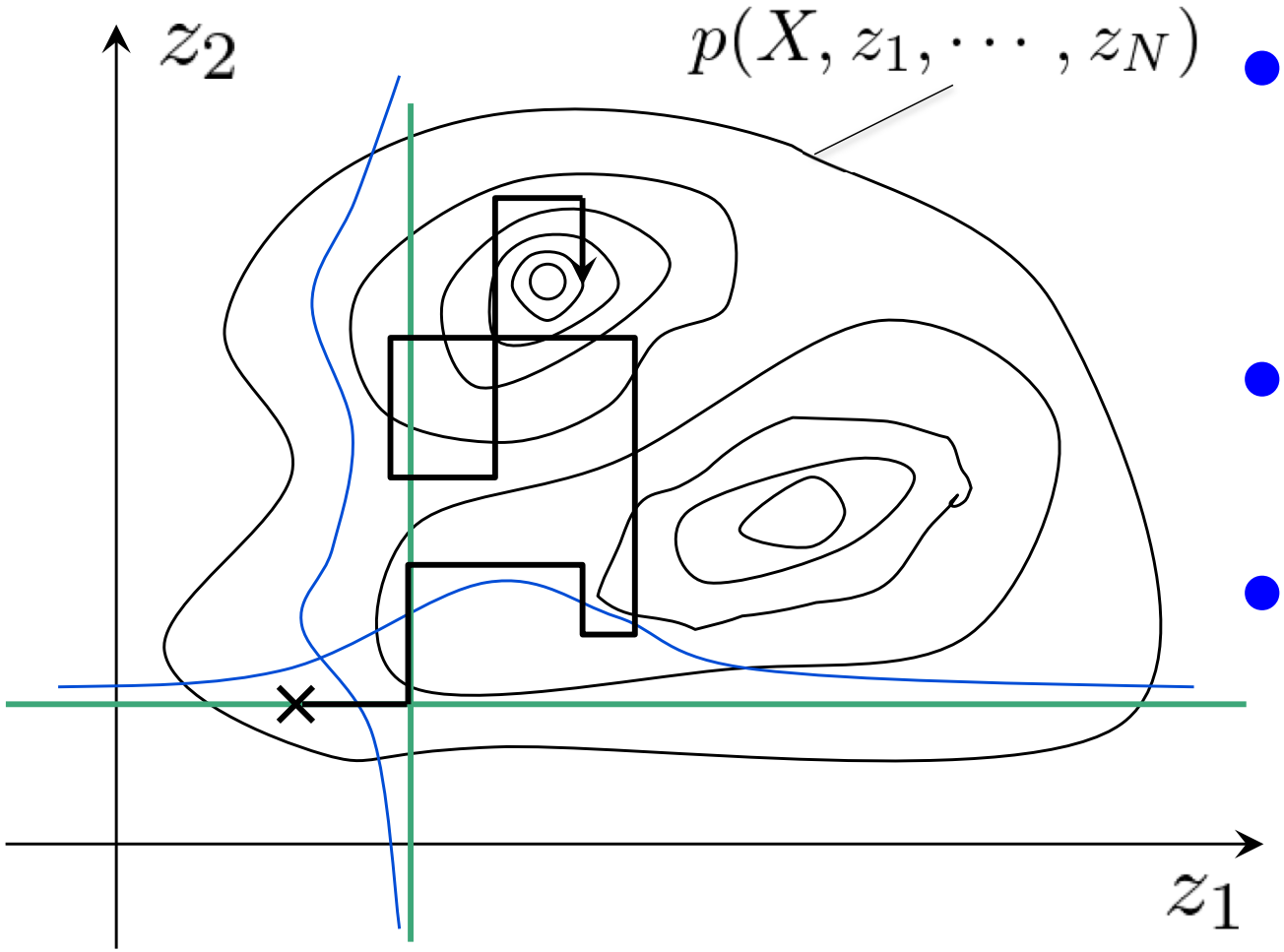
があるとき、各z_iを「考え直す」、つまり、
条件付き分布

$$z_i \sim p(z_i | X, z_1, \cdots, z_{i-1}, z_{i+1}, \cdots, z_N)$$

からランダムにサンプリングすることを繰り返す

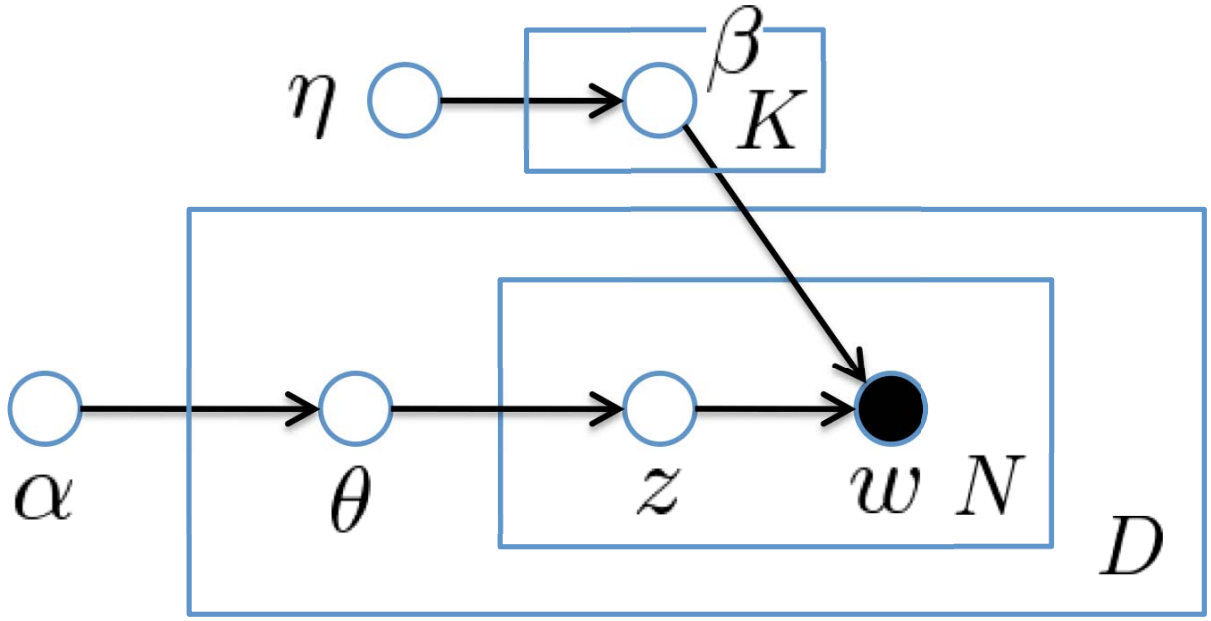
真の分布に収束.

Gibbs Samplerのイメージ



- 条件つき確率に基づく、確率的な山登り
- 局所最適に陥らない
- 確率最大の一点に収束するわけではない
  - 分布全体からのサンプル

LDAのGibbs Sampler



- LDAの潜在変数: θ (文書のトピック分布)と z (各単語のトピック)→実は z だけでよい
  - $z_i \sim p(z_i|\mathbf{w}, z_{-i}, \alpha, \eta)$ から、z_i を次々とサンプルして更新.

LDAのGibbs Sampler (2) (Griffiths+ 2004)

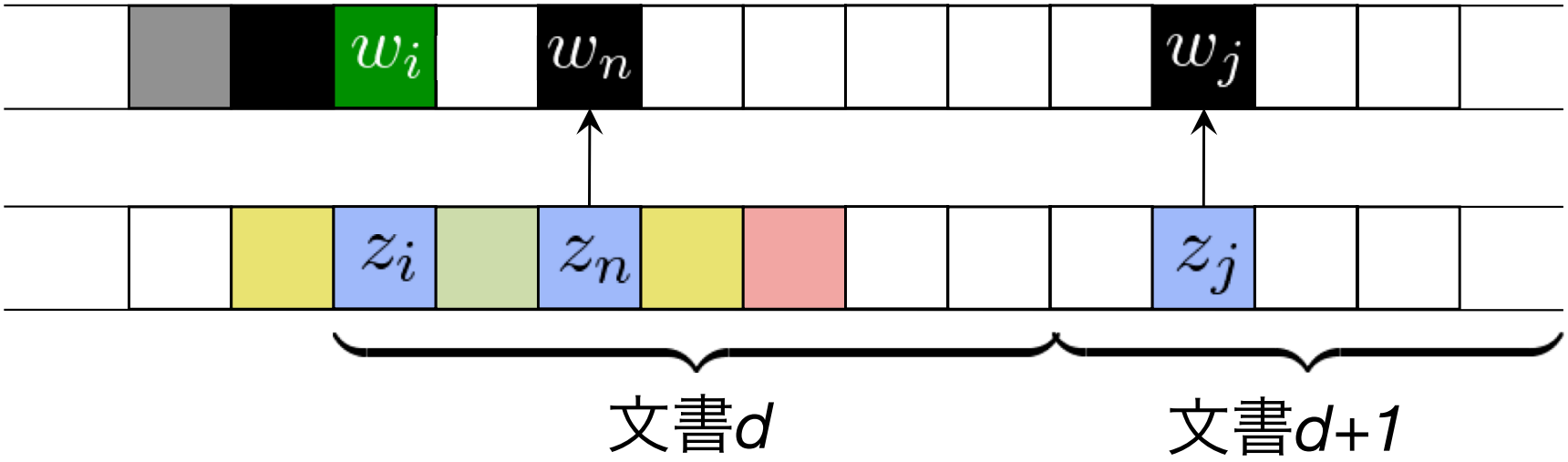
$$
\begin{aligned}
p(z_i = k|\mathbf{w}, z_{-i}) &\propto p(z_i = k, w_i|\mathbf{w}_{-i}, z_{-i}) \\
&= p(w_i|z_i = k, w_{-i}, z_{-i})\, p(z_i = k|\mathbf{w}_{-i}, z_{-i}) \\
&= \frac{\eta + n^{(w_i)}_{-i,k}}{\sum_w \left(\eta + n^{(w)}_{-i,k}\right)} \cdot \frac{\alpha_k + n^{(d)}_{-i,k}}{\sum_k \left(\alpha_k + n^{(d)}_{-i,k}\right)}
\end{aligned}
$$

$n^{(w)}_{-i,k}$ データ全体で単語wがトピックkに割り当てられた回数 (w_i除く)

$n^{(d)}_{-i,k}$ 文書d中でトピックkに割り当てられた単語数 (w_i除く)

- $p(z|w,d) \propto p(z,w|d) = p(w|z)p(z|d)$ のような意味
  - 第2項では、$\int p(z_i = k|\theta)p(\theta|\alpha, \mathbf{w}_{-i}, z_{-i})d\theta$ として θ を積分消去

LDAのGibbs Sampler (3) : イメージ



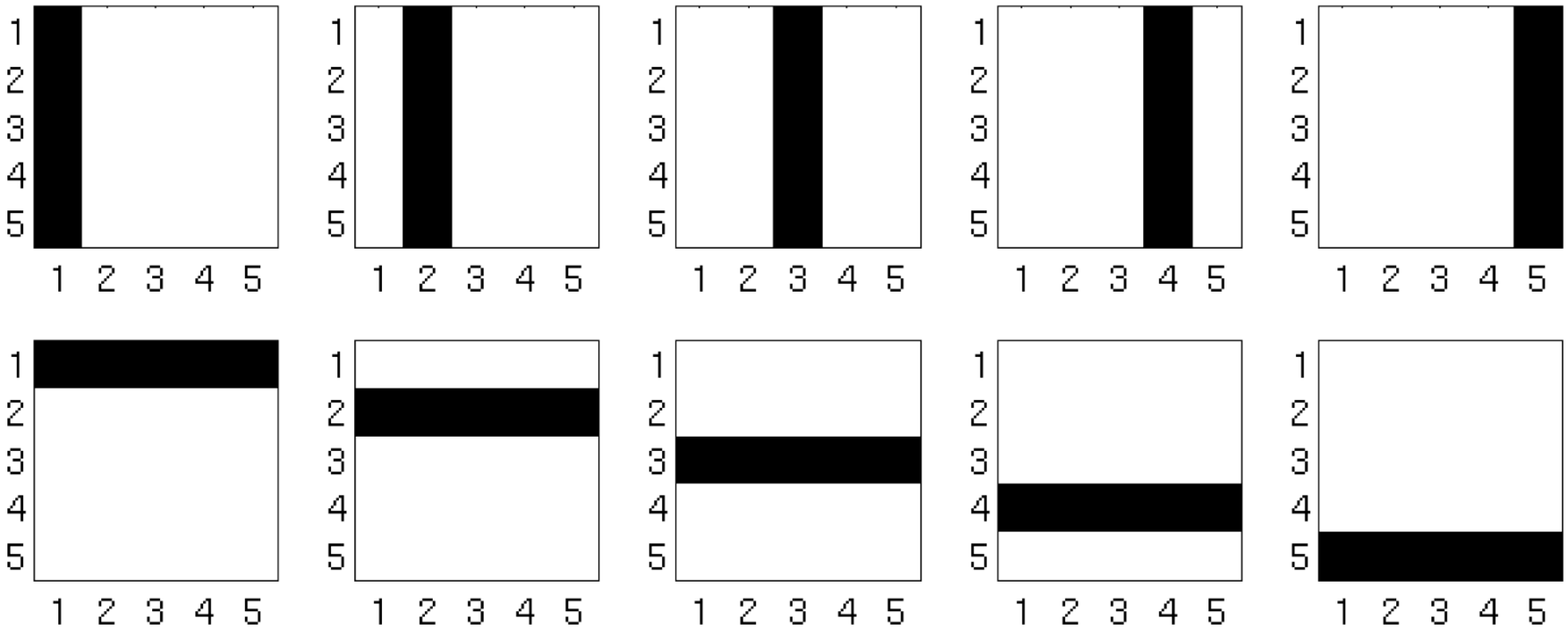
$$p(z_n = \blacksquare) \propto p(\blacksquare \mid \blacksquare)\, p(\blacksquare \mid d)$$

$$= p(w_n = \blacksquare \mid z_n = \blacksquare)\, p(z_n = \blacksquare \mid d)$$

- 色の確率 ∝ (その文書内での色の割合)
  × (その色から単語の出る確率)

