

---

第2回IBISMLチュートリアル

機械学習に基づく自然言語処理  
—教師なし学習と最近の話題—

持橋大地

統計数理研究所

*daichi@ism.ac.jp*

2013-11-10(日)

IBIS 2013

# 本チュートリアルの目的

---

- 「教師あり学習」と「教師なし」学習の関係について、実際の立場から
- 自然言語処理における教師なし学習は、どんなものが可能か
  - 自然言語処理を題材にした教師なし学習入門
- 教師なし学習の可能性と展望
  - 教師なし学習 = クラスタリングではない!
- 扱わないもの: 連続系の確率モデル、音・画像などの教師なし学習 (非常に重要)

# 目次

---

- 教師なし学習 (Unsupervised Learning) とは
- 簡単なモデルの教師なし学習
- 複雑なモデルの教師なし学習
- 自然言語処理研究の先端での教師なし学習 & 関連する統計モデルについて

---

# 教師なし学習の概要

# 教師あり学習

---

- $\mathbf{x} \in \mathbb{R}^n$  : 入力値

- 例  $\mathbf{x} = (0 \ 2 \ 1 \ 0 \ 0 \ \dots \ 1 \ 0 \ 4)$

ある文書における単語の出現回数

$$\mathbf{x} = (17 \ 5 \ 3 \ 2 \ 108 \ 91 \ 2 \ 34 \ \dots)$$

*When he was a young boy, a book ...*

ある文を単語のIDの列に直したものの

- $\mathbf{y} \in \mathbb{R}^m$  : 出力値

- 非常に多くの場合、 $m = 1$  (スカラー)かその系列 / 組み合わせ

- 例  $\mathbf{y} = y \in \{0, 1\}$  ある文書が迷惑メールか否か

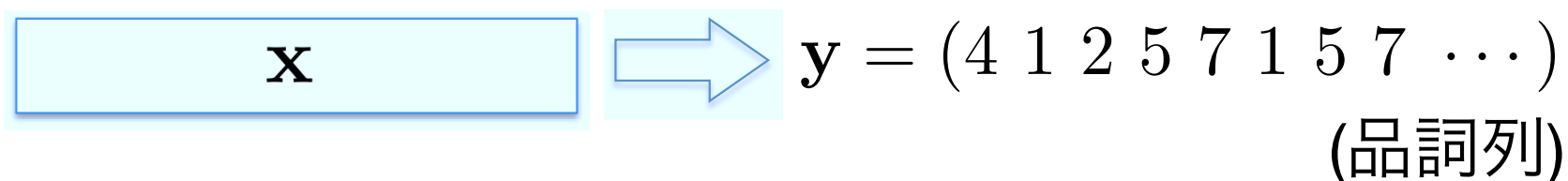
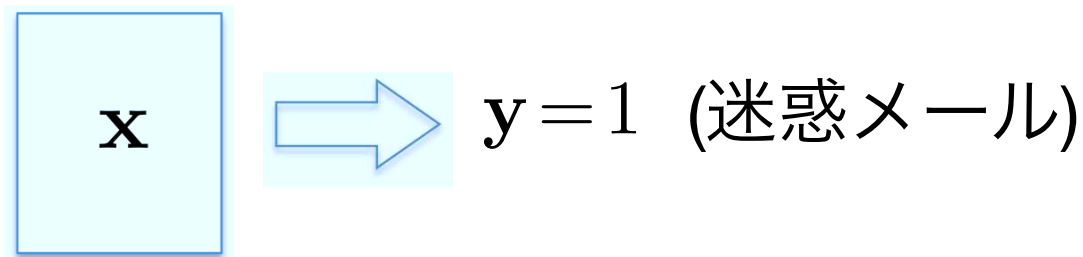
$$\mathbf{y} = (4 \ 1 \ 2 \ 5 \ 7 \ 1 \ 5 \ 7 \ \dots)$$

上の文に対応する品詞列 (接続詞, 名詞, 動詞, ...)のID

## 教師あり学習 (2)

---

- 教師あり学習の目的:  $\mathbf{x}$  から  $y$  を予測する



- 確率モデル:  $p(\mathbf{y}|\mathbf{x})$ 
  - 回帰 / 分類問題 (を系列や木に拡張したもの)
  - $\mathbf{x} \mapsto \mathbf{y}$  の写像さえ学習すればよい!
    - ex. MeCab (CRF), Cabocha(SVM), Jubatus (線形), ...

## 教師あり学習 (3)

---

- $p(\mathbf{y}|\mathbf{x})$  の学習には大量の教師データ  $\{\mathbf{y}_n\}_{n=1}^N$  が必要
  - 「文に対する正しい形態素解析」の数万文の集合
    - 話し言葉、崩れた言葉の場合は？
    - 「正しい品詞」とは？
    - 人手の解析ミスはないか？ [→クラウドソーシング(鹿島さん)]
  - 「この文書が属する正しいカテゴリ」の集合
    - “正しいカテゴリ”が常に一意に決まるか？
    - そもそも、カテゴリの定義は？
  - 正解が簡単には定義できない場合が多い



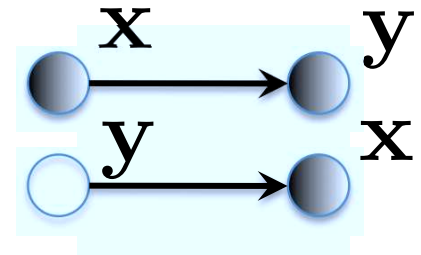
一般には、こちらの方が難しい問題

「データそのもの」からの学習 = 教師なし学習

## 二つの学習の関係について

- 教師あり学習 :  $p(\mathbf{y}|\mathbf{x})$

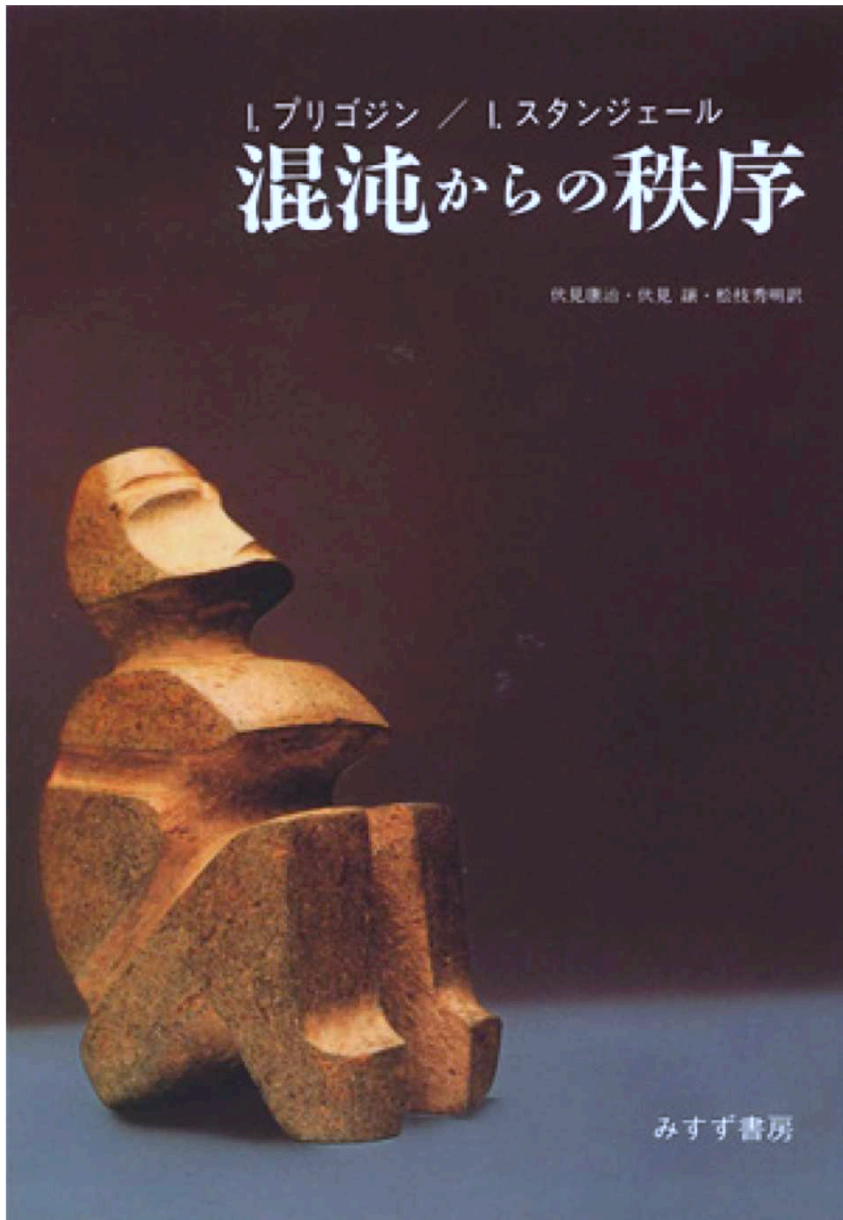
- 教師なし学習 :  $p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$



$$= \sum_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$$

- 教師なし学習は、教師あり学習で既知の“ラベル” を推定すべき潜在変数とおいたもの
- ちなみに、 $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$  は密度推定なので教師なし学習に属するが、実際はハイブリッド
  - $p(\mathbf{y}|\mathbf{x})$ は回帰モデル、 $p(\mathbf{x})$ は密度推定
  - 「教師」の定義の問題・・・人手で作るのではなく、自然な $\mathbf{y}$ を教師データにすればよい
    - amazonの星の数、旅行の行き先、動画のコメント、...





## Intermission

---

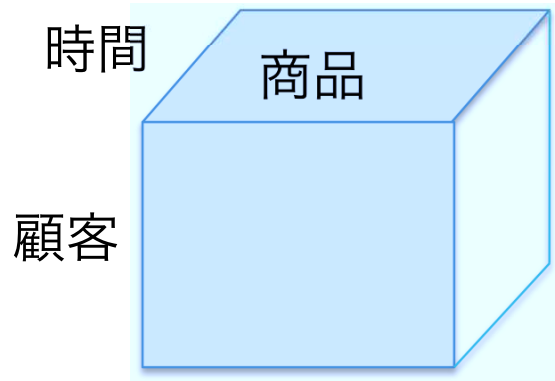
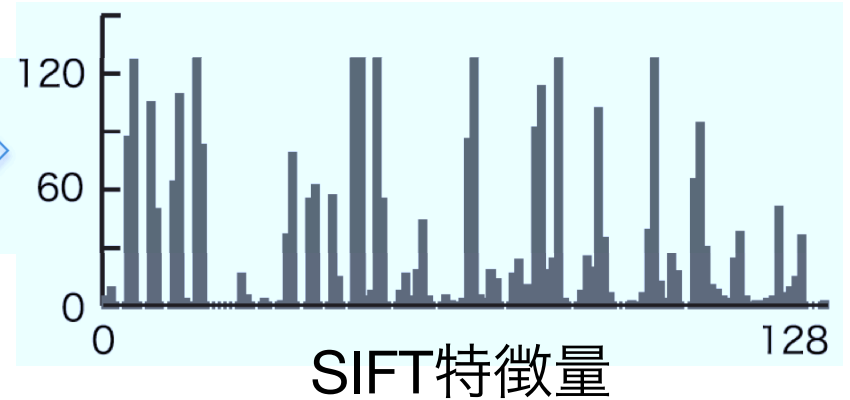
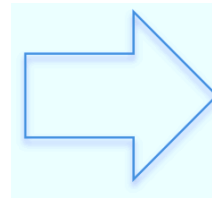
# 簡単なモデルの教師なし学習

# Bag of words: 最も簡単なデータ

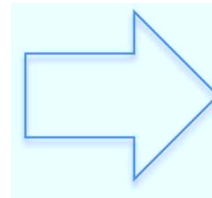
- 多くのデータは、特徴量(素性)とその値の集合で表せる



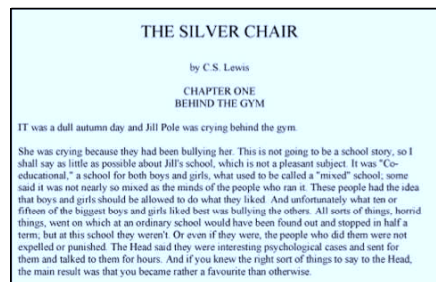
画像/映像  
データ



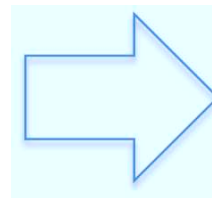
購買データ



紅茶:2, バター:1, CD:4,  
文庫本:3, ……



文書データ



シリア:2, 和平:4, 国際:1,  
条約:1, 締結:1, ……

# ナイーブベイズ法、テキスト分類

- 各文書  $\mathbf{w}_n$  にラベル  $y_n$  がついているデータ  
 $\{\mathbf{w}_1, y_1\}, \{\mathbf{w}_2, y_2\}, \{\mathbf{w}_3, y_3\}, \dots$  があるとする
  - $\mathbf{w}_n = (w_{n1}, w_{n2}, \dots, w_{nT})$  :  $n$ 番目の文書の単語をIDにして並べたもの (例:  $\mathbf{w} = (17 \ 5 \ 3 \ 2 \ 1 \ 8 \ 91 \ 2 \ 34 \ \dots)$ )
- 目的: 新しい文書  $\mathbf{w}$  に対するラベル  $y$  を予測したい

$$p(y = k | \mathbf{w}, \{\mathbf{w}_n, y_n\}_{n=1}^N, \Theta)$$

- 識別器のアプローチ: SVM
- 確率モデルのアプローチ: ナイーブベイズ, ロジスティック回帰

実はロス関数の定義が違うだけ!

# ナイーブベイズ法

---

- ベイズの定理から、

$$p(y|\mathbf{w}) \propto p(\mathbf{w}|y) p(y)$$

- $p(y)$ は教師データの経験分布からほぼ明らかなので、 $p(\mathbf{w}|y)$ だけが問題!

- 単純(ナイーブ)な仮定：クラス $y=k$ の下で、各単語 $w$ の生起は独立で、 $p(w|k)$ に従う

$$\begin{aligned} p(\mathbf{w}|y=k) &= p(w_1|k) \cdot p(w_2|k) \cdot p(w_3|k) \cdots p(w_T|k) \\ &= \prod_{i=1}^T p(w_i|k) \end{aligned}$$

ナイーブベイズ法のパラメータ:  $p(k)$ および $p(w|k)$

## ナイーブベイズ法 (2)

---

- 推定するパラメータ:  $p(k)$ ,  $p(w|k)$ 
  - 学習データでの単純な数をカウントするだけ

$$\begin{cases} p(k) \propto \sum_{n=1}^N \mathbb{I}(y_n = k) \\ p(w|k) \propto p(w, k) \propto \sum_{n=1}^N \sum_{i=1}^{T_n} \mathbb{I}(y_n = k) \mathbb{I}(w_{ni} = w) \end{cases}$$

- 導出: データ全体の尤度

$$p(\mathbf{W}, \mathbf{y}) = p(\mathbf{y})p(\mathbf{W}|\mathbf{y}) = \prod_{n=1}^N p(y_n) \prod_{i=1}^{T_n} p(w_{ni}|y_n)$$

の対数をとって、ラグランジュの未定乗数法

# ナイーブベイズ法 (3)

---

- よって、

$$p(y=k|\mathbf{w}) \propto p(k) \prod_{i=1}^T p(w_i|k)$$
$$\iff \underbrace{\log p(y=k|\mathbf{w})}_{\text{ラベルkの事後スコア}} \propto \underbrace{\log p(k)}_{\text{事前スコア}} + \sum_{i=1}^T \underbrace{\log p(w_i|k)}_{\text{各単語のスコア}}$$

でラベル $y$ の事後確率分布がわかる。

# ナイーブベイズ法 (4)

---

Subject: Powerful growth formula

From: jessica@susie.schubert.de

Keep your loved one contented at night <http://...>

- $p(y=\text{迷惑メール}|w) \propto p(\text{迷惑メール}) \times p(\text{keep}|1) \times p(\text{your}|1) \times p(\text{loved}|1) \times p(\text{one}|1) \times p(\text{contented}|1) \times p(\text{at}|1) \times p(\text{night}|1)$   
 $= 0.1 \times 0.01 \times 0.01 \times 0.1 \times 0.01 \times 0.1 \times 0.01 \times 0.1$   
 $= 1 \times 10^{-12}$
- $p(y=\text{普通メール}|w) \propto p(\text{普通メール}) \times p(\text{keep}|0) \times p(\text{your}|0) \times p(\text{loved}|0) \times p(\text{one}|0) \times p(\text{contented}|0) \times p(\text{at}|0) \times p(\text{night}|0)$   
 $= 0.9 \times 0.01 \times 0.01 \times 0.01 \times 0.01 \times 0.01 \times 0.01 \times 0.01$   
 $= 9 \times 10^{-14}$
- よって、 $p(y=\text{迷惑メール}) = 10^{-12} / (10^{-12} + 9 \times 10^{-14}) = 0.91743$



# ナイーブベイズ法→UM

---

- もしラベル  $y_n$  がなく、 $\mathbf{w}_n$  だけが与えられたら？

- 一般にはこちらの方がよくある状況

→  $y_n$  を潜在変数にすればよい!

