音楽と言語への
ベイズ統計的アプローチ

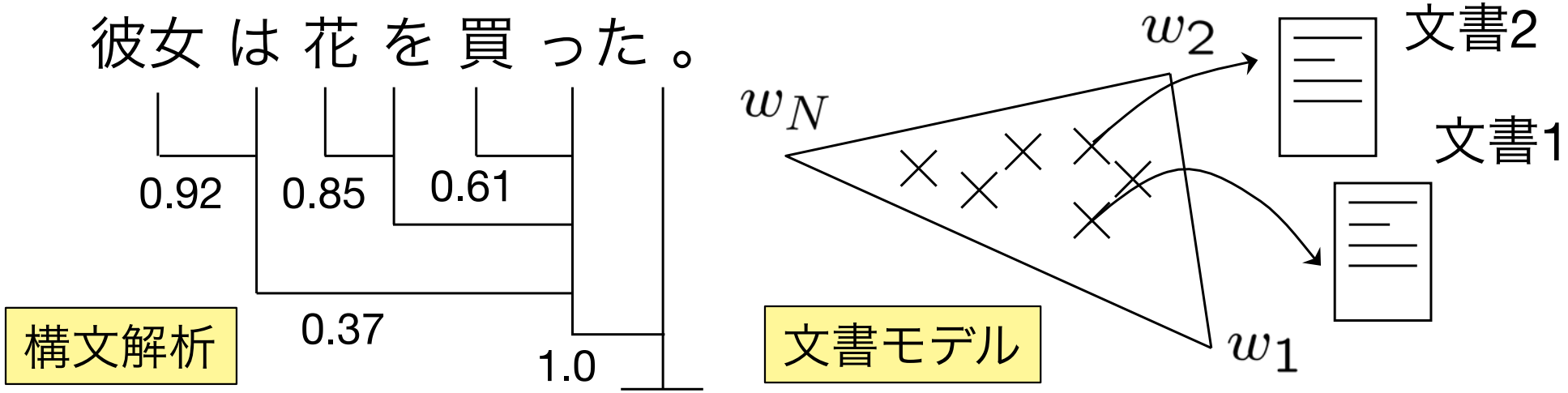
持橋大地
統計数理研究所 数理・推論研究系 准教授
daichi@ism.ac.jp

統計数理研究所オープンハウス 2014
2014-6-13 (Fri)

統計的自然言語処理とは

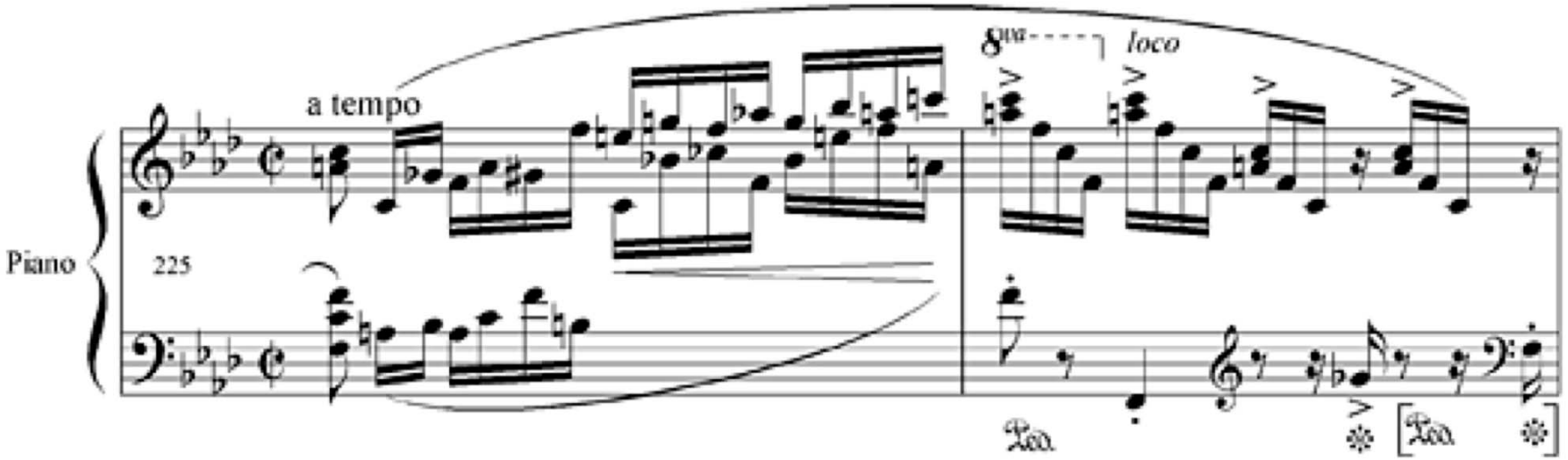
- 言語の統計的な取り扱い
  (＝計算言語学)
  - 1990年代後半以降、Webによる電子テキストの増大によって、加速的に進歩
- 2014年: 大きく進歩したが、まだ解けていない基本問題もある

統計的自然言語処理とは (2)



- 代表的な応用:
  構文解析、形態素解析、文書モデル、意味極性分類、
  照応解析、言語進化モデル、・・・・

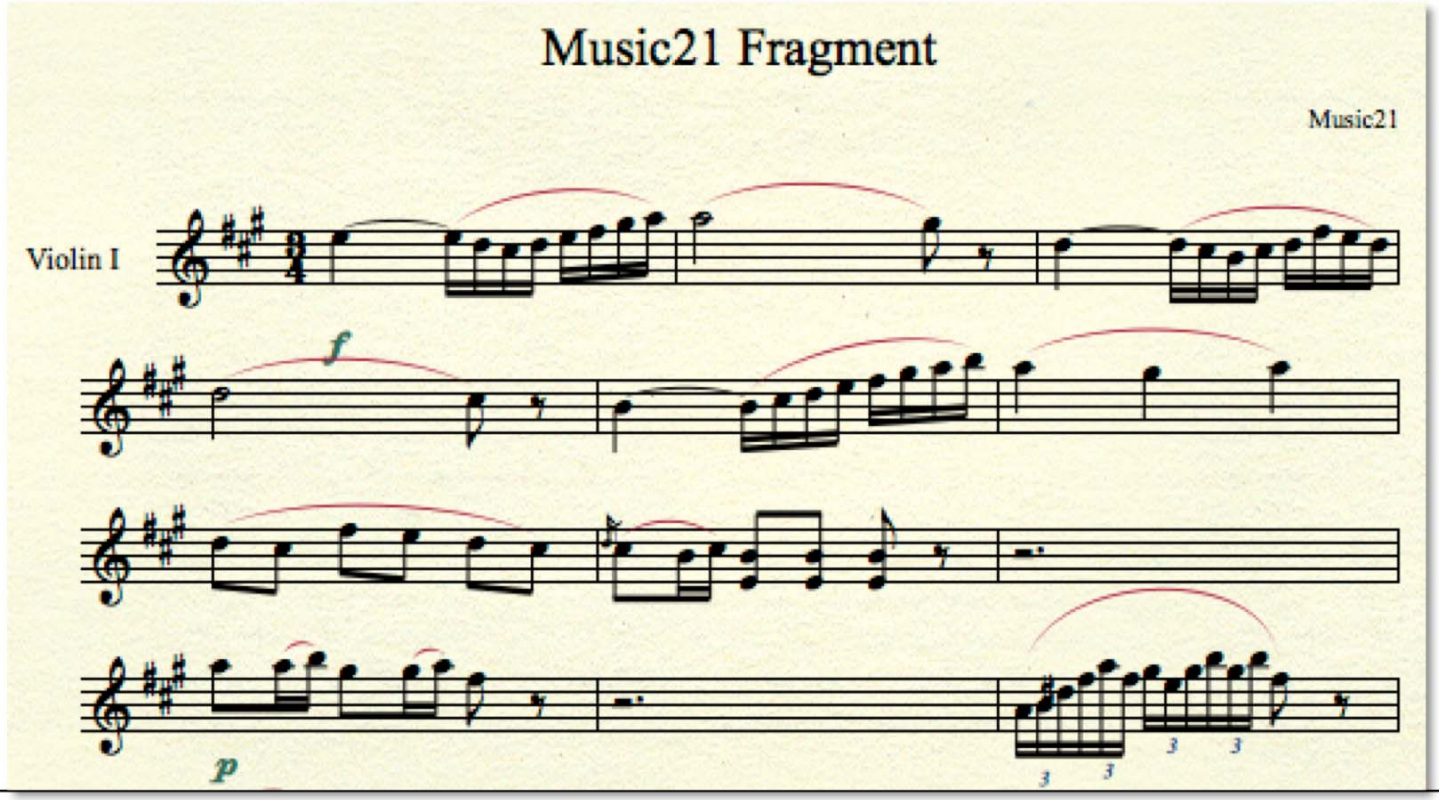
音楽との共通性



- 音楽は楽譜をもち、それ自身の構造を持っている
  ‥‥言語と同じ
- 音響処理だけからはわからない！

例：Mozart, ヴァイオリン協奏曲

- 音楽情報処理のためのPythonパッケージである Music21 (http://web.mit.edu/music21/) 付属のコーパスの一部



例：Mozart, ヴァイオリン協奏曲 (2)

- 記号列に直してみる (mozart-notes.py)

<tune>	note:5/4/0.25	····
note:5/4/1	note:5/9/0.25	
note:5/4/0.25	note:5/9/2	
note:5/2/0.25	note:5/8/0.5	
note:5/1/0.25	note:5/2/0.25	
note:5/2/0.25	note:5/2/0.25	
note:5/4/0.25	note:5/2/2	
note:5/6/0.25	note:5/1/0.5	
note:5/8/0.25	note:4/11/0.25	
note:5/9/0.25	note:5/11/0.25	

- 隠れ状態がある····? → 言語と同じ！

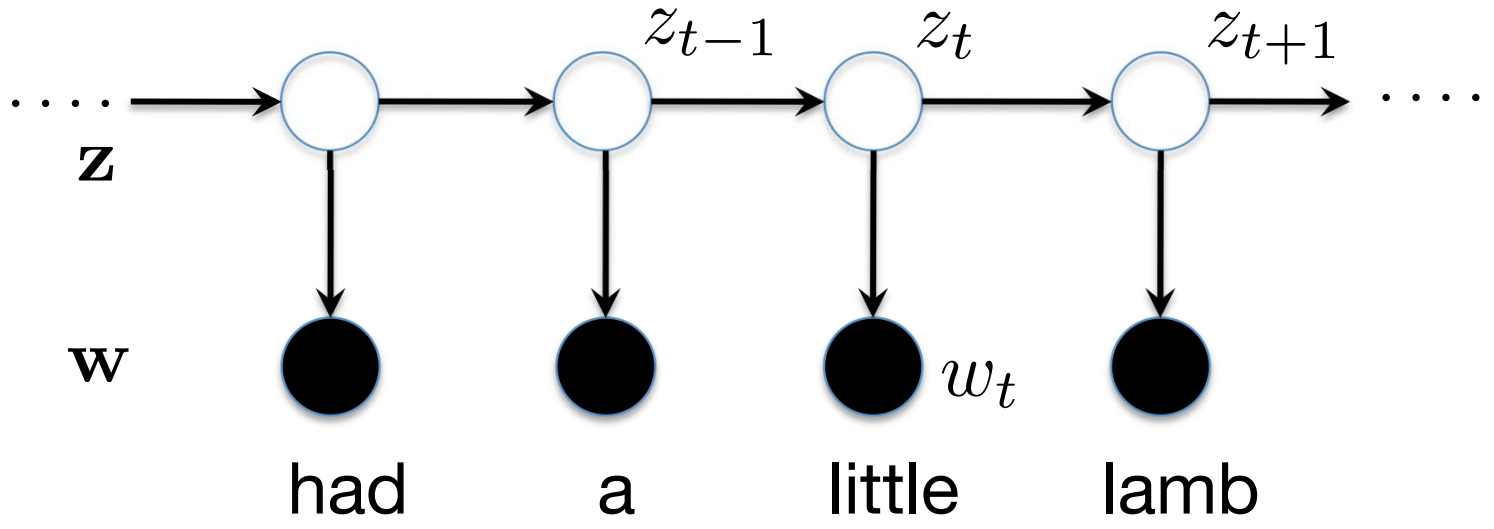
教師なし品詞解析

When she arrived at the hotel, he realized that the era ..

CONJ N V P DT N N V CONJ DT N

- 言語には品詞があり、われわれは品詞を認識している
  - 名詞、動詞、形容詞、冠詞、接続詞、‥
- どうやって品詞がわかるのか?
  → 隠れMarkovモデル (Merialdo 1994, van Gael+ 2009)

隠れMarkovモデル



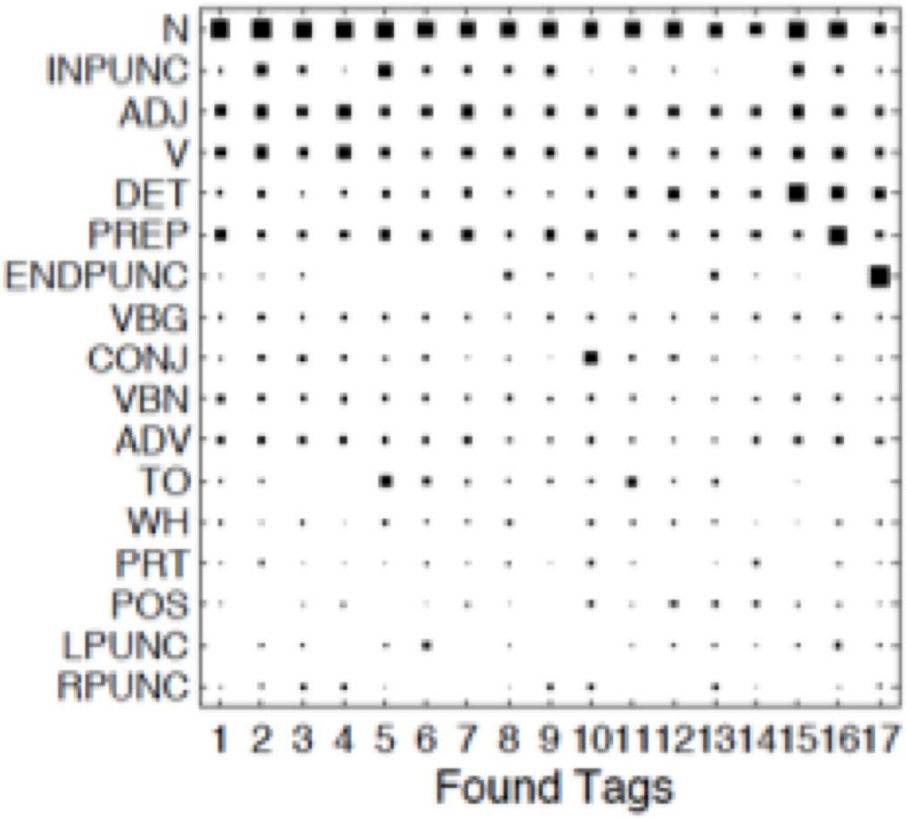
- 観測データ: 単語列 $\mathbf{w} = w_1 w_2 w_3 \cdots w_T$
- 潜在変数　: 品詞列 $\mathbf{z} = z_1 z_2 z_3 \cdots z_T$
  - データ全体の確率:

$$p(\mathbf{w}, \mathbf{z}) = \prod_{t=1}^{T} p(w_t|z_t)\, p(z_t|z_{t-1})$$

隠れMarkovモデルの学習

Baum-Welchともいう

- EMアルゴリズム: Forward-Backward (動的計画法)
- しかし‥



- 学習された「品詞」間の状態遷移行列
- うまく学習できていない!
- モデルが悪いのか?

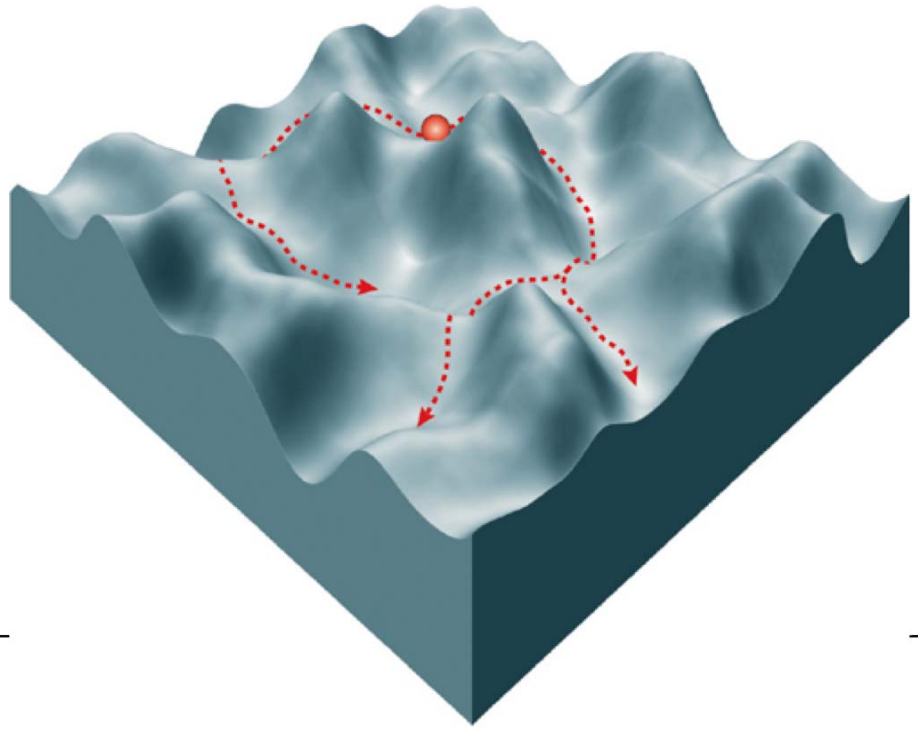
No!

隠れMarkovモデルのベイズ学習

- EMアルゴリズムは最尤推定

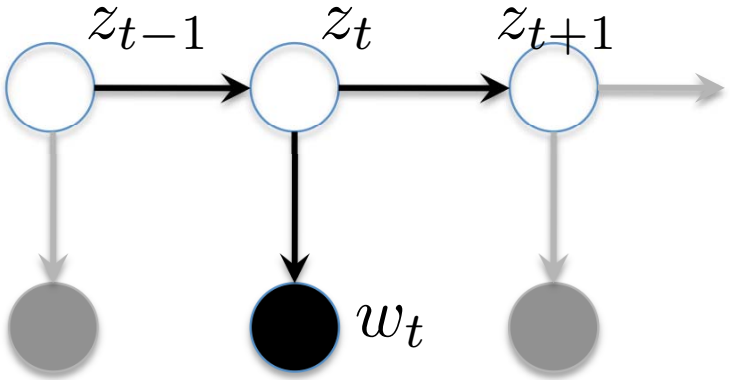
$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\mathbf{w}|\theta)$$

- 実際のデータでは、多数の局所解



隠れMarkovモデルのベイズ学習 (2)

- MCMCで解けばよい! (Johnson, Goldwater 2007)
  - $p(z_t|z_{t-1}) \sim \mathrm{Dir}(\gamma)$
  - $p(w|z) \sim \mathrm{Dir}(\eta)$
- このとき、

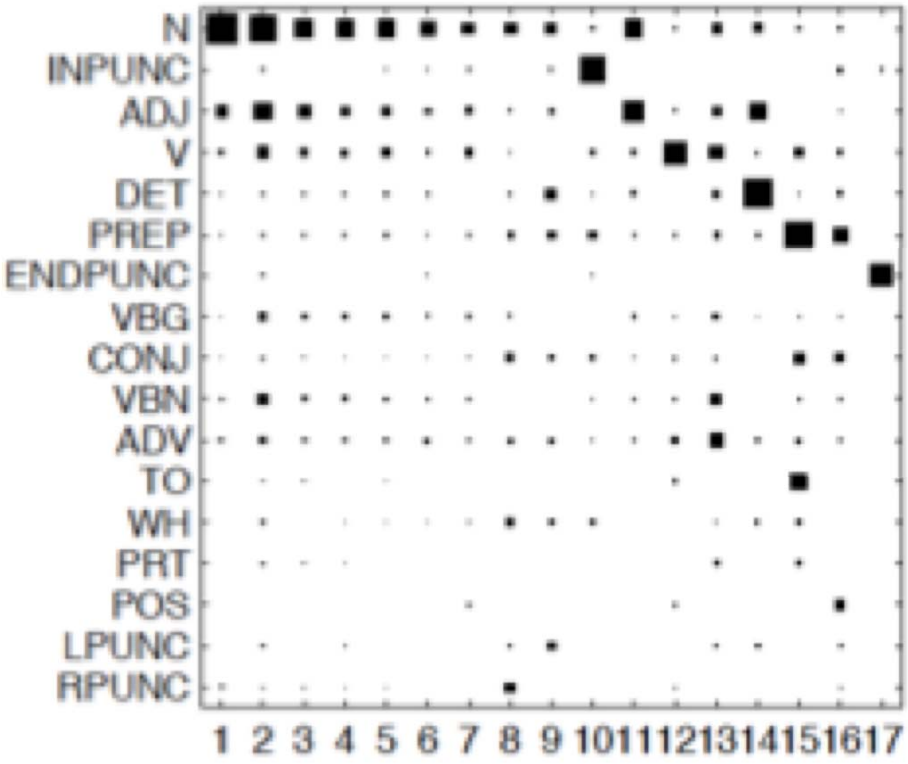


$$p(z_t|w_t, z_{t-1}, z_{t+1}, \text{others}) \propto p(w_t|z_t)\, p(z_{t+1}|z_t)\, p(z_t|z_{t-1})$$