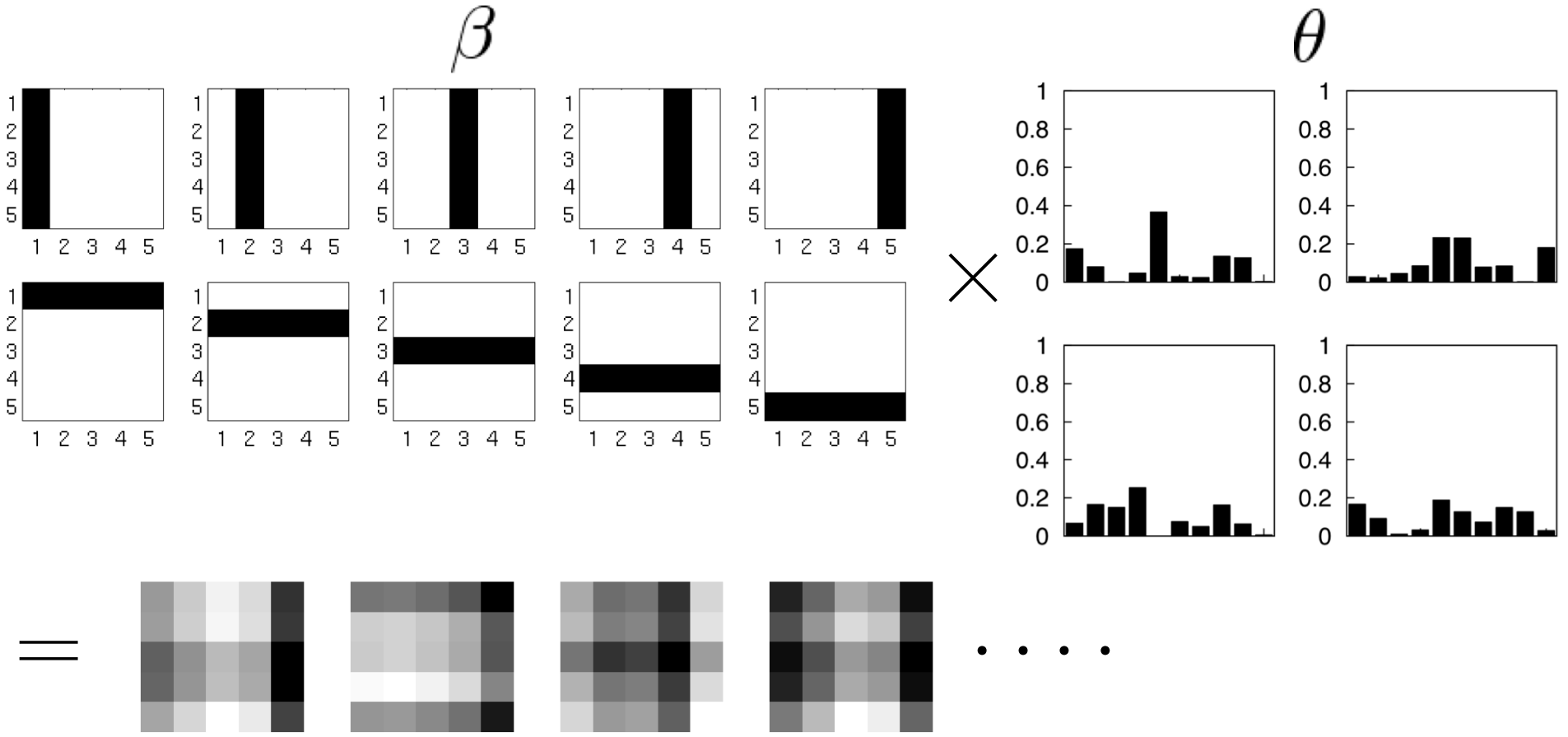
LDAのGibbs Sampler (4)：学習例

- 5x5=25単語上に10個の「トピック」β



  - 白は、その単語が出る確率が0
  - 黒は、その単語が出る確率が0.2

- θ ~Dir(1,1,…,1) で混ぜ合わせた単語分布



これらは真の値なので未知

実際に学習に使ったデータ

→ 単語

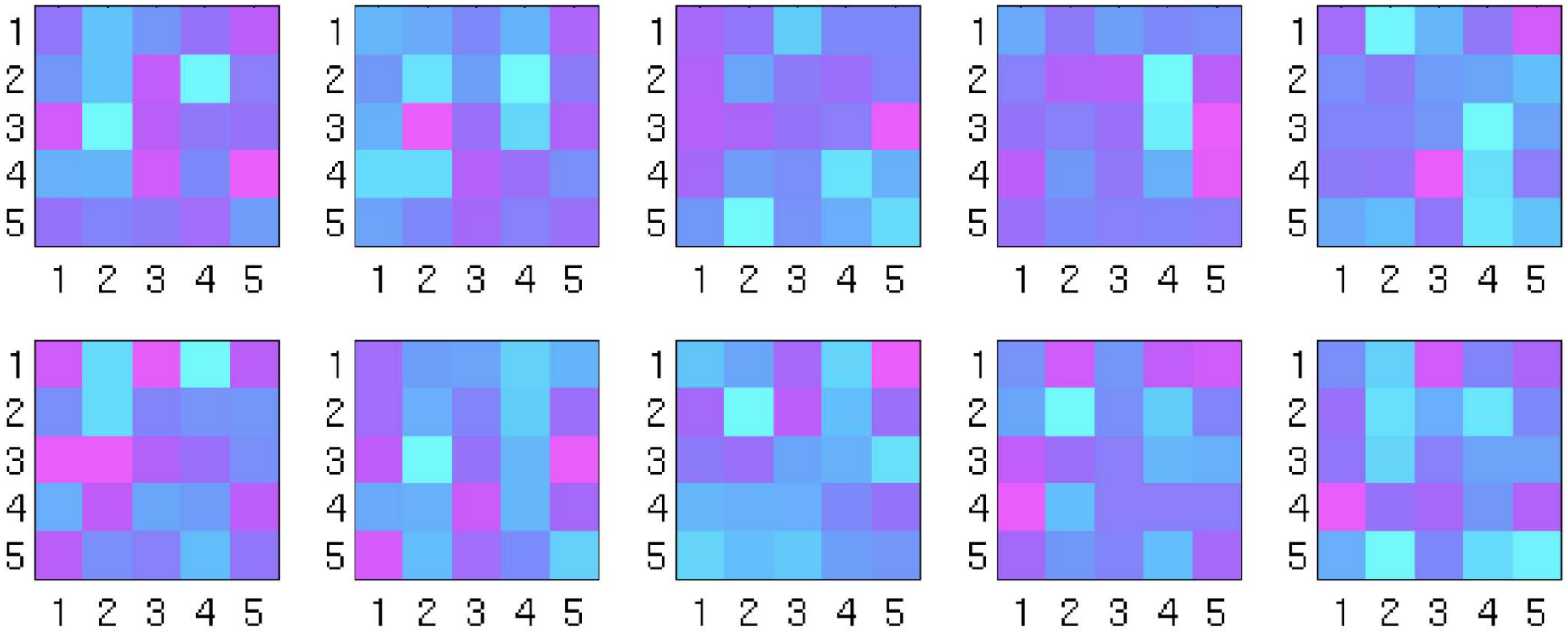
↓ 文書

10 10 08 12 07 06 09 11 04 01 00 00 05 03 00 05 02 06 10 01 19 16 25 17 13
14 04 05 01 09 03 02 10 04 05 10 08 07 02 07 21 07 03 04 05 23 09 13 14 10
07 05 07 08 03 12 10 11 06 08 09 05 15 14 03 14 17 19 06 08 04 04 03 02 00
09 08 19 09 05 10 09 11 06 03 08 05 06 06 03 08 04 10 09 02 07 10 12 13 08
07 00 03 03 01 06 03 01 02 05 13 16 29 16 13 03 02 07 01 04 09 13 13 10 20
03 06 00 14 02 17 14 10 20 09 05 11 06 07 05 09 23 07 11 08 02 05 00 06 00
05 03 09 06 07 15 05 07 06 09 15 03 08 07 05 08 06 03 09 09 24 06 09 09 07
04 03 08 05 04 08 08 14 05 12 07 03 11 03 05 11 10 11 04 10 19 08 14 06 07
06 00 03 02 04 09 02 08 06 04 04 01 11 11 04 13 08 07 08 06 15 17 18 24 09
11 11 29 07 11 05 01 15 00 03 06 06 12 03 01 08 04 19 07 08 06 07 15 04 01
07 08 11 10 03 03 03 06 03 01 13 05 10 05 04 04 07 09 05 01 21 17 12 16 16
05 04 04 06 15 14 03 12 15 23 04 01 01 04 12 06 01 11 06 15 02 04 09 06 17
:

- 真の値から1,000文書を生成
- 観測値は上の頻度データだけ

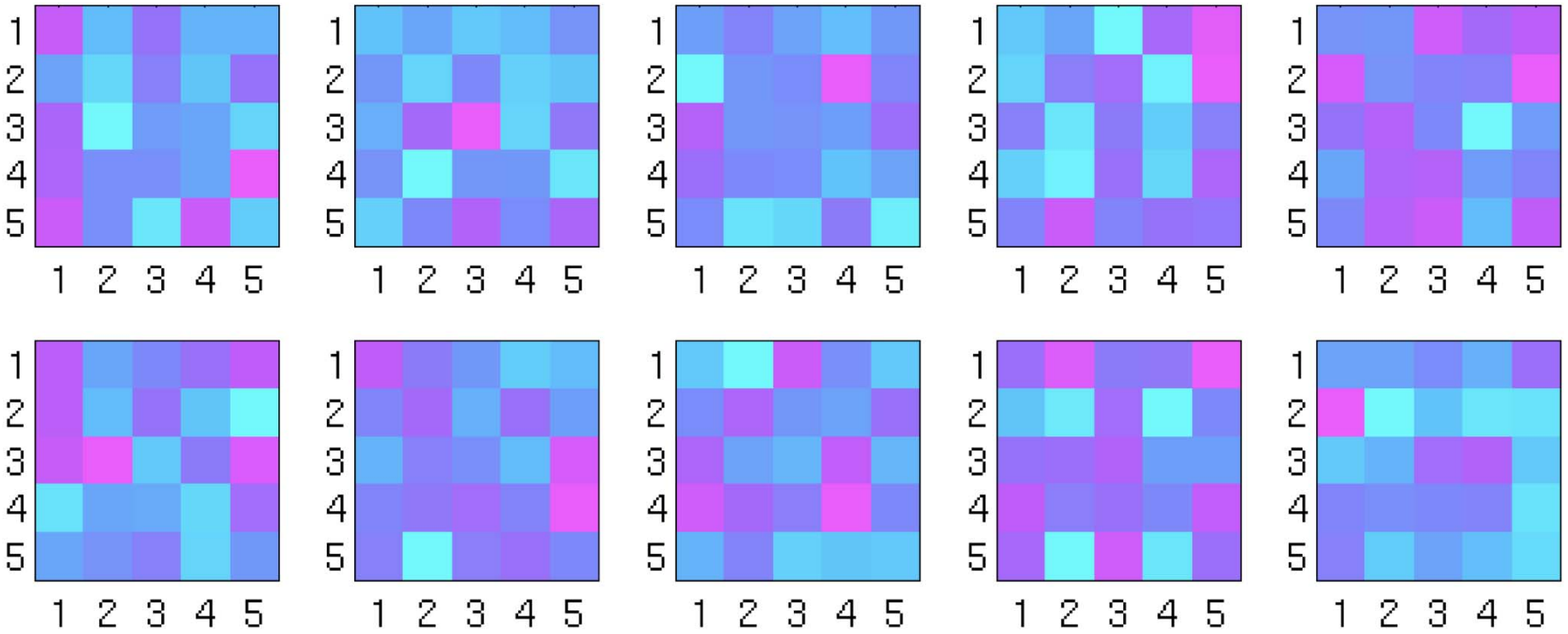
トピック分布 β の学習経過

- Gibbs iteration = 1 (乱数で初期化)