- 潜在変数の周辺化

$$p(\mathbf{w}_n) = \sum_{y_n} p(\mathbf{w}_n, y_n) \left( = \sum_{y_n} p(\mathbf{w}_n | y_n) p(y_n) \right)$$

- ナイーブベイズ法の教師なし版・・・ Unigram Mixtures (Nigram+ 2000)

# Unigram Mixtures (UM)

---

- $y_n$  がわからなくても、EMアルゴリズムで学習できる

0. パラメータ  $p(k), p(w|k)$  を適当に設定

1. Eステップ

各文書  $\mathbf{w}_n$  について、

$$p(y_n = k | \mathbf{w}_n) \propto p(k) \prod_{i=1}^T p(w_{ni} | k)$$

を計算

2. Mステップ

$$p(k) \propto \sum_{n=1}^N p(y_n | \mathbf{w}_n)$$

$$p(w | k) \propto \sum_{n=1}^N \sum_{i=1}^T p(y_n = k | \mathbf{w}_n) \mathbb{I}(w_{ni} = w)$$

を更新

3. 収束していなければ1に戻る

ナイーブベイズの学習を繰り返し行うだけ！

# EMアルゴリズムの直感的な説明

$$p(\mathbf{x}|\Theta) = \sum_y p(\mathbf{x}, y|\Theta)$$

- $y$  の値が潜在変数でわからないので、
  - Eステップ: 現在のモデルから、各  $\mathbf{x}_i$  のもつ  $y_i$  の確率分布  $p(y_i|\mathbf{x}_i, \Theta)$  を計算
  - Mステップ: 上の確率分布を重みづけに使って、パラメータ  $\Theta$  を最尤推定  
Eステップに戻る
- 基本はこれだけ!



# Unigram Mixturesの学習例

---

- UMの学習ツール: `um-0.1.tar.gz`  
<http://www.ism.ac.jp/~daichi/dist/um/um-0.1.tar.gz>

```
mondrian:~/work/um/src% ./um
```

```
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] [-l 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)
```

```
mondrian:~/work/um/src% ./um -M 10 cran.dat model
```

```
iteration 1/100.. (1397/1397)PPL = 626
```

```
iteration 2/100.. (1397/1397)PPL = 511.64
```

```
iteration 3/100.. (1397/1397)PPL = 482.871
```

```
iteration 4/100.. (1397/1397)PPL = 480.815
```

```
iteration 5/100.. (1397/1397)PPL = 480.53
```

```
iteration 6/100.. (1397/1397)PPL = 480.277
```

```
iteration 7/100.. (1397/1397)PPL = 480.178
```

```
iteration 8/100.. (1397/1397)PPL = 480.123
```

```
iteration 9/100.. (1397/1397)PPL = 480.112
```

```
converged.
```

```
writing model..
```

```
done.
```

# 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,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

# 実際的な問題とベイズ的な解決

---

- 実際にナイーブベイズ/UMを適用すると、問題が発生
  - 単語 $w$ がラベル $k$ の文書で一度も現れなければ、

$$p(w|k) \propto \sum_{n=1}^N \sum_{i=1}^T \mathbb{I}(y_n = k) \mathbb{I}(w_{ni} = w) = 0$$

このとき、 $w$ を含む文書がラベル $k$ から生成される確率は完全に0になってしまう

$$p(\mathbf{w}|k) = p(w_1|k)p(w_2|k) \cdots p(w|k) \cdots p(w_T|k) = 0$$

- 簡単な対策: 小さな値 $\alpha$ を足す

$$p(w|k) \propto \sum_{n=1}^N \sum_{i=1}^T \mathbb{I}(y_n = k) \mathbb{I}(w_{ni} = w) + \alpha$$

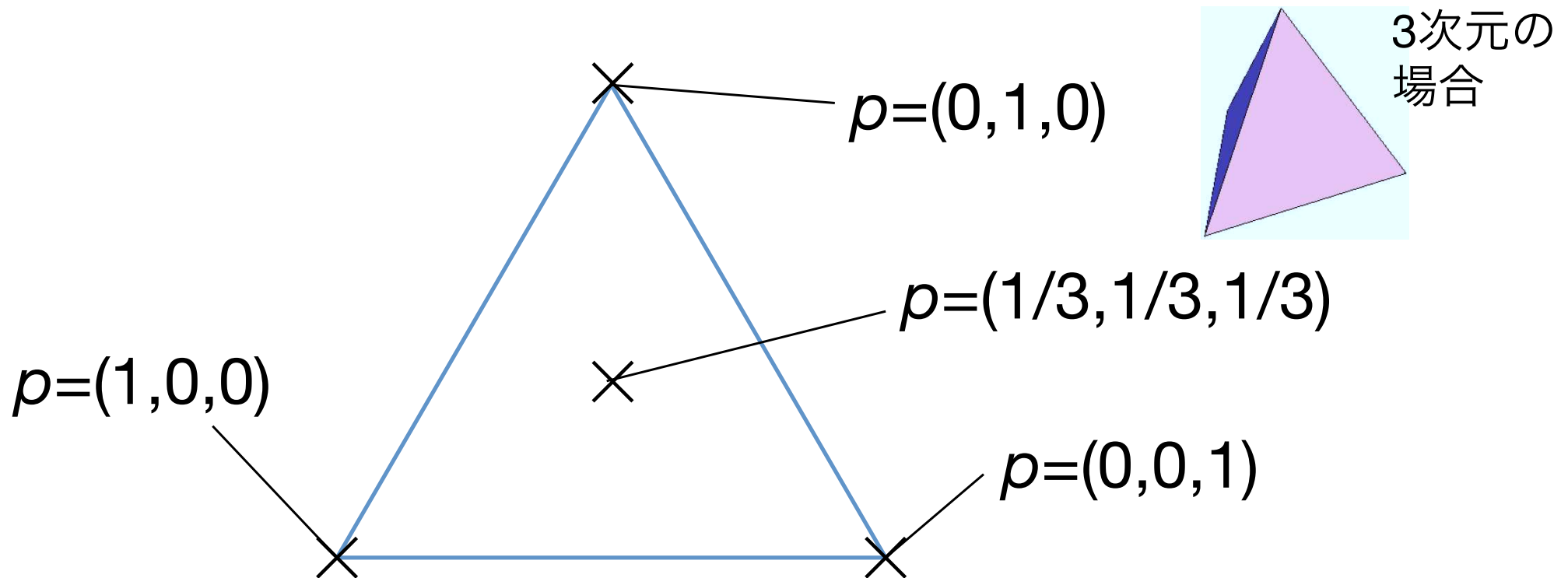
- どういう意味がある?
- どうやって $\alpha$ を求めればよい?

# 多項分布と単体

- K次元の多項分布

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

は、単体(Simplex)とよばれるK-1次元の図形の中に存在

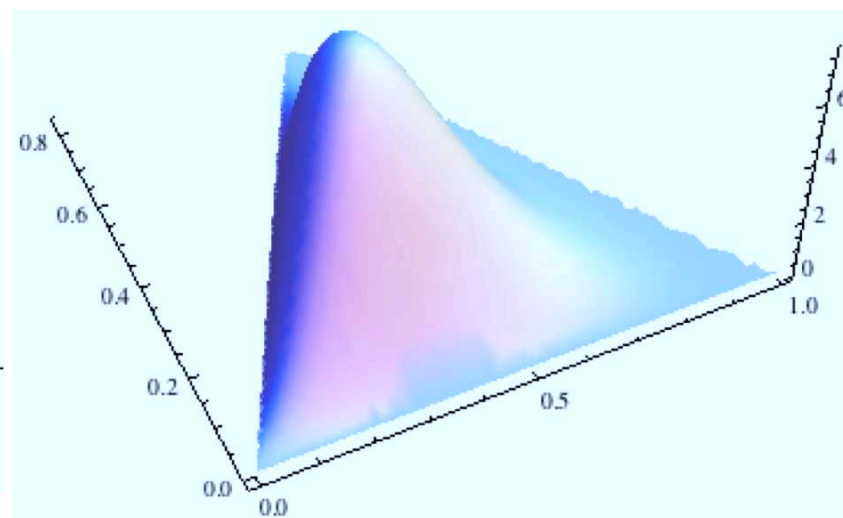




# ディリクレ分布

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

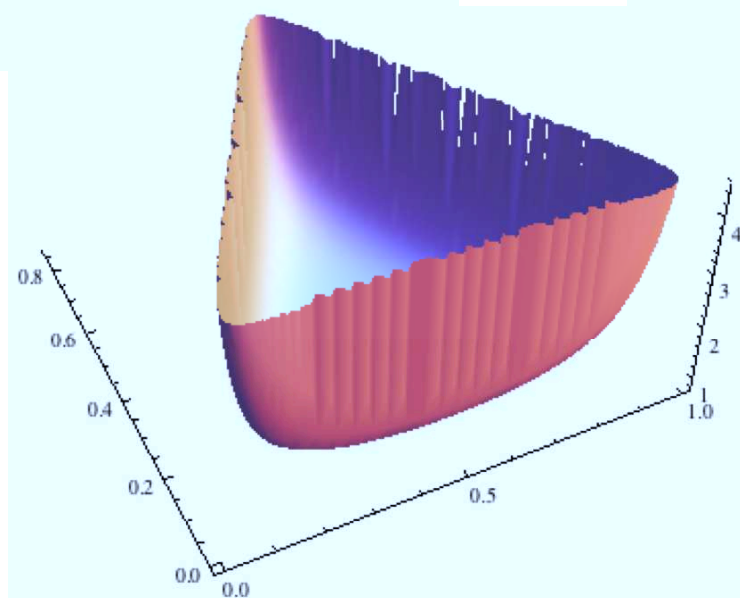
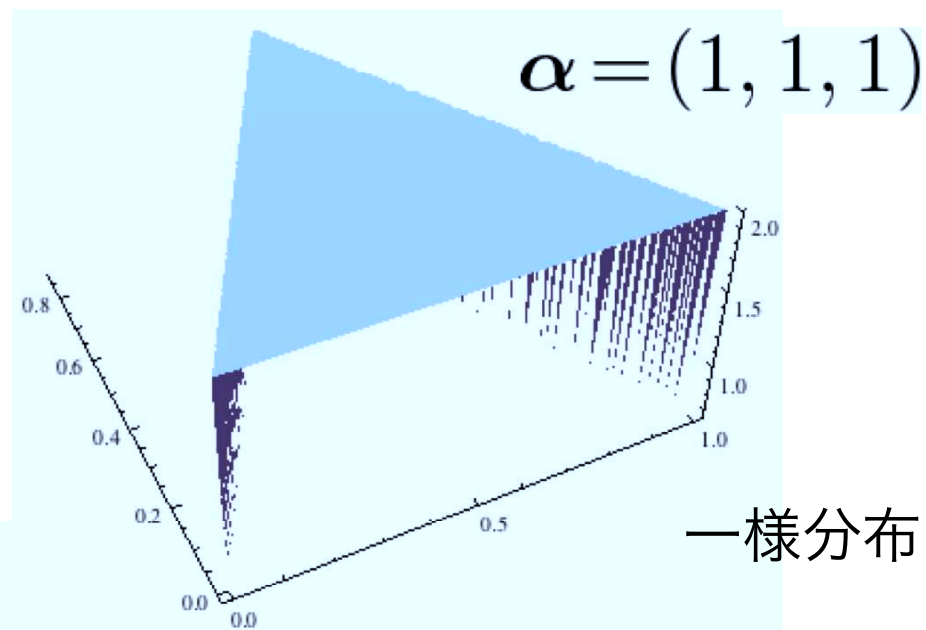
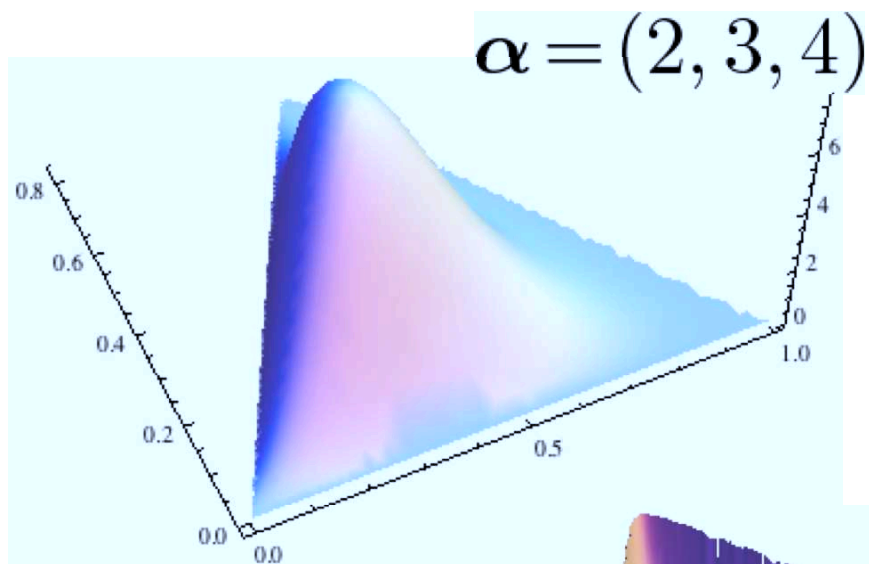
$$\begin{aligned} p(\mathbf{p}|\boldsymbol{\alpha}) &\propto \prod_{k=1}^K p_k^{\alpha_k - 1} \\ &= \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^K p_k^{\alpha_k - 1} \end{aligned}$$



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

# ディリクレ分布 (2)

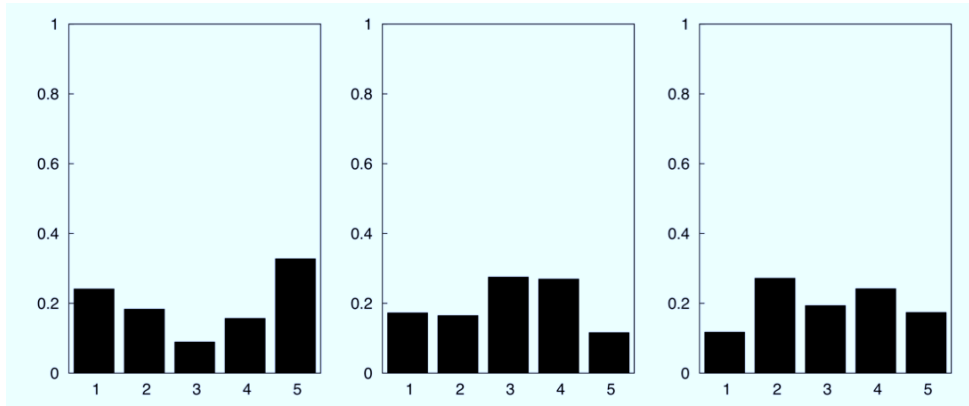
- ディリクレ分布のパラメータ  $\alpha$  と分布の形



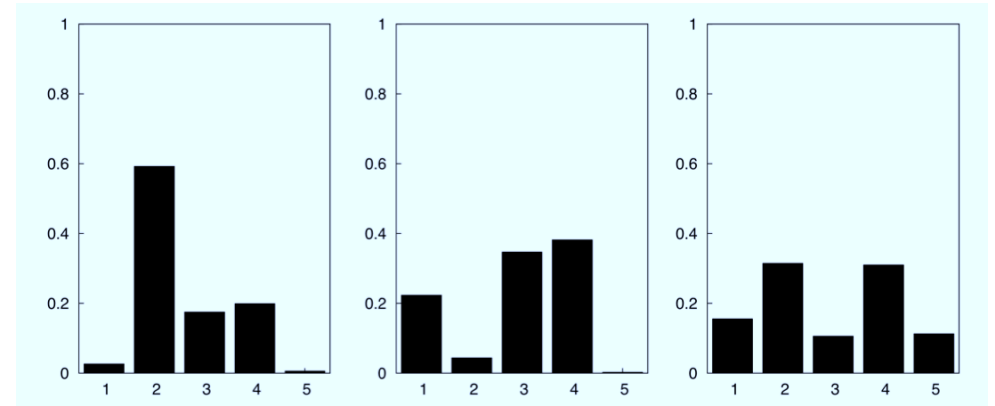
$\alpha = (0.5, 0.5, 0.5)$

# ディリクレ分布 (3)

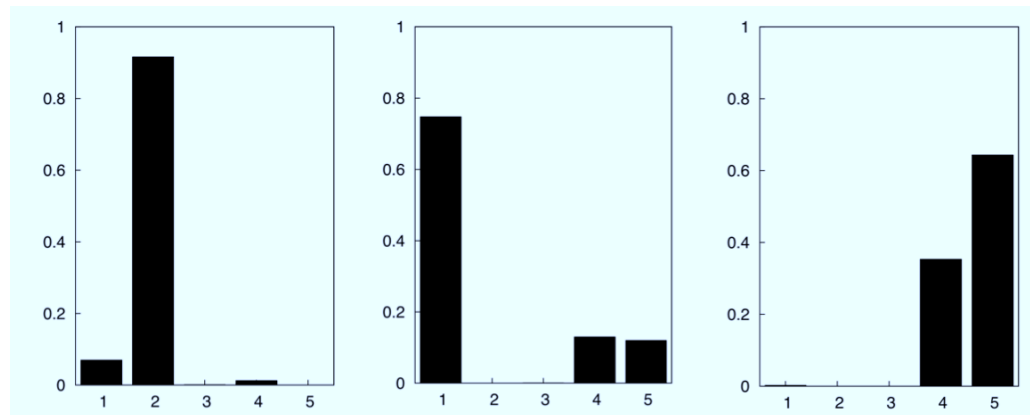
- ディリクレ分布からのサンプル  $\mathbf{p}$



$$\alpha = (10, 10, \dots, 10)$$



$$\alpha = (1, 1, \dots, 1)$$



$$\alpha = (0.1, 0.1, \dots, 0.1)$$

# 最尤推定とベイズ推定

---

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

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

- これは  $\text{Dir}(\mathbf{n} + \boldsymbol{\alpha})$  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 (2)