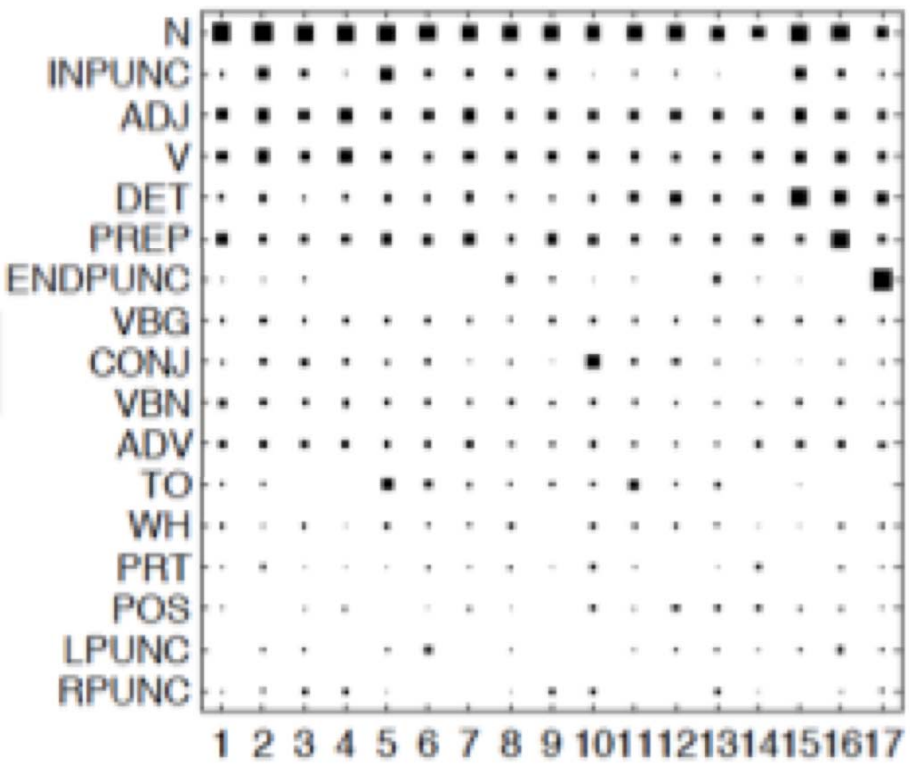
$$p(z_t|z_{t-1}, z_{t+1}, w_t, \text{others}) \propto \left(\frac{n(w_t, z_t) + \eta}{\sum_w n(w, z_t) + \eta}\right) \cdot \left(\frac{n(z_t, z_{t-1}) + \gamma}{n(z_{t-1}) + K\gamma}\right) \cdot \left(\frac{n(z_{t+1}, z_t) + I(z_{t+1} = z_t = z_{t-1}) + \gamma}{n(z_t) + I(z_t = z_{t-1}) + K\gamma}\right)$$

隠れMarkovモデルのベイズ学習 (3)

- 結果: 劇的に改善



ベイズ推定+MCMC



最尤推定+EM

隠れMarkovモデルのベイズ学習 (3)

- 問題: 隠れクラス数(＝品詞数) Kは?
  → infinite HMM (Beal 2002; Teh+ 2006)

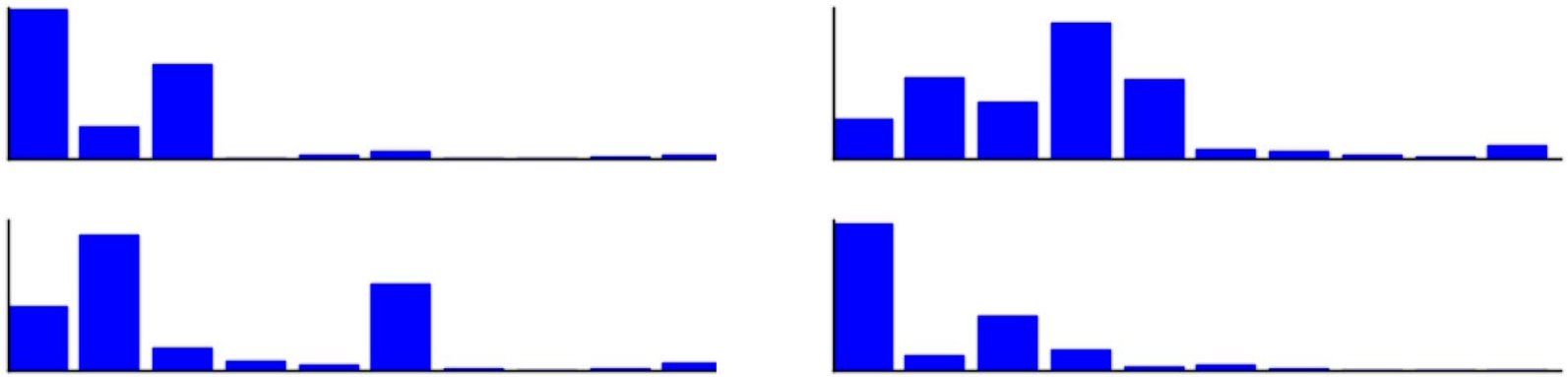
infinite HMM

- HMMのパラメータは、$p(w|z)$ と $p(z_{t+1}|z_t)$
- z を生成する $p(z_{t+1}|z_t)$ が、無限次元のGEM分布

$$p(z_{t+1}|z_t) \sim \mathrm{GEM}(\gamma)$$

  に従うとする.

- GEM分布からのサンプル:

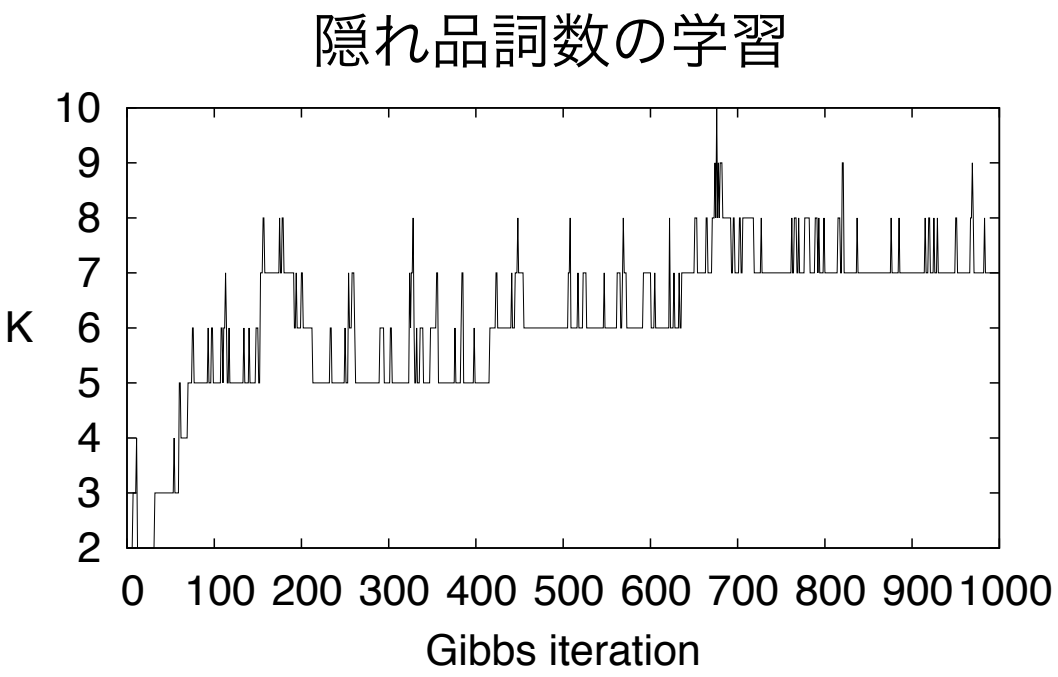


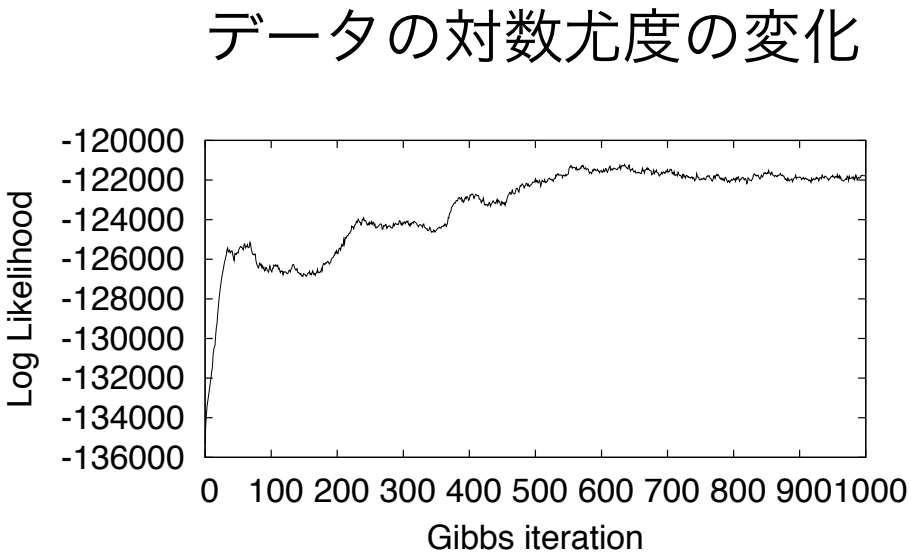
infinite HMM (2)

- このままだと学習に∞個の次元を調べないといけないが、
  (1) CRP (中国料理店過程, Aldous 1985)
  (2) Slice Sampling (Neal 2003, van Gael+ 2008)
  を使うと、有限次元で計算できる
- 注意: データ数*N*以上のクラス数は必要ない
  - 自然数*N*の分割問題 (確率分割)

infinite HMM (3)

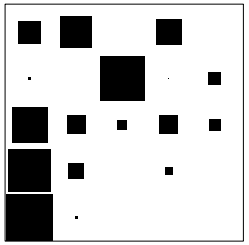
- 「不思議の国のアリス」(26689語,1431行)を学習データにしてiHMMを学習





Infinite HMM (2)

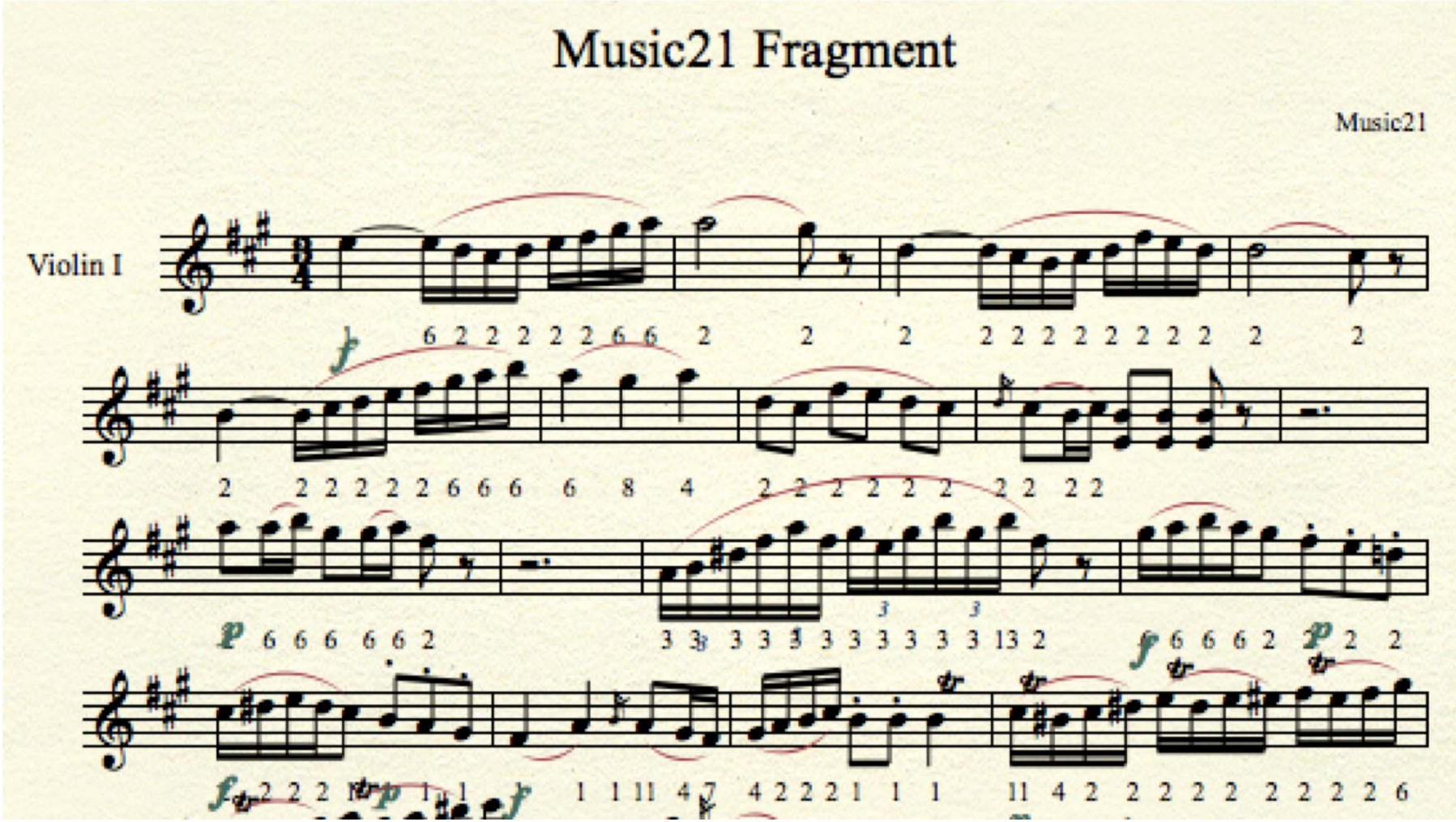
状態遷移行列



1		2		3		5	
she	432	the	1026	was	277	way	45
to	387	a	473	had	126	mouse	41
i	324	her	116	said	113	thing	39
it	265	very	84	$	87	queen	37
you	218	its	50	be	77	head	36
alice	166	my	46	is	73	cat	35
and	147	no	44	went	58	hatter	34
they	76	his	44	were	56	duchess	34
there	61	this	39	see	52	well	31
he	55	$	39	could	52	time	31
that	39	an	37	know	50	tone	28
who37		your	36	thought	44	rabbit	28
what	27	as	31	herself	42	door	28
i'll	26	that	27	began	40	march	26

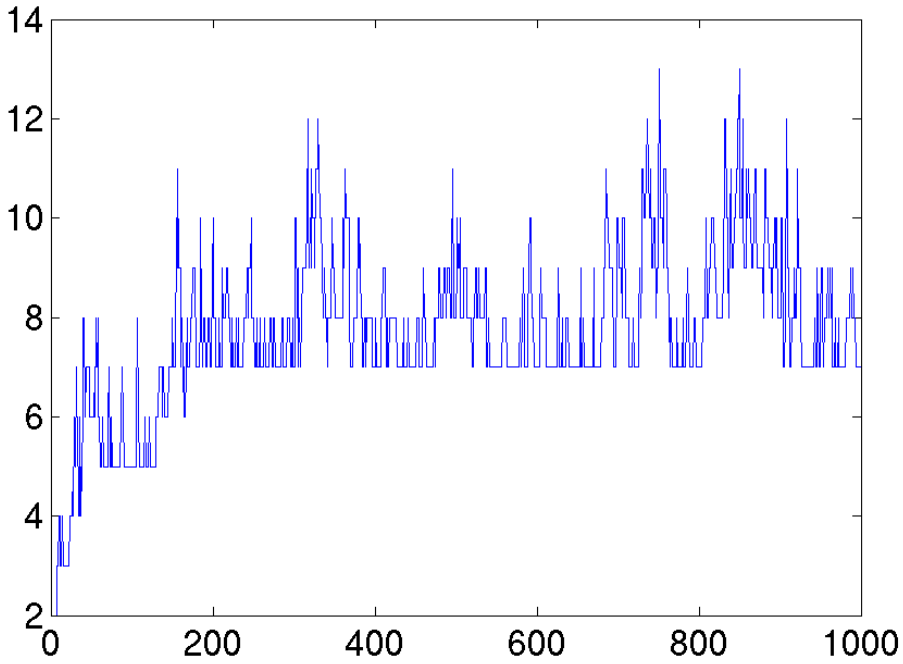
- 教師なしで、品詞に相当するものが学習できている!

Infinite Mozart?

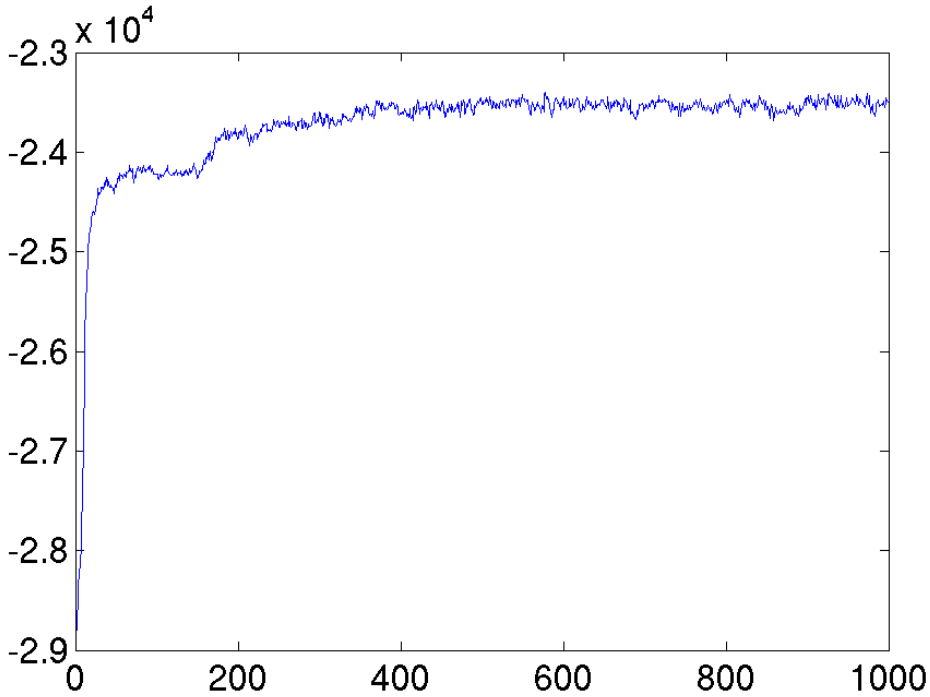


- フレーズのカテゴリがわかる! (実験はまだ不完全)

Infinite Mozart? (2)



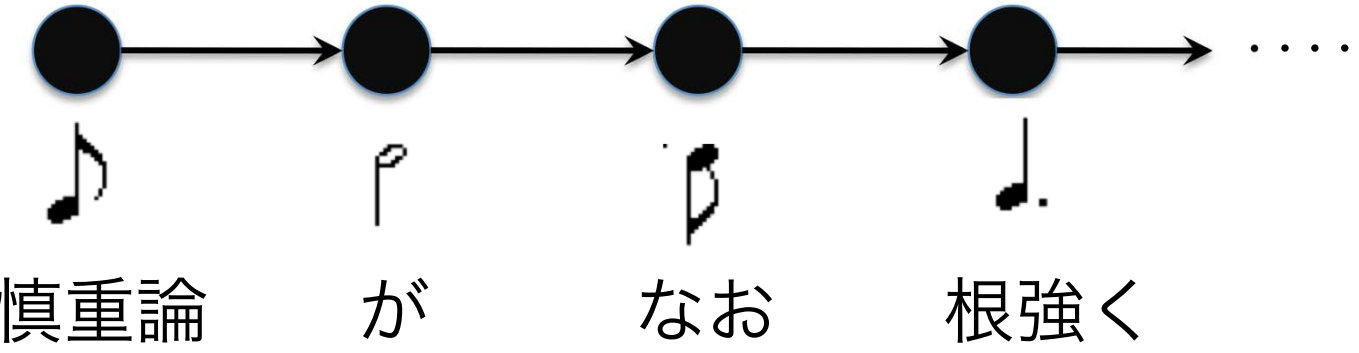
潜在クラス数K



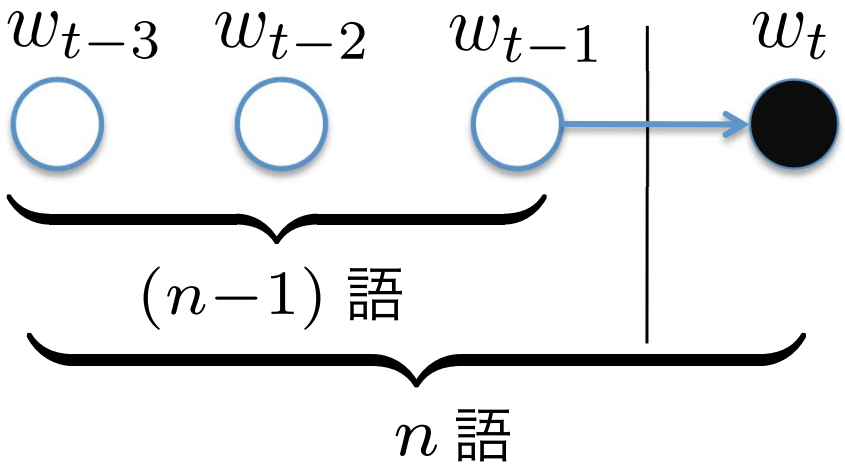
Joint Log Likelihood

- MCMC 400 iteration程度でほぼ収束

音符のn-gramモデル



- 音符や単語には、直接状態遷移があるのでは?
  → n-gramモデル



(n-1)語を見た後、次に来る語の条件付き確率

$$p(w_t | w_{t-1}, \cdots, w_{t-(n-1)})$$

を計算する

n-gramモデルの問題

$$p(w_t | w_{t-1}, w_{t-2}, \cdots, w_{t-(n-1)})$$

組み合わせが指数的に増大！

– 語彙の数 V=10,000のとき、4-gramでは原理的に $10000^3=10^{12}=1000000000000$個のパラメータ

nグラムモデルのベイズ学習

- nグラムモデル‥‥古典的だが、音声認識や機械翻訳では未だ重要、基本的 (言葉のMarkovモデル)

$$p(\text{彼女 が 見る 夢}) = p(\text{彼女}) \cdot p(\text{が} \mid \text{彼女}) \cdot p(\text{見る} \mid \text{が}) \cdot p(\text{夢} \mid \text{見る})$$