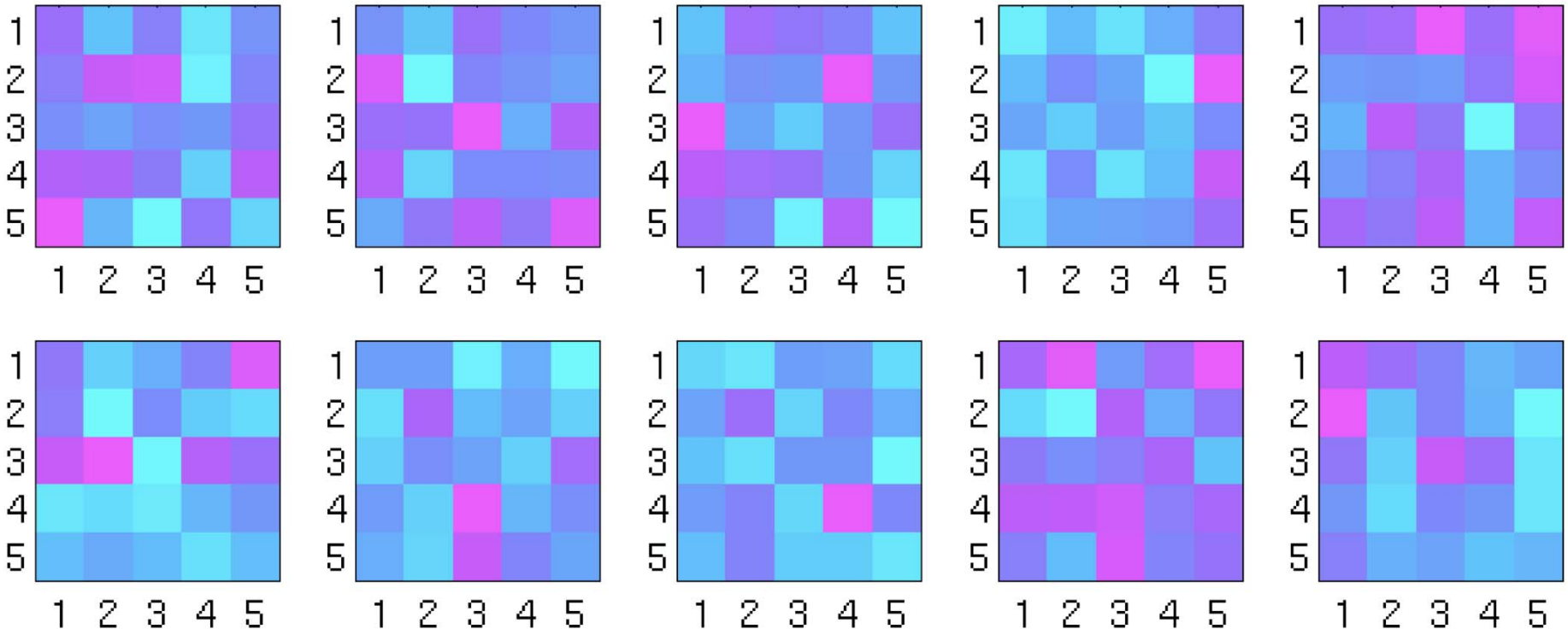
トピック分布 β の学習経過

- Gibbs iteration = 2



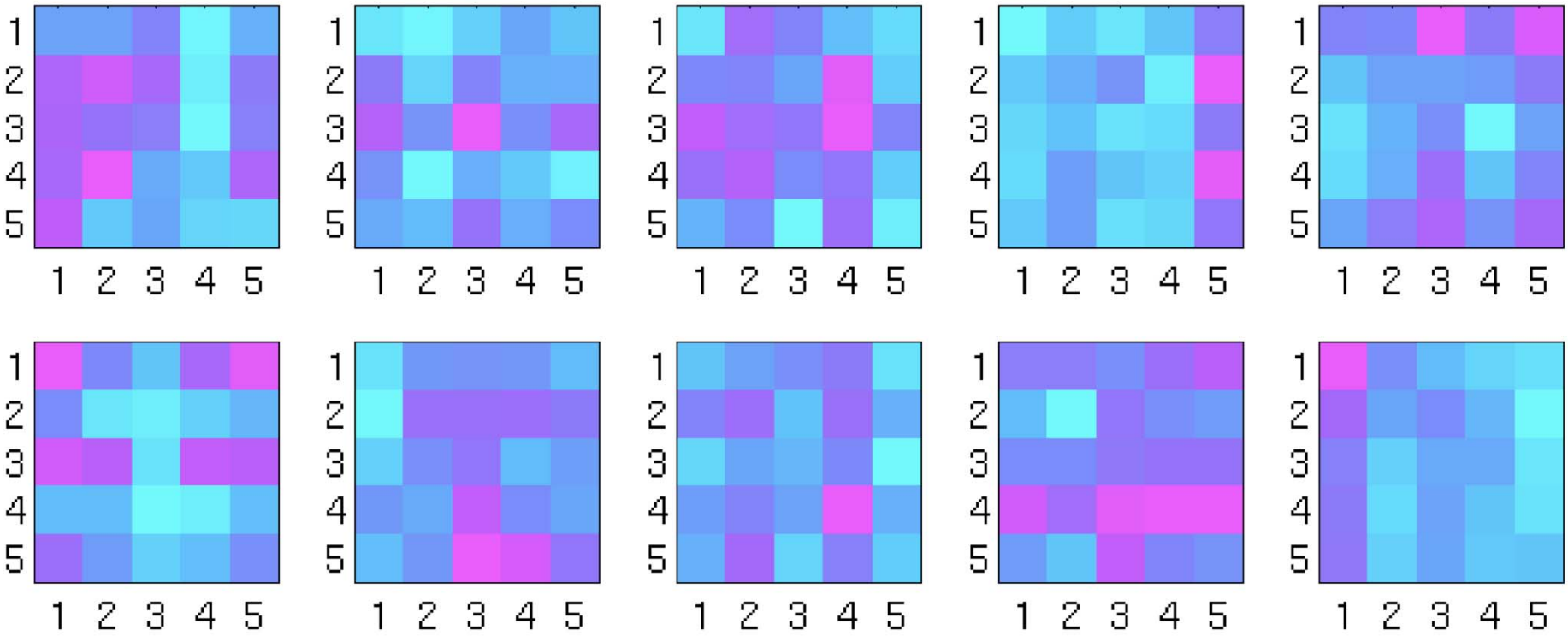
トピック分布 β の学習経過

- Gibbs iteration = 4



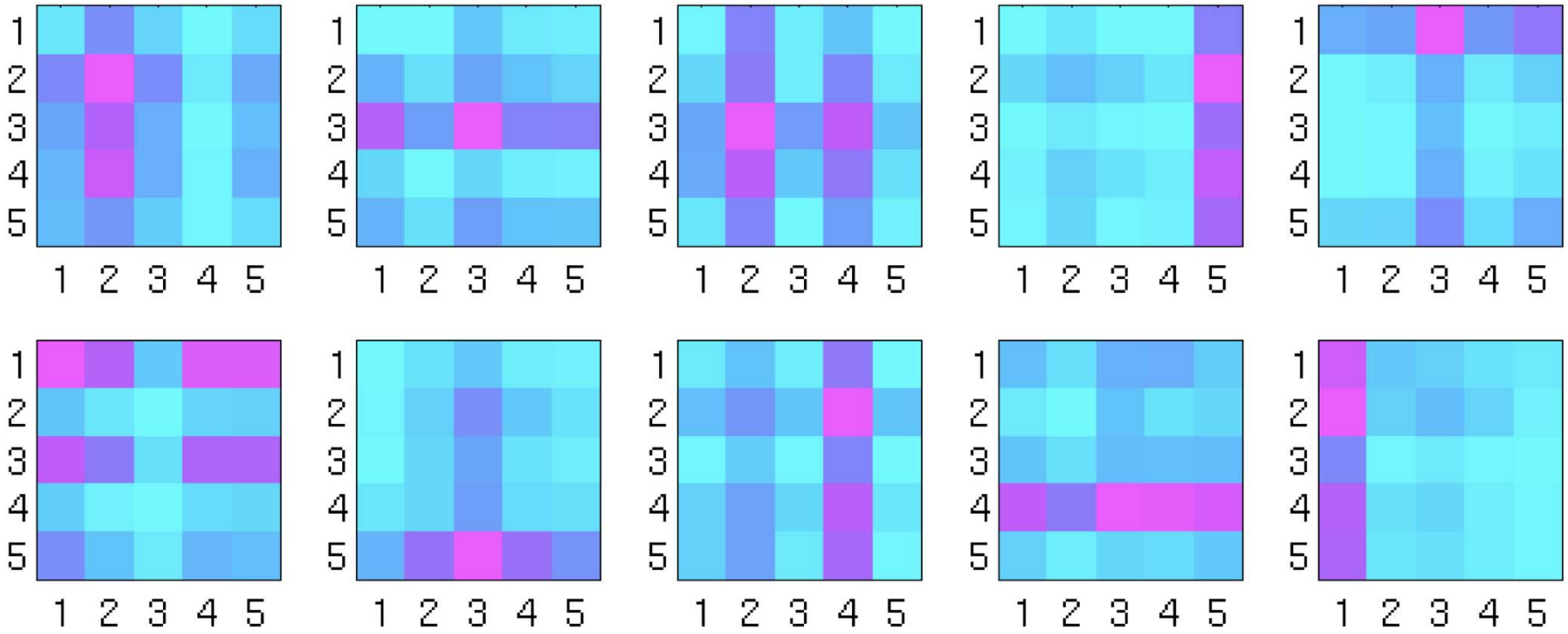
トピック分布 β の学習経過

- Gibbs iteration = 8



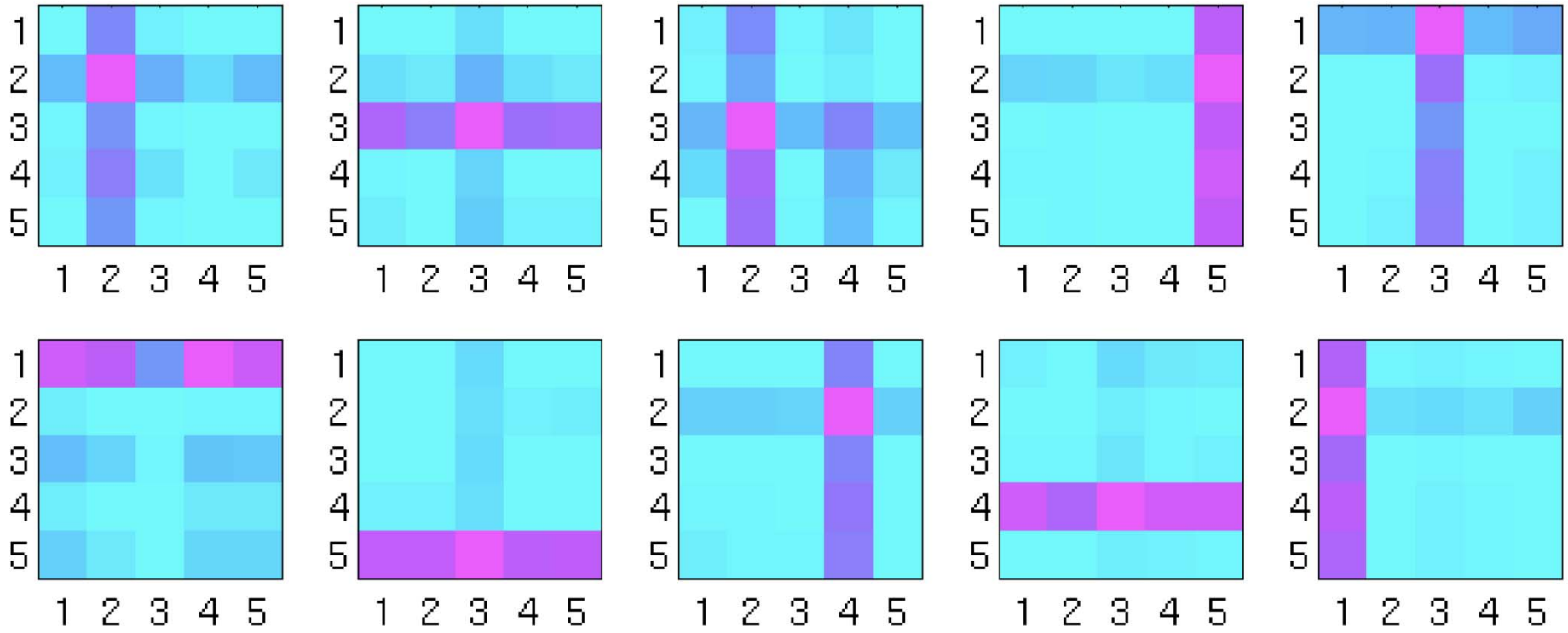
トピック分布 β の学習経過

- Gibbs iteration = 16



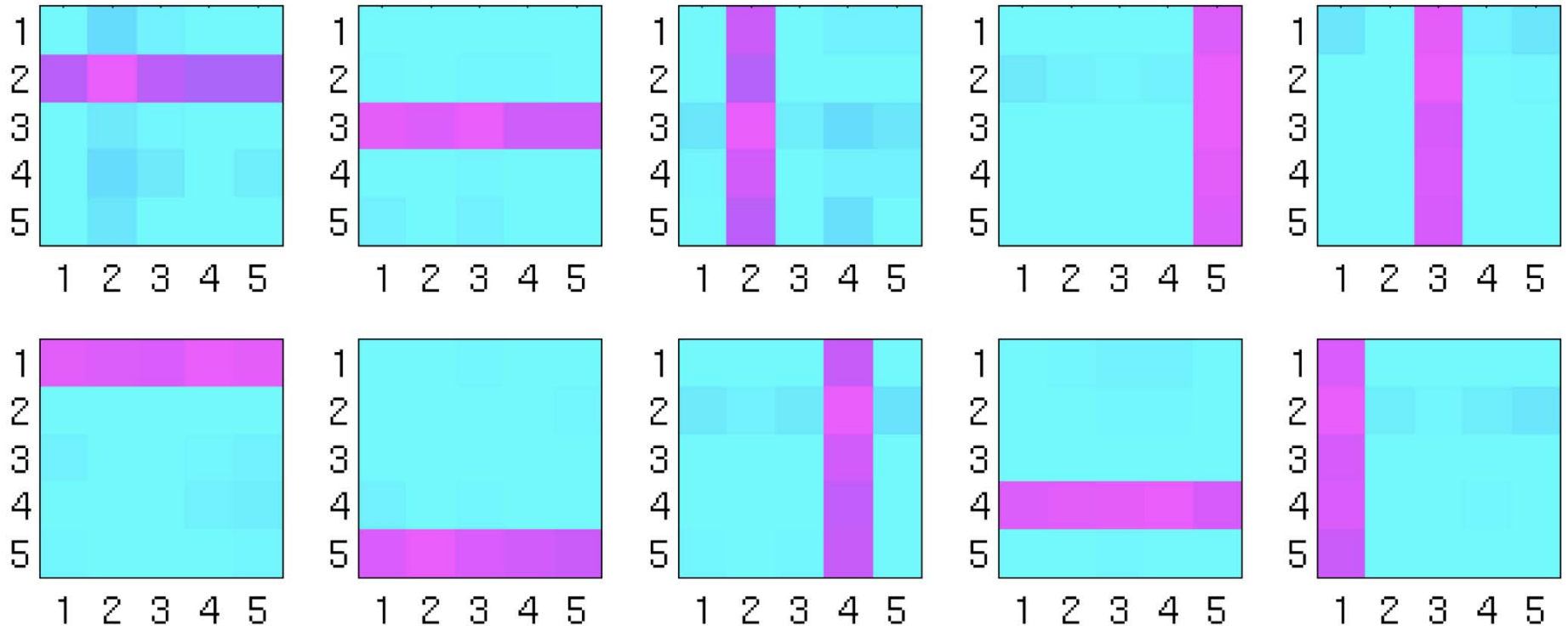
トピック分布 β の学習経過

- Gibbs iteration = 32



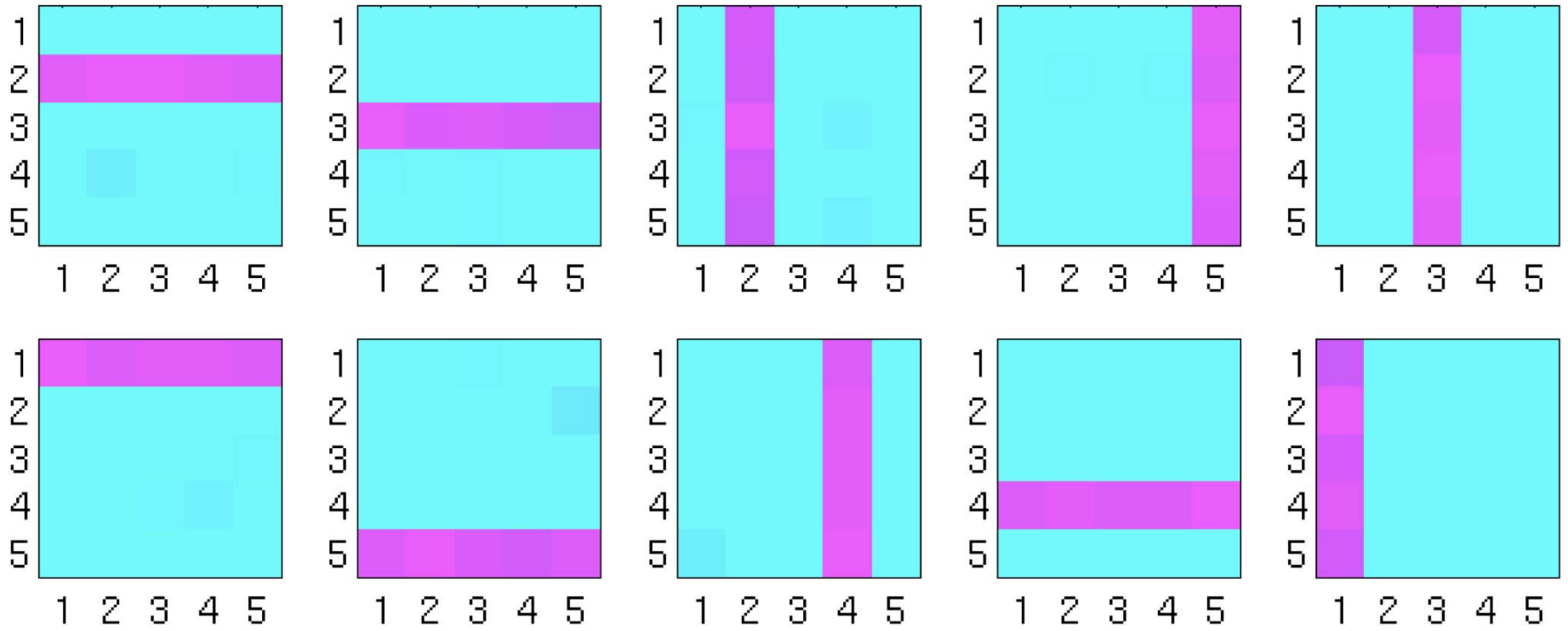
トピック分布 β の学習経過

- Gibbs iteration = 64



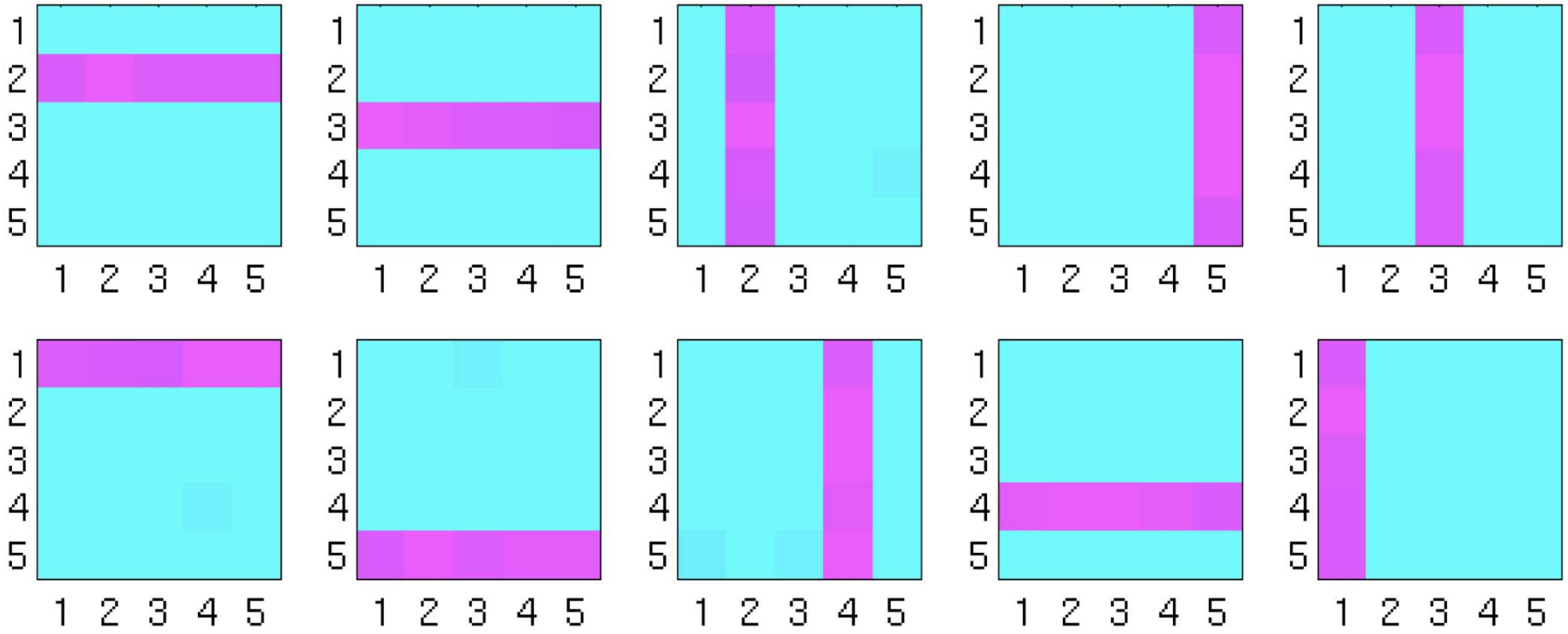
トピック分布 β の学習経過

- Gibbs iteration = 128

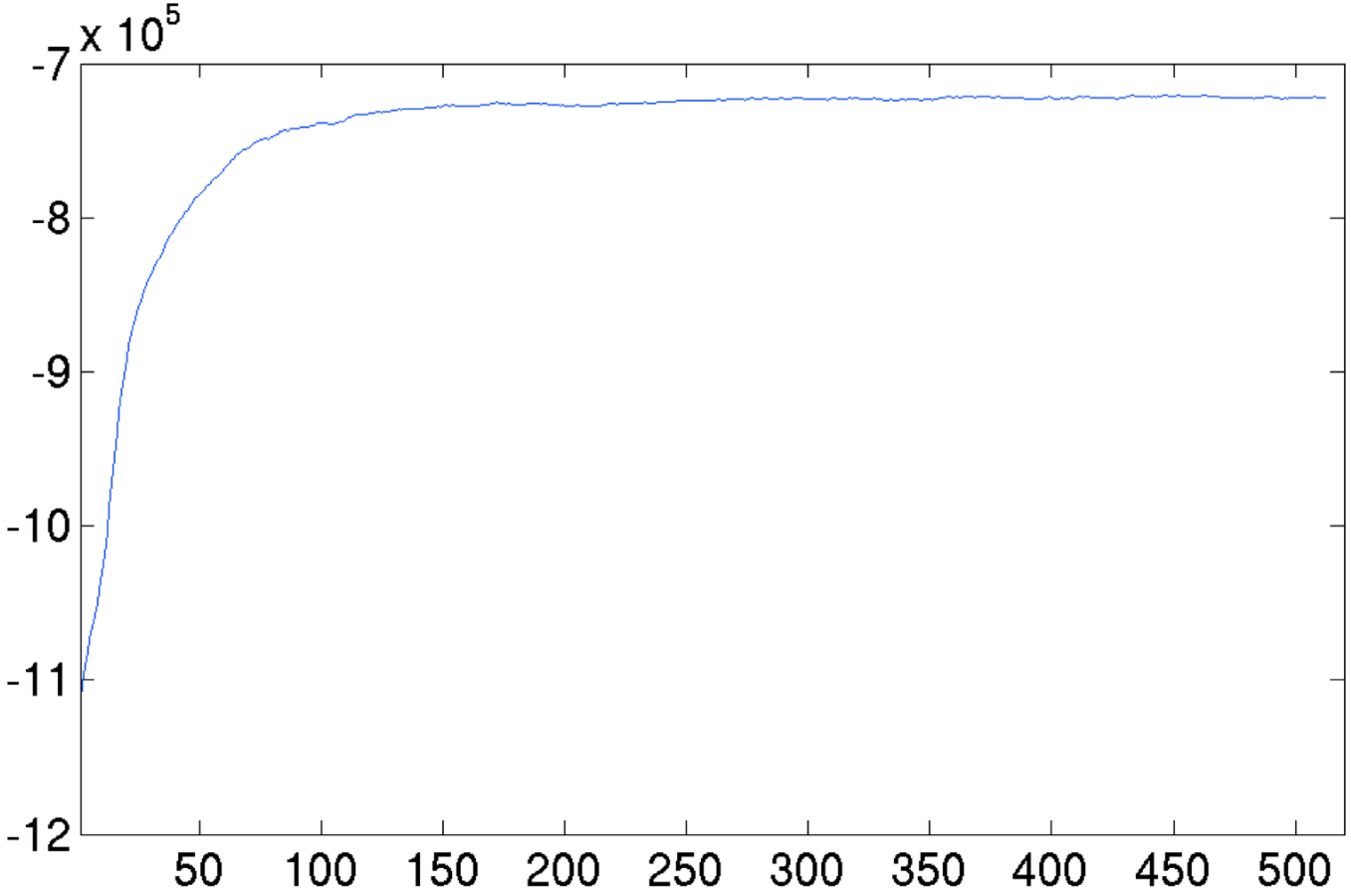


トピック分布 β の学習経過

- Gibbs iteration = 256



対数尤度の変化



↑
データの
対数尤度

Iterations

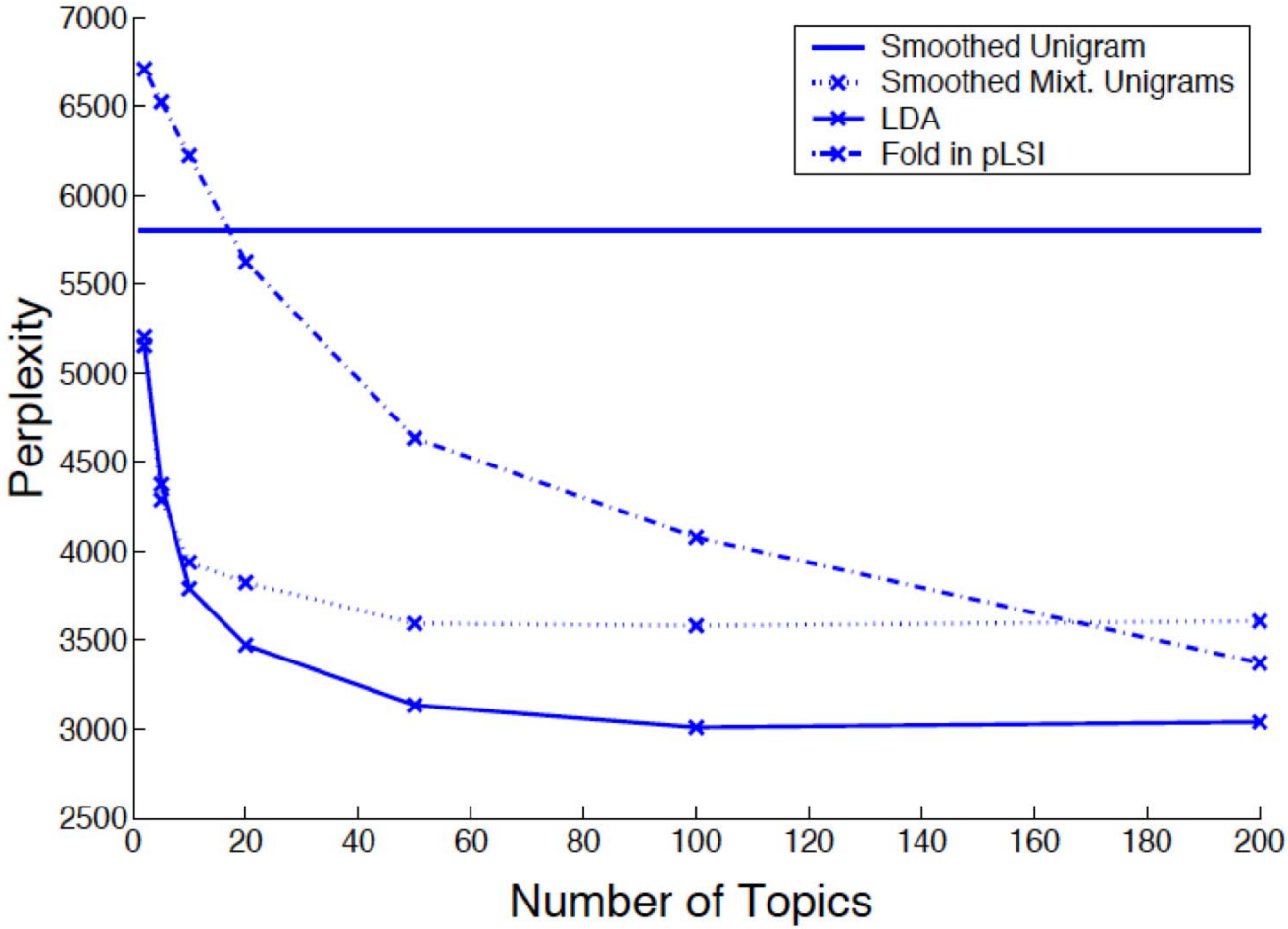
- 200 iterationあたりでほぼ収束

LDAとPLSIの比較

- LDAは、PLSIのベイズ化
- 様々な良い点 (デファクトスタンダード)
  - オーバーフィットしない
  - 完全な階層ベイズ生成モデル
    → 様々な拡張が可能 (明日の講義)
    - PLSIでは p(z|d) に由来がないため、これ以上手をつけることができない
  - 計算量のオーダーはほとんど同じ
    - モデルが完全なので、色々なオンライン推定法も提案されている

LDAとPLSIの比較 (2)

- PPLの比較 (Blei+ 2003より)

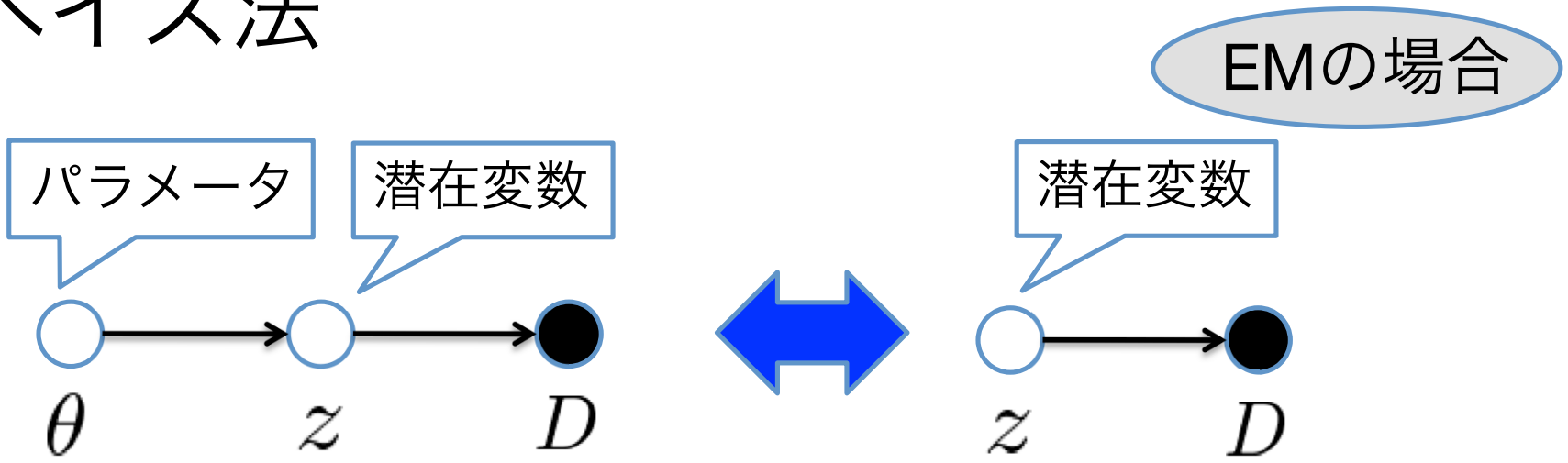


現在では、単純なLDAよりさらに性能の良いモデルが開発されている

LDAのEM学習: 変分ベイズ法

- LDAの潜在変数‥ θ と z