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  $\alpha_k$  を足して正規化することは、ディリクレ事前分布  $\text{Dir}(\alpha)$  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

# ハイパーパラメータ $\alpha$ の推定

---

- $\alpha$  はどうやって決める?
  - 観測データの尤度  $p(X|\alpha)$  を最大化する  $\alpha$
  - 経験ベイズ法(Empirical Bayes)とよばれる方法

$$\begin{aligned} p(X|\alpha) &= \int p(X|\mathbf{p}) p(\mathbf{p}|\alpha) d\mathbf{p} \\ &= \int \prod_k p_k^{n_k} \cdot \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k p_k^{\alpha_k - 1} d\mathbf{p} \\ &= \frac{\Gamma(\sum_k \alpha_k)}{\Gamma(\sum_k n_k + \alpha_k)} \prod_k \frac{\Gamma(n_k + \alpha_k)}{\Gamma(\alpha_k)} \end{aligned}$$

- これはPolya分布 / DCM分布とよばれる

Dirichlet Compound Multinomial

## ハイパーパラメータ $\alpha$ の推定 (2)

- ナイーブベイズの場合

- あるラベルに属する文書群  $X_1, \dots, X_N$  があるとき、

$$\begin{aligned} p(X_1, \dots, X_N | \alpha) &= \prod_{i=1}^N p(X_i | \alpha) \\ &= \prod_{i=1}^N \frac{\Gamma(\sum_k \alpha_k)}{\Gamma(\sum_k n_{ik} + \alpha_k)} \prod_k \frac{\Gamma(n_{ik} + \alpha_k)}{\Gamma(\alpha_k)} \end{aligned}$$

- これは  $\alpha$  に関して凸なので、Newton法で最適化できる (Minka 2000)

$$\alpha'_k = \alpha_k \cdot \frac{\sum_i \Psi(n_{ik} + \alpha_k) - \Psi(\alpha_k)}{\sum_i \Psi(\sum_k n_{ik} + \alpha_k) - \Psi(\sum_k \alpha_k)}$$

$(\Psi(x) = \frac{d}{dx} \log \Gamma(x))$

## ハイパーパラメータ $\alpha$ の推定 (3)

---

- 注意: この場合  $\mathbf{p}$  を点推定していないので、 $p(X|k)$  は DCM分布になる ( $\alpha$  が  $k$  ごとに存在)

$$p(X|\alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\Gamma(\sum_k n_k + \alpha_k)} \prod_k \frac{\Gamma(n_k + \alpha_k)}{\Gamma(\alpha_k)}$$

- 各単語が独立(ナイーブ)ではない
- 「キャッシュ」効果がある → 同じ単語が再び出やすい
- UMの場合も上の拡張は可能
  - … Dirichlet Mixtures (Sjölander+96, 山本+03,05)
  - 導出はやや複雑
  - 実装: <http://chasen.org/~daiti-m/dist/dm/>



# 単語の時系列データ

---

- 本当は、言語の入力は時系列

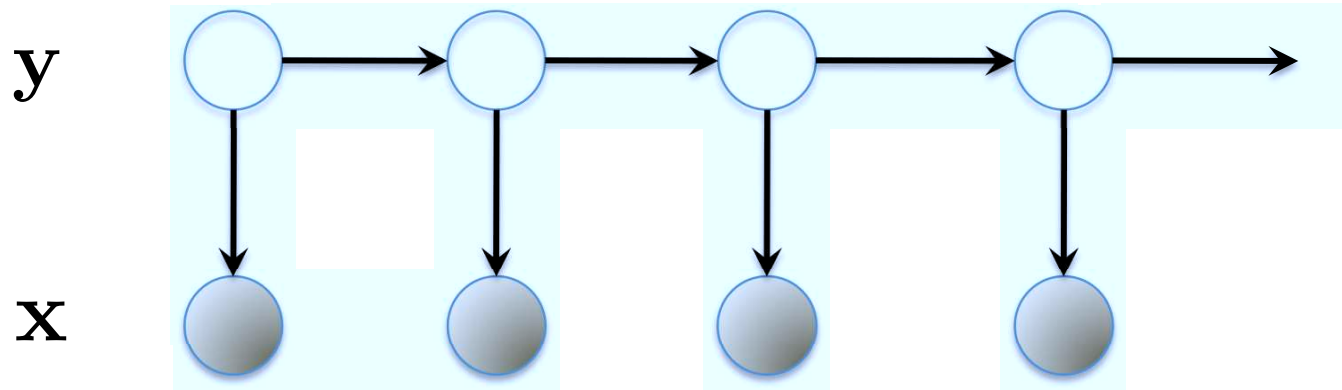
$$\mathbf{x} = (17 \ 5 \ 3 \ 2 \ 108 \ 91 \ 2 \ 34 \ \dots)$$

*When he was a young boy, a book ...*

- これをどのようにモデル化するか？
  - 面白い複雑なモデルは色々考えられるが、
  - 最も簡単な隠れマルコフモデル (HMM) について

# HMMの基礎

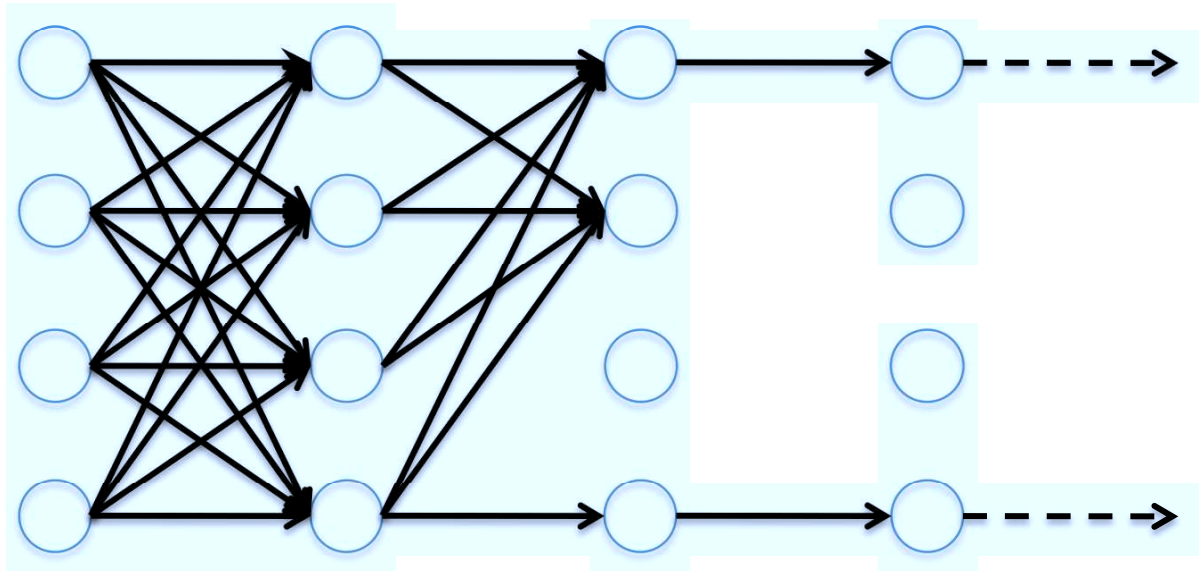
---



- 各時刻 $t$ の観測値  $x_t$  に、隠れ状態  $y_t$  が存在
  - 一般には  $x_t \in \mathbb{R}^d$ ,  $y_t \in \mathbb{R}^K$
  - 自然言語処理での最も簡単な場合は、  
 $x_t = w_t \in \{1 \dots V\}$  : 単語、 $y_t \in \{1 \dots K\}$  : 隠れ状態
- 時系列の確率モデル

$$p(\mathbf{x}, \mathbf{y}) = p(y_0) \prod_{t=1}^T p(x_t | y_t) p(y_t | y_{t-1})$$

# HMMの学習法: 最尤推定



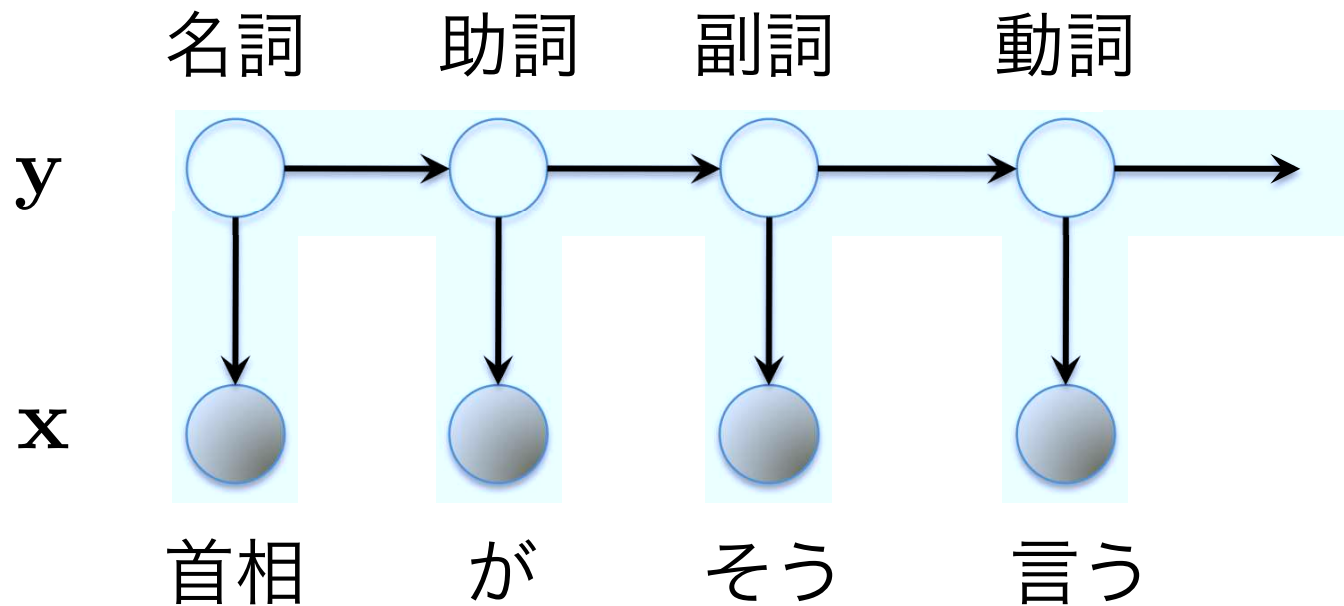
- 可能なパスは指数的( $K^T$ 個)に存在・・・動的計画法

$$\alpha_t(s) = p(y_t = s, x_1 \cdots x_t) \quad (\text{内側確率})$$

$$= \sum_k p(x_t | y_t = s) p(y_t = s | y_{t-1} = k) \alpha_{t-1}(k)$$

- デコード時には、確率最大のパスを1つだけ、動的計画法で求める (Viterbiパス)

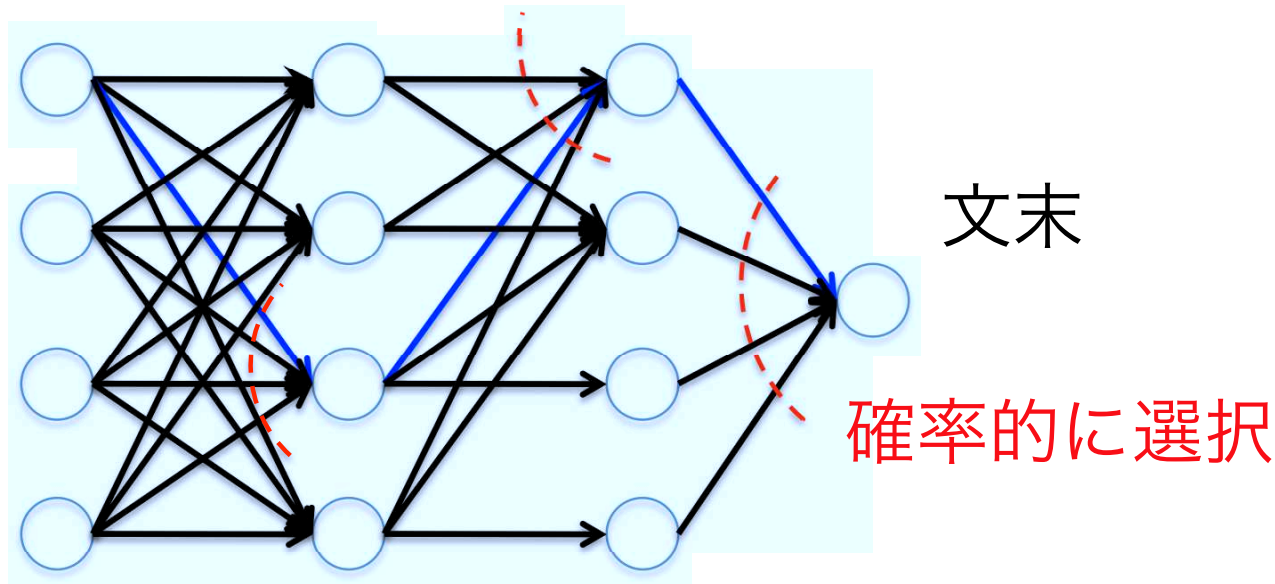
# 自然言語処理でのHMM



- 各単語の持つ「状態」 = 品詞
- chasenはHMMを教師あり学習に使用 (竹内&松本1997)
- 品詞の教師なし学習は? → Merialdo (1994)
  - しかし、あまり高い性能が出なかった
  - EMの局所解の影響が大きかった

# HMMのベイズ学習

- MCMC: 各データの持つ状態系列を実際にサンプリング
- Forward Filtering-Backward Sampling (Scott 2002)