- nグラムモデルの問題: スムージング

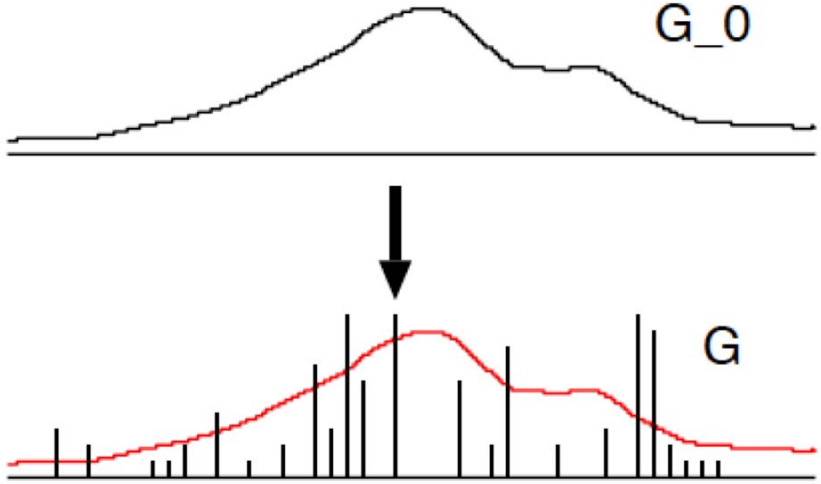
$$\hat{p}(\text{yield}|\text{maximum likelihood will often}) = \frac{n(\text{maximum likelihood will often yield})}{n(\text{maximum likelihood will often})} = 0$$

現在のGoogleカウント

  - 頻度そのままでなく、何か値を足したりする必要!

Pitman-Yor過程 (Pitman and Yor 1997)

- ディリクレ過程とは、自然言語の1次元の場合、無限次元の多項分布を生成する分布のこと。

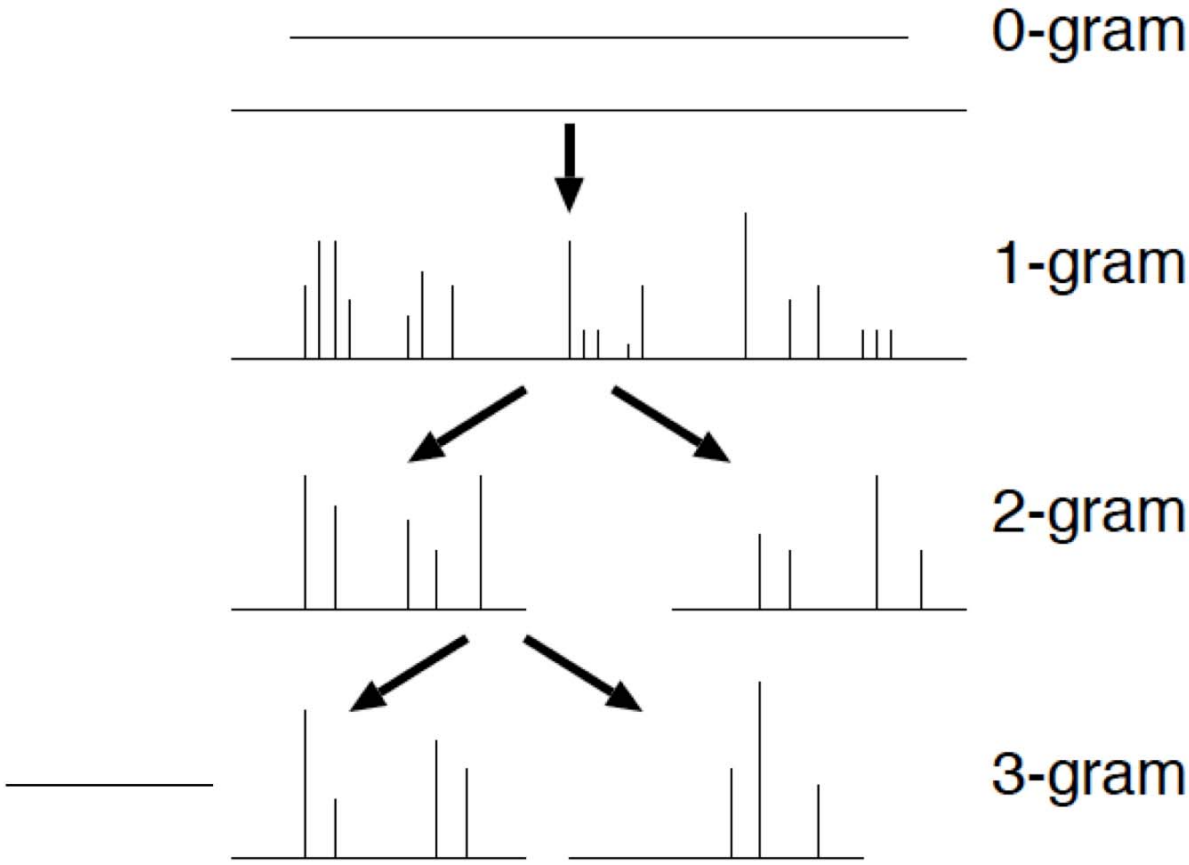


横軸:
可能な単語の種類

  - 元となる(連続)分布G_0に少し似た、無限次元の離散分布Gを生成
  - $G \sim \mathrm{DP}(\alpha, G_0)$　と表記 (α: 集中度パラメータ)
  - この2パラメータ拡張がPitman-Yor過程 $\mathrm{PY}(\alpha, d, G_0)$

階層Pitman-Yor過程

- nグラム分布が、階層的に(n-1)グラム分布からのPitman-Yor過程によって生成されたと仮定
  - 最初はUniform, だんだん急峻になる



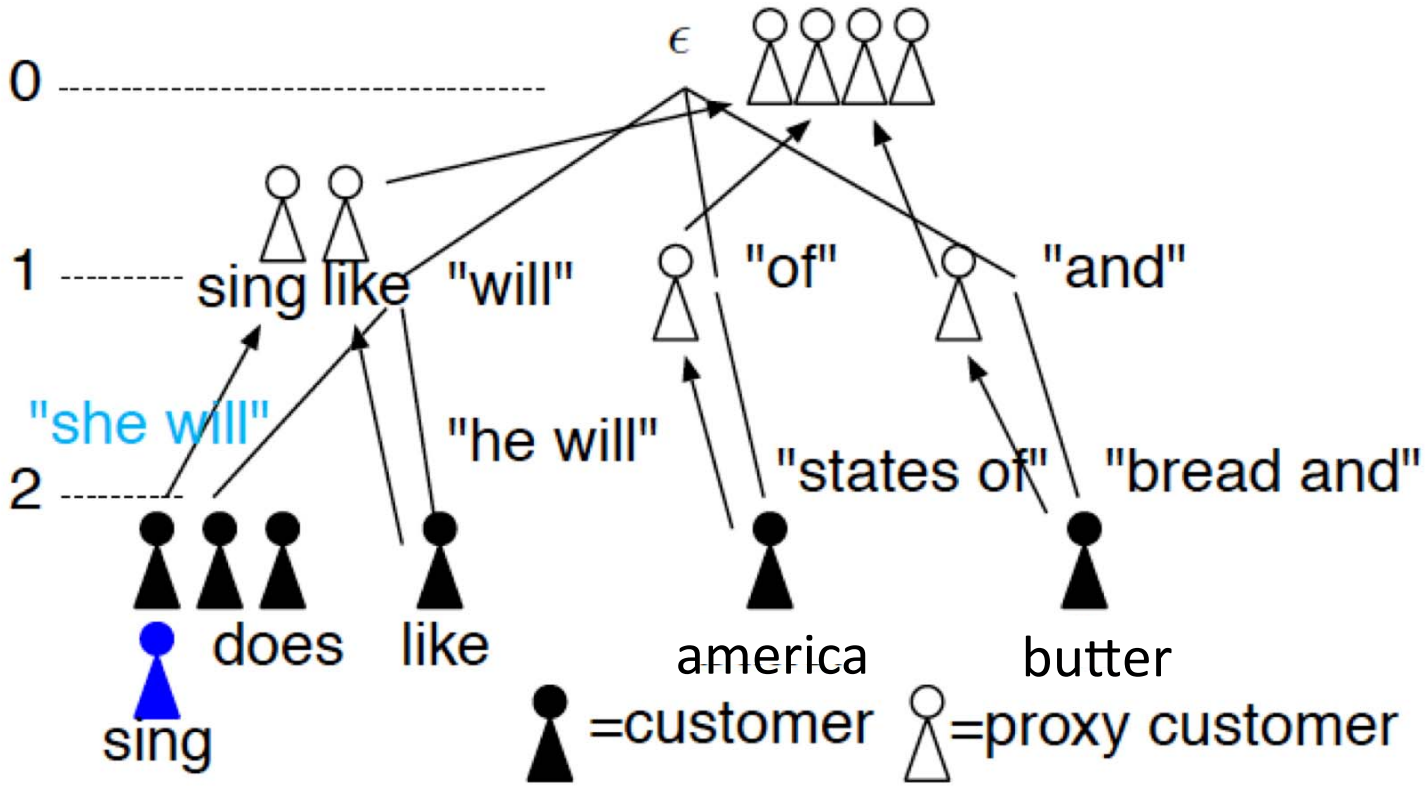
Prior: $(n-1)$ gram

$\downarrow \mathrm{PD}(d_n, \theta_n)$

Posterior: (n) gram

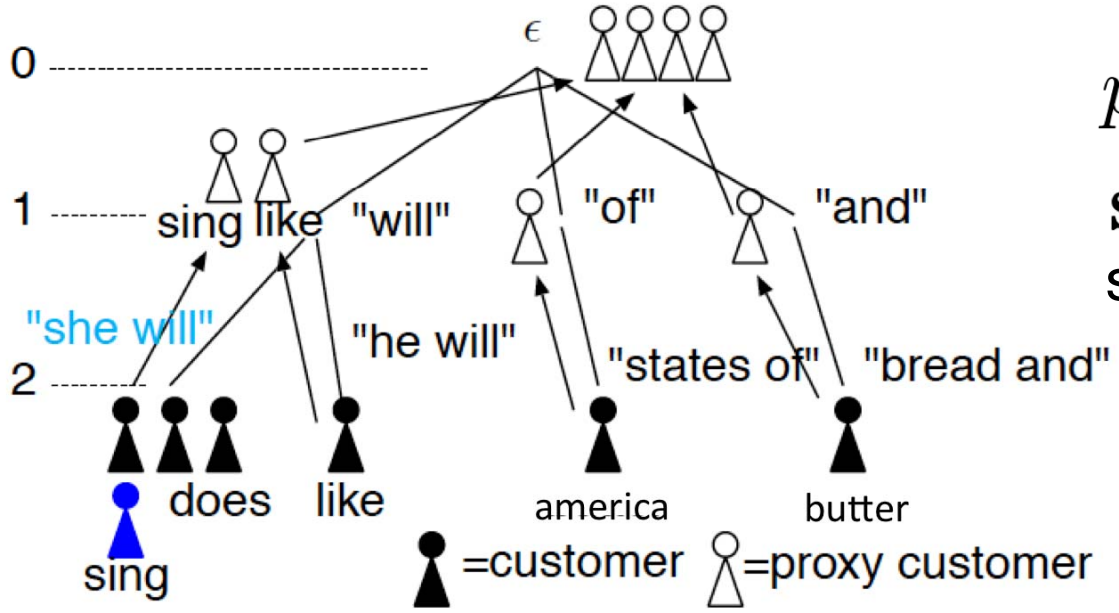
階層CRP表現

- 測度を直接扱う代わりに、カウントで離散表現する
  - 一人の「客」が1単語分のカウントに対応
  - 下の青い客は、文脈 “she will” の後に “sing” が1回現れたことを意味する (全部で2回)



HPYLMの学習

- HPYLM (hierarchical Pitman-Yor language model) の学習＝潜在的な代理客の最適配置
- Gibbs sampling: 客を一人削除して再追加、を繰り返す
  - For each w = randperm(all counts in the corpus),
    - 客 w と関連する代理客をモデルから削除
    - 客 w をモデルに追加＝代理客を再サンプル

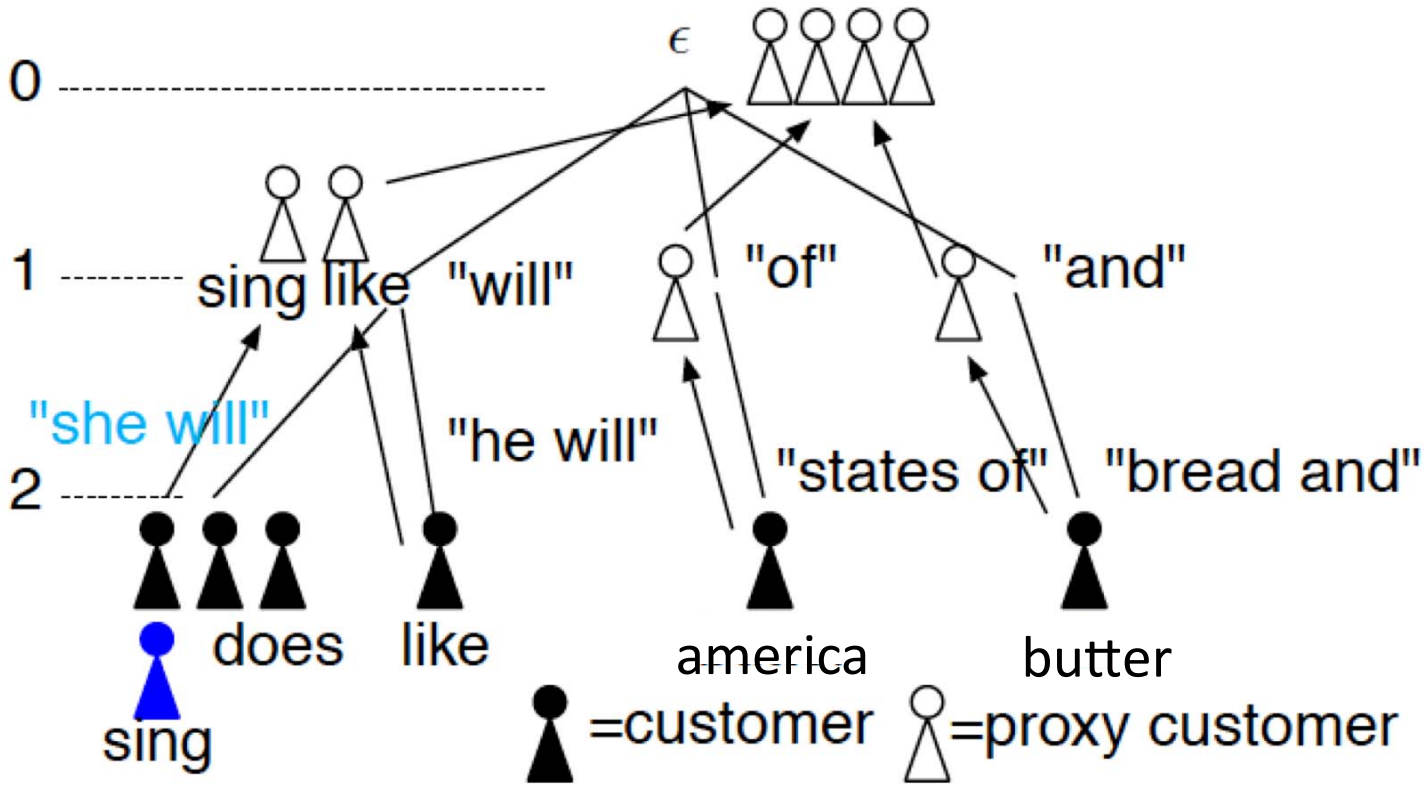


$$p(\mathbf{w}) = \sum_{\mathbf{s}} p(\mathbf{w}, \mathbf{s})$$

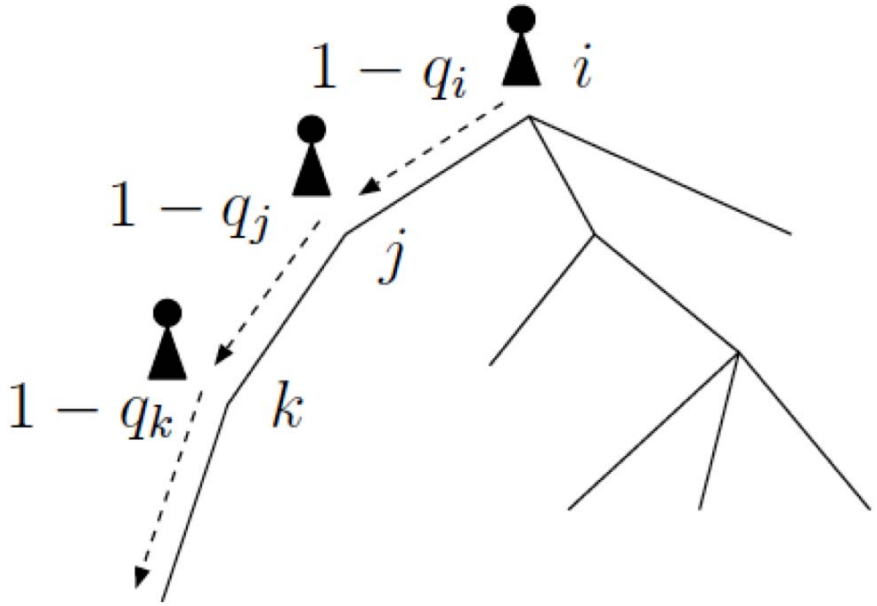
$\mathbf{s}$: 白い代理客の seating arrangements

HPYLM=nグラムモデルの問題

- 常に客を深さn-1に配置していいのか?
  - “other than”, “the united states of america” など、必要なnグラムのオーダーは本来異なるはず
  - HPYLMではどうすればいい?