  - 二層なので、普通のEMアルゴリズムが使えない!
    - (LDAの場合は、Gibbs Samplerのように、θ を積分消去すればいいのでは?→Collapsed変分ベイズEM)
  - 階層ベイズ学習では、こういう状況はよくある
- どうすればよい?

変分ベイズ法



- パラメータ θ とzが両方とも潜在変数になっている確率モデル
  - θ とzに依存関係がある [組み合わせ爆発]
  - この下で、データDの確率

$$p(D) = \int\int p(D, z, \theta) dz d\theta$$

を最大化したい

変分ベイズ法 (2)

- Variational Bayes: Attias (UAI 1999)により提案
- まず、Jensenの不等式より、最大化したい確率は

$$\begin{aligned}\log p(D) &= \log \int\int p(D, z, \theta) dz d\theta \\ &= \log \int\int q(z, \theta|D) \frac{p(D, z, \theta)}{q(z, \theta|D)} dz d\theta \\ &\geq \int\int q(z, \theta|D) \log \frac{p(D, z, \theta)}{q(z, \theta|D)} dz d\theta\end{aligned}$$

と下限を作れることに注意する

変分ベイズ法 (3)

$$\log p(D) \geq \int\int q(z, \theta|D) \log \frac{p(D, z, \theta)}{q(z, \theta|D)} dz d\theta$$

- ここで、z, θ には本来依存関係があるが、
  近似分布として

$$q(z, \theta|D) = q(z|D)q(\theta|D)$$

  という分解 (因子化仮定, 平均場近似) を仮定すると

qは近似なので、
任意に作れる

変分ベイズ法 (4)

$$\log p(D) \geq \int\int q(z|D)q(\theta|D) \log \frac{p(D,z,\theta)}{q(z|D)q(\theta|D)} dz d\theta$$
$$= F(q(z|D), q(\theta|D))$$

- この下限 (変分下限、Variational lower bound) $F(q(z|D), q(\theta|D))$ は $q(z|D), q(\theta|D)$ について逐次最大化できる！

VB-EMアルゴリズム.