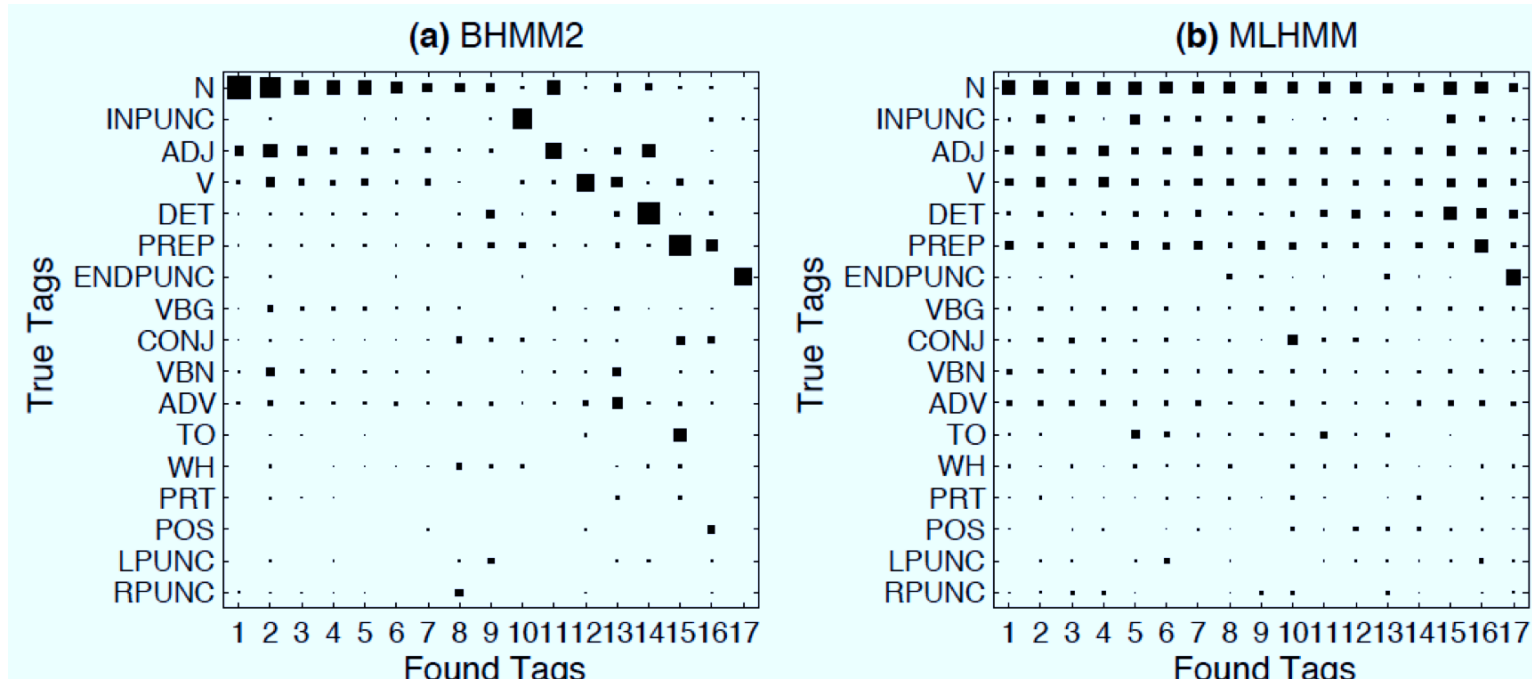
- 内側確率を計算しておいて、文末から確率的に選択 (確率的 Viterbi)

# HMMのベイズ学習 (2)

- Goldwater&Griffiths (2007): 最尤推定に比べて大きな改善を報告

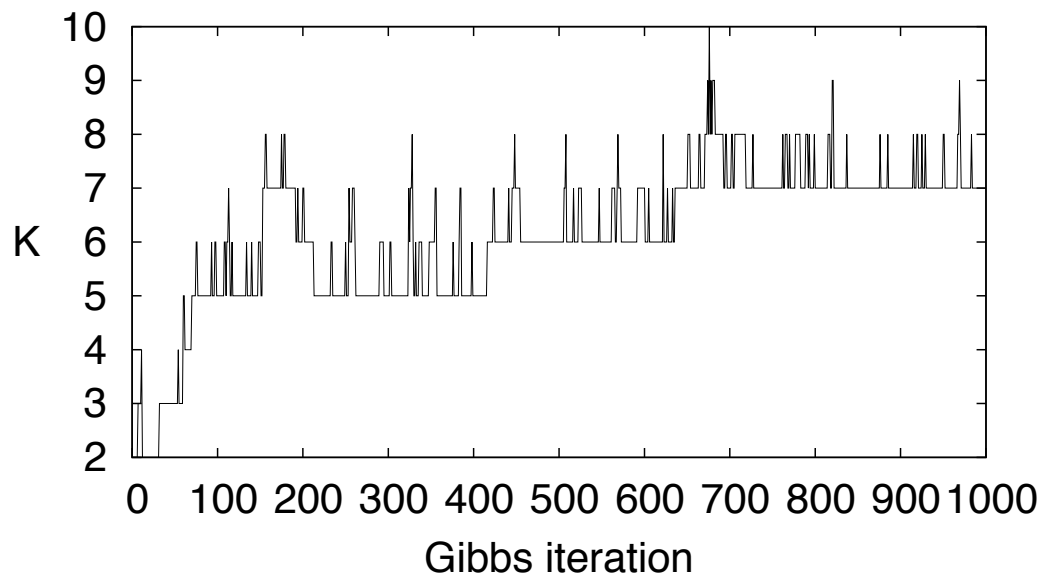
Accuracy	Corpus size			
	12k	24k	48k	96k
random	64.8	64.6	64.6	64.6
MLHMM	71.3	74.5	76.7	78.3
CRF/CE	86.2	88.6	88.4	89.4
BHMM1	85.8	85.2	83.6	85.0
BHMM2	85.8	84.4	85.7	85.8

- 推定された状態遷移行列が最尤推定とは全く異なる



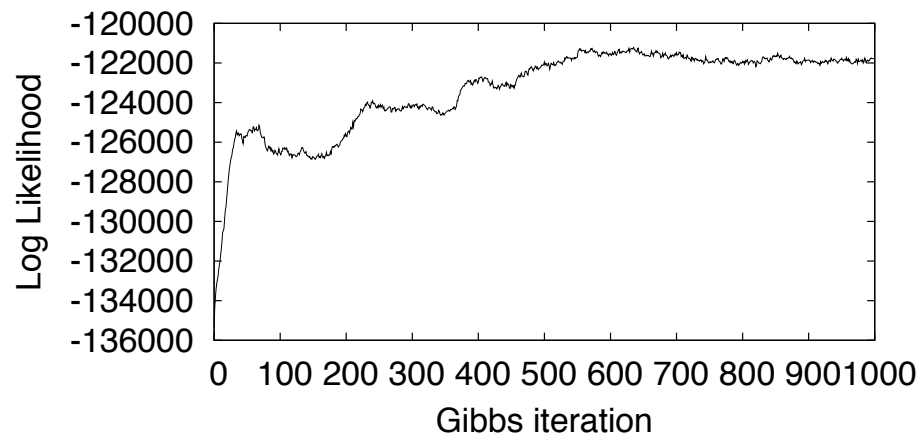
# Infinite HMM

- ノンパラメトリックベイズ法(複雑なため今回は割愛)を使うと、HMMの状態数も同時に推定できる
  - Beal 2001, Teh 2006, van Gael 2008
- 下は、「不思議の国のアリス」を学習テキストにして実際に動かしてみた結果



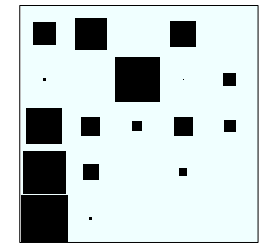
隠れ品詞数の学習

データの対数尤度の変化



# 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
who	37	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!



# Online HMM

---

- 通常のHMMの学習: 繰り返しが必要・・計算量が大きい

- EMの場合

- For (収束するまで) {

- Eステップ:

- for (n=0; n<N; n++) {

- Forward-Backwardで  $p(y_n | \mathbf{x}_n, \Theta)$  を計算

- }

- Mステップ:

- 上の  $p(y_n | \mathbf{x}_n, \Theta)$  から、パラメータ  $\Theta$  を更新

- }

- データを全部見た後でしか  $\Theta$  を更新しない!

## Online HMM (2)

---

- Online EM (Cappe&Moulines 09;Liang&Klein 09)の適用
- SGDで充分統計量を更新: HMMの場合、
  - 状態遷移  $j \rightarrow k$  の回数  $c(j,k)$
  - $k$ から単語 $w$ を生成した回数  $n(k,w)$  がわかればよい

For  $t = 1 \dots T$ , {

For  $n = \text{randperm}(1 \dots N)$  {

Forward-Backwardで  $p(y_n | \mathbf{x}_n, \Theta)$  を計算  
文 $n$ 内での  $c(j,k)$ ,  $n(k,w)$  の期待値  $s_n$  を求める

$$\mu = (1 - \eta_k) \mu + \eta_k s_n \quad \eta_k : \text{学習率}$$

$$k = k + 1$$

}  
}

パラメータ更新の回数が多い!  
→ 収束が速い、局所解回避

# Online HMM (3)

- PFIの岡野原氏による ohmm-0.02 が公開されている



The screenshot shows a web browser window with the title "Ohmm: Online Training for Hidden Markov Model". The address bar contains the URL "http://web.archive.org/web/20090529142908/http://www-tsujii.is.s.u-tc". The page content includes a main heading "ohmm: Online training for Hidden Markov Model", a link for "English", a "概要" (Overview) section, a "ダウンロード" (Download) section, and a "更新情報" (Update Information) section.

**ohmm: Online training for Hidden Markov Model**

[English](#)

## 概要

ohmmは隠れマルコフモデルにおいて、Online EMアルゴリズム[1]を用いて学習するためのライブラリです。大規模なデータを利用した学習に対応しており数十万語規模の学習データを利用した学習を行うことができます。また学習結果を他用途で利用できるような形で出力することができます。

## ダウンロード

ohmmはフリーソフトウェアです。BSD ライセンスに従って本ソフトウェアを使用、再配布することができます。

- ohmm-0.02.tar.gz: [HTTP](#)

## 更新情報

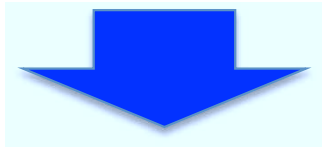
---

# 複雑なモデルの教師なし学習

# Bag of wordsふたたび

---

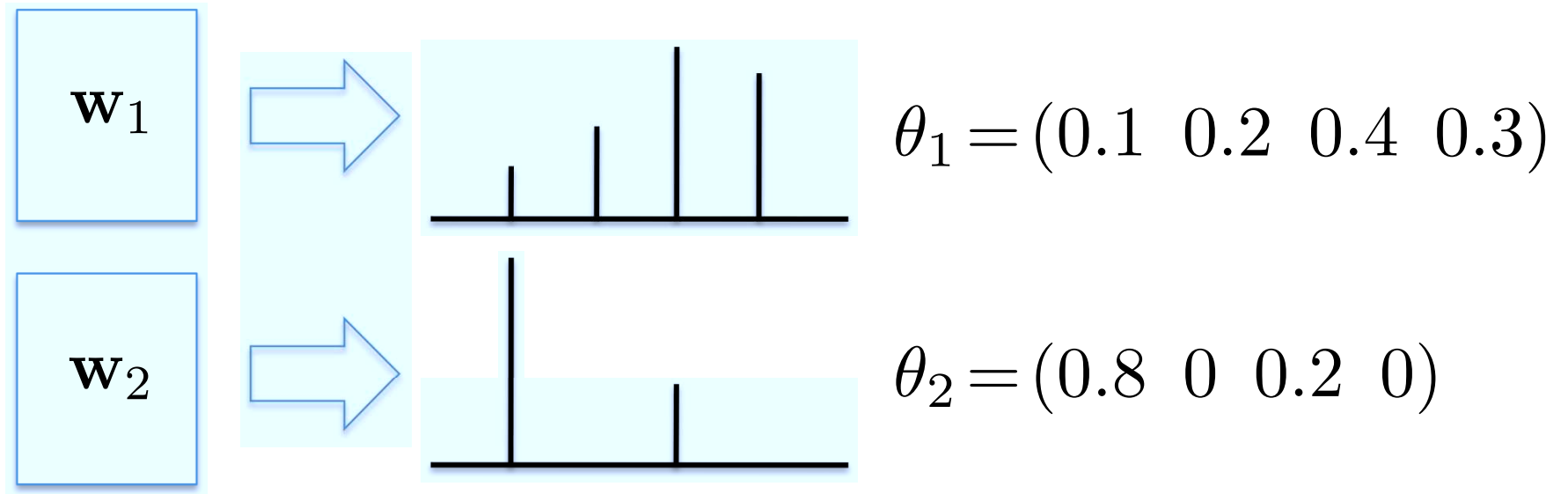
- NB/UMでは、 $p(y_n | \mathbf{w}_n, \Theta)$ を計算した
  - データ点  $\mathbf{w}_n$  をラベル  $y_n$  で表現



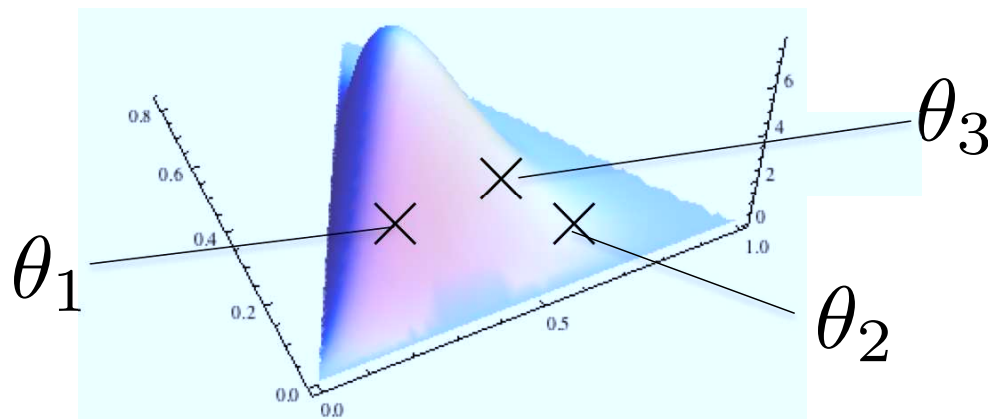
- 実際には、 $\mathbf{w}_n$  の内容はそう一言では言えない例)
  - Amazonのレビュー文
    - よい評価の箇所
    - 悪い評価の箇所
  - 新聞記事
    - “科学分野の予算”の記事
    - “伝統芸能の国際化”の記事
    - ..

# トピックモデル: LDA (Blei+ 01,03)

- 解決(の1つ): 文書  $w$  を話題(トピック)の混合で表現する



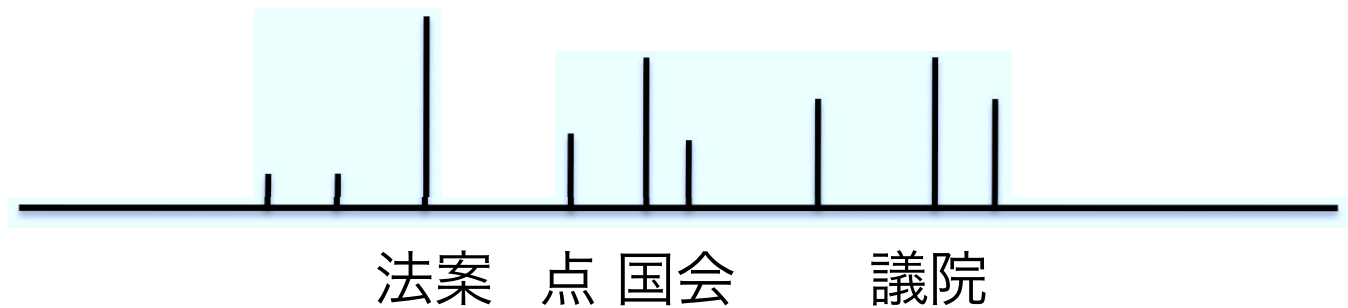
- 混合比  $\theta$  をディリクレ事前分布から生成



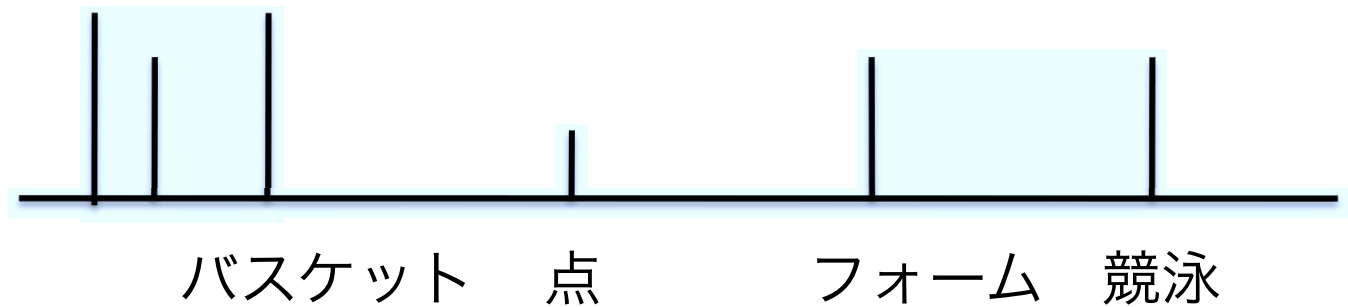
# トピックモデル (2)

- 「話題」とは? → 単語の生起確率分布  $\beta_k = \{ p(w|k) \}$   
( $w = 1 \dots V$ )