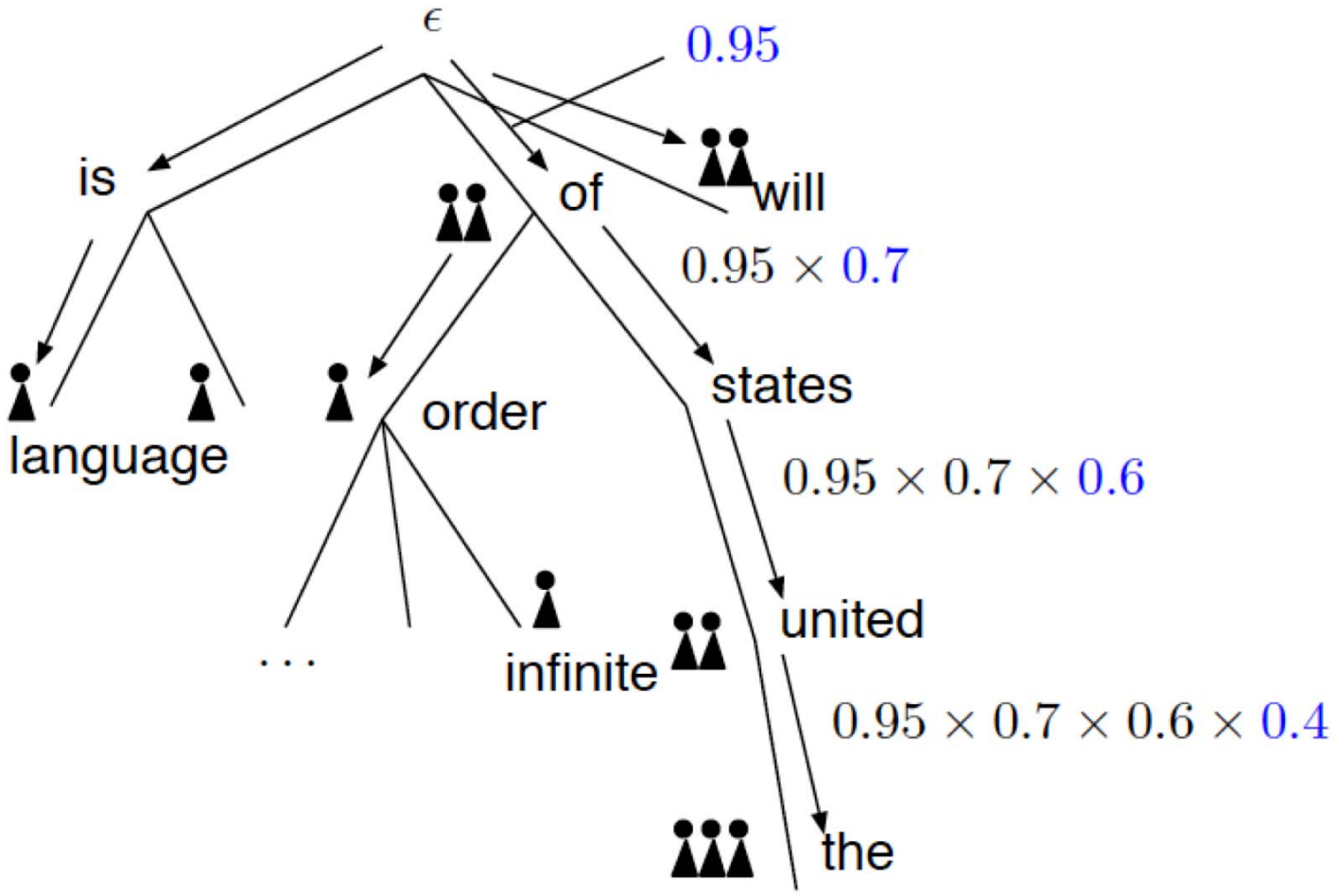
VPYLM (Variable-order HPYLM)



- 客を、木の根から確率的にたどって追加
- ノード i に,そこで止まる確率 q_i がある ($1-q_i$:通過確率)
  - q_i は、ランダムにベータ事前分布から生成
  - ゆえに、深さnで止まる確率は

$$p(n|h) = q_n \prod_{i=0}^{n-1} (1-q_i).$$

VPYLM, Variable-order HPYLM (2)



- “通過確率” ($1-q_i$)が大きい→深いノードに到達できる
- “通過確率” ($1-q_i$)が小さい→短いMarkov依存性を持つ

VPYLMの学習

- 学習データの各単語 $\mathbf{w} = w_1 w_2 \cdots w_N$ に, それを生んだ隠れたMarkovオーダー $\mathbf{n} = n_1 n_2 \cdots n_N$ が存在

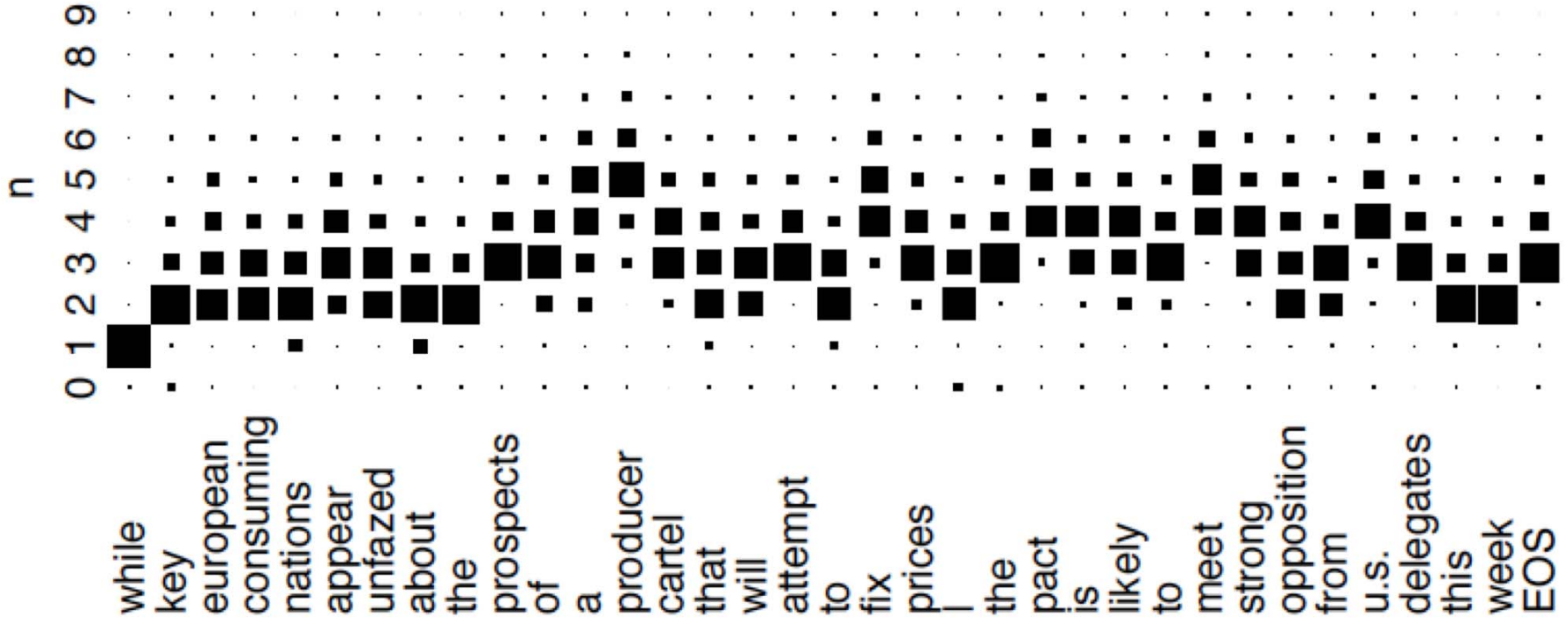
$$p(\mathbf{w}) = \sum_{\mathbf{n}} \sum_{\mathbf{s}} p(\mathbf{w}, \mathbf{n}, \mathbf{s})$$

- Gibbs (MCMC)で $\mathbf{n}$ を推定

$$p(n_t | \mathbf{w}, \mathbf{n}_{-t}, \mathbf{s}_{-t}) \propto \underbrace{p(w_t | n_t, \mathbf{w}_{-t}, \mathbf{n}_{-t}, \mathbf{s}_{-t})}_{n_t\text{グラム予測確率}} \cdot \underbrace{p(n_t | \mathbf{w}_{-t}, \mathbf{n}_{-t}, \mathbf{s}_{-t})}_{\text{深さ}n_t\text{に到達するprior}}$$

  - 2つの項のトレードオフ (深いn_tにペナルティ)
  - 第二項の事前確率はどう計算する？

VPYLMの学習結果



- NAB (WSJ) コーパスの各単語が生成されたMarkovオーダーの推定結果
  - 情報量の多い語の後は短く、連語の後は長いなどの傾向が学習されている

VPYLMの予測

- 従来と異なり、nグラムオーダーnを事前に知らないので、nに関して積分消去

$$p(w|h) = \sum_n p(w, n|h)$$
$$= \sum_n p(w|h, n)p(n|h)\,.$$

  - $p(n|h)$ は、先の計算で q_i から計算できる
- Suffix tree 上の Stick-breaking process になっている
  - 説明省略、NIPS 2011にほぼ同じアイデアがこの話を引かずに掲載

VPYLMの性能

n	SRILM	HPYLM	VPYLM	Nodes(H)	Nodes(V)
3	118.91	113.60	113.74	1,417K	1,344K
5	107.99	101.08	101.69	12,699K	7,466K
7	107.24	N/A	100.68	*27,193K*	10,182K
8	107.21	N/A	100.58	*34,459K*	10,434K
∞	—	—	100.36	—	10,629K

- SRILM: SRI言語モデルツールキット (Kneser-Ney)
- 少ないノード数で、高い性能
  - パープレキシティ＝平均予測確率の逆数 (smaller is better)
  - ∞-gram が可能!! (今や、nは不要)

VPYLMからの生成

‘how queershaped little children drawling-desks, which would get through that dormouse!’ said alice; ‘let us all for anything the secondly, but it to have and another question, but i shalled out, ‘you are old,’ said the you’re trying to far out to sea.

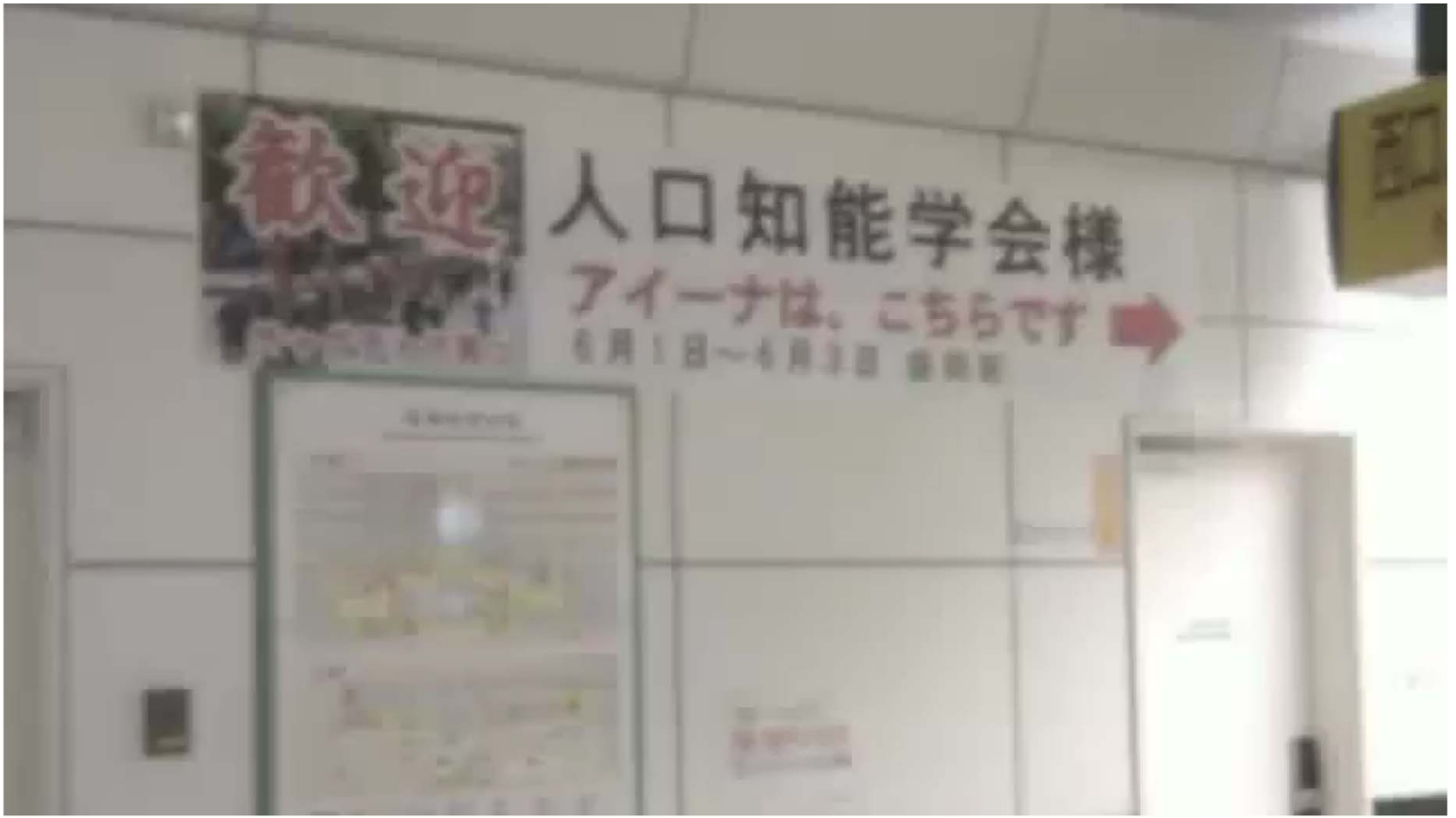
(a) Random walk generation from a character model.

Character	s a i d ␣ a l i c e ; ␣ ‘ l e t ␣ u s ␣ a l l ␣ f o r ␣ a n y t h i n g ␣ ···
Markov order	5 6 5 4 7 10 6 5 4 3 7 1 4 8 2 4 4 6 5 5 4 4 5 5 6 4 5 6 7 7 5 3 3 4 5 9 ···

(b) Markov orders used to generate each character above.

- 「不思議の国のアリス」の∞-gram文字モデルからのランダムウォーク生成
  - 生成では、気をつけないと元データがそのまま再生されてしまう

∞-gramによるメロディ生成 (白井&谷口2011)



- 旋律のトピック適応等、様々な確率的技法が使われているようです

∞-gramに基づくコード進行認識 (Yoshii+2011)

- C7→F7→C7のようなコード進行は、特定のMarkovオーダーでは記述できない

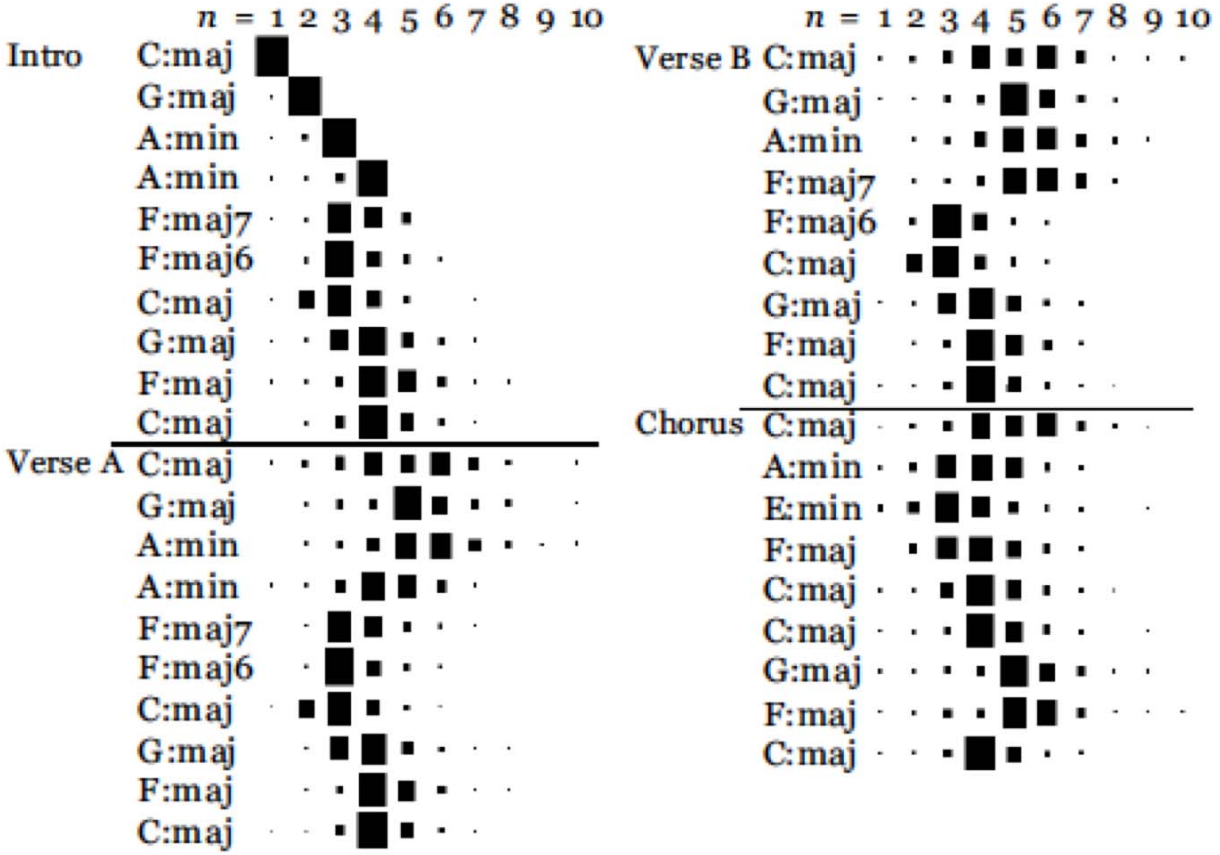


Figure 4. Hinton-diagram representation of posterior distributions over n at the beginning of the Beatles' "Let It Be."

コード進行のパープレキシティ:

モデル	PPL
Good-Turing	38.3
Kneser-Ney	18.5
HPYLM	18.0
VPYLM	15.8
VF-VPYLM	14.6

音楽と歌詞

(Facebookより[6/10], 公開記事)

白土 由佳
23時間前

ジャニーズの歌詞分析しているよ。
しろい話を紹介します。

今ジャニーズ事務所に所属している
は別格なので除く)の全曲、1,823曲
形態素解析してあげると、頻度の上
像のようになります。

そして、その30単語について、その
たら1点として、各曲に点数付けをし
点は20点、嵐の「5×10」です。つま
嵐の「5×10」は、一番ジャニーズの
詞の曲だと言えます。

実はこの曲、なんと、嵐が10周年で
こめて作詞しています。もし嵐が、
照せず、アドバイスも受けずにこの
たとしたら、アイドルなのに天然の
ね。すごい！

シェア

共通の頻出語TOP30

Lylics Analysis

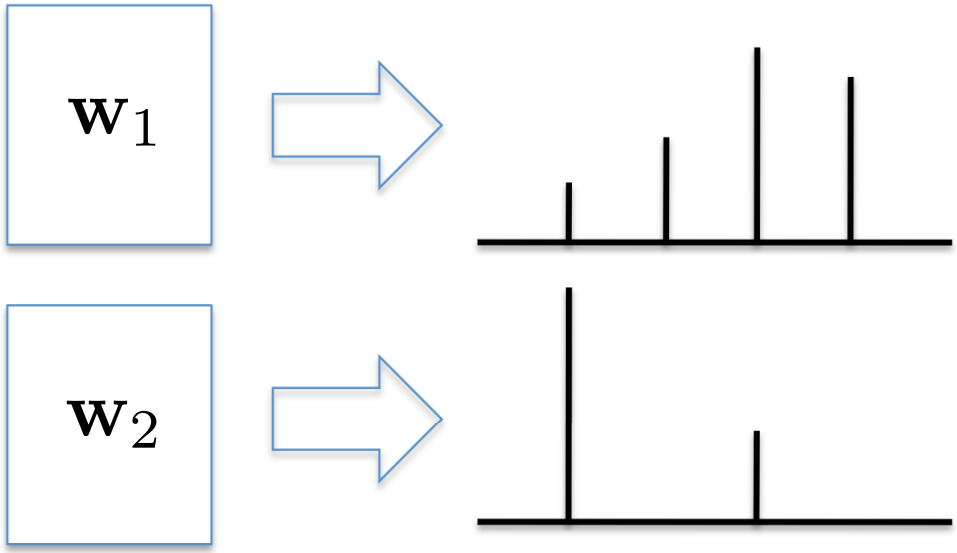
順位	語	合計	順位	語	合計	順位	語	合計
1	君	19,282	11	僕ら	11,517	21	忘れる	10,366
2	僕	15,982	12	明日	11,250	22	行く	10,264
3	夢	14,532	13	胸	11,002	23	言う	10,124
4	今	13,911	14	空	10,765	24	夜	9,989
5	愛	13,395	15	いつ	10,599	25	知る	9,847
6	誰	13,030	16	信じる	10,956	26	抱きしめる	9,707
7	手	12,686	17	未来	10,808	27	変わる	9,569
8	心	12,381	18	笑う	10,679	28	男	9,427
9	見る	12,077	19	キミ	10,578	29	恋	9,276
10	何	11,795	20	生きる	10,471	30	風	9,130