– *F*は、変分自由エネルギーともよばれる

変分ベイズ法 (5)

- $q(z)$ について最大化 ($q(z|D)=q(z)$ と書いた)

$$L = F + \lambda\left(\int q(z)dz - 1\right)$$

$$= \int\int q(z)q(\theta)\log\frac{p(D,z,\theta)}{q(z)q(\theta)}dzd\theta$$

$$\frac{\delta L}{\delta q(z)} = \int\int q(\theta)\Big[\log p(D,z,\theta) - \log q(\theta) - \log q(z) - 1\Big]dzd\theta + \lambda$$

$$= \int\int q(\theta)\Big[\log p(D,z|\theta) + \log p(\theta) - \log q(\theta) - \log q(z) - 1\Big]dzd\theta + \lambda$$

$$= \Big\langle \log p(D,z|\theta)\Big\rangle_{q(\theta)} - \log q(z) + (\text{const.}) + \lambda = 0$$

$$q(z) \propto \exp\Big\langle \log p(D,z|\theta)\Big\rangle_{q(\theta)}.$$

変分ベイズ法 (6)

- $q(\theta)$ について最大化 ($q(\theta|D)=q(\theta)$ と書いた)

$$
\begin{aligned}
L &= F + \lambda\left(\int q(\theta)d\theta - 1\right) \\
&= \int\int q(z)q(\theta)\log\frac{p(D,z,\theta)}{q(z)q(\theta)}dzd\theta + \lambda\left(\int q(\theta)d\theta - 1\right) \\
\frac{\delta L}{\delta q(\theta)} &= \int\int q(z)\Big[\log p(D,z,\theta) - \log q(\theta) - \log q(z) - 1\Big]dzd\theta + \lambda \\
&= \int\int q(z)\Big[\log p(D,z|\theta) + \log p(\theta) - \log q(\theta) - \log q(z) - 1\Big]dzd\theta + \lambda \\
&= \Big\langle \log p(D,z|\theta)\Big\rangle_{q(z)} + \log p(\theta) - \log q(\theta) + (\text{const.}) + \lambda = 0
\end{aligned}
$$

$$
q(\theta) \propto p(\theta)\exp\Big\langle \log p(D,z|\theta)\Big\rangle_{q(z)}.
$$

変分ベイズ法 (7)

- $q(z)$ と $q(\theta)$ に依存関係→変分ベイズEMアルゴリズム (VB-EM)

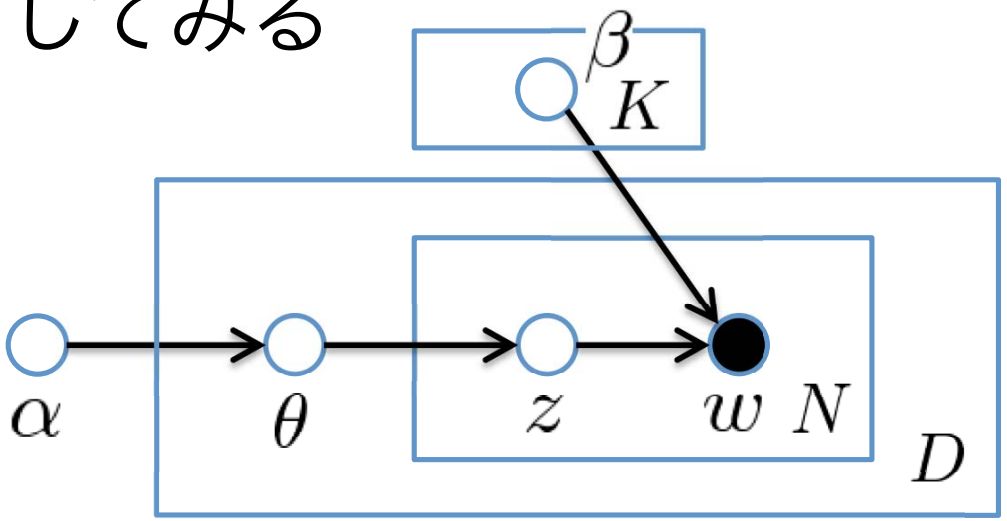
  Eステップ： $q(z) \propto \exp\left\langle \log p(D, z|\theta) \right\rangle_{q(\theta)}$.

  Mステップ: $q(\theta) \propto p(\theta) \exp\left\langle \log p(D, z|\theta) \right\rangle_{q(z)}$.

  - 下限が収束するまで繰り返す
  - 変分下限を逐次最大化している
  - Mステップが点推定 $\hat{\theta}$ の場合は、普通のEMアルゴリズムと同じ

LDAのVB-EM学習

- VB-EMを、LDAに適用してみる



$$p(\mathbf{w}) = \int \sum_z p(\mathbf{w}, z, \theta) d\theta$$

$$= \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \int \left(\prod_k \theta_k^{\alpha_k - 1} \right) \prod_n \sum_k p(w_n|k) \theta_k d\theta$$

LDAのVB-EM学習 (2)

$$\log p(\mathbf{w}|\alpha,\beta) = \log \int \sum_z p(\mathbf{w},z,\theta|\alpha,\beta)d\theta$$
$$= \log \int \sum_z q(z,\theta) \log \frac{p(\mathbf{w},z,\theta)}{q(z,\theta)} d\theta$$
$$\geq \int \sum_z q(z)q(\theta) \log \frac{p(\mathbf{w},z,\theta)}{q(z)q(\theta)} d\theta$$

- $q(z)$ は各単語の持つ潜在トピックの近似推定、
  $q(\theta)$ は各文書の持つ潜在トピック分布の近似推定

LDAのVB-EM学習 (3)

$$p(\mathbf{w}, z, \theta|\alpha, \beta) = p(\theta|\alpha) \prod_n p(z_n|\theta)p(w_n|z_n)$$

より、

$$\int \sum_z q(z)q(\theta) \log \frac{p(\mathbf{w}, z, \theta|\alpha, \beta)}{q(z)q(\theta)} d\theta$$

$$= \left\langle \log p(\theta|\alpha) \right\rangle_{q(\theta)} + \sum_n \left\langle \log p(z_n|\theta) \right\rangle_{q(\theta),q(z)}$$

$$+ \sum_n \left\langle \log p(w_n|z_n) \right\rangle_{q(z)} - \left\langle \log q(\theta) \right\rangle_{q(\theta)} - \sum_n \left\langle \log q(z) \right\rangle_{q(z)}$$

- 具体的にp, qにディリクレ分布や多項分布を代入して計算

LDAのVB-EM学習 (4)

- 頑張って解くと、

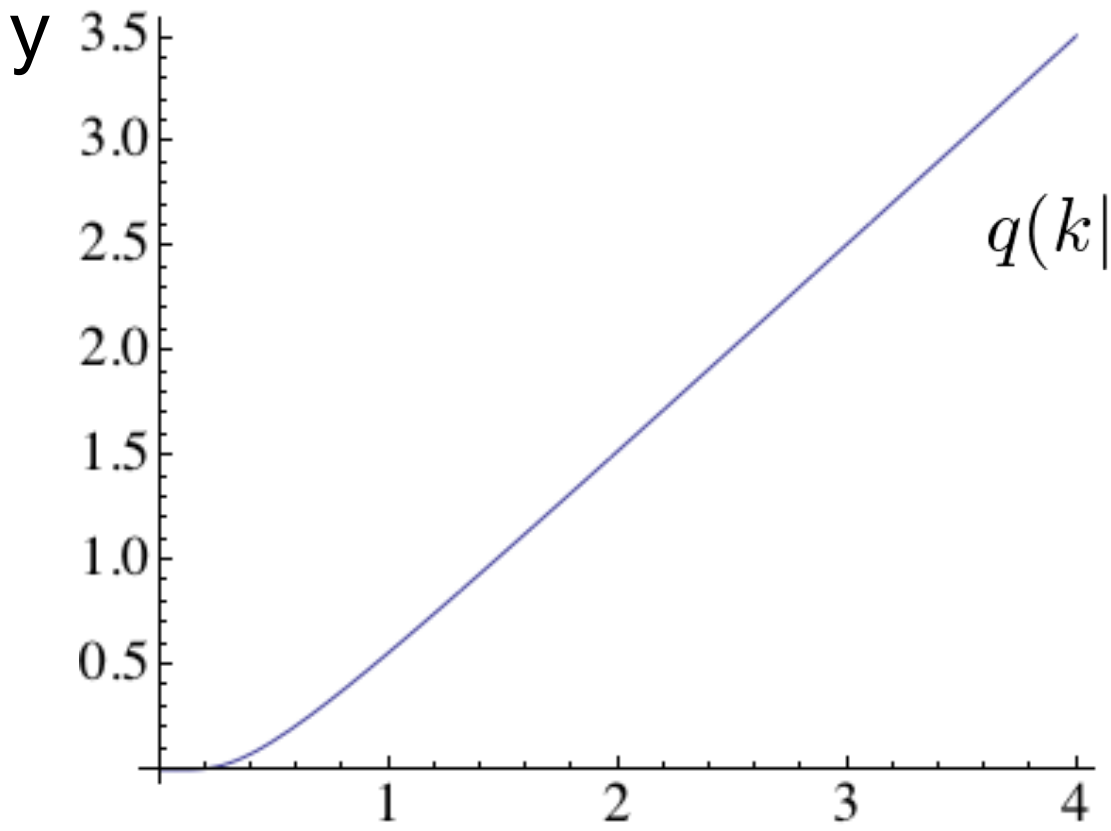
$$\begin{cases} q(k|w) \propto p(w|k) \exp\left(\Psi(\gamma_k)\right) \\ q(\theta|\mathbf{w}) \sim \mathrm{Dir}(\gamma), \ \ where\ \gamma_k = \alpha_k + \sum_{w \in \mathbf{w}} q(k|w) \end{cases}$$

  - ここで、$\Psi(x) = \dfrac{d}{dx} \log \Gamma(x)$ はΨ関数とよばれる

- α, β については、Newton法などで最尤推定
  - 実装:
    http://www.ism.ac.jp/~daichi/dist/lda/lda-0.1.tar.gz

LDAのVB-EM学習 (5)

- y=exp(Ψ(x))のグラフ



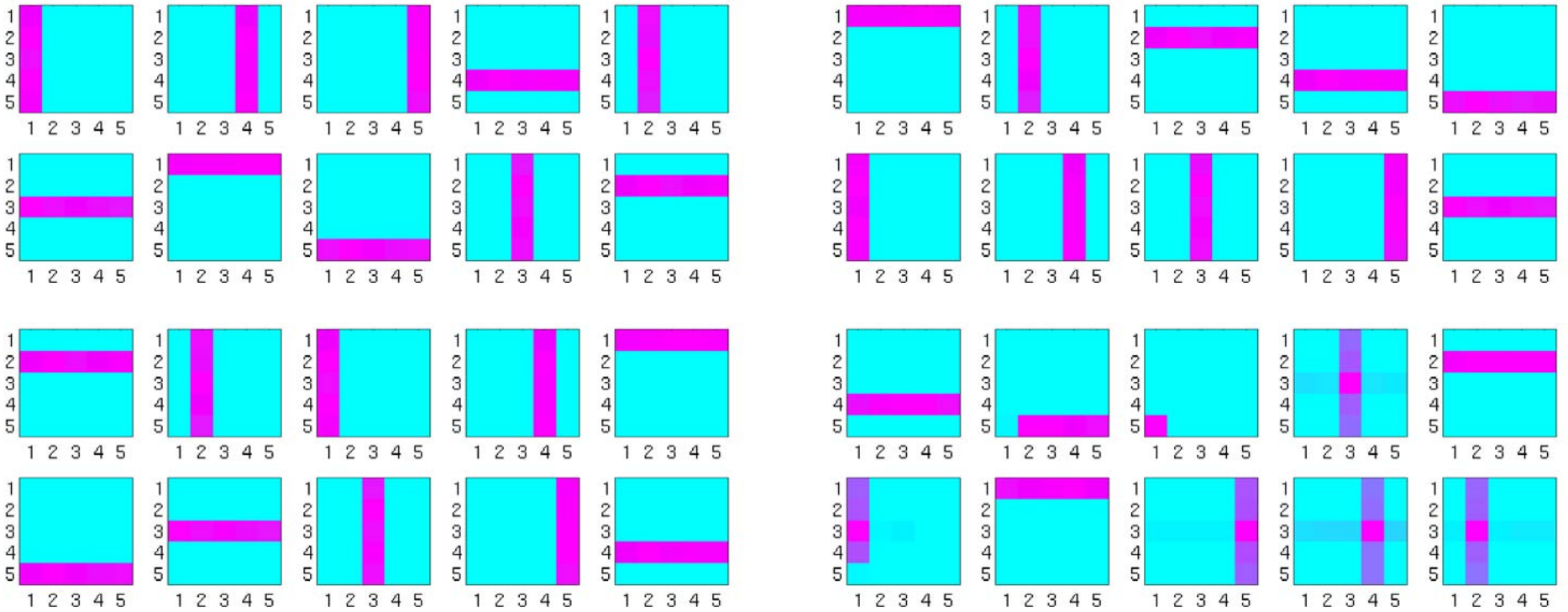
exp(Ψ(x))～x-0.5

$$q(k|w) \propto p(w|k)\exp\left(\Psi(\gamma_k)\right)$$

は、小さい頻度を0につぶしてスパースにしている効果

LDAのVB-EM学習 (例)

- 前のGibbsの例をVBで計算したもの



初期化により、まれに局所解

LDAのVB-EM学習 (例.cont)