$\beta_1$  「政治」

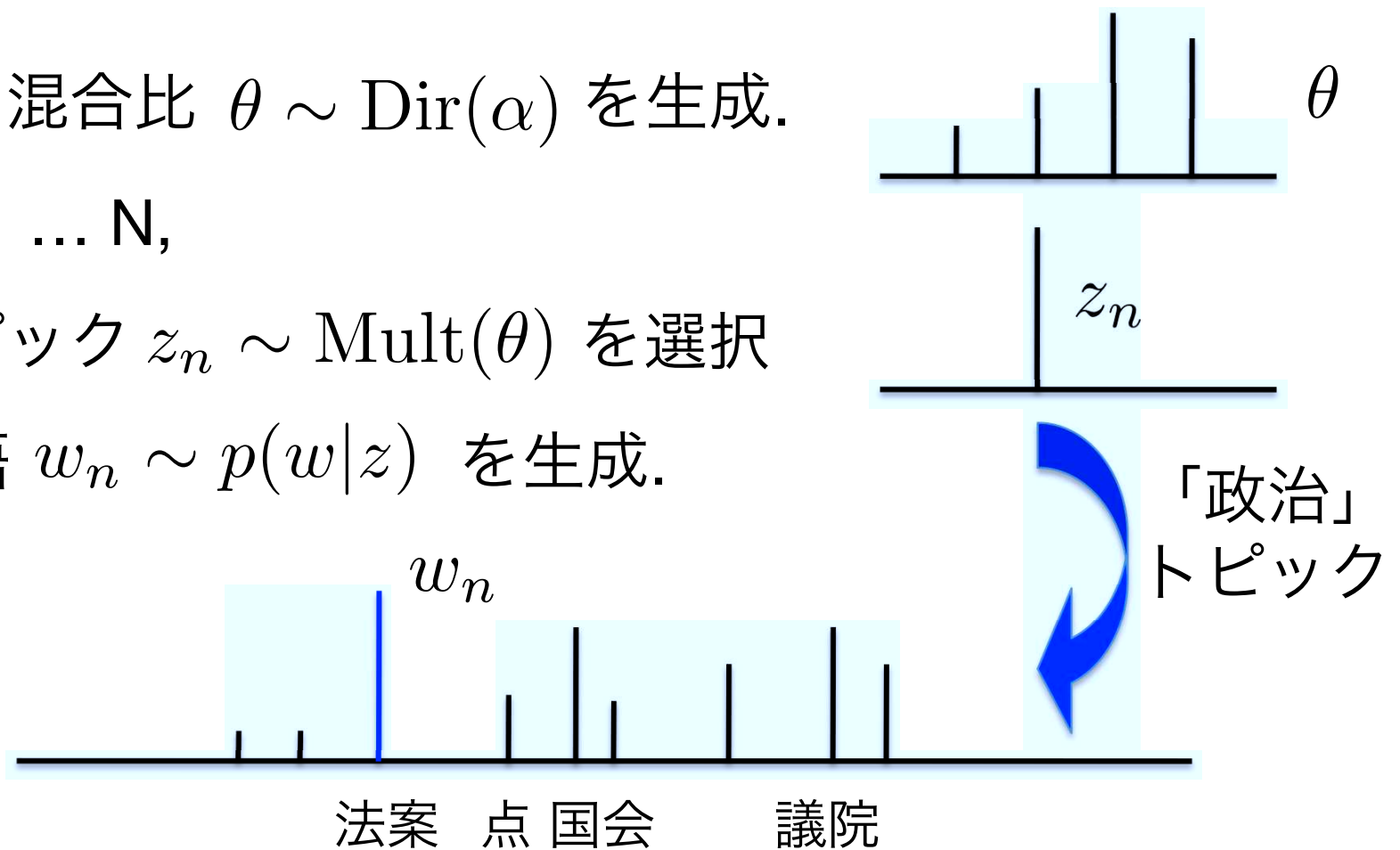


$\beta_2$  「スポーツ」



# LDAの文書生成モデル

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





# LDAの学習例

---

- 川端康成「雪国」の冒頭

国境の長いトンネルを抜けると雪国であった。  
夜の底が白くなった。信号所に汽車が止まった。  
向側の座席から娘が立って来て、島村の前のガラス  
窓を落した。雪の冷気が流れこんだ。...

- 2000年度毎日新聞記事全文 (2,887万語) で学習したモデルで分析

- 青色のトピックは冬に関する
- 緑色のトピックは電車に関する
- 黒色は地の文

# LDAの確率モデル

---

- 式で書くと、 $\mathbf{w} = (w_1, w_2, \dots, w_T)$  について

$$\begin{aligned} p(\mathbf{w}|\alpha, \beta) &= \int \sum_{\mathbf{z}} p(\mathbf{w}, \mathbf{z}, \theta) d\theta \\ &= \int \sum_{\mathbf{z}} p(\mathbf{w}|\mathbf{z}) p(\mathbf{z}|\theta) d\theta \\ &= \int \prod_n \sum_k p(w_n|k) \theta_k \cdot \text{Dir}(\theta|\alpha) d\theta \\ &= \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \int \left( \prod_k \theta_k^{\alpha_k - 1} \right) \prod_n \sum_k \theta_k p(w_n|k) d\theta \end{aligned}$$

- 推定すべきパラメータは $\alpha$ と $\beta = \{p(w|k)\}$ 
  - パラメータの数はナイーブベイズと同じ

# LDAの学習

---

- これを解く方法は色々あるが、標準的なVB-EMアルゴリズムでは (導出略):

- VB-E step:

$$p(z=k|w_{ni}) \propto p(w_{ni}|k) \exp\left(\Psi\left(\alpha_k + \sum_i p(z=k|w_{ni})\right)\right)$$

- VB-M step:

$$p(w|k) \propto p(k|w)p(w) \propto \sum_n \sum_i p(z=k|w_{ni})$$

- 全体の学習アルゴリズム
  - 各文書 $n$ の各単語 $i$ について、 $p(z=k|w_{ni})$  を計算
  - その結果から、 $\beta = p(w|k)$ と $\alpha$ を更新
  - 以上を繰り返す.

## LDAの学習 (2)

---

- 全体の学習アルゴリズム:

VB-E step:

For  $n = 1 \dots N$  {

For  $i = 1 \dots T$  {

$$p(z = k | w_{ni}) \propto p(w_{ni} | k) \exp\left(\Psi\left(\alpha_k + \sum_i p(z = k | w_{ni})\right)\right)$$

}

}

を計算

VB-M step:

$\alpha$  を更新;

$$\beta = p(w | k) \propto p(k | w) p(w) \propto \sum_n \sum_i p(z = k | w_{ni}) \text{ を更新}$$

- 実は途中で  $\alpha, \beta$  を更新できるのでは? → オンライン化

# Online LDA (Sato+ 2010)

---

- 1文書を見るごとに、 $\alpha, \beta$ を更新

While (収束するまで) {

For  $n = 1 \dots N$  {

For  $i = 1 \dots T$  {

$$p(z = k | w_{ni}) \propto p(w_{ni} | k) \exp\left(\Psi\left(\alpha_k + \sum_i p(z = k | w_{ni})\right)\right)$$

}

を計算

$\alpha$ を更新

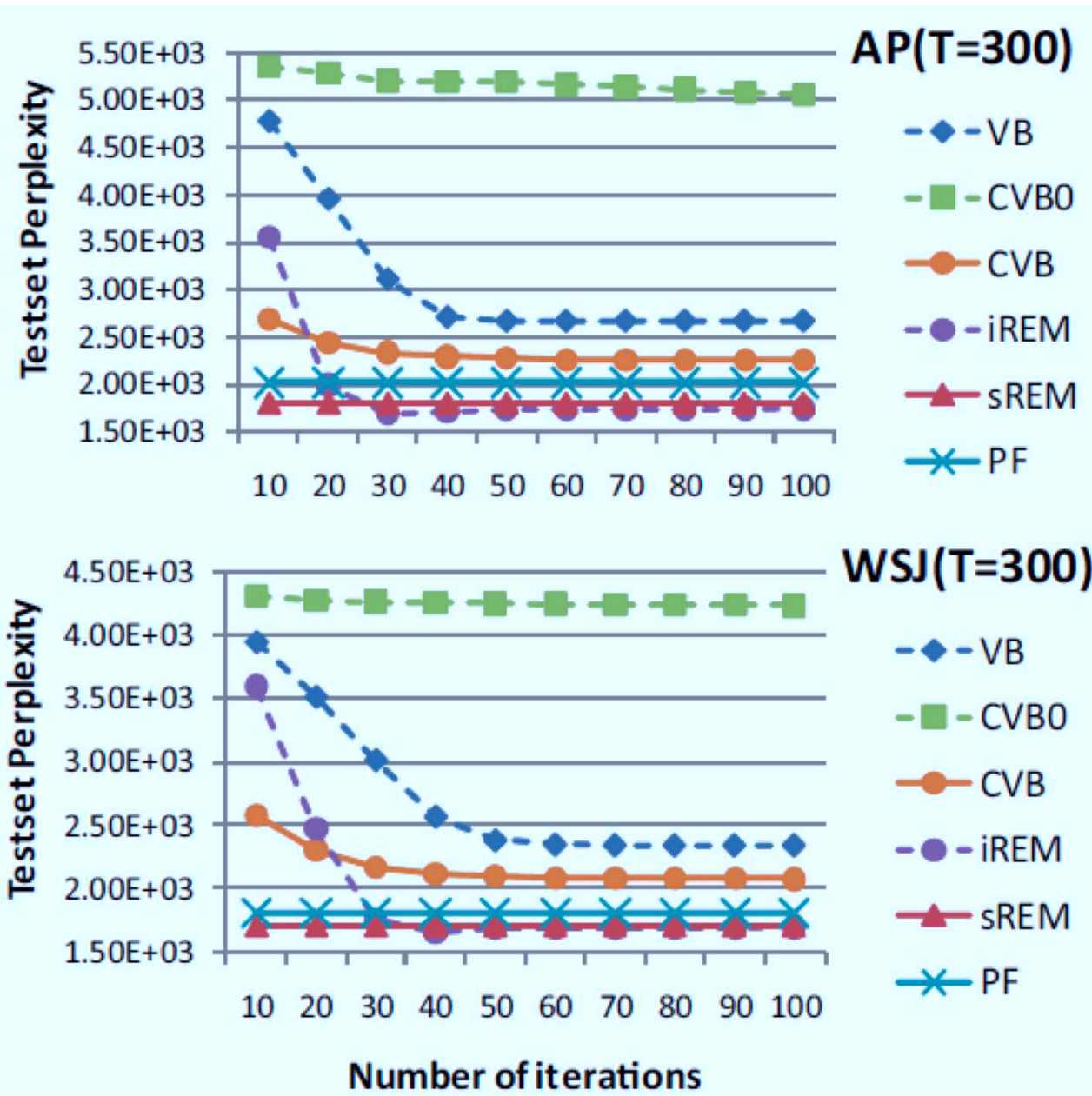
$$\beta = p(w | k) \propto p(k | w) p(w) \propto \sum_n \sum_i p(z = k | w_{ni}) \text{ を更新}$$

}

}

- 一番外側のループはなくてもよい → オンライン学習
  - 1文書を見て学習…データを捨ててしまってもよい

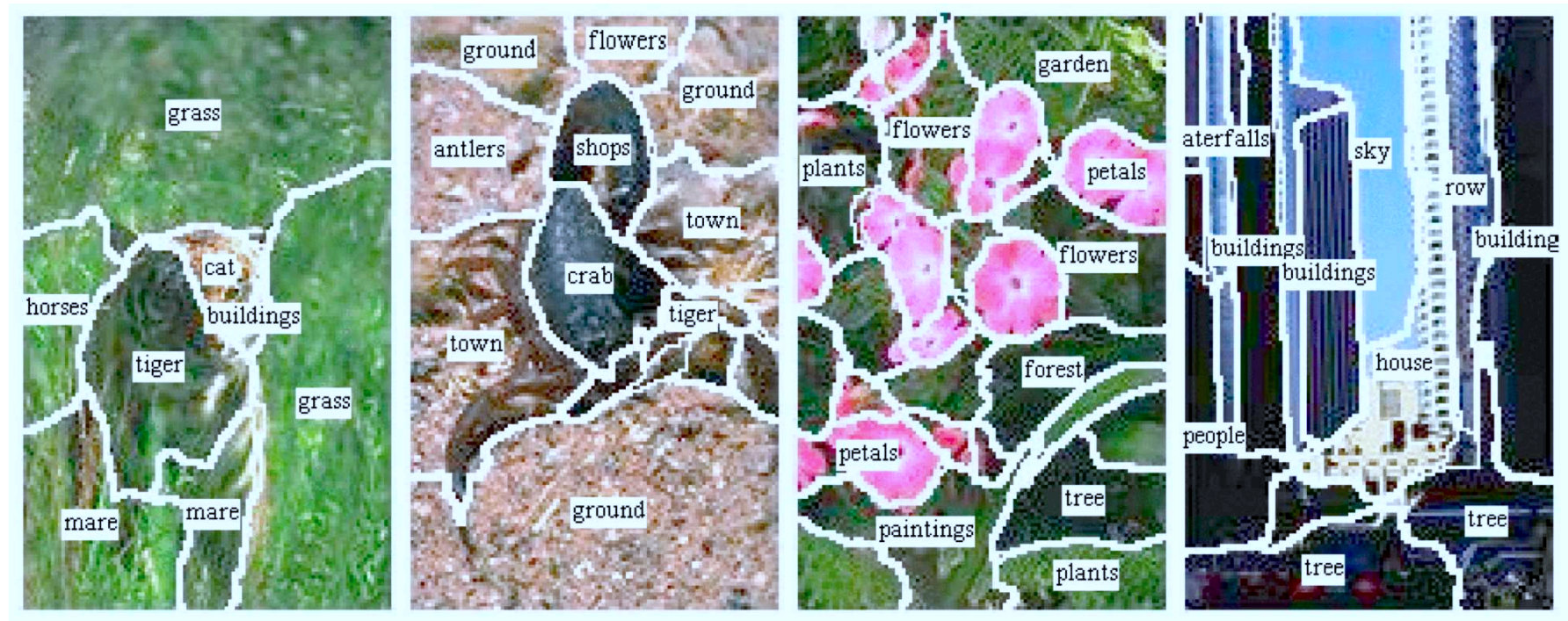
# Online LDA (Sato+ 2010): 実験結果



- 紫の▲(sREM)が Online LDA
- AP: Associated Press コーパス
- WSJ: Wall Street Journal コーパス

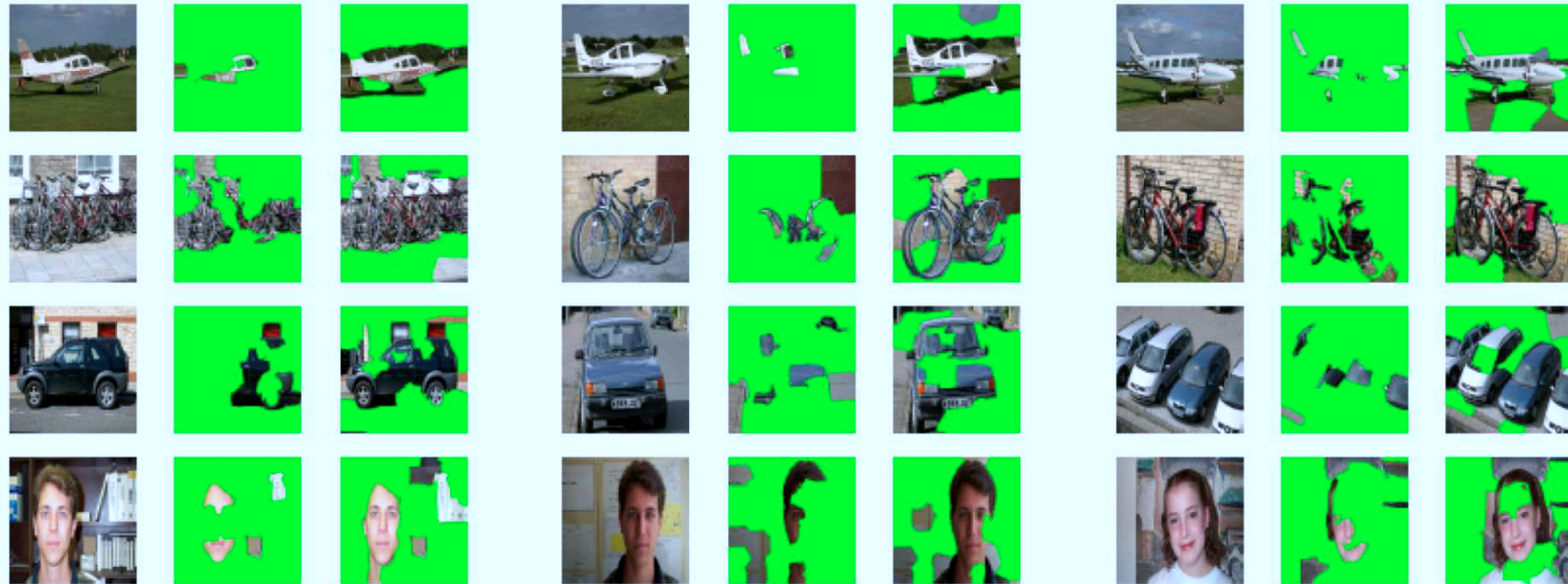
# 画像処理への応用

- 古典的な適用: “Matching words and Pictures”  
(K.Barnard, ICCV 2001/JMLR 2003)