音楽と歌詞

- 統計モデルにできるか? ⇨ もちろん!
- 有名なモデル: トピックモデル

LDA: トピックモデル

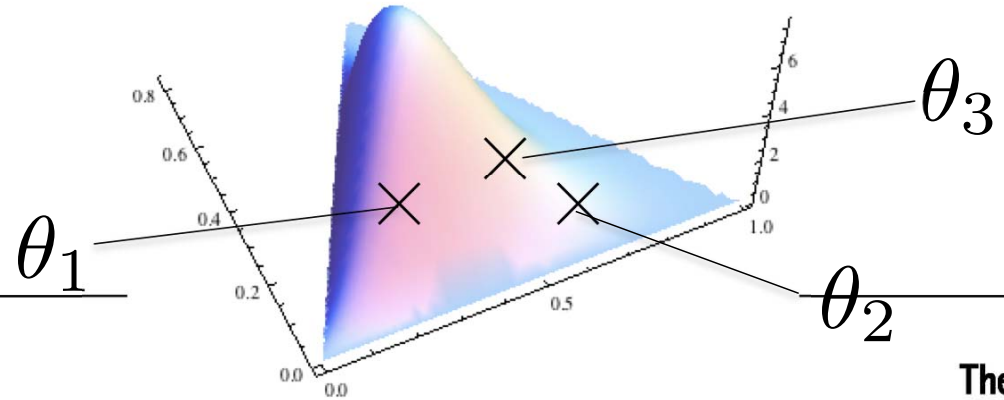
- 文書$\mathbf{w}$を話題(トピック)の混合で表現



$$\theta_1 = (0.1 \;\; 0.2 \;\; 0.4 \;\; 0.3)$$

$$\theta_2 = (0.8 \;\; 0 \;\; 0.2 \;\; 0)$$

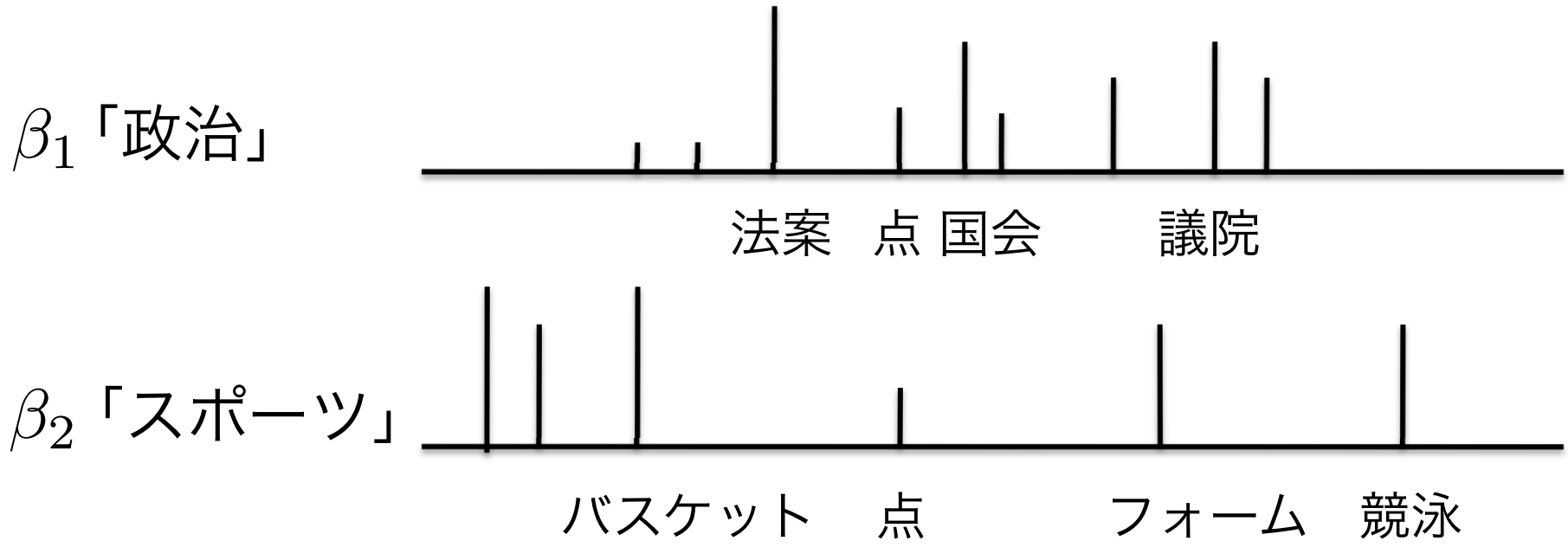
- 混合比 θ をディリクレ事前分布から生成



トピックモデル (2)

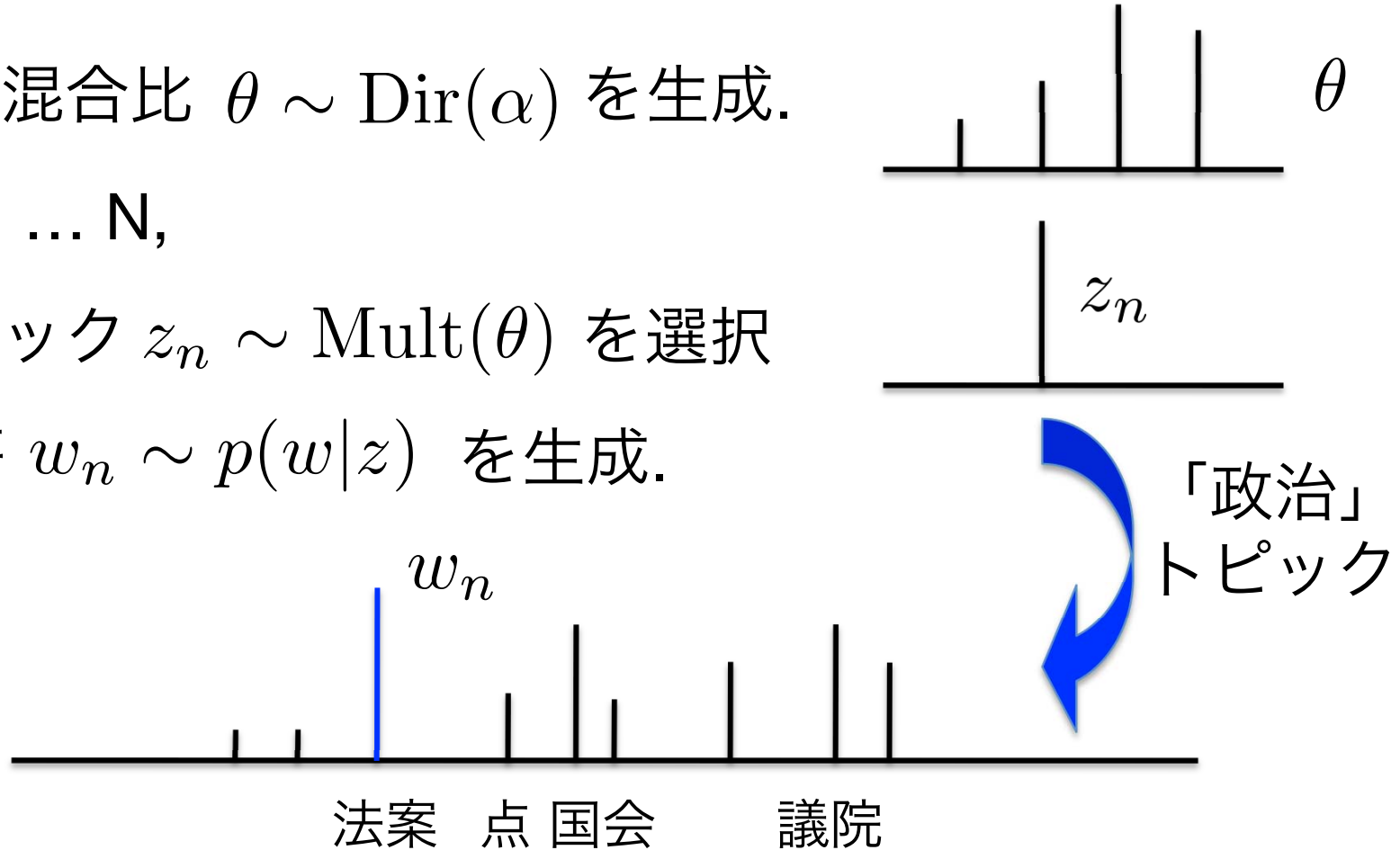
- 「話題」とは?→単語の生起確率分布 $\beta_k = \{ p(w|k) \}$

$(w = 1 \cdots V)$



LDAの文書生成モデル

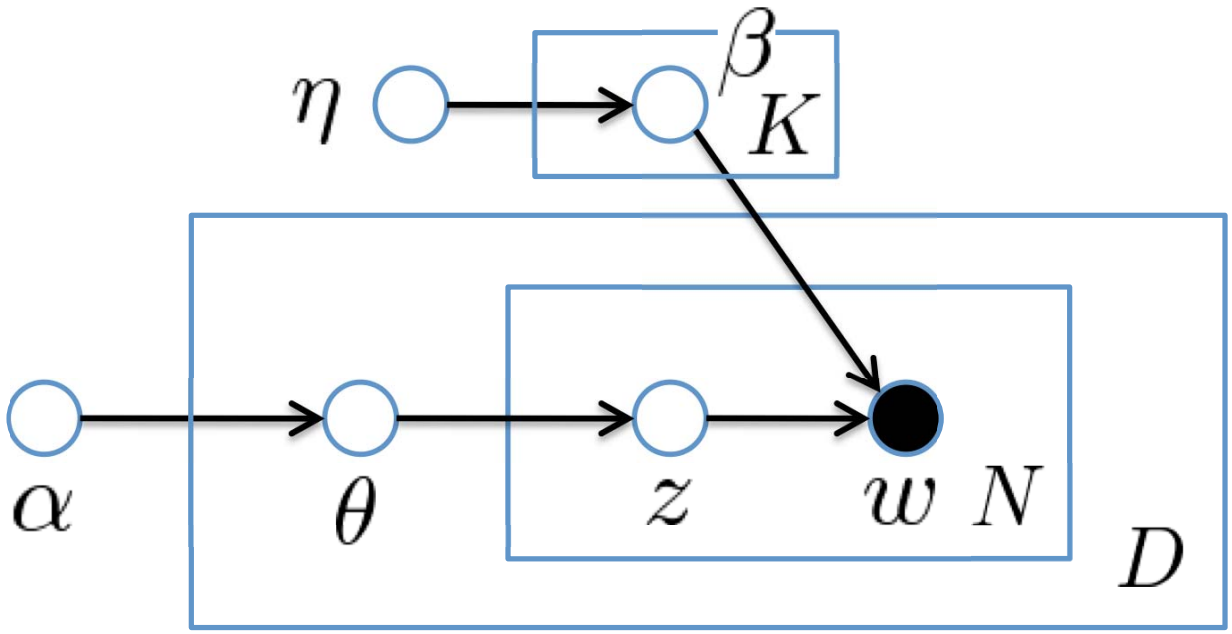
1. トピック混合比 $\theta \sim \mathrm{Dir}(\alpha)$ を生成.
2. For n = 1 … N,
   a. トピック $z_n \sim \mathrm{Mult}(\theta)$ を選択
   b. 単語 $w_n \sim p(w|z)$ を生成.



LDAの学習: Gibbs Sampler

- 導出や実装が簡単で、高性能
- 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

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

Last.fm データセット

- “Million Song Dataset” http://labrosa.ee.columbia.edu/millionsong/ 中の Last.fm データセットのうち、タグの付けられた1,611曲の歌詞
  - Bag of Words形式
  - 頻度順で上位5000語を使用

Last.fm in LDA

Topic 1: “german”		Topic 3: “love”		Topic 17: “young”	
0.064031	ich	0.050848	go	0.107093	danc
0.041963	und	0.050427	love	0.060974	the
0.029936	die	0.047225	let	0.022725	kill
0.025735	du	0.044963	babi	0.018697	cherri
0.021566	der	0.036644	me	0.018126	night
0.020731	ist	0.035958	no	0.016975	lyric
0.019416	in	0.032467	one	0.015383	pop
0.018470	das	0.029634	the	0.015153	jag
0.017061	es	0.024699	more	0.013968	to
0.016384	nicht	0.023832	my	0.013483	no
0.016217	mich	0.022584	time	0.013464	i
0.015953	na	0.018943	in	0.011929	som
0.015548	demain	0.018062	and	0.010176	more
0.015046	auf	0.014692	again	0.009995	kan

Last.fm in LDA

Topic 3: "stopwords"		Topic 9: "french"		Topic 10: "general"	
0.048826	the	0.031964	de	0.184127	i
0.037000	to	0.026116	la	0.074498	me
0.032441	and	0.023961	et	0.069192	you
0.020244	in	0.020578	le	0.050677	to
0.019731	it	0.019688	je	0.032629	my
0.019236	a	0.017437	pas	0.022816	have
0.018974	way	0.016745	a	0.021597	know
0.016048	they	0.016607	les	0.021543	be
0.015976	no	0.016585	que	0.018998	and
0.014010	up	0.014540	un	0.016740	would
0.011777	have	0.013672	tu	0.016288	for
0.011509	with	0.013404	qui	0.016179	love
0.011457	them	0.012692	ce	0.015297	want
0.011078	good	0.012599	e	0.015199	that

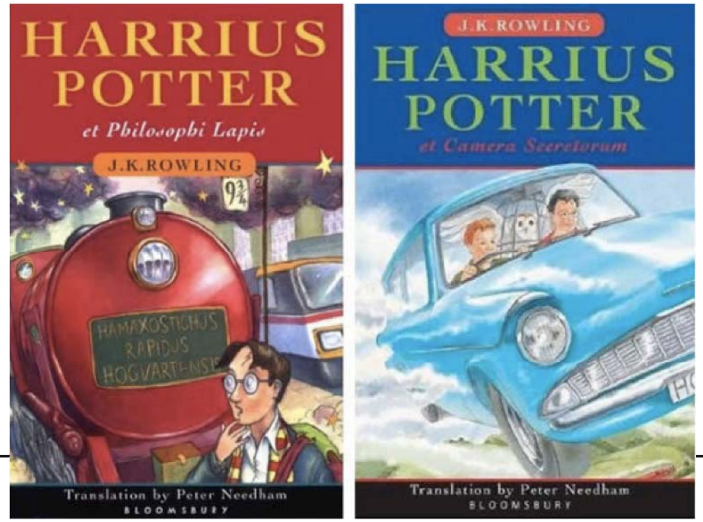
しかし‥

- LDAのGibbsサンプラーの更新式:

$$p(z_{dn}=k|W,Z_{-dn}) \propto \frac{\alpha_k + n(d,k)}{\sum_k \alpha_k + n(d,k)} \cdot \frac{\eta + n(w_{dn},k)}{\sum_w \eta + n(w,k)}$$

  - 各単語は1つのクラスタにしか属さない → 本当?

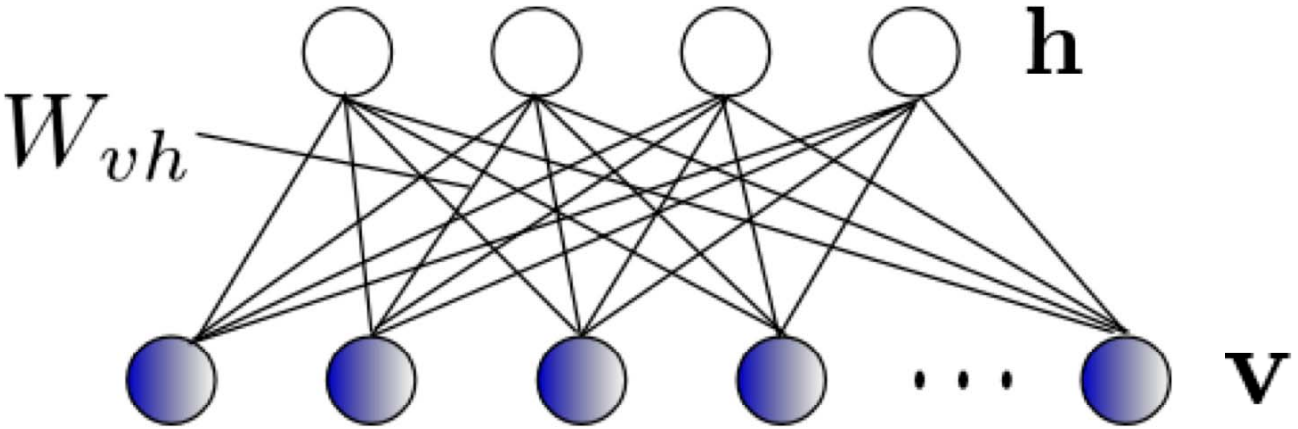
- 文書＝人、単語＝商品と考えてみる
(協調フィルタリング)



さまざまな属性:
小説／本／若者向け／挿絵あり／ラテン語／‥‥

単なるクラスタリングでは表現できない！

Restricted Boltzmann Machines



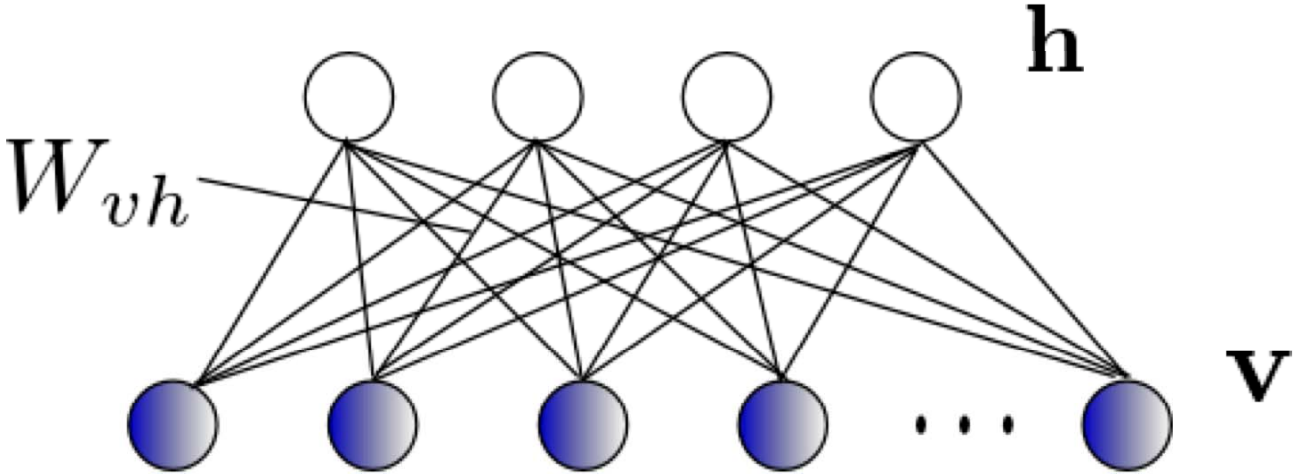
- “Deep Learning”の最も基本的なモデル
  - 出力層 *v* と0/1の潜在層 *h* が重み *W* で結ばれたニューラルネット

Hinton (2002)

- 混合モデルではなく、積モデル (Product of Experts)

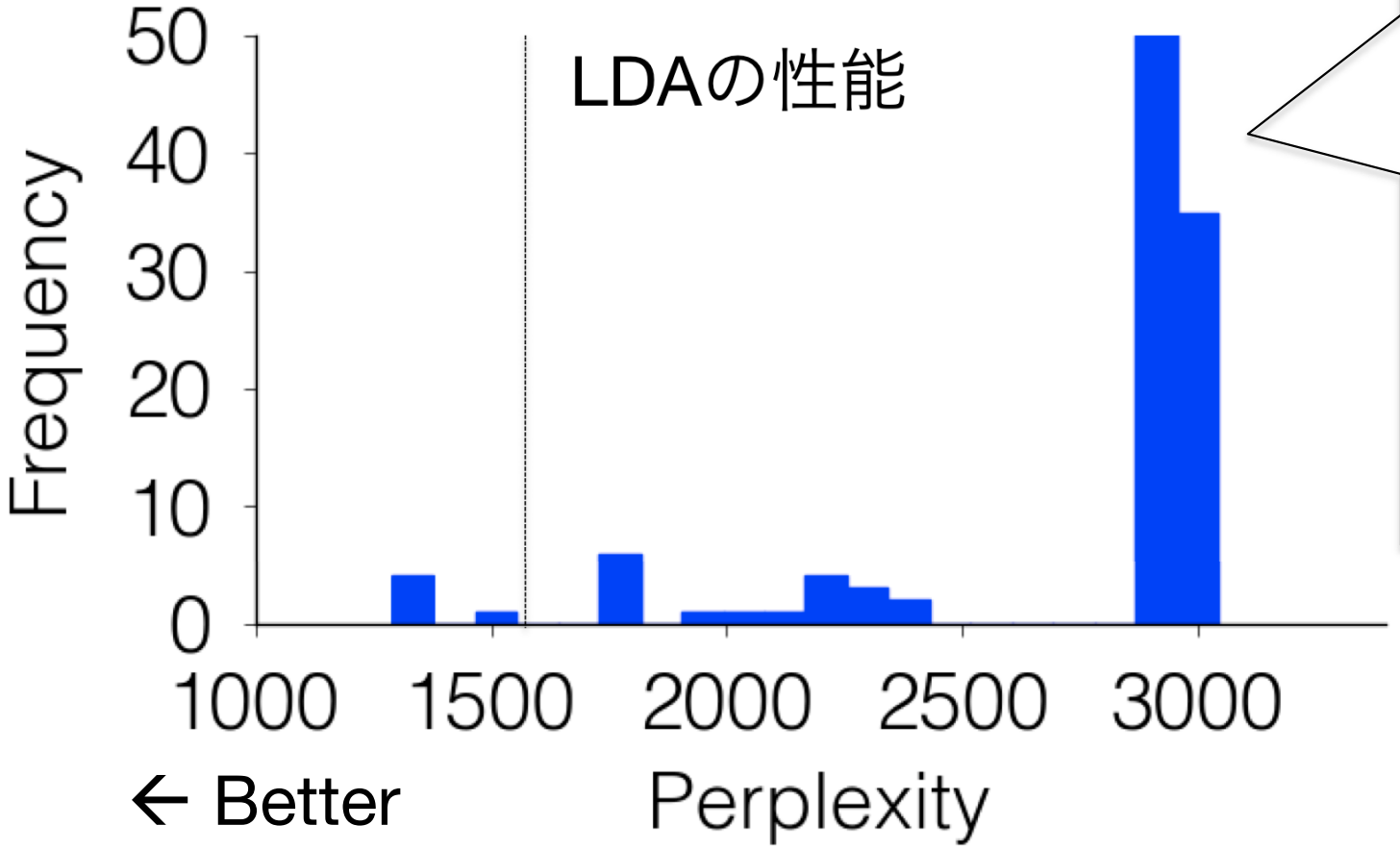
$$p(\mathbf{v},\mathbf{h}) = \frac{\exp(\mathbf{v}^T W \mathbf{h})}{\sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp(\mathbf{v}^T W \mathbf{h})} \propto \prod_i \prod_j e^{W_{ij} v_i h_j}$$