- ただし、収束した尤度を見れば局所解か判断可能

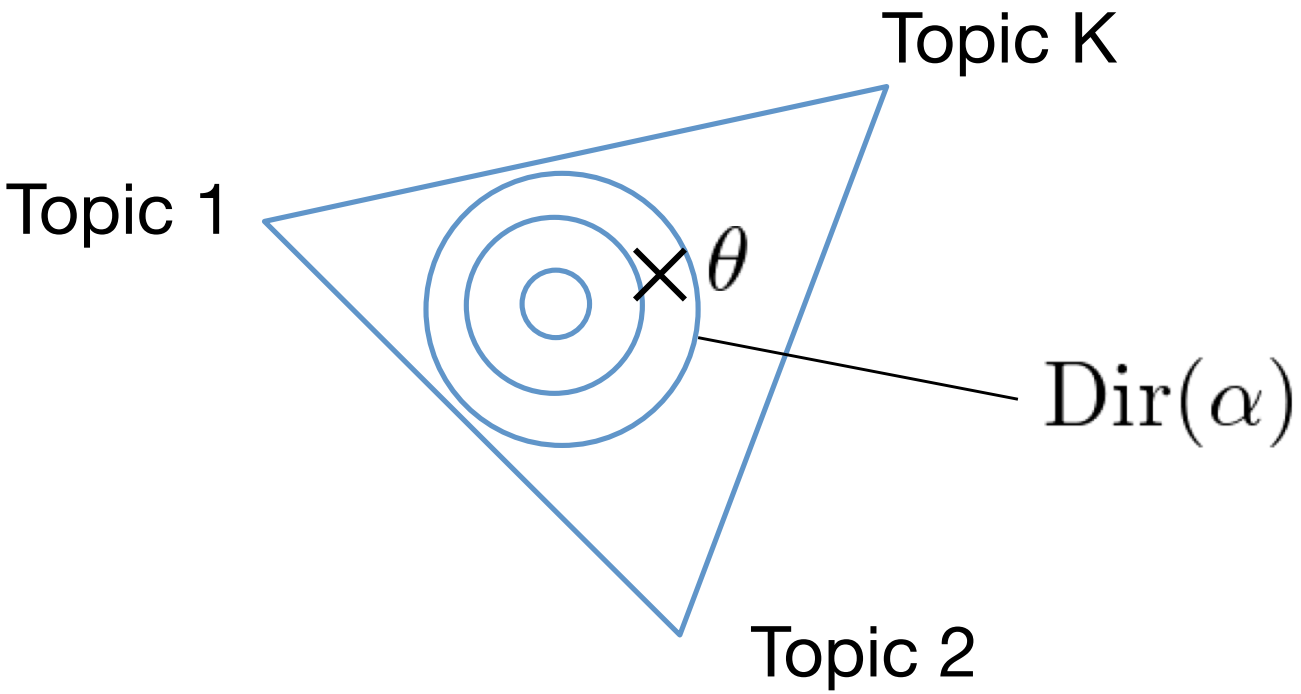
```
% ./lda -N 10 -I 200 -E 1e-6 piclda.dat piclda.model
iteration 78/200..      likelihood = -603842    ETA: 0:00:04 (0 sec/step)
converged. [ 0:00:03]
writing model..
done.
% ./lda -N 10 -I 200 -E 1e-6 piclda.dat piclda.model
writing model..00..     likelihood = -604862    ETA: 0:00:00 (0 sec/step)
done.
```

  - 計算はかなりVBが速い (言語に依存)

LDAの計算法について

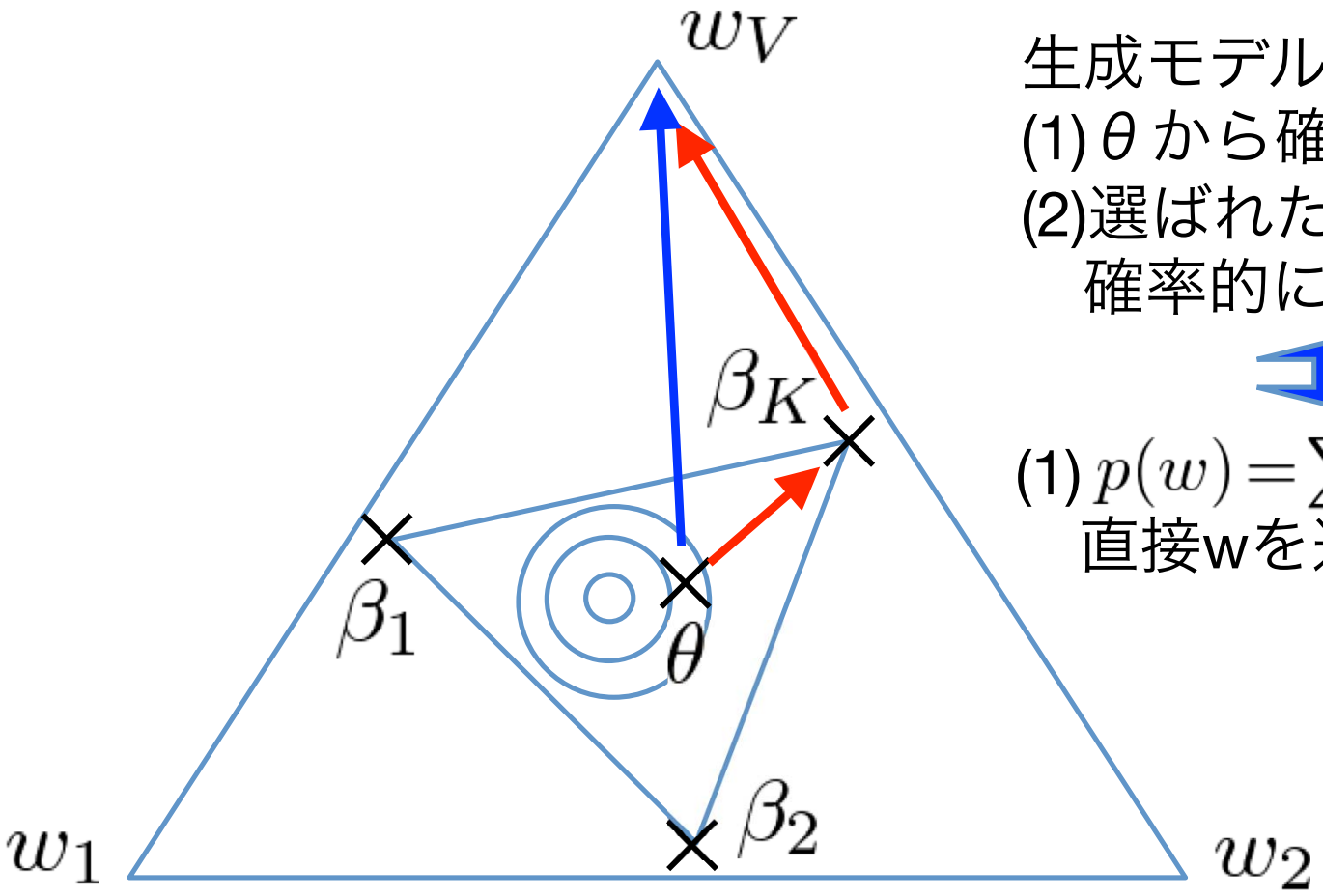
- 現代的には、素のLDAの学習にはCVB (Collapsed Variational Bayes) または Gibbs が薦められる
  - CVBについては、参考文献を参照
- オンライン学習法もある (Hofmann10,Sato 10など)
- 複雑な拡張モデルについては、VBまたはGibbsをそれぞれ構成 (明日の話)
  - VBは下限の導出が難解だが、高速に動作
  - Gibbsは導出が易しく性能が高いが、収束判定が難しい

LDA in Geometry



- LDA‥文書ごとに、トピック単体上のディリクレ分布からトピック分布θを選ぶ
  - 単体の角がトピックに対応

LDA in Geometry (2)



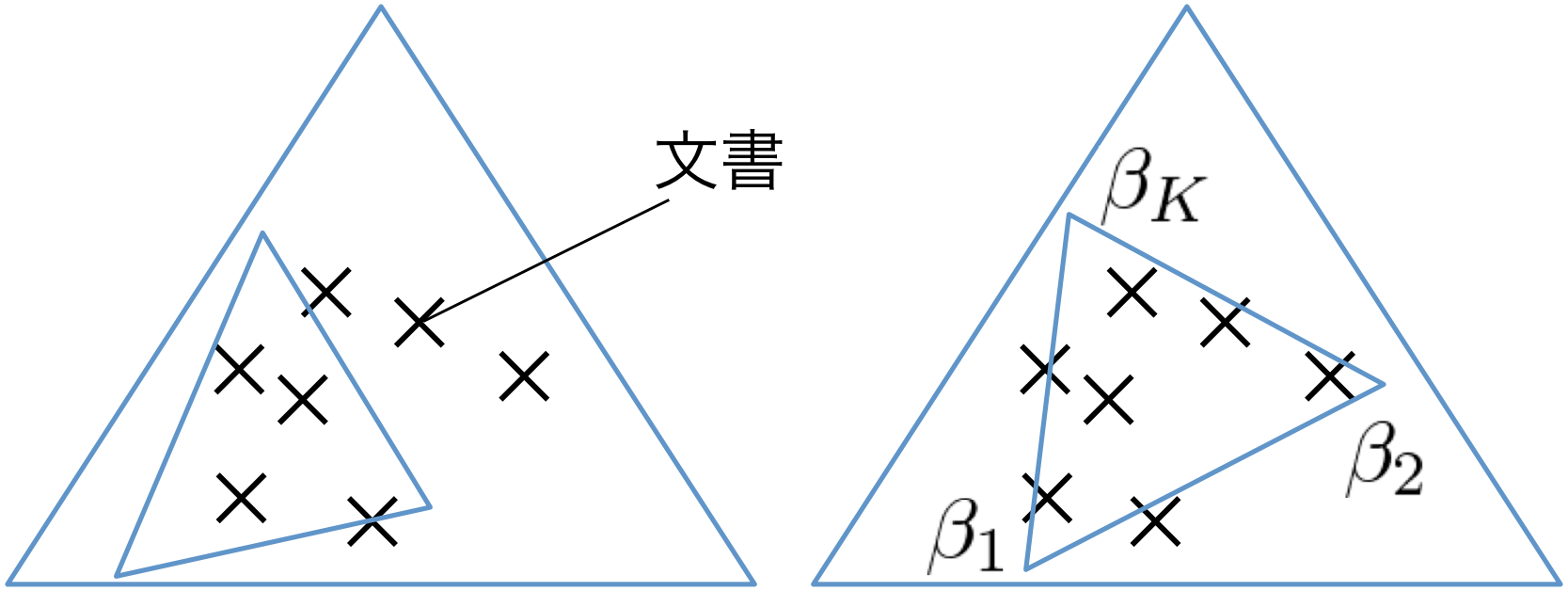
生成モデル:
(1) θ から確率的に頂点 β を選ぶ
(2)選ばれた頂点 β =p(w|k)から、
確率的に単語wを選ぶ

⇔

(1) $p(w) = \sum_k \theta_k p(w|k)$ から、
直接wを選ぶ

- 各トピックは、単語単体上の一点
  →トピック単体は、単語単体に埋め込まれている

LDAの最適化



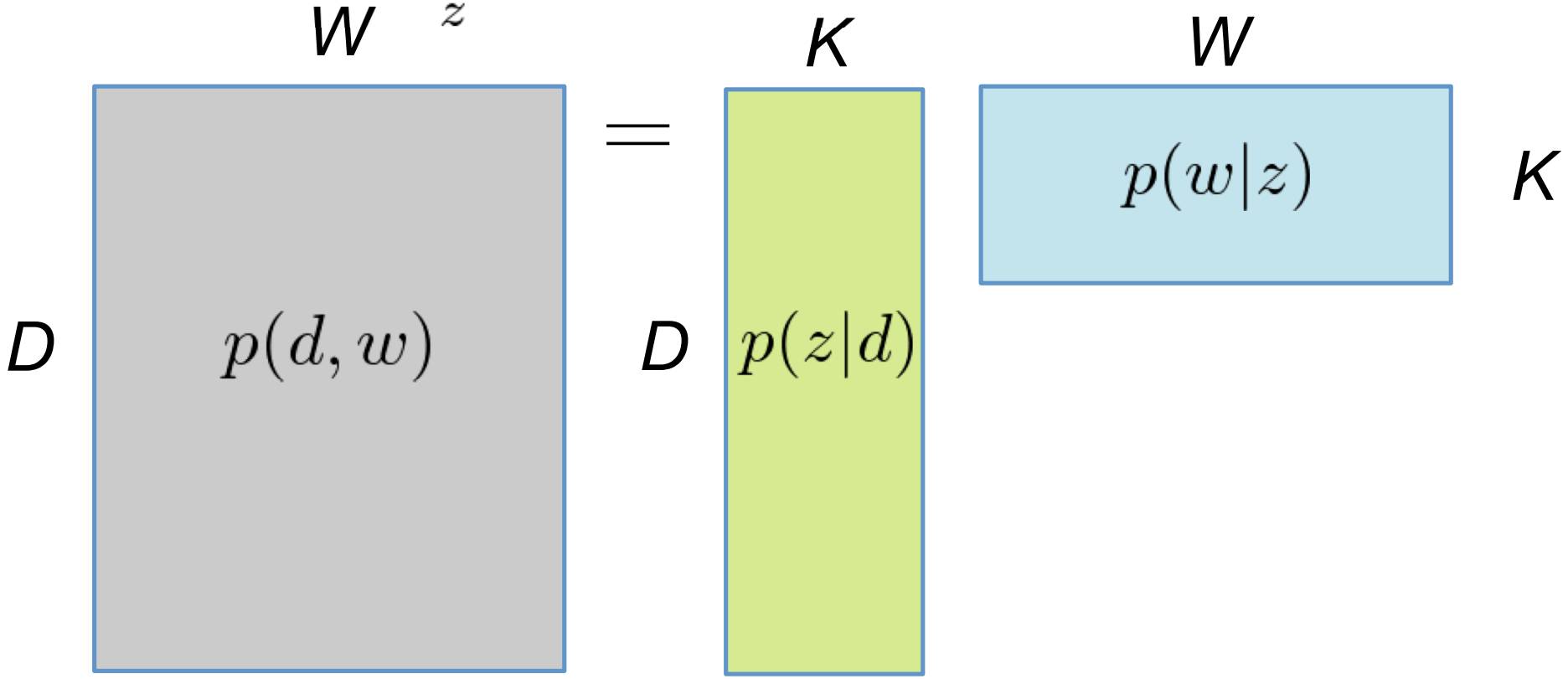
- LDAは、文書群を表現できる低次元のトピックSubsimplexを見つける問題.
  - トピック分布 $\beta_1, \beta_2 \sim \beta_K$ の最適化

Lecture 4

より進んだトピックモデル

Matrix view of PLSI (LDA)