# 比較的最近の画像への適用

- Topic Random Field (Fei-Fei Li+, ECCV 2010)








**Fig. 5.** (Best viewed in color). Segmentation results of the MSRC database. From left to right: original image, segmentation result of spatial LDA and TRF.

$$p(\mathbf{z}^d | \boldsymbol{\theta}^d, \sigma) = \frac{1}{A(\boldsymbol{\theta}^d, \sigma)} \exp \left[ \sum_n \sum_k z_{nk}^d \log \theta_k^d + \sum_{n \sim m} \sigma I(z_n^d = z_m^d) \right] \quad (1)$$



# Geographic topic model (Eisenstein+ 2010)

	“basketball”	“popular music”	“daily life”	“emoticons”	“chit chat”
	PISTONS KOBE LAKERS game DUKE NBA CAVS STUCKEY JETS KNICKS	album music beats artist video #LAKERS ITUNES tour produced vol	tonight shop weekend getting going chilling ready discount waiting iam	:) haha :d :( ;) :p xd :/ hahaha hahah	lol smh jk yea wyd coo ima wassup somethin jp
Boston 	CELTICS victory BOSTON CHARLOTTE	playing daughter PEARL alive war comp	BOSTON	:p gna loveee	<i>ese</i> exam suttin sippin
N. California 	THUNDER KINGS GIANTS pimp trees clap	SIMON dl mountain seee	6am OAKLAND	<i>pues</i> hella koo SAN fckn	hella flirt hut iono OAKLAND
New York 	NETS KNICKS	BRONX	iam cab	oww	wasssup nm
Los Angeles 	#KOBE #LAKERS AUSTIN	#LAKERS load HOLLYWOOD imm MICKEY TUPAC	omw tacos hr HOLLYWOOD	af <i>papi</i> raining th bomb coo HOLLYWOOD	wyd coo af <i>nada</i> tacos messin fasho bomb
Lake Erie 	CAVS CLEVELAND OHIO BUCKS od COLUMBUS	premiere prod joint TORONTO onto designer CANADA village burr	stink CHIPOTLE tipsy	:d blvd BIEBER hve OHIO	foul WIZ salty excuses lames officer lastnight

# LDAの拡張

---

- 他にも無数にある(現在も発展)が、中でも識別学習との結合モデルを以下で紹介
  - Titov&McDonald (2008): “A Joint Model of Text and Aspect Ratings for Sentiment Summarization”

# Titov&Mcdonald (2008)

- 背景: レビューサイトでのレビュー文には、評価ポイントごとの点がついていることが多い

価格.com - SONY サイバースhoot DSC-QX100 レビュー評価・評判

http://review.kakaku.com/review/K0000575947/#tab

さん  
累計支持数: 0人 | ファン数: 0人

2013年10月27日 17:18 [643874-1]

デザイン	★★★★☆ 3	スマホの画質に物足りなくて
画質	★★★★★ 5	
操作性	★★★★☆ 3	
バッテリー	★★★★☆ 2	
携帯性	★★★★☆ 2	
機能性	★★★★☆ 4	
液晶	無評価	
ホールド感	★★★★☆ 3	
満足度	★★★★★ 4	

露出優先でややアンダー気味にして撮影しました。

一昨日到着、早速iPhone5に取り付けて使ってみました。近くのお城の庭園で菊の花や風景・ゆるキャラなんか撮影してみました。スマホの画質がどうも物足りなくて、といってわざわざ重たい一眼レフを持って行くのが嫌なので、スマホにいたるときだけ取りつけて、そこそこの写真が得られる使い方にはびったりです。その場できれいな画像をメールで送れるのもいいですね。Aモードで撮影するのが好きなのでスマホの大きな画面で露出を確認出来て

アスペクト

- 問題: どの評価ポイント(アスペクト)がレビュー文のどこに対応しているかわからない!
  - しかし、統計的には相関があるのでわかるはず

# MAS (Multi-Aspect Sentiment model)

- 解決: アスペクト ∈ トピックとみて、アスペクトに割り当てられた語を使った回帰モデル
  - トピックモデル+ロジスティック回帰

This hotel has a good location and great service. Lunch is also great, especially with a café style desserts. We can reach any spots from this hotel by walk or a light rails. The most prominent feature of this hotel is its silence; it is a bit far away from the downtown. However, during our stay, we could enjoy fabulous restaurants located in this hotel. ...

Logistic  
Regression

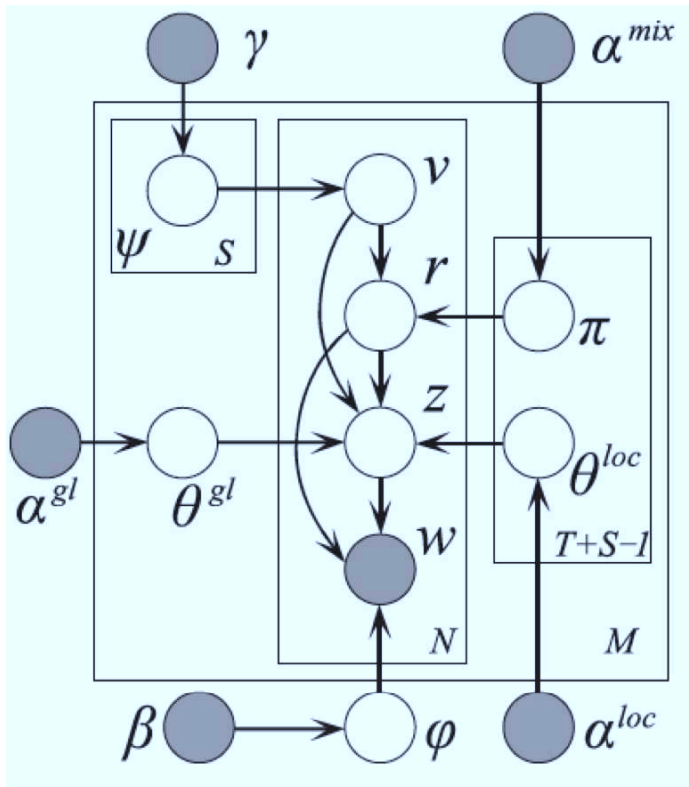


**Food:**

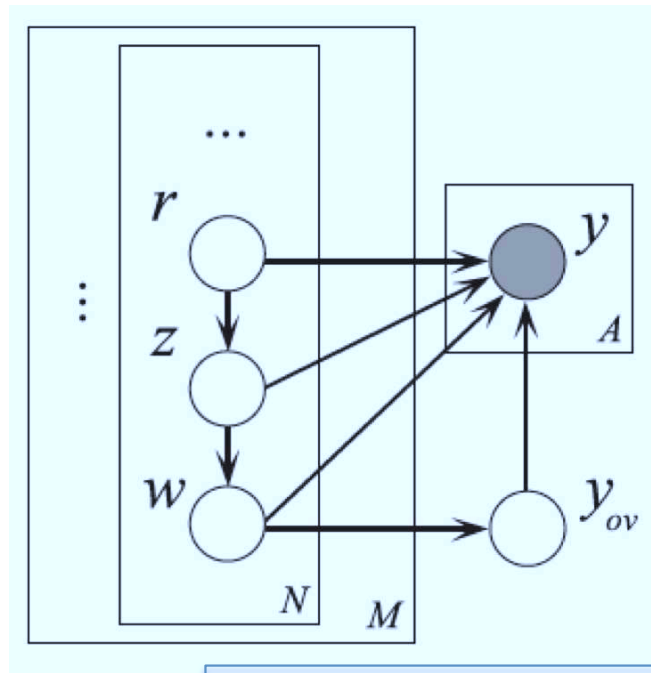


# MAS (2)

全体像



回帰モデル部



nグラムfが評価yを生む重み

$$p(y^{(a)} = y | \mathbf{w}, \mathbf{r}, \mathbf{z}) \propto \exp\left(b_y^{(a)} + \sum_f \lambda_{f,y} + p(a|f) \lambda_{f,y}^{(a)}\right)$$

- トピックをサンプルする際にも、この重みを用いる (同時学習)

---

# 自然言語処理の先端での教師なし学習 & 関連する統計モデル

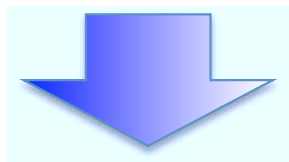
# 混合モデル(Mixture model)の復習

- 混合モデル: データがある1つの分布から生成

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



- ナイーブベイズ、Unigram Mixtures:  
文書全体が  $p(w|z)$  から生成



- LDA: 各単語ごとにトピック  $z$  があり、 $p(w|z)$  から生成

# 混合モデルには限界がある

---

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

- 現実のデータ: さまざまな制約が満たされて生成されている
  - 自然言語の場合: トピック以外に、
    - 文法的な制約 [主語は1つ, 係り結びが完結, ...]
    - 時制の一致
    - 文体が適正か [ですます / である, 女言葉, ...]
  - 購買データの場合: 中身以外に、
    - デザインの各個人の嗜好
    - 広告効果、メーカー信頼度 [Sonyファンなど]
    - 緊急性 ...

これを混合モデルで扱うのは困難!

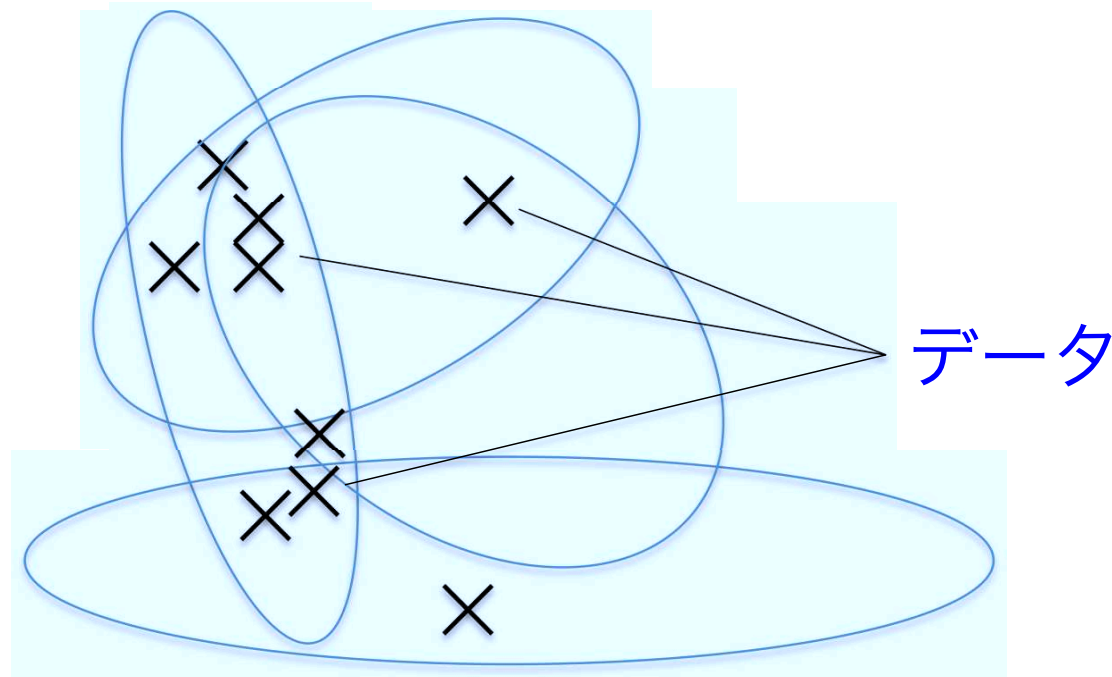


# 積モデル (Product Model)

- 制約を確率(でなくてもよい)の積で表現 (Hinton 2002)

$$p(\mathbf{w}|\theta) = \frac{\prod_k p(\mathbf{x}|\theta_k)}{Z}, \quad Z = \sum_{\mathbf{w}} \prod_k p(\mathbf{x}|\theta_k)$$

- データは、すべての制約  $p(\mathbf{x}|\theta_k)$  を満たされて生成



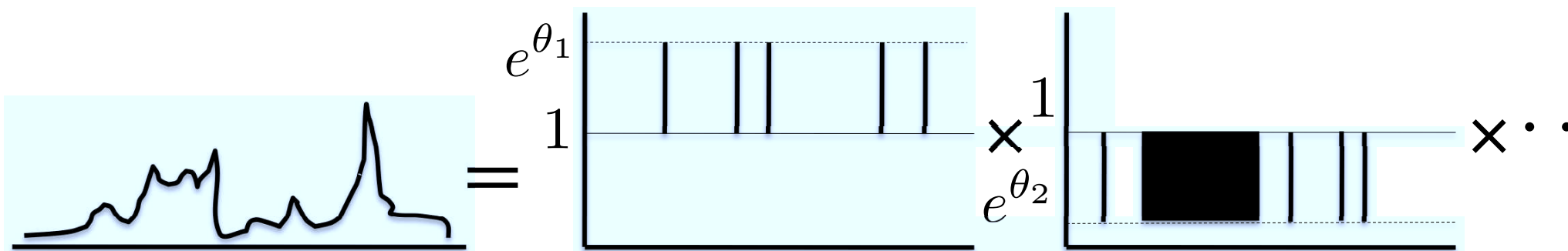
# Log-Linearモデル / 最大エントロピー法

- 対数線形モデルは、Product Modelの一種

$$p(\mathbf{w}|\theta) = \frac{\exp(\sum_k \theta_k f_k(\mathbf{w}))}{Z} = \frac{\prod_k e^{\theta_k f_k(\mathbf{x})}}{Z}$$

$$\begin{aligned} p(\mathbf{w}|\theta_k) &= e^{\theta_k f_k(\mathbf{x})} \\ &= \begin{cases} e^{\theta_k} & \text{if } f_k(\mathbf{x}) = 1 \\ 1 & \text{if } f_k(\mathbf{x}) = 0 \end{cases} \end{aligned}$$

とおけば、  
これは  
Product Model



# Product Modelの学習

---

$$p(\mathbf{w}|\theta) = \frac{\prod_k p(\mathbf{w}|\theta_k)}{Z}$$

- 分配関数  $Z = \sum_{\mathbf{w}} \prod_k p(\mathbf{w}|\theta_k)$  が容易には求まらない!
  - Zは「可能な文すべてについての膨大な和」
  - 10,000単語種×20単語= $(10^4)^{20}=10^{80}$  !! [全宇宙の電子の総数]
  - CRFなどは、Markov性でZが計算できる特別な場合

## Product Model の学習 (2)

---

- 一般に、
$$p(\mathbf{w}|\theta) = \frac{f(\mathbf{w}|\theta)}{Z}, \quad Z = \sum_{\mathbf{w}} f(\mathbf{w}|\theta)$$
を考える.

- モデル $p$ のもとでの $\mathbf{w}$ の平均的な対数尤度 (確率) を最大化したい

$$\begin{aligned} L &= \left\langle \log p(\mathbf{w}|\theta) \right\rangle_{\hat{p}(\mathbf{w})} \\ &= \sum_{i=1}^N \hat{p}(\mathbf{w}_i) \log p(\mathbf{w}_i|\theta) \quad \rightarrow \text{最大化} \end{aligned}$$

# Product Model の学習 (3)

- 勾配法で  $\theta$  を最適化

$$\begin{aligned}\frac{\partial L}{\partial \theta} &= \left\langle \frac{\partial}{\partial \theta} \log p(\mathbf{w}|\theta) \right\rangle_{\hat{p}(\mathbf{w})} \\ &= \left\langle \frac{\partial}{\partial \theta} \left\{ \log f(\mathbf{w}|\theta) - \log Z \right\} \right\rangle_{\hat{p}(\mathbf{w})} \\ &= \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{w}|\theta) \right\rangle_{\hat{p}(\mathbf{w})} - \left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{w}|\theta) \right\rangle_{p(\mathbf{w}|\theta)}\end{aligned}$$

今求めようとしているモデル  
 $p(\mathbf{w}|\theta)$  自体による期待値！  
(どうする?)