Restricted Boltzmann Machines (2)



- LDAと異なり、意味を分散表現できる
  - 国際経済＝“国際”×“経済”
  - 海外サッカー＝“国際”×“サッカー”
  - 自然言語処理＝ “数学”×“言語学” ・・・・
- しかし、

RBMの最適化の難しさ



Replicated Softmax (Salakhut-dinov 2009) のNIPSコーパスでの実験結果

- RBMには、学習率、ミニバッチサイズ、モーメント、CD iterations、‥などの多数のメタパラメータ
- ほとんどの場合、非常に悪い性能しか出ない

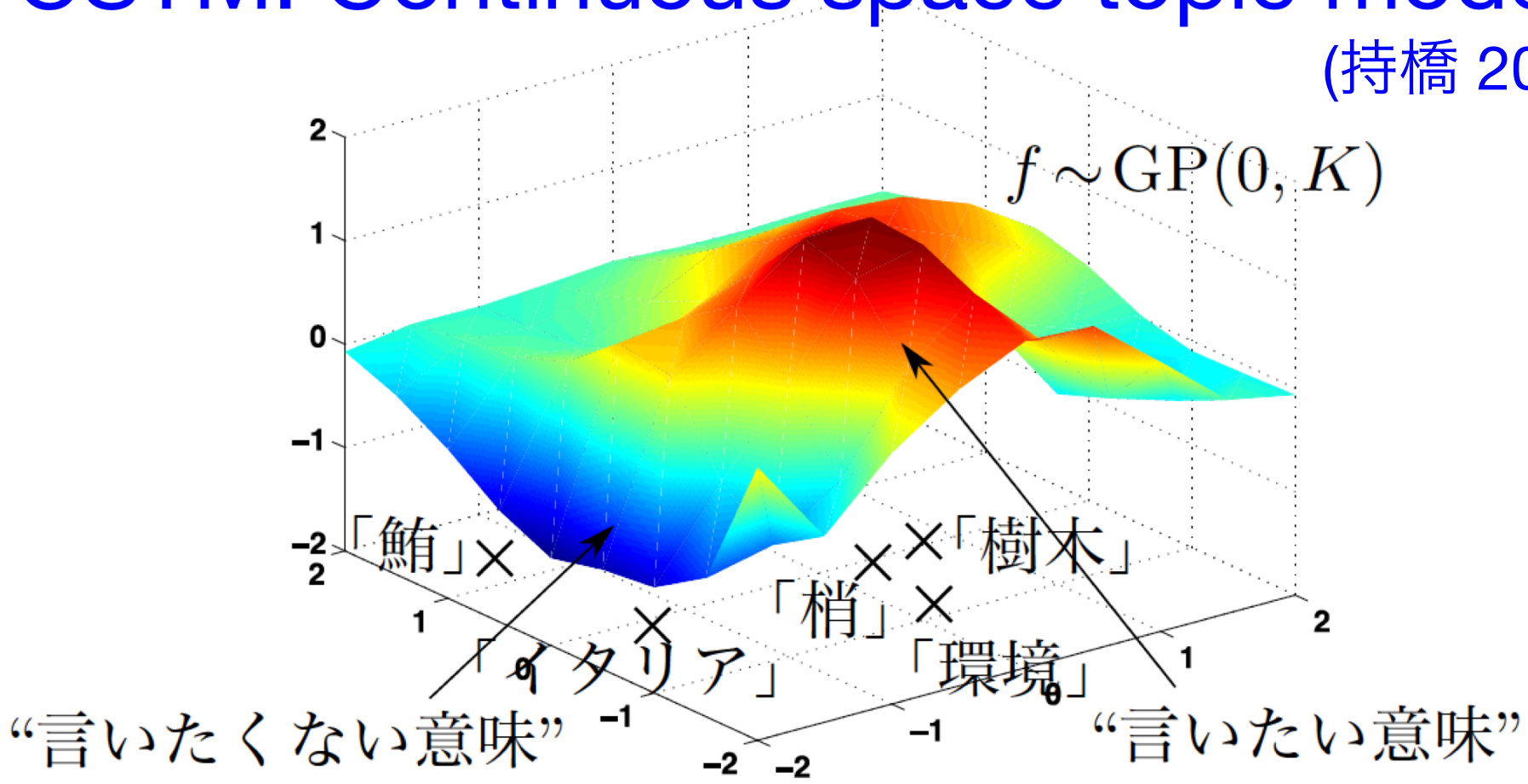
何が問題か?

- RBMは生成モデルがなく、0/1の潜在変数と
  シグモイド関数で強引に正則化している
- RBM, LDAとも、語彙の情報が非常に重要
  - RBM: ニューラルネットの重み W_{vk}
  - LDA: 単語のトピック分布 $p(z|w) \propto p(w|z)p(z)$

- 単語に潜在座標を明示的に与えるモデル.
  - 実は、統計学では Latent space models (Hoff 2002)
    として知られている (社会ネットワーク解析)

CSTM: Continuous space topic models

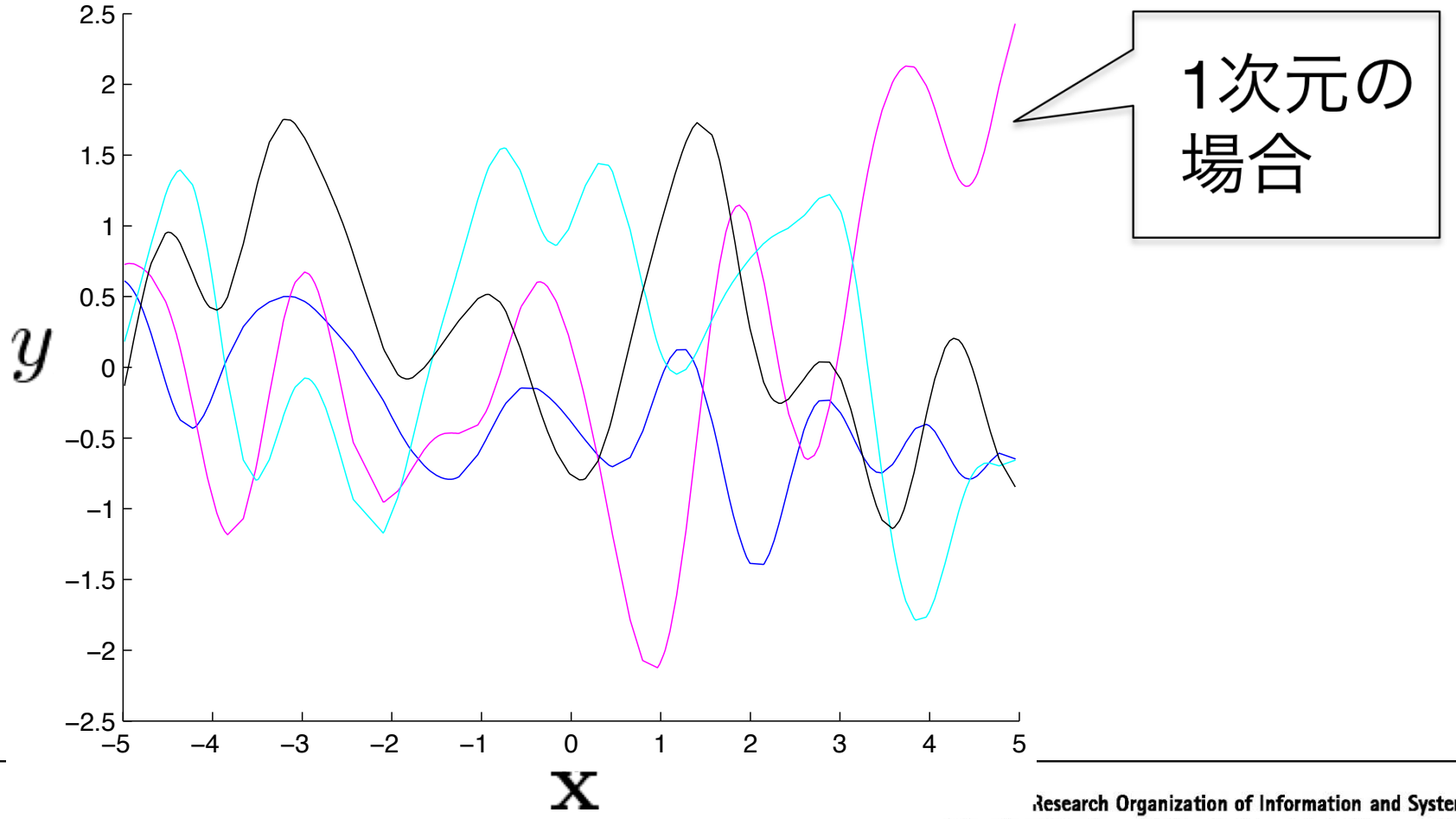
(持橋 2013)



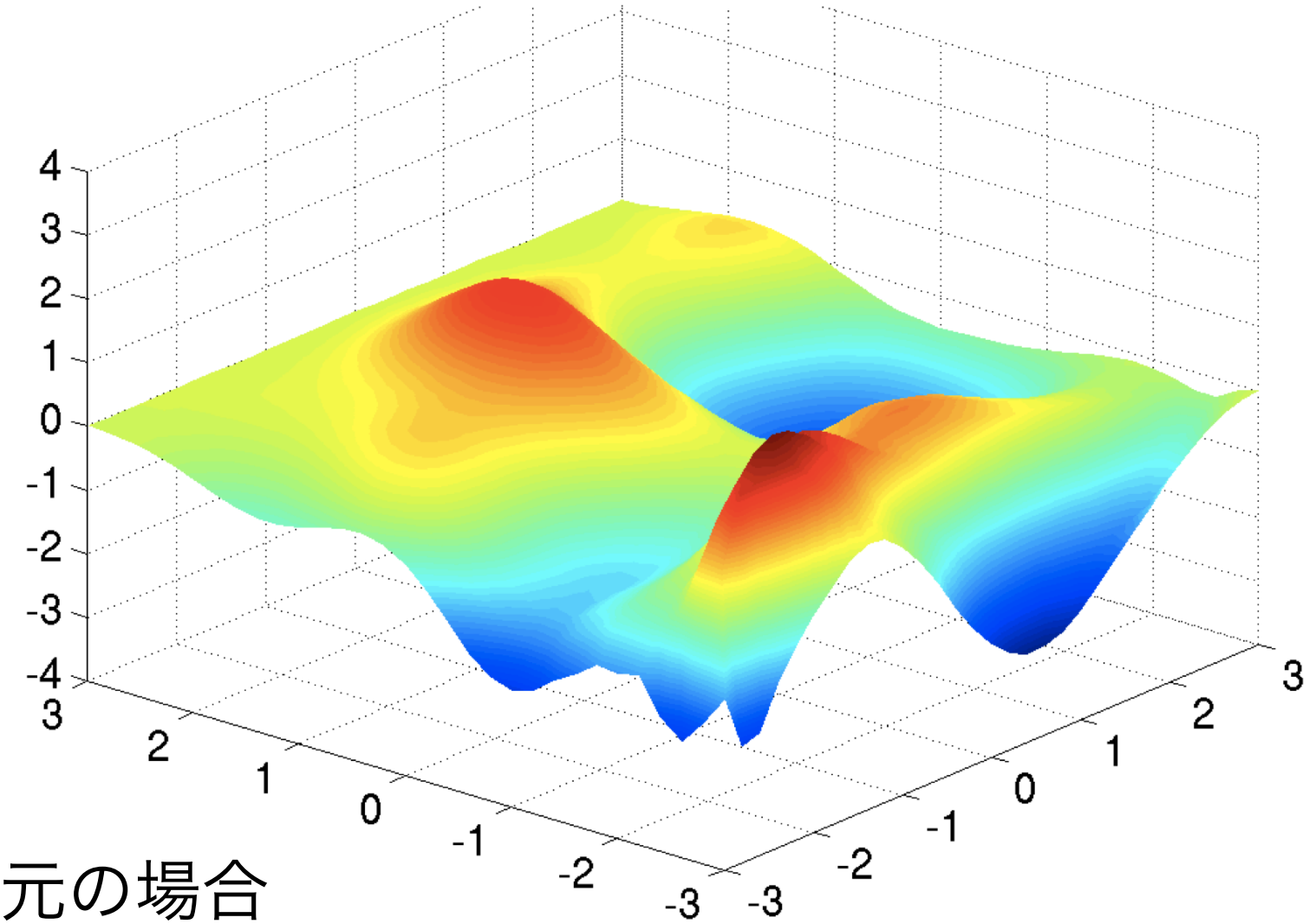
- 単語wはd次元の潜在座標 $\phi(w) \sim \mathrm{N}(0, I_d)$ をもつ
- この上に、ガウス過程 $f \sim \mathrm{GP}(0, K)$ を生成

Gaussian process とは

- ガウス過程: $\mathbf{x} \mapsto y$ への回帰関数を生成する確率分布
  - 実際には、無限次元のガウス分布

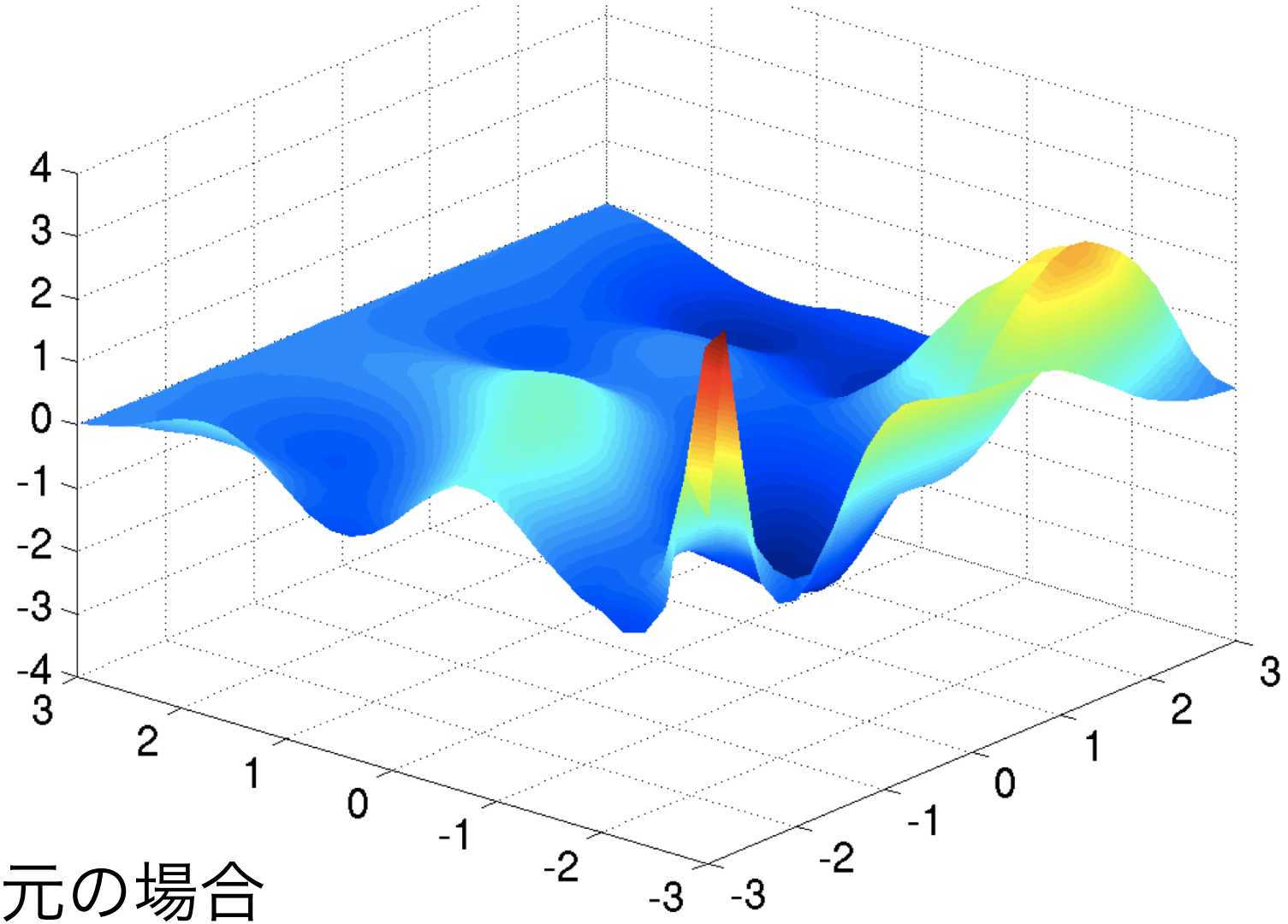


Gaussian processとは (2)



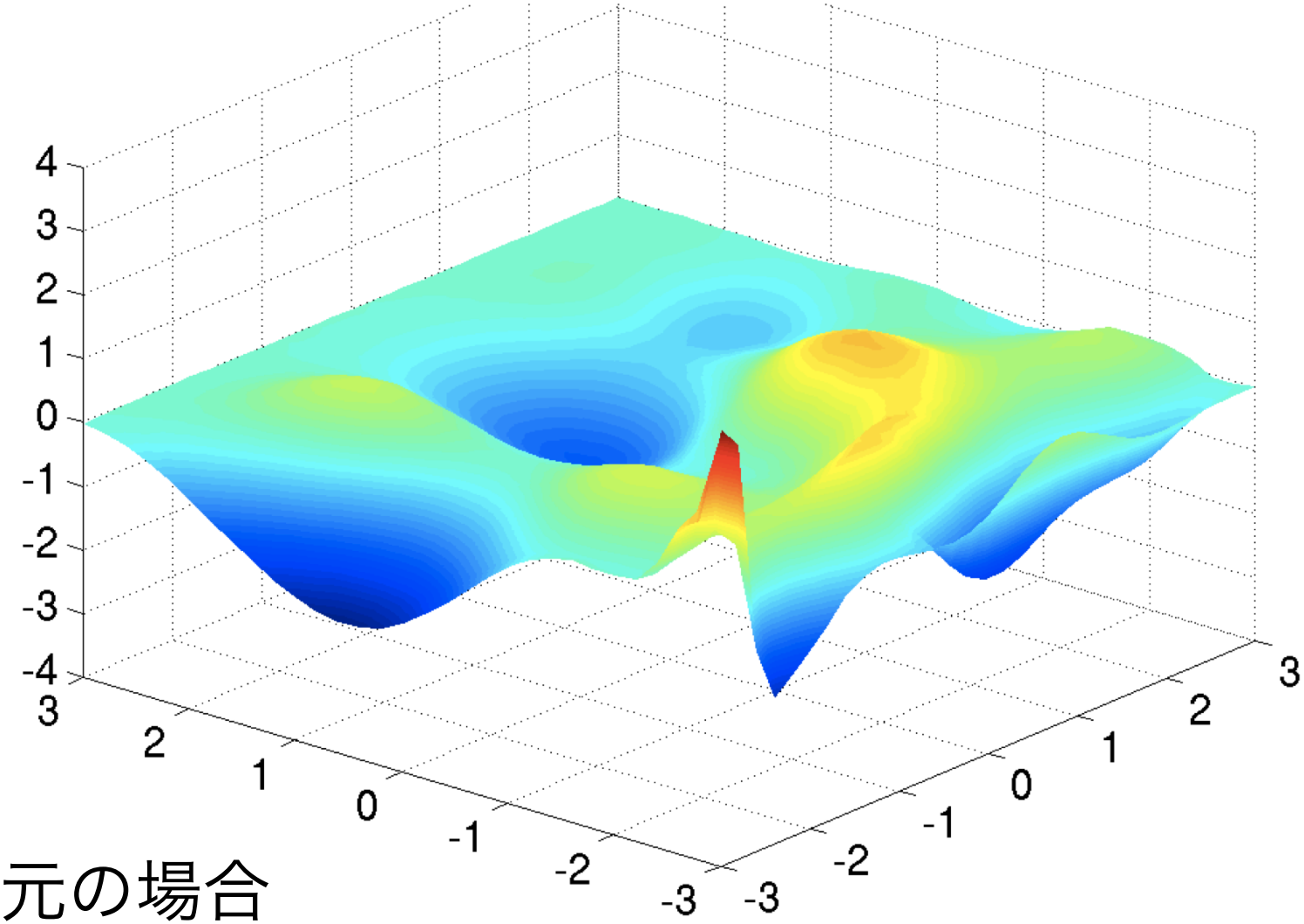
- 2次元の場合

Gaussian processとは (2)



- 2次元の場合

Gaussian processとは (3)