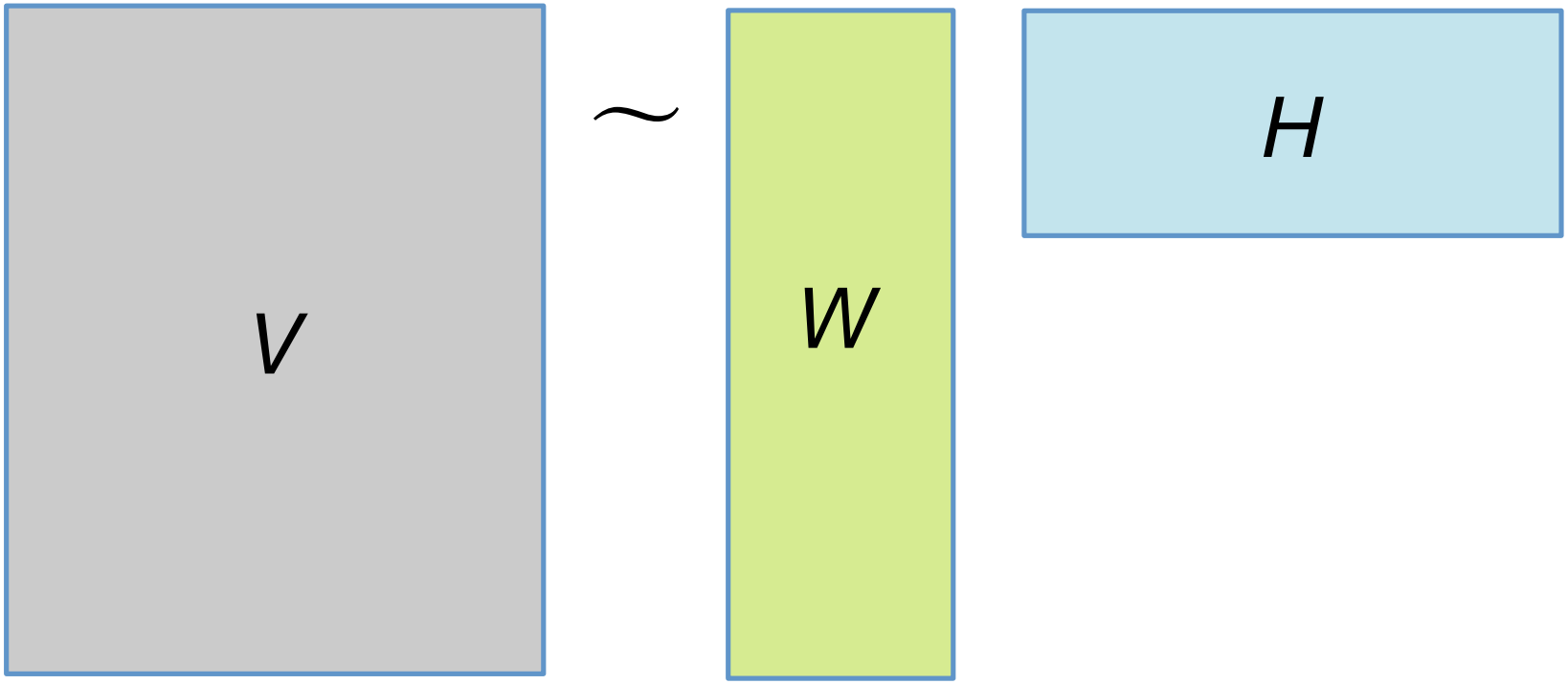
$$p(d,w) = \sum_{z} p(w|z)\, p(z|d)$$



- PLSIは、観測行列Xの低次元行列による分解

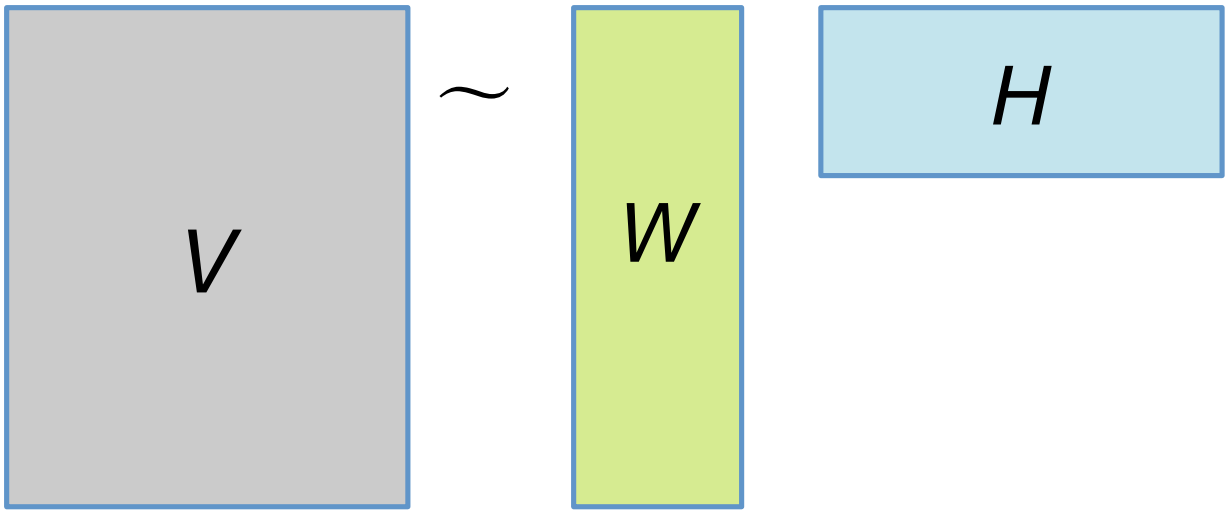
NMF (Nonnegative Matrix Factorization)

Lee & Seung (2001)



- $D(V||WH)$ を最小化→EMアルゴリズム
  - W,Hが正規化されていないが、PLSIとほぼ同じ

NMF (2)



$$I = \sum_{ij} v_{ij} \log\left(\sum_{k} w_{ik} h_{kj}\right) - \sum_{k} w_{ik} h_{kj}$$ →最小化

- ここで $\log \mathrm{Po}(x|\lambda) = x \log \lambda - \lambda - \log x!$ より、
  NMFはポアソン分布の下で、$V \sim \mathrm{Po}(WH)$
  となる低次元のW,Hを求めていることに相当する

NMF→GaP

- NMFは最尤推定‥‥オーバーフィットの危険
- Hにガンマ事前分布を入れる
  → GaP (Gamma-Poisson)モデル (Canny 2004)

E Step:

$$X_{ik} = X_{ik} \left(\sum_{j=1}^{m} \frac{F_{jk}}{Y_{jk}} \Lambda_{ji} + \frac{a_i - 1}{X_{ik}} \right) / \left(\sum_{j=1}^{m} \Lambda_{ji} + \frac{a_i}{c_i} \right)$$

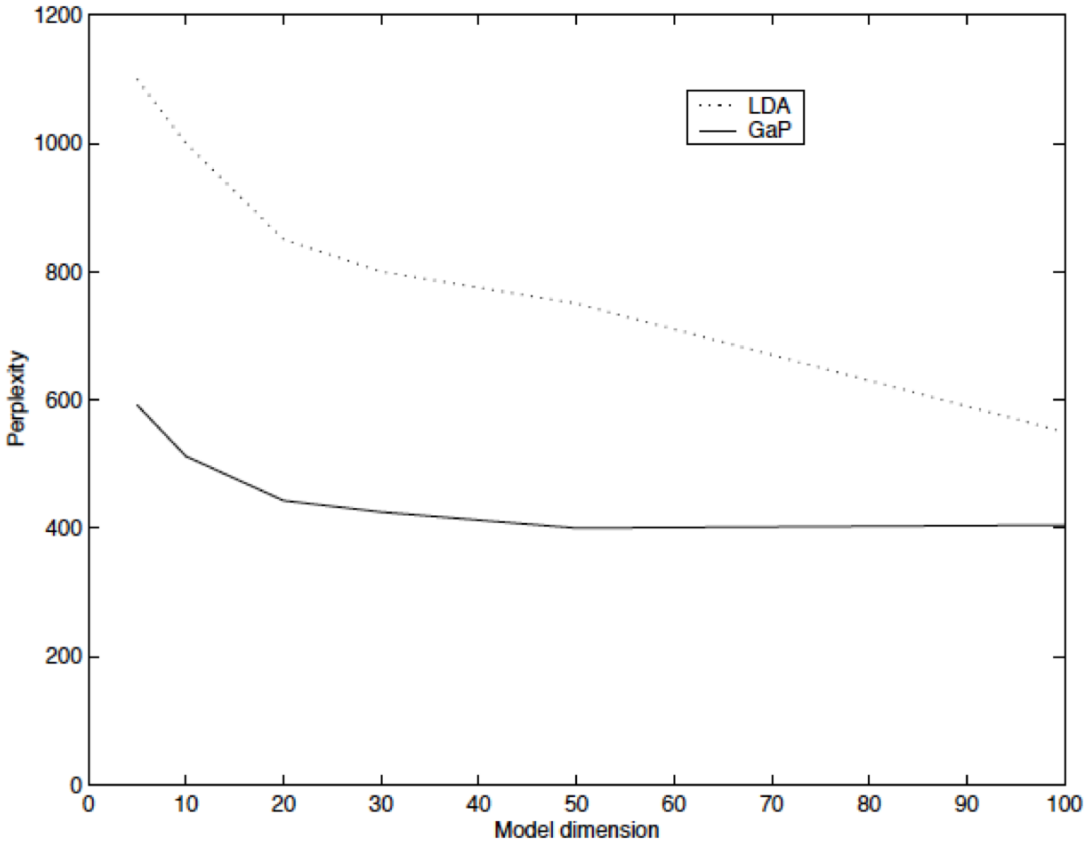
M Step:

$$\Lambda_{ij} = \Lambda_{ij} \left(\sum_{k=1}^{n} \frac{F_{ik}}{\overline{Y}_{ik}} \overline{X}_{jk} \right) / \left(\sum_{k=1}^{n} \overline{X}_{jk} \right)$$

正則化あり

GaP: 実験結果

- LDAとの比較



- 実装面でも、行列の積の更新で済むので非常に高速
- Intel MKL or Atlasで高速に書けるらしい
- LDAよりパラメータ数が多い

Discrete PCA (Buntine 2005)

- 一般化して、

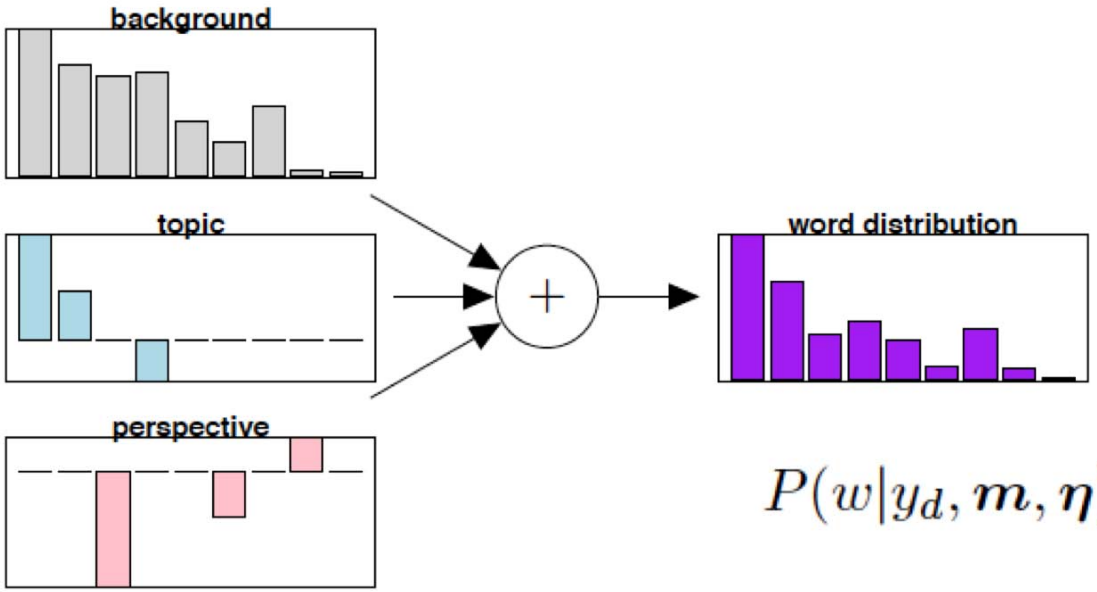
$$\begin{cases} \mathbf{w} \sim P_D(\boldsymbol{\beta\theta}, \alpha) \\ \boldsymbol{\theta} \sim P_C(\eta) \end{cases} \quad \text{where } \boldsymbol{\beta\theta} = \langle \mathbf{w} \rangle_{p(\mathbf{w}|\boldsymbol{\theta})}$$

$$P_D : \text{Discrete}, \quad P_C : \text{Continuous}$$

	P_D	P_C	Constraint
NMF	—	—	$\theta_k \geq 0$
PLSI	Mult	—	$\theta_k \geq 0, \sum_k \theta_k = 1$
LDA	Mult	Dir	$\theta_k \geq 0, \sum_k \theta_k = 1$
GaP	Poisson	Gamma	$\theta_k \geq 0$

Sparse Additive Generative Model (SAGE)

- LDAの $\beta = \{ p(w|z) \}$ のパラメータ数:
  K×V=e.g. 200x50000=10,000,000個
- 本当に各トピックに関係する語はごく一部
  なのでは..?



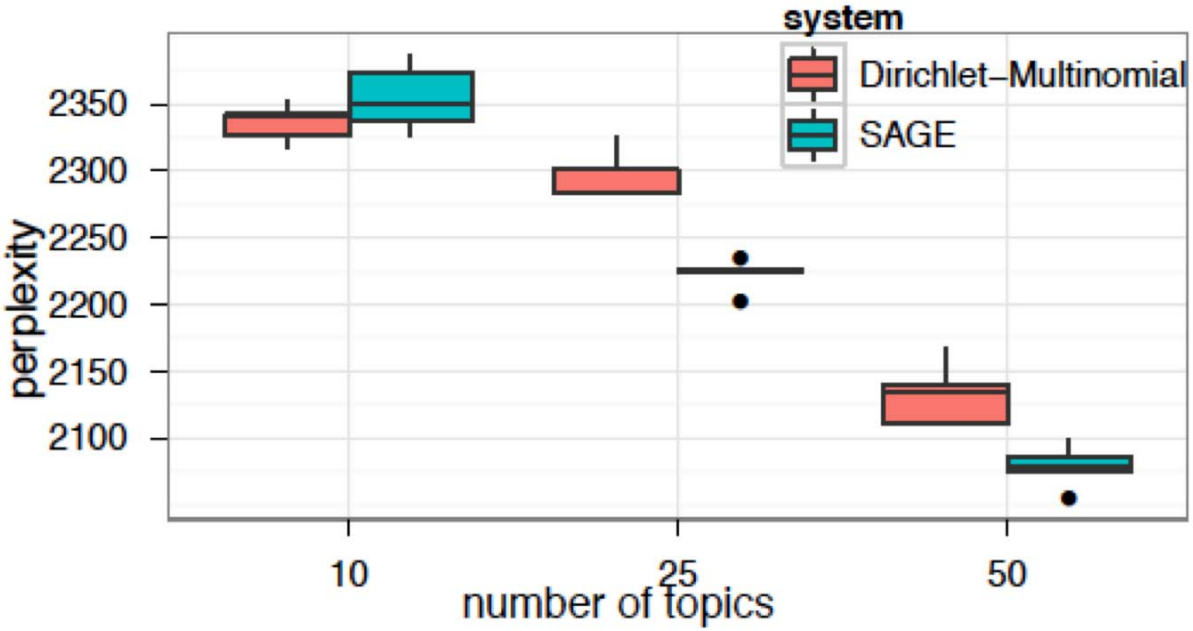
単語の確率を背景との差分で表現する
(Eisenstein+ 2011)
差分はベクトル空間に存在する正規分布

$$P(w|y_d, \boldsymbol{m}, \boldsymbol{\eta}) = \frac{\exp\left(\boldsymbol{m} + \boldsymbol{\eta}_{y_d}\right)}{\sum_i \exp\left(m_i + \eta_{y_d,i}\right)}$$

SAGE: 変分下限と結果

- LDAとほぼ同じ変分下限

$$\ell = \sum_d \langle \log P(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \rangle + \sum_n^{N_d} \left\langle \log P(w_n^{(d)}|\boldsymbol{m}, \boldsymbol{\eta}_{z_n^{(d)}}) \right\rangle$$
$$+ \left\langle \log P(z_n^{(d)}|\boldsymbol{\theta}_d) \right\rangle + \sum_k \langle \log P(\boldsymbol{\eta}_k|\mathbf{0}, \boldsymbol{\tau}_k) \rangle$$
$$+ \sum_k \langle \log P(\boldsymbol{\tau}_k|\gamma) \rangle - \langle \log Q(\boldsymbol{\tau}, \boldsymbol{z}, \boldsymbol{\theta}) \rangle .$$