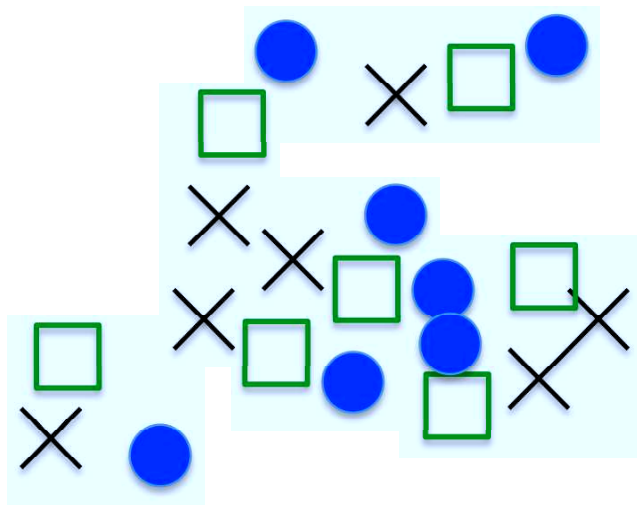
# Contrastive Divergence 学習

$$\left\langle \frac{\partial}{\partial \theta} \log f(\mathbf{w}|\theta) \right\rangle_{p(\mathbf{w}|\theta)}$$

の期待値を、データ点から始めた  
MCMC 1回分で近似  
( $\infty$ 回すればモデル分布)

||

擬似的な「負例」, fantasy data



- × : 実際のデータ点
- : モデルからの真のサンプル
- : MCMC1回分のサンプル (fantasy data)

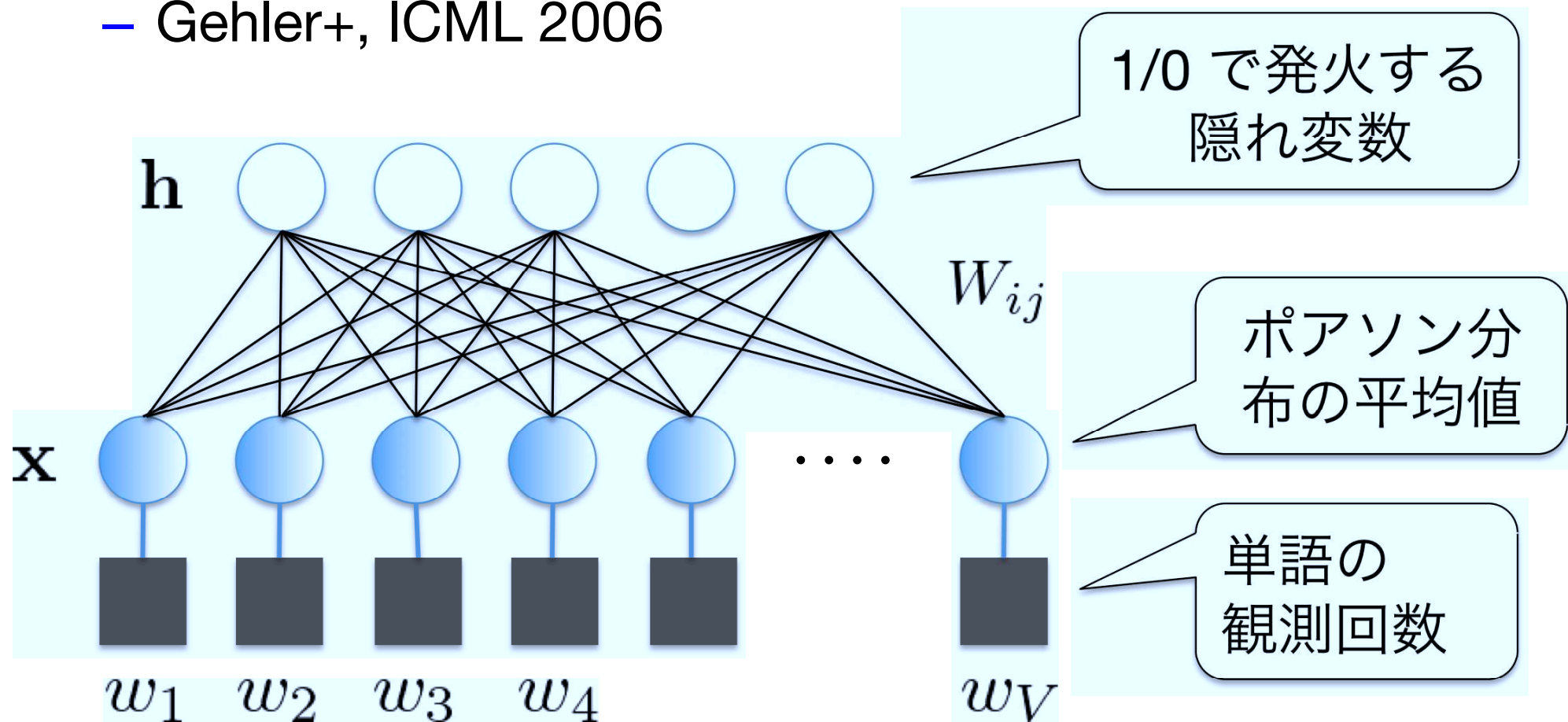
- PRML4章, ロジスティック回帰(教師あり) (4.93)式

$$\nabla E(\theta) = - \sum_n (t_n - \theta^T \phi_n) \phi_n$$

正解とモデル予測との差

# テキストのProduct Model

- RaP (Rate Adapting Poisson) モデル
  - Gehler+, ICML 2006



Restricted Boltzmann Machine (**RBM**)とよばれるニューラルネット

# RaPの確率モデル

---

- RaPでは、潜在層 $\mathbf{h}$ と観測層 $\mathbf{v}$ に以下の結合確率を仮定

$$\begin{aligned} p(\mathbf{v}, \mathbf{h}) &= \frac{\exp\left(\sum_{ij} W_{ij} v_i h_j + f(\mathbf{v}) + g(\mathbf{h})\right)}{Z} \\ &= \frac{\prod_{ij} \exp\left(W_{ij} v_i h_j\right) \cdot e^{f(\mathbf{v})} e^{g(\mathbf{h})}}{Z} \end{aligned}$$

- RaP(一般に、こうしたRBM)はProduct Model !



## RaPの確率モデル (2)

---

- 潜在層と観測層が条件付き確率で結ばれる

$$p(\mathbf{x}|\mathbf{h}) = \prod_i \text{Po}\left(x_i \mid \log(\lambda_i) + W_{ij}h_j\right)$$
$$p(\mathbf{h}|\mathbf{x}) = \prod_j \text{Bin}\left(h_j \mid \sigma\left(\log\left(\frac{p_j}{1-p_j}\right)\right) + \sum_i W_{ij}x_i\right)$$

- 学習:  $x$ から $h$ をサンプル/ $h$ から $x$ をサンプル,  
をMCMCで繰り返して勾配を計算
  - Contrastive Divergence 学習!

# RaPの解釈

- 潜在トピック層を周辺化して消去すると,

$$p(\mathbf{x}) \propto \prod_i \lambda_i \frac{e^{x_i}}{x_i!} \cdot \prod_j (1 + \exp(\underbrace{\sum_i W_{ij} x_i - \beta_j}_{\text{トピック } j \text{ に関する } x \text{ の "activation" }}))$$

x の Poisson  
事前確率

トピック  $j$  に関する  $x$  の  
“activation”

トピック  $j$  の励起度  $\geq 1$

- ポアソン分布×トピック別の  
励起度の積

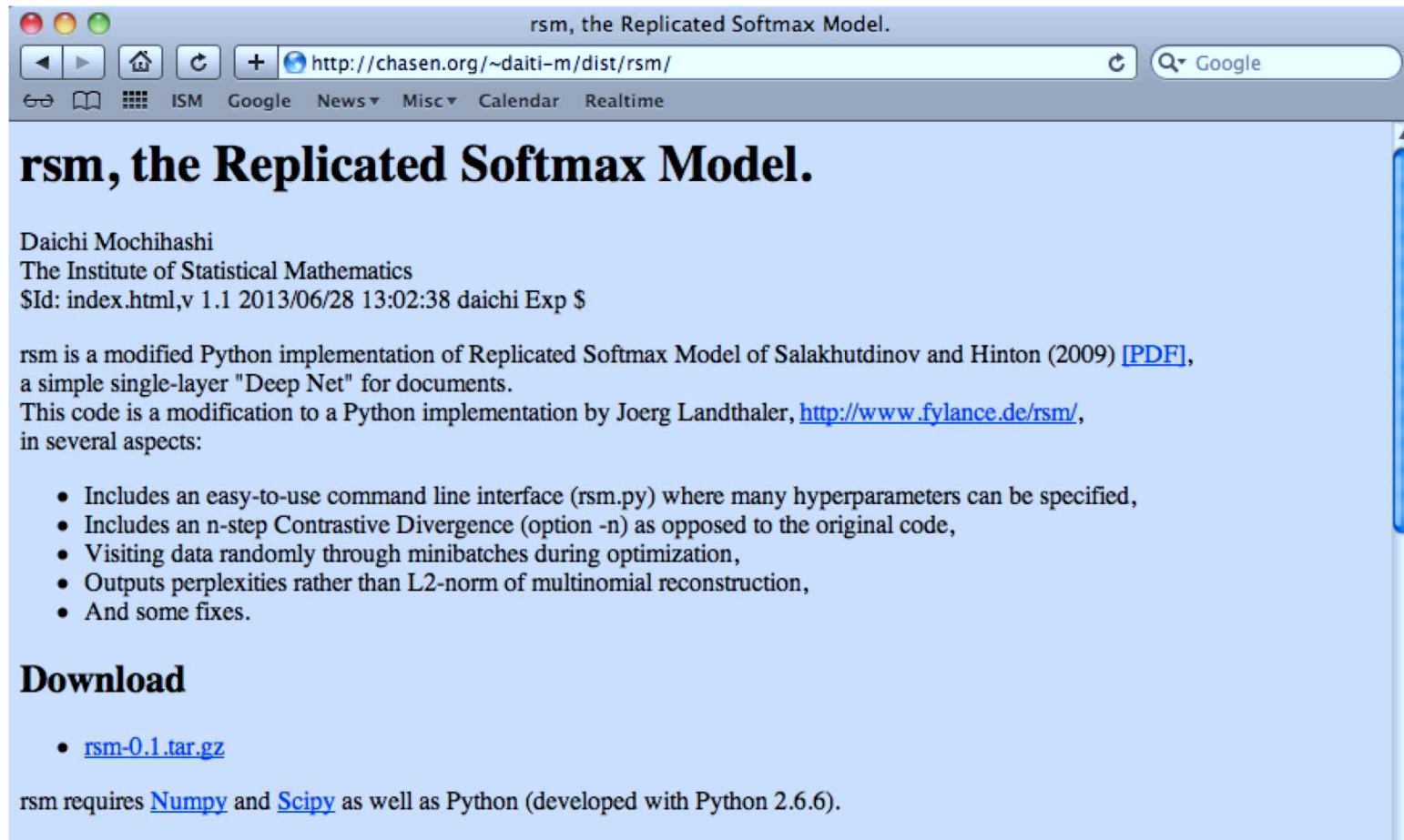
$$\beta_j = -\log\left(\frac{p_j}{1-p_j}\right)$$

とした

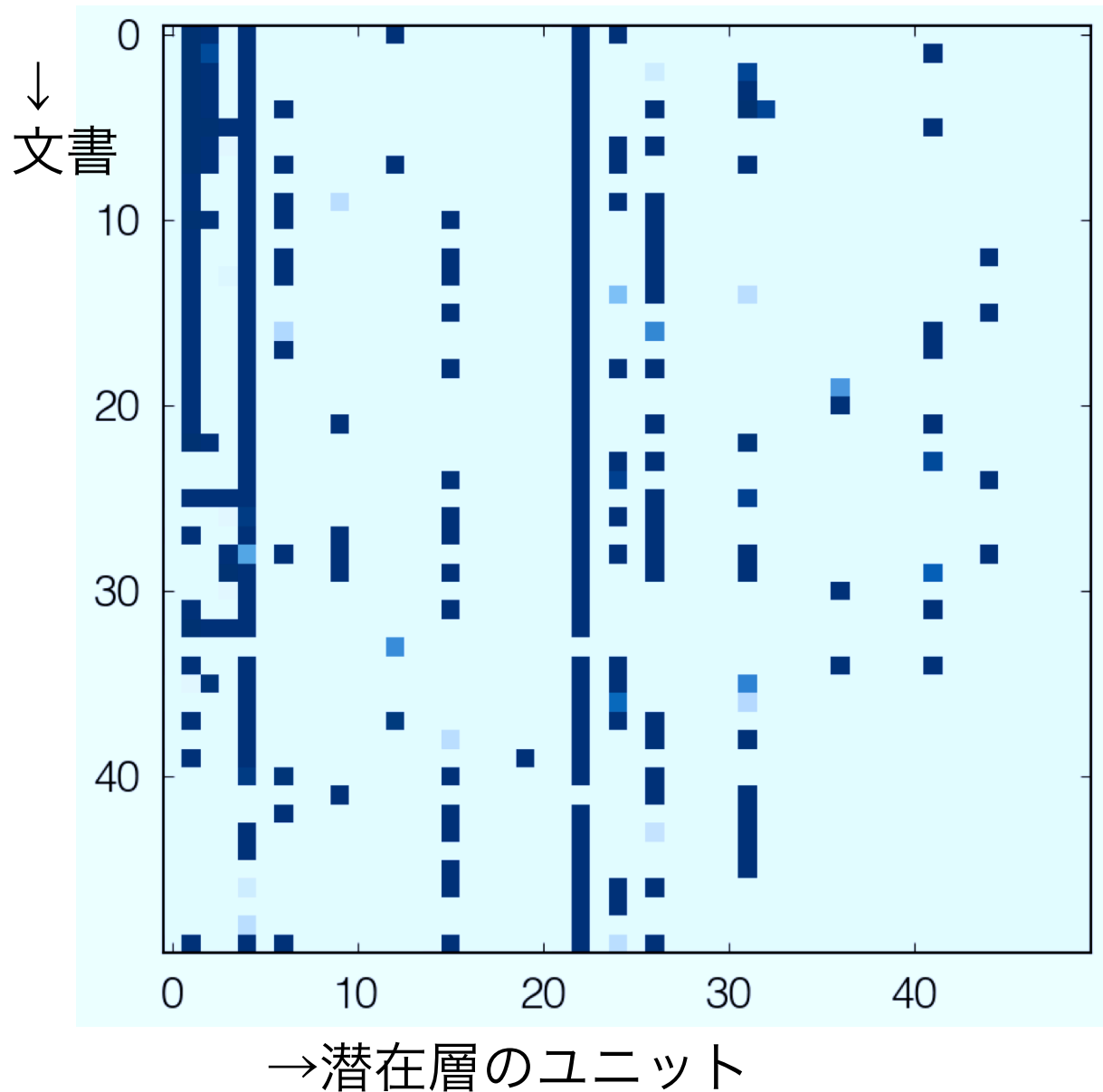
# Replicated Softmax Model

---

- RaPを固定長以外の文書に拡張 (Salakhutdinov+ 09)
  - モデルや学習方法はほぼ同じ、State of the art
- 実装: <http://www.ism.ac.jp/~daichi/dist/rsm/>