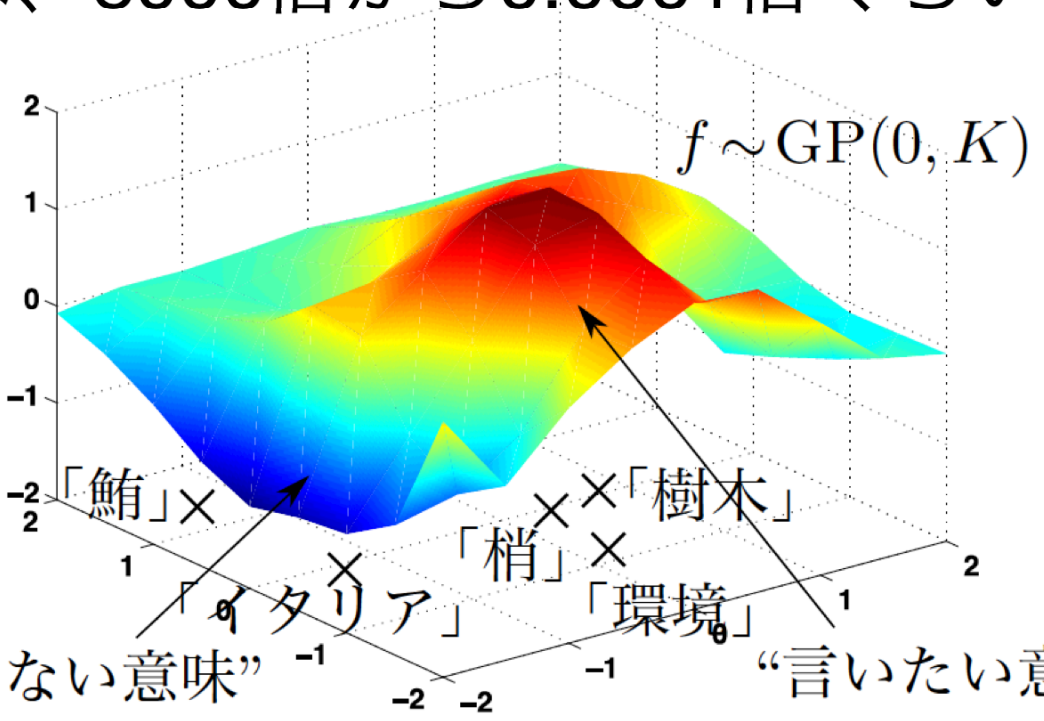
- 2次元の場合

CSTM: 最初のモデル

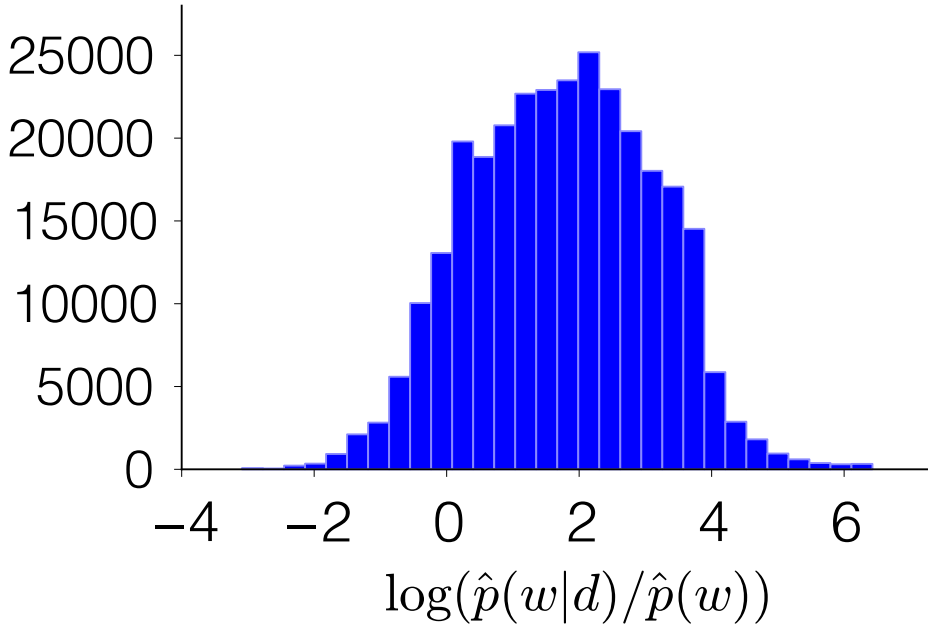
- 単語の平均的な確率(最尤推定) $G_0(w)$を、ガウス過程 $f(w)$でモジュレート

$$p(w|d) \propto e^{f(w)} G_0(w) = \frac{e^{f(w)} G_0(w)}{\sum_w e^{f(w)} G_0(w)}$$

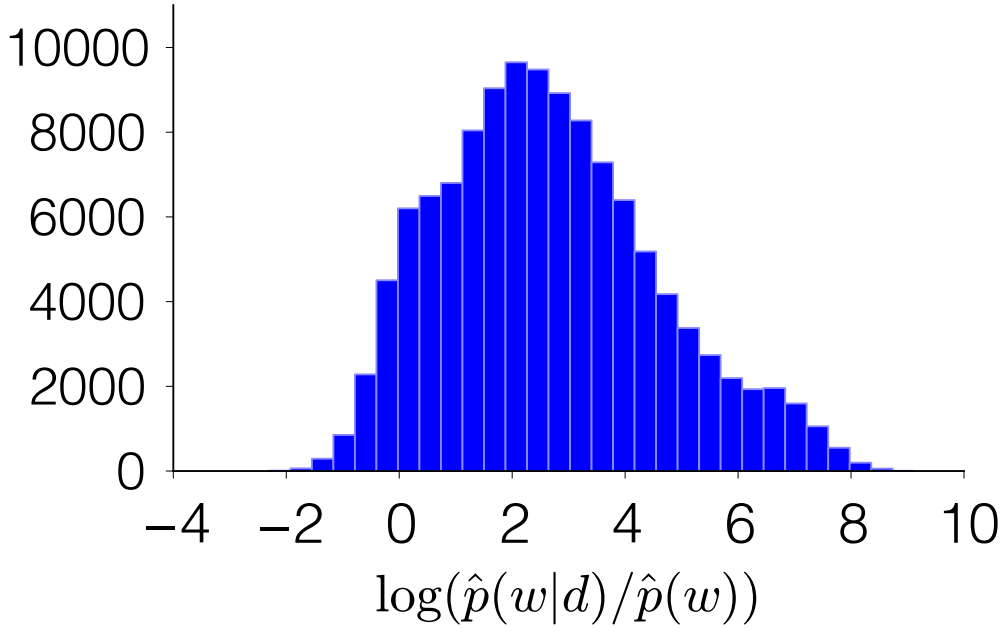
  - $e^{f(w)}$ は、8000倍から0.0001倍くらいの値



Empirical Evidence



Brownコーパス



Cranfield コーパス

- $p(w|d) \propto e^{f(w)} p(w) \Longleftrightarrow f(w) \propto \log\left(\frac{p(w|d)}{p(w)}\right)$ を最尤推定で計算してプロット
- 確率の比はほぼGaussianで分布している!

Polya分布による拡張

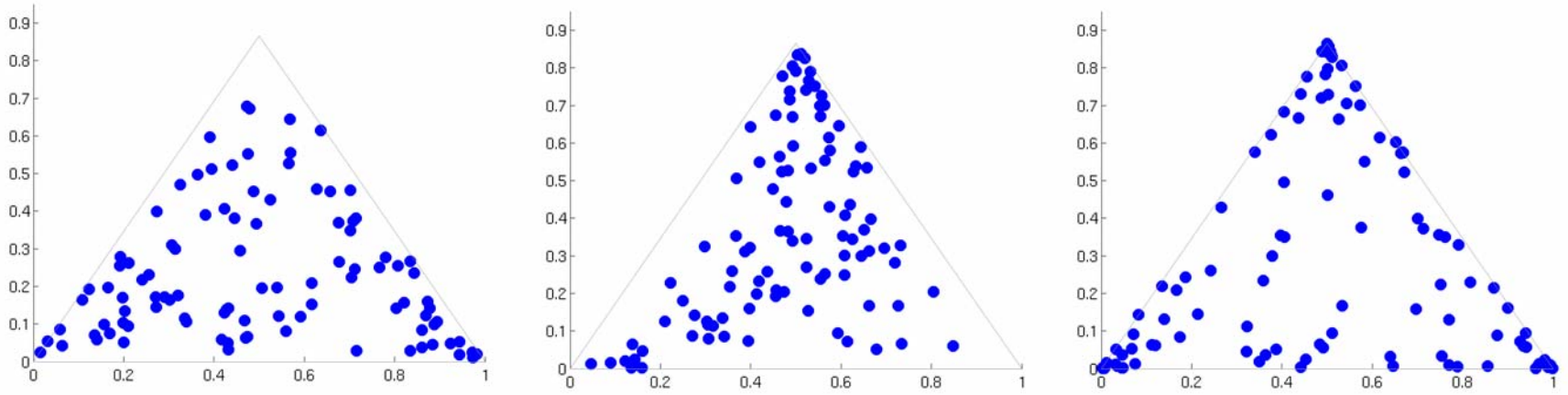
- 言語にはバースト性がある→Polya (DCM)分布

$$\mathrm{DCM}(\alpha) = \int p(\mathbf{w}|\mathbf{p})p(\mathbf{p}|\alpha)d\mathbf{p} = \frac{\Gamma(\sum_k \alpha_k)}{\Gamma(n+\sum_k \alpha_k)} \prod_k \frac{\Gamma(n_k+\alpha_k)}{\Gamma(\alpha_k)}$$

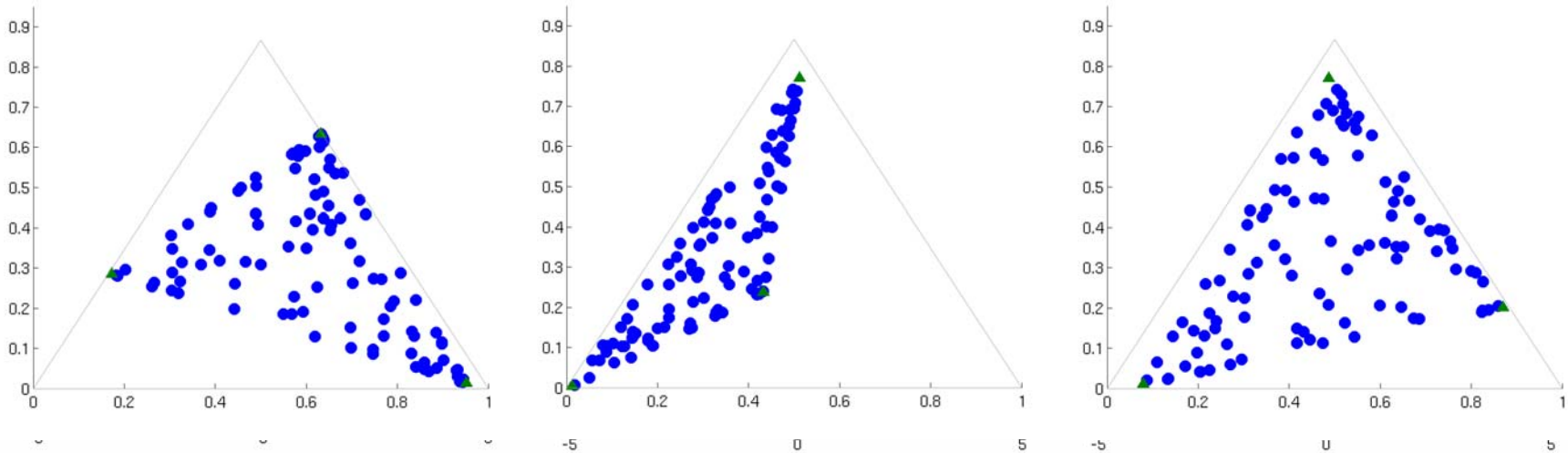
  - Draw $\mathbf{p} \sim \mathrm{Dir}(\alpha)$
  - For n=1..N, Draw $w_n \sim \mathbf{p}$.
- $\alpha = (\alpha(w_1), \cdots, \alpha(w_V))$ を文書ごとに下で生成
  - Draw $f \sim \mathrm{GP}(0, K)$
  - Set $\alpha(w) = \alpha_0 G_0(w) e^{f(w)}$
  - Draw $\mathbf{w} \sim \mathrm{DCM}(\alpha)$.

CSTMとLDAの単語確率分布

CSTM



LDA



- CSTMは全単語Simplexを網羅 (和が1の制約がない)

学習

- ガウス過程から生成した関数*f*は文書ごとに無限次元
  → 学習不可能
- DILN (Paisley+ 2012)と同様に、補助変数uを導入
  - 単語座標の行列を $\Phi = (\phi(w_1), \cdots, \phi(w_V))$ とする
  - $u \sim \mathrm{N}(0, I_d)$ のとき、$f = \Phi u$ はuを積分消去して

$$f|\Phi \sim \mathrm{N}(0, \Phi^T \Phi) = \mathrm{N}(0, K)$$

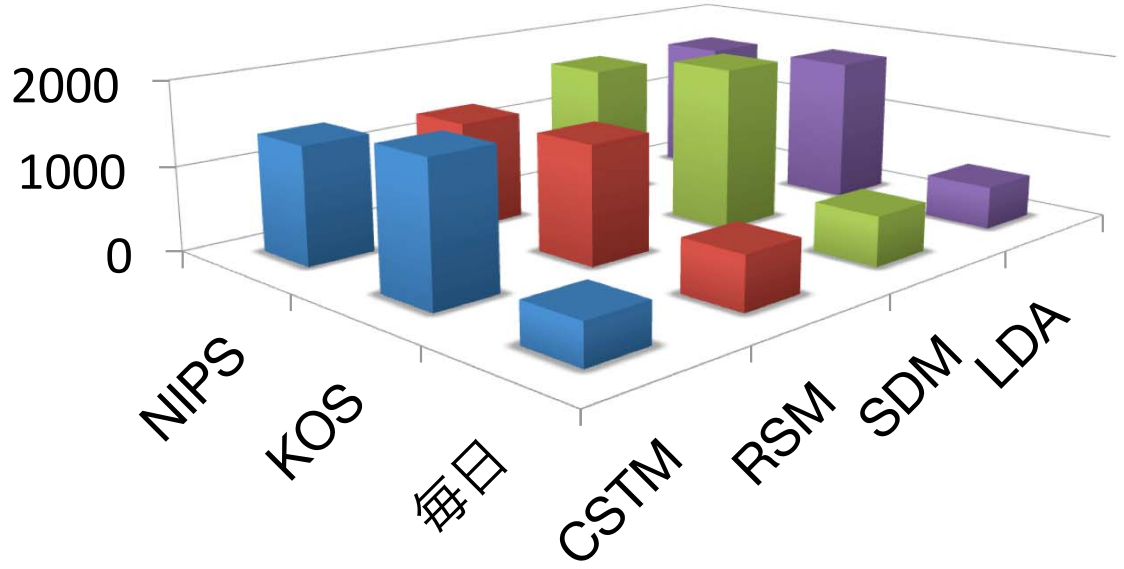
  - これは、線形カーネル $k(w_i, w_j) = \phi(w_i)^T \phi(w_j)$ を使ったGPと等価なことを意味する

- $\alpha(w) = \alpha_0 G_0(w) e^{u^T \phi(w)}$ として、u と $\phi(w)$ の学習問題!

学習 (2)

- 通常のMH MCMCで、単語と文書の潜在座標を学習
  - For j = 1 .. J,
    - for i = randperm(1 .. D),
      - Draw u' ~ N(u, σ^2) & MH-accept(u'); Update Z
    - For w = randperm(1 .. W),
      - Draw ϕ'(w) ~ N(ϕ(w), σ^2) & MH-accept(u'); Update Z_1..Z_N
    - z ~ N(0, σ^2); α_0' = α_0 ・ exp(z)
      - If MH-accept(α_0') then α_0=α_0'
  - 実際は、uとϕ(w)の更新をランダムに混合
- 単語間に強い相関があるため、勾配法では局所解

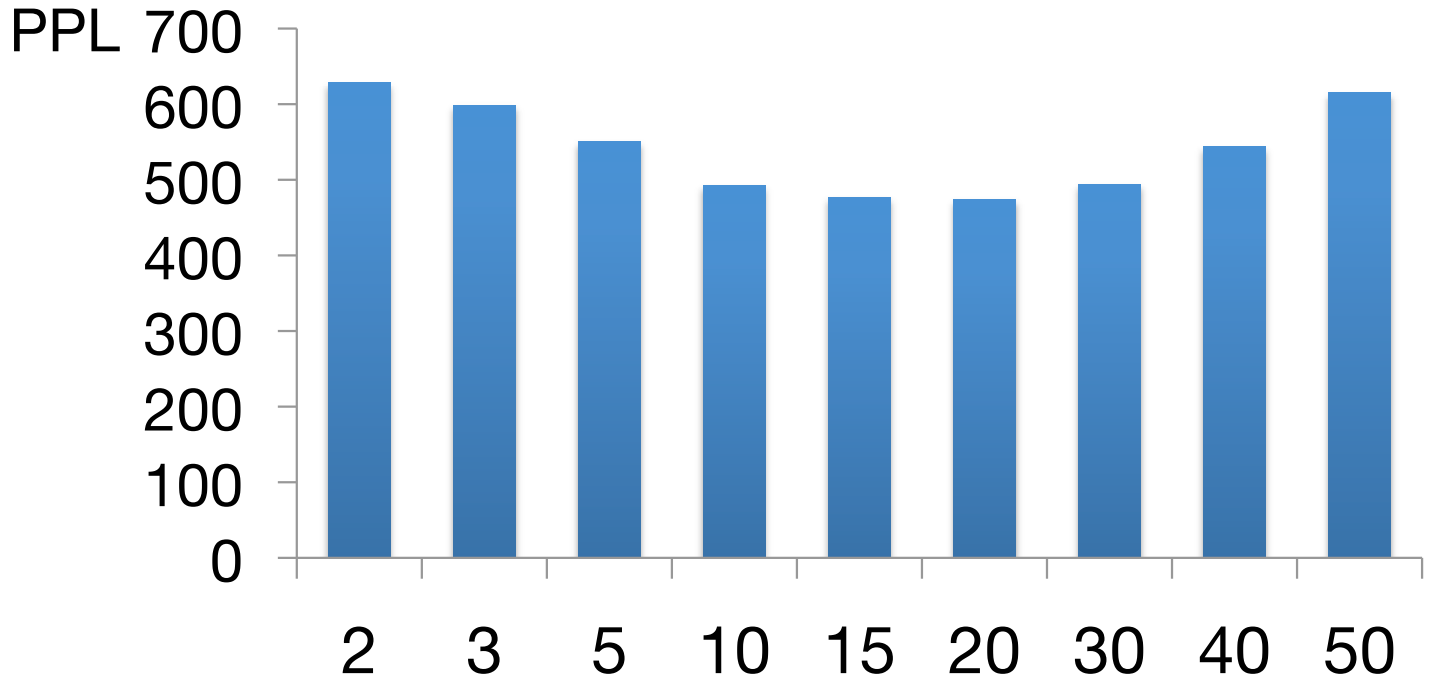
実験結果 (予測パープレキシティ)