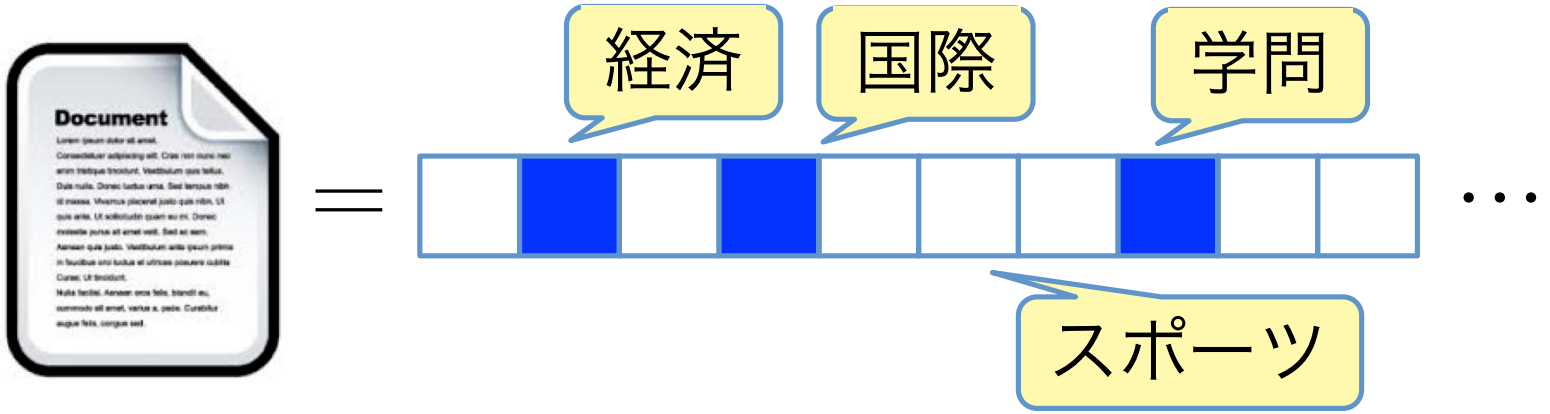


NIPSコーパスでの実験結果

Factorized Topic Modeling

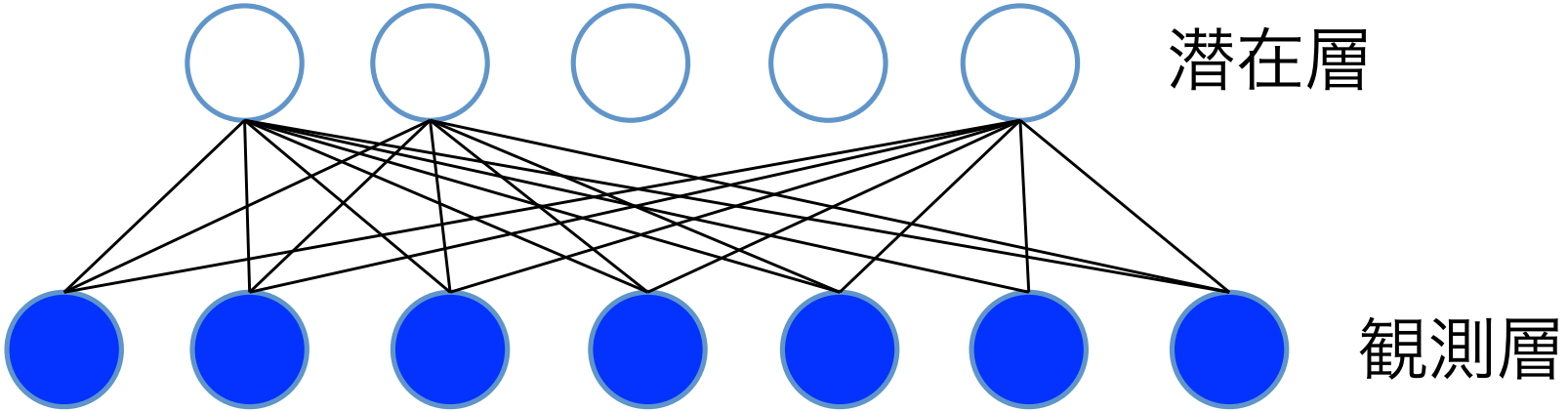
- 今までのトピックモデルは、トピックの「どれか一つ」が使われるものだった

- 実際には、組み合わせ表現が非常に有用



  - 1/0の組み合わせのトピック10個＝1024通り!

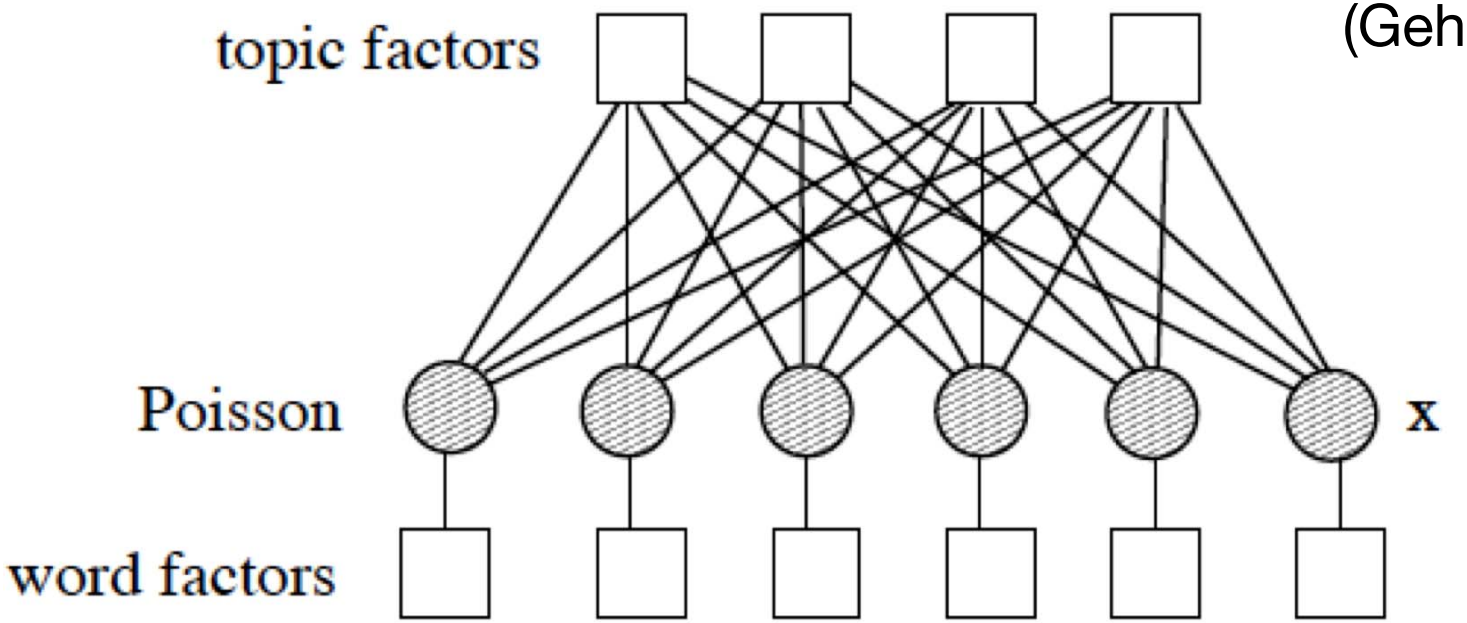
RBM (Restricted Boltzmann Machines)



- 観測層、潜在層で自分自身にリンクがない ニューラルネット
  - 潜在層のニューロンは、0/1で発火
  - リンクの重みを学習する
- 最近流行のDeep Networkは、これの多層化

Rate-Adapting Poisson Model (RaP)

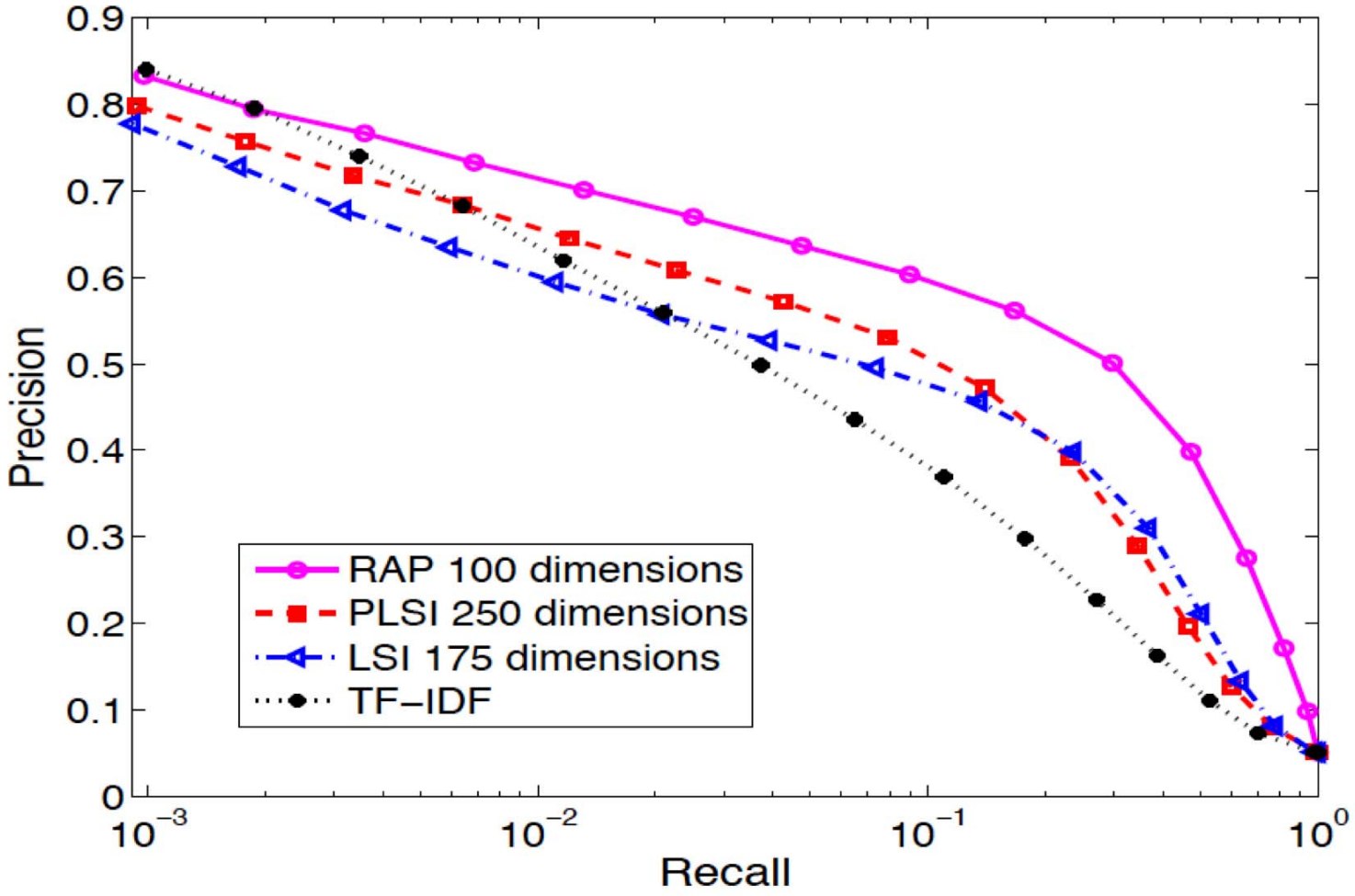
(Gehler+ 2006)



- 観測層がポアソン分布の期待値→単語の観測頻度ベクトルから、潜在層の発火とリンクの重みを求める
  - 学習には、特別なMCMCを使用

RaP: 実験

- PLSI, LSIとの比較 on 20-newsgroup データセット



RBMs on Topic Modeling

- Pros: コンパクトな表現、ユークリッド空間
  (制約が少ない)
- Cons: 時系列など、他の拡張が難しい(色々あるが、ややアドホック)
  - ただし、例えば言語モデルでは、RBMによる Neural Probabilistic Language Model が性能では最高性能といわれている
  - Research Theme!

最前線の話題 (の一部)

- Beta-Negative Binomial process (Zhou+, arXiv 2012)
  - “Beta-Negative Binomial process and Poisson Factor Analysis”, arXiv.
- Dependent Hierarchical Normalized Random Measures (Chen+, ICML 2012)
  - “Dependent Hierarchical Normalized Random Measures for Dynamic Topic Modeling”, icml.cc
- しかし、まだ Bag of words だけでいいのかは疑問

参考文献 (1)

- 統計的機械学習全般の教科書
  - 「パターン認識と機械学習: ベイズ理論による統計的予測」(上)(下). C. M. Bishop著, Springer, 2007,2008.
  - “Information Theory, Inference, and Learning Algorithms”. David J. C. MacKay. Cambridge University Press, 2003.
  - “Machine Learning: A Probabilistic Perspective”. Kevin P. Murphy. MIT Press, 2012.
    - 最新の、包括的な教科書
- ベイズ統計について
  - “Bayesian Data Analysis”, second edition. Andrew Gelman et al., Chapman&Hall/CRC, 2003.
  - 「ベイズ統計と統計物理」(岩波講座 物理の世界 物理と情報(3)), 伊庭幸人. 岩波書店, 2003.

参考文献 (2)

- LDA, PLSIについて
  - “Latent Dirichlet Allocation”. David M. Blei, Andrew Y. Ng, Michael I. Jordan. Journal of Machine Learning Research, vol.3, pp. 993-1022, 2003.
  - “Probabilistic Latent Semantic Indexing”. Thomas Hofmann, SIGIR 1999, pp.50-57, 1999.
- EMアルゴリズム、VB-EMアルゴリズムについて
  - “A view of the EM algorithm that Justifies Incremental, Sparse, and other Variants”. Radford Neal, Geoffrey Hinton. Learning in Graphical Models, pp.355-368, 1998.
  - “Inferring Parameters and Structure of Latent Variable Models by Variational Bayes”. Hagai Attias. UAI 1999, pp.21-30, 1999.

参考文献 (3)

- トピックモデルのGibbs Sampling
  - “Finding Scientific Topics”, Thomas L. Griffiths, Mark Steyvers. PNAS, vol.101, pp.5228-5235, 2004.
- Unigram Mixtures, Dirichlet Mixtures
  - “Text Classification from Labeled and Unlabeled Documents using EM”. Kamal Nigam, Andrew McCallum, Sebastian Thrun, Tom Mitchell. Machine Learning. vol.39, no.2/3, pp.103-134, 2000.
  - “Dirichlet Mixtures in Text Modeling”. Mikio Yamamoto, Kugatsu Sadamitsu. CS Technical Report CS-TR-05-1, University of Tsukuba, 2005.
- トピックモデルの最近の参考文献の紹介
  - 「私のブックマーク: Latent Topic Model (潜在的トピックモデル)」. 佐藤一誠, 人工知能学会誌 vol.27, no.3, 2012.