	CSTM	RSM	SDM	LDA
NIPS	1383.66	1290.74	1638.94	1648.3
KOS	1632.35	1396.61	1936.25	1730.7
毎日新聞	466.83	622.69	582.37	507.39

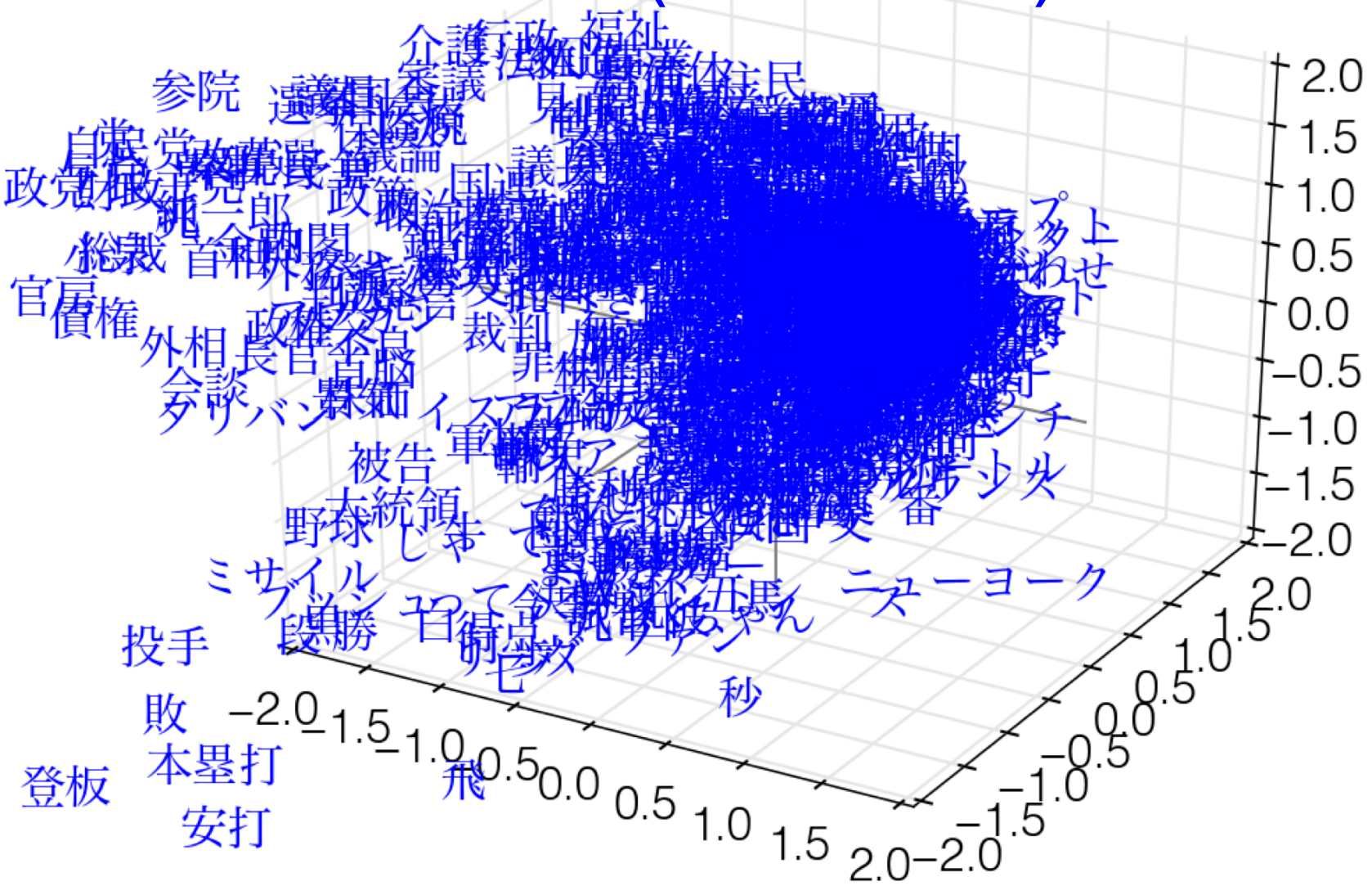
CSTMの次元選択

- 毎日新聞データでの性能と潜在次元数



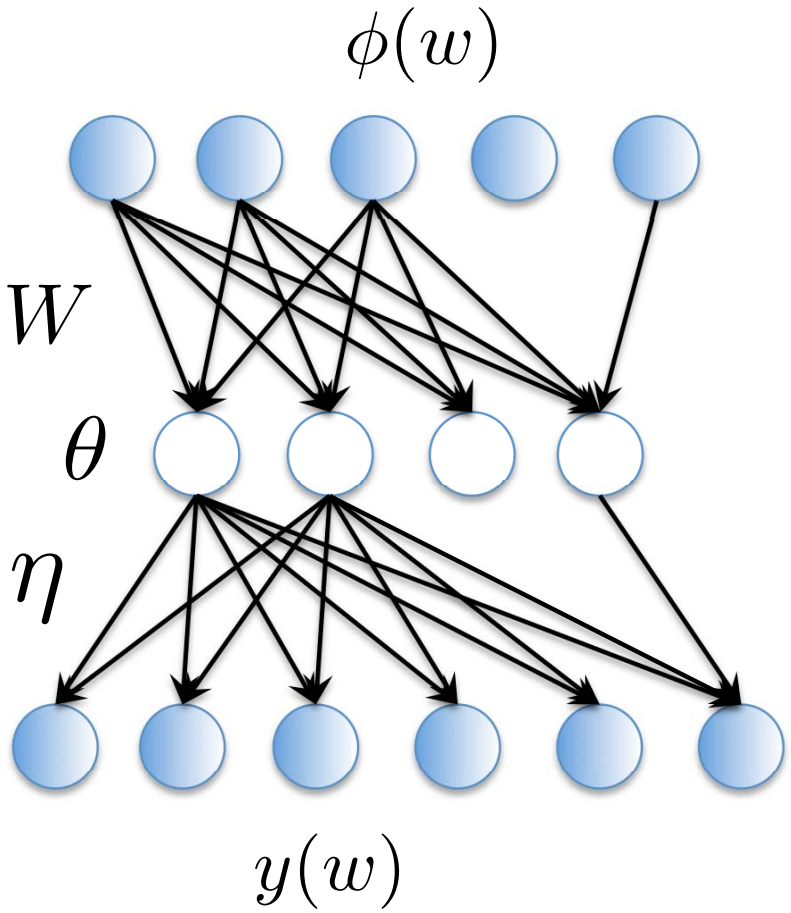
  - 文書の潜在次元が連続なため、小さい値で高性能
  - 次元選択を行う簡単な方法はない (Beta FA?)

毎日新聞テキスト (2000年度)



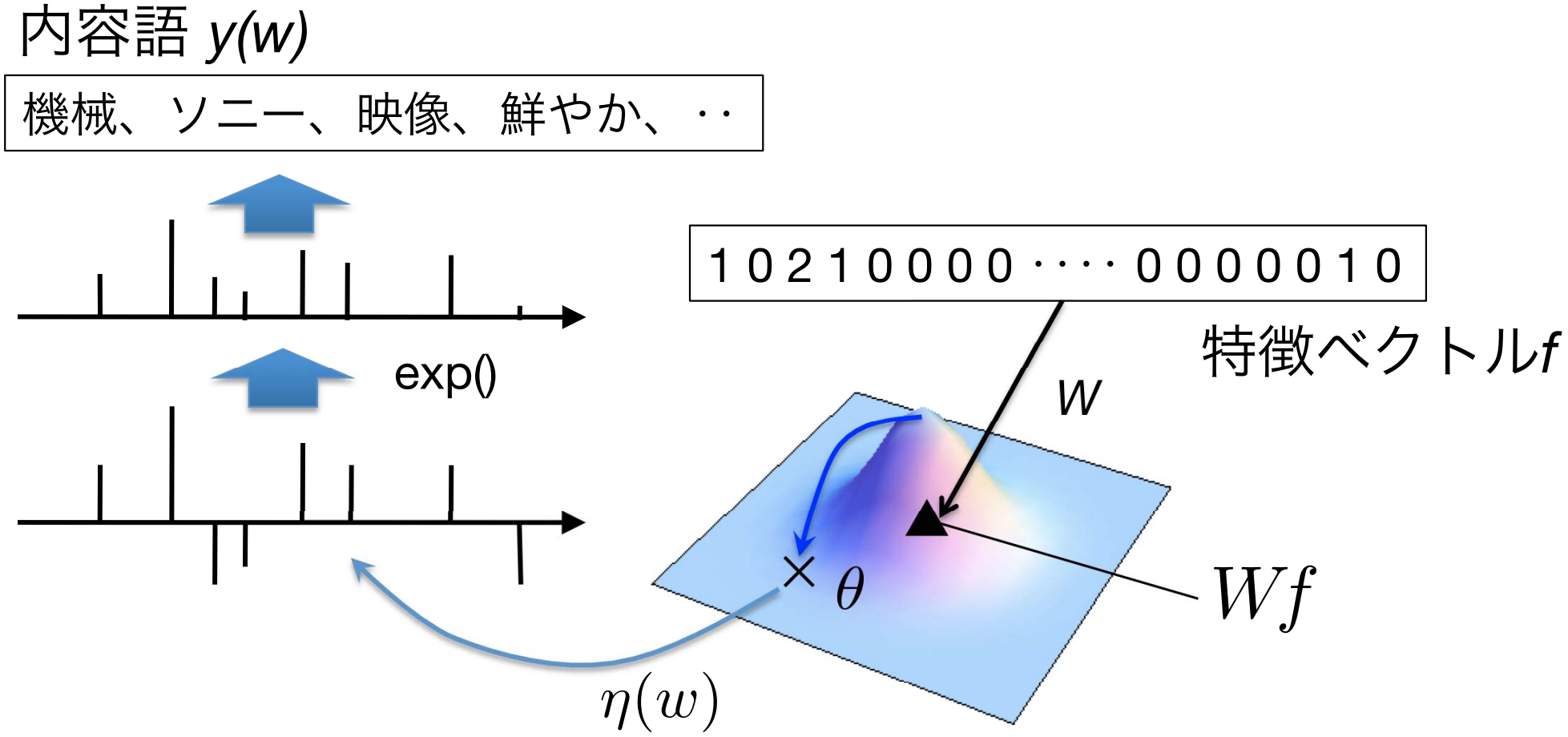
- 出現に偏りの大きい語ほど原点から遠くに位置する

潜在的回帰モデル



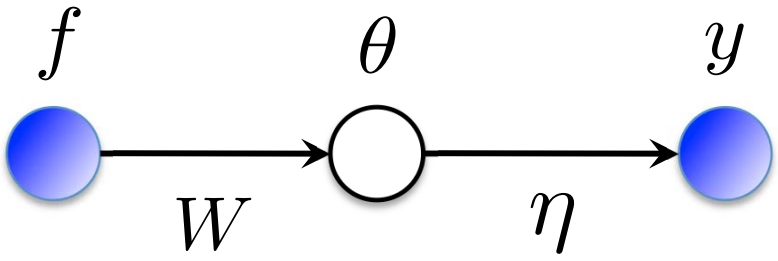
- テキストの共変量φ(w)と内容単語y(w)を直接リンクさせるのは難しい
  →潜在層θに‘意味’を集約
- まずφ(w)からの線形回帰+ノイズでθが生成され、θからさらに内容単語たちy(w)が確率的に生成される

Latent Linear topic model (lltm)



- 共変量の特徴からの回帰+ノイズで、観測された語y(w)が生成される

Latent Linear topic model (2)



- 確率で表すと、

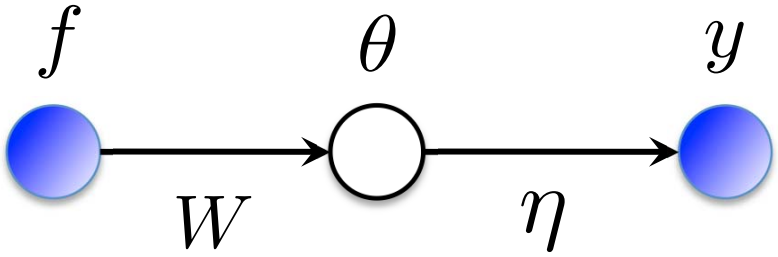
$$p(y|f) = \int p(y|\theta)\, p(\theta|f) d\theta$$

$$\propto \int \prod_w \left(\frac{e^{\eta(w)^T \theta} G_0(w)}{Z} \right)^{c(w)} \cdot \exp\left(-\frac{\beta}{2} (Wf - \theta)^2 \right)$$

yの中に単語wが現れた頻度

  - $G_0(w)$ は単語wの “デフォルト”確率で最尤推定する

Latent Linear Topic Model (3)



- 学習はMCMC(θ および η)+ベイズ線形回帰(W)
  - θ, η は普通のランダムウォークMH
  - Wは θ を目的変数とした回帰モデルのガウス事後分布からサンプル

$$p(y|f) = \int p(y|\theta)p(\theta|f)d\theta$$

$$\propto \int \prod_w \left(\frac{e^{\eta(w)^T\theta} G_0(w)}{Z} \right)^{c(w)} \cdot \exp\left(-\frac{\beta}{2}(Wf - \theta)^2 \right)$$

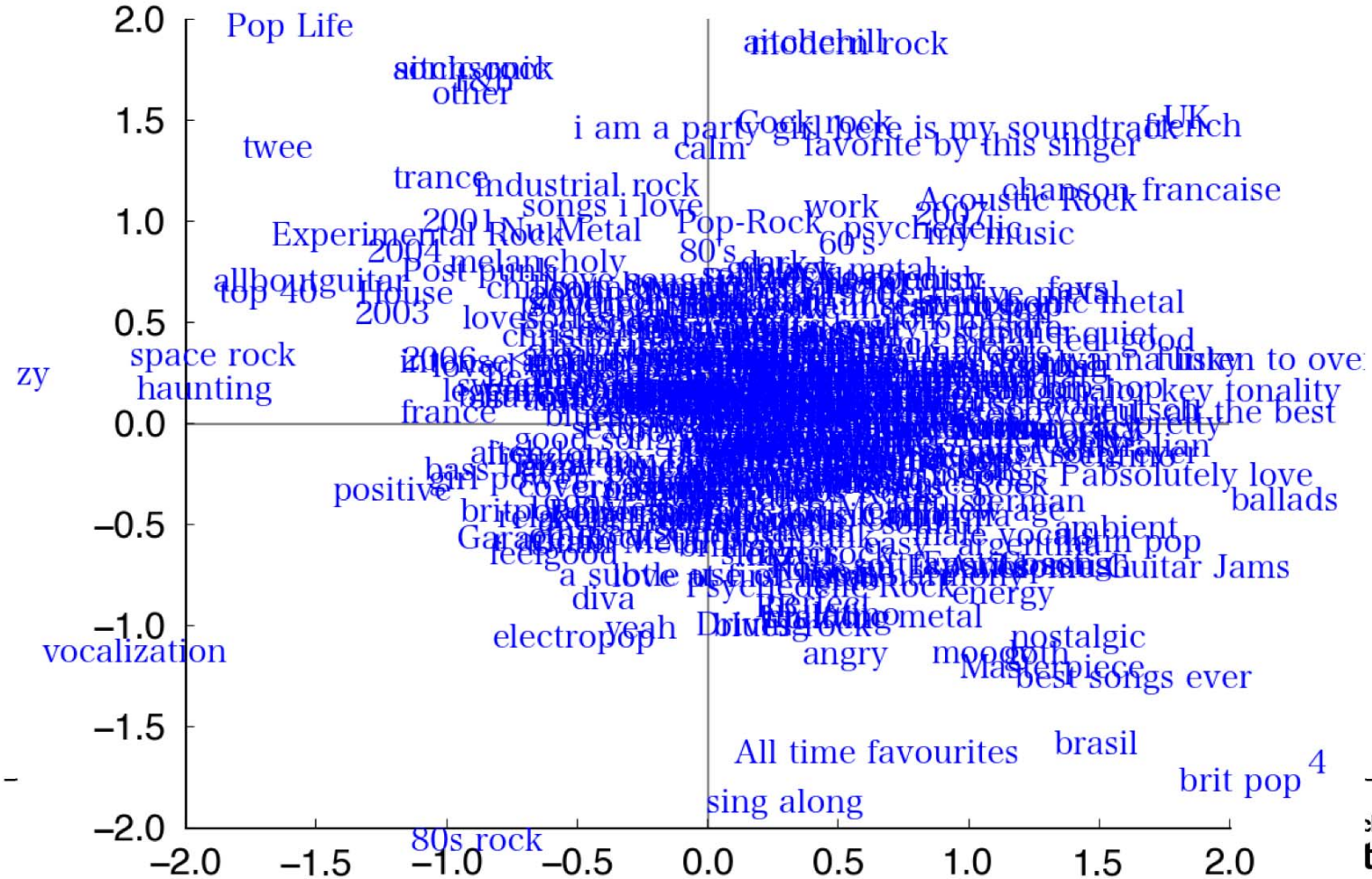
β も確率変数

Last.fmデータ

- Last.fmの各曲についているタグ(Rock,80s,Electro pop, …)を入力の特徴として使用
  - 上位5,000個の特徴
  - 5000次元の離散データ(タグ)→10000次元の離散データ(歌詞)への回帰問題
  - MCMC 100 iterations, K=2,10

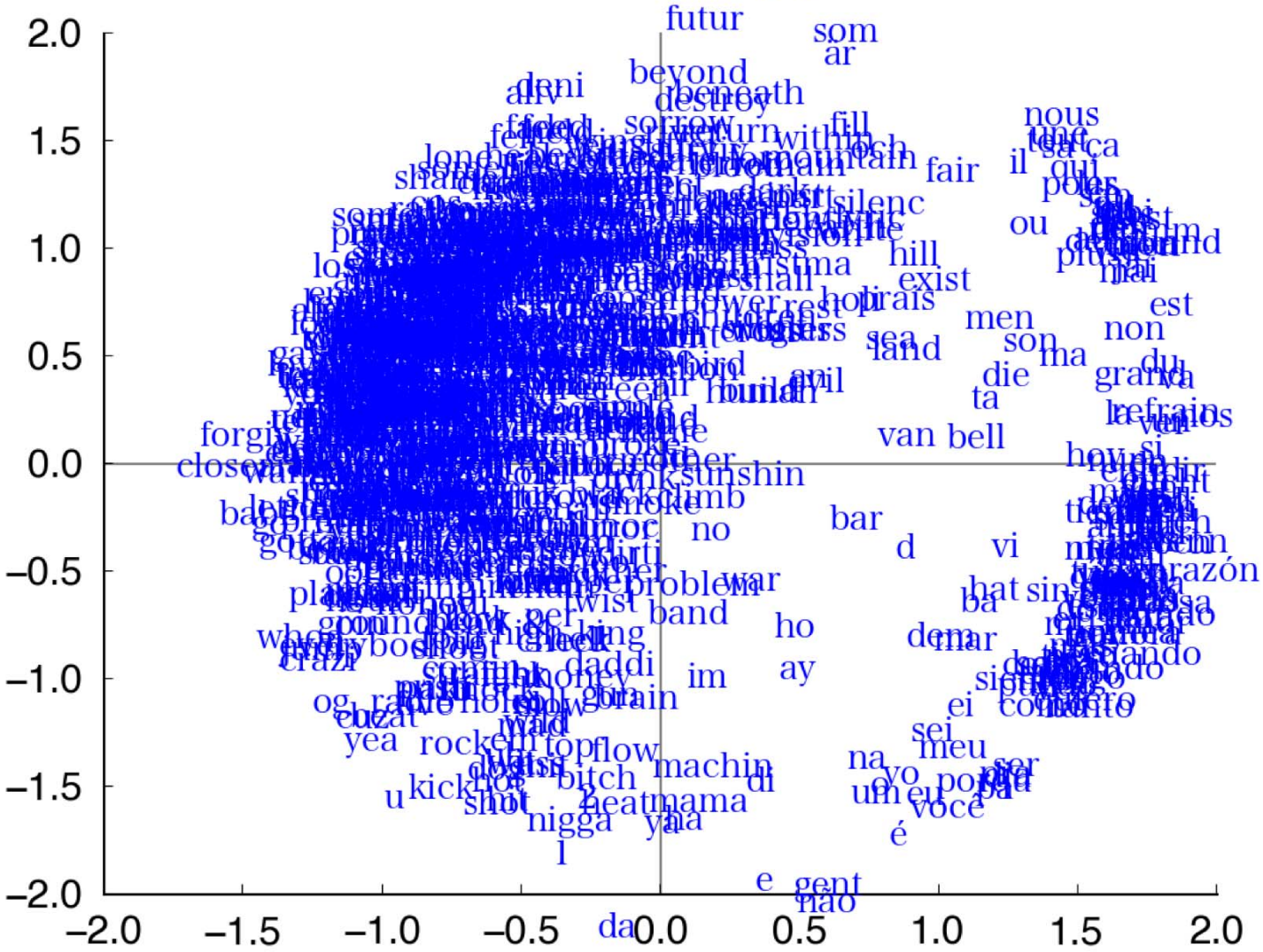
Last.fm regression

- タグ特徴の潜在層への回帰係数をプロット
  - 図示のため、K=2次元に圧縮して学習



Last.fm regression (2)

- 歌詞の単語の潜在座標 $\phi(w)$ をプロット (K=2)



Last.fm 歌詞予測

- タグから潜在的回帰を通じて、歌詞を予測
  - 「普通より確率の高くなる語」の上位語

タグ “rock”		タグ “love”		タグ “female vocalists”	
2.096279	donc	2.069825	rum	2.298850	illumin
2.053631	mere	2.025971	dancin	2.189100	independ
2.008083	mississippi	2.024850	famous	2.185653	crawl
1.964316	toni	2.007292	anybodi	2.150131	comprehend
1.943512	modern	1.971674	cancer	2.131693	hustl
1.881520	brooklyn	1.937310	whoa	2.108225	carv
1.843006	losin	1.913502	wretch	2.101845	spite
1.838629	rewind	1.904969	glimps	2.099663	fade
1.828743	juli	1.904207	spell	2.096050	depress
1.825501	hug	1.880279	lane	2.090748	wrath
1.816417	sleepless	1.855865	kneel	2.085099	gypsi
1.761052	goodby	1.846672	dizzi	2.081990	shallow

まとめ

- ベイズ統計の手法を用いることで、言語と同様に、記号を用いる楽曲データが解析できる
  - 複雑な階層モデル
  - 音響信号だけからは分からない知識
- 音響と言語をつなぐ手法が必要→潜在的回帰モデル
  - 回帰モデルの目的変数自体が未知の潜在変数
  - パラメータのベイズ事後分布からのサンプリング
- 歌詞のより緻密なモデル化が課題

今後の研究課題

- 歌詞を自動生成する統計モデル
  - n-gram (∞-gram)だけでなく、文法に基づいた生成
  - 楽譜情報からの回帰 (離散時系列への回帰問題!)
- 音響信号の教師なし学習との接続

終わり

- ご清聴ありがとうございました。