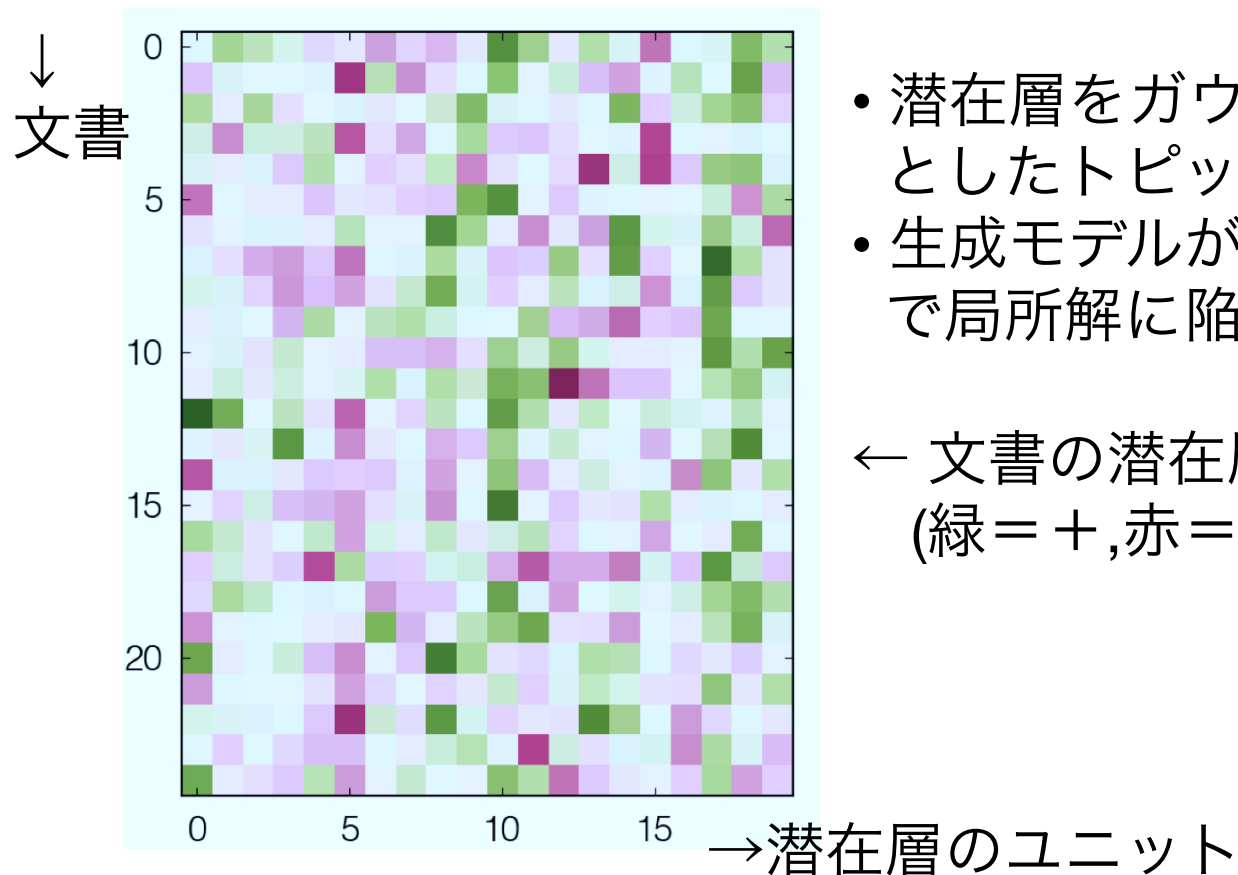
# RSMの学習結果



- RSMで学習した文書の潜在層 (NIPSコーパスの一部)
- 潜在層は $[0,1]$ だが、ほぼ0か1になる
  - テキストの bit coding

# RBM: ただし...

- RBMのContrastive Divergenceによる勾配法は、最適化が非常に難しい
  - きわめて多数の局所解: 学習率、モーメント、初期値……
- 潜在層が二値である必要は、本当はない



- 潜在層をガウス分布 (正負両方)の連続値としたトピックモデル (持橋+ 2013)
- 生成モデルがあるため、最適化はMCMCで局所解に陥らない

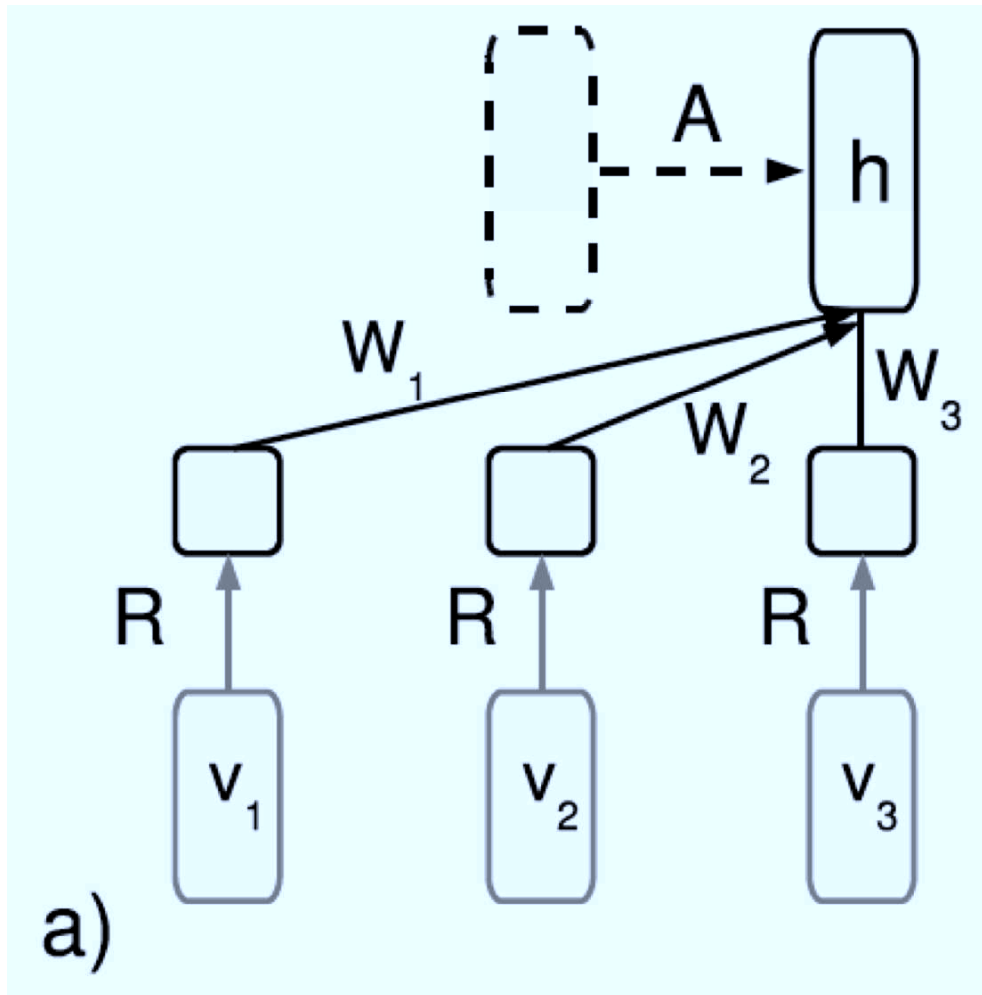
← 文書の潜在層を可視化したもの  
(緑 = +, 赤 = -)

# 言語モデルへの拡張

---

- RBMを時系列の言語データに拡張できないか?
- 言語モデル: 文の確率  $p(w_1, w_2, \dots, w_N)$  を計算
  - $p(w_1, \dots, w_N) = \prod_{n=1}^N p(w_n | w_1 \dots w_{n-1})$  より、
  - $p(w_n | w_1 \dots w_{n-1})$  がわかればよい
- Neural probabilistic language model (NPLM) (Bengio 2003)に近い
  - NPLMはn-gramより高性能

# 単純な拡張 (Mnih+ 2007)

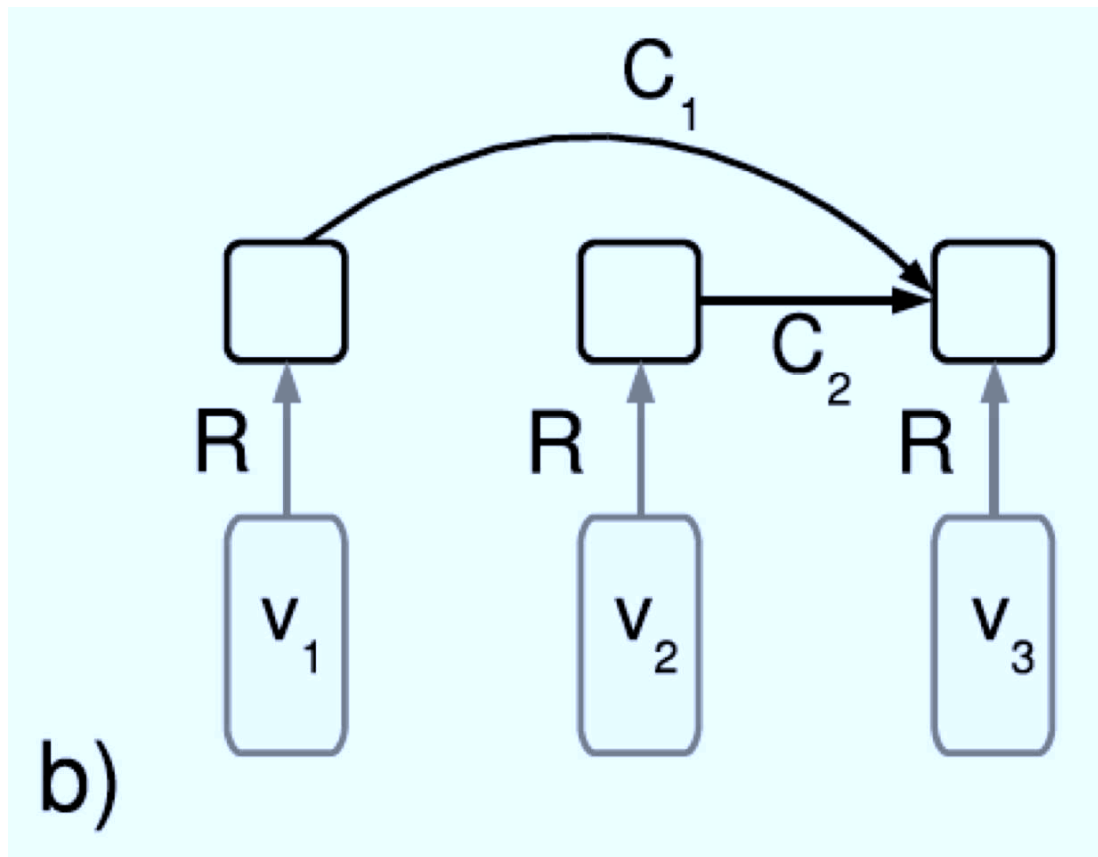


- 各文脈に隠れ層 $h$ あり
- 単語 $v_i$ の連続表現  
 $v_i^T R$ と $h$ を重み行列  
 $W_i$ で内積  
→全体のエネルギー

$$E(w_1, \dots, w_n, h)$$
$$= - \sum_{i=1}^n (v_i^T R) W_i h$$

+ (正則化項).

# LBL (Log-Bilinear Language model)



- 隠れ層  $h$  を消去
- 予測語  $w_n$  と文脈  $w_i$  の連続表現を、位置依存の  $C_i$  で内積

$$\begin{aligned} E(w_1, \dots, w_n, h) &= - \left( \sum_{i=1}^{n-1} v_i^T R C_i \right) R^T v_n \\ &= - \sum_{i=1}^{n-1} \vec{w}_i^T C_i \vec{w}_n \end{aligned}$$

– これに正則化項



# Word embeddingの例 (Mirowski+10)

**Table 8.** Examples of 10 closest neighbors in the latent word embedding space on the Reuters dataset, using an LBLN architecture with 500 hidden nodes,  $|Z_W| = 100$  dimensions for the word representation and  $|Z_X| = 5$  dimensions for the POS features representation. The notion of distance between any two latent word vectors was defined as the cosine similarity. Although word representations were initialized randomly and WordNet::Similarity was not enforced, functionally and semantically (e.g. both synonymic and antonymic) close words tended to cluster.

debt	aa	decrease	met	slow
financing	aaa	drop	introduced	moderate
funding	bbb	decline	rejected	lower
debts	aa-minus	rise	sought	steady
loans	b-minus	increase	supported	slowing
borrowing	a-1	fall	called	double
short-term	bb-minus	jump	charged	higher
indebtedness	a-3	surge	joined	break
long-term	bbb-minus	reduction	adopted	weaker
principal	a-plus	limit	made	stable
capital	a-minus	slump	sent	narrow

# LBL > n-gram

---

Table 2. Perplexity scores for the models trained on the 14M word training set. The mixture test score is the perplexity obtained by averaging the model's predictions with those of the Kneser-Ney 5-gram model. The log-bilinear models use 100-dimensional feature vectors.

Model type	Context size	Model test score	Mixture test score
Log-bilinear	5	117.0	97.3
Log-bilinear	10	107.8	92.1
Back-off KN3	2	129.8	
Back-off KN5	4	123.2	
Back-off KN6	5	123.5	
Back-off KN9	8	124.6	

- LBLはKneser-Ney n-gramよりかなり高性能

# LBL/NPLMの最近の話

---

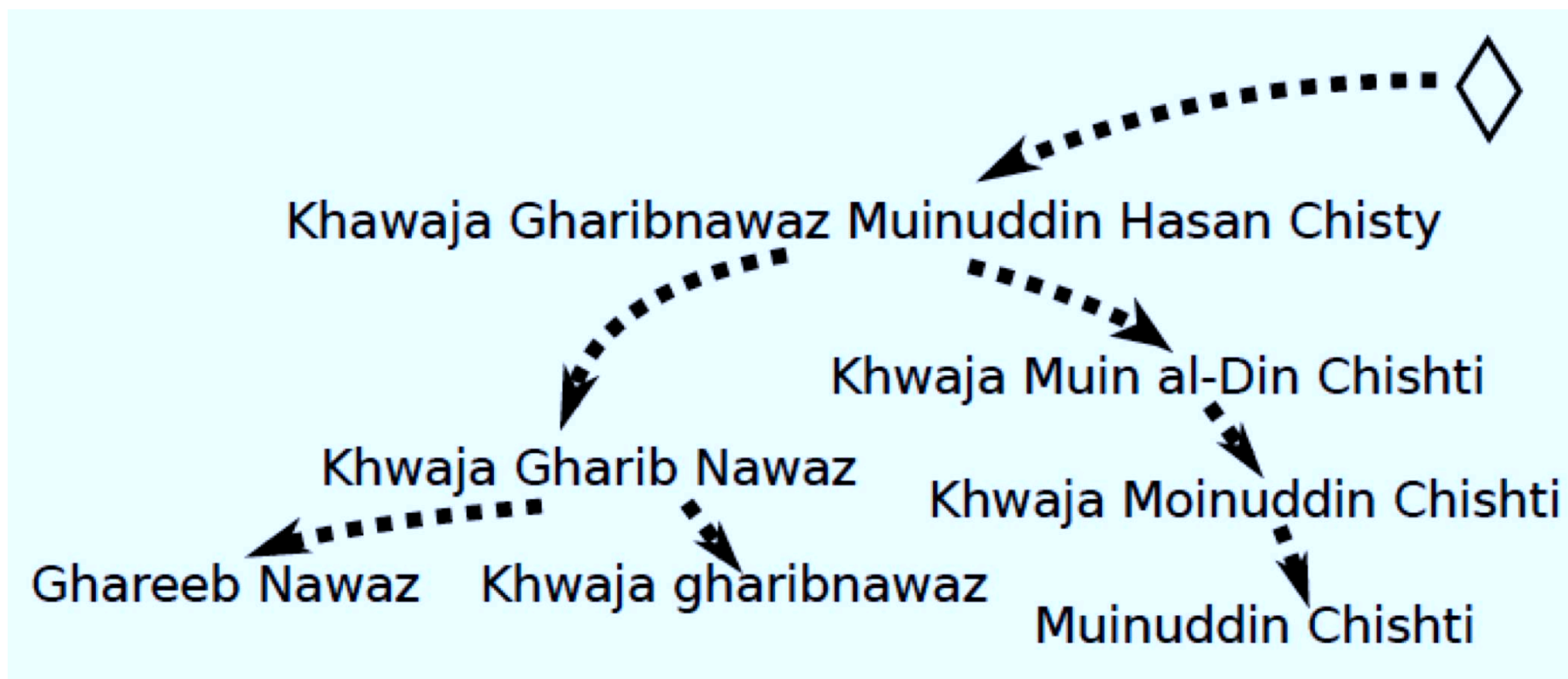
- Hierarchical LBL (HLBL)
  - (Mnih&Hinton, NIPS 2008)
  - 語彙を階層クラスタリングして計算量削減
- LBLの学習高速化 (Mnih&Teh, ICML2012)
  - Contrastive estimationで勾配を計算
- 音声認識への適用 (Mirowski+ 2010)

Table 7. Speech recognition results on TV broadcast transcripts, using the same training set and test set as in Table 6, but with the true sentence to be predicted included among the n-best candidates.

Method	Accuracy
Back-off KN 4-gram	86.9 %
<b>LBLN+POS+init</b>	<b>94 %</b>
“Oracle”	100 %

# 教師なし学習はRBMには限らない

- Deep Netは、教師なし学習のごく一部
- 最近の例: 文字列の Phylogenetic Inference (Andrews+ EMNLP2012)



文字列の変異の系統樹が知りたい

# Andrews+ (2012) “Name Phylogeny”

---

The latent variables in the model are<sup>4</sup>

- The spanning tree over tokens  $\mathbf{p}$
- The token permutation  $\mathbf{i}$
- The topics of all named-entity and context tokens  $\mathbf{z}$

Inference requires marginalizing over the latent variables:

$$\Pr_{\phi, \theta}(\mathbf{x}) = \sum_{\mathbf{p}, \mathbf{i}, \mathbf{z}} \Pr_{\phi, \theta}(\mathbf{x}, \mathbf{z}, \mathbf{i}, \mathbf{p})$$

- どの文字列がどの文字列に書き変わったのかをEMで推定した後、文字列の Transducer (書き換え器) のパラメータを更新  
→EMを繰り返す

# まとめ

---

- 自然言語の教師なし学習の初歩は混合モデル (クラスタリング): NB, UM, LDA, ...
  - さまざまな拡張がある、基本モデル
  - 識別モデルとも統合できる (研究の frontline)
- 混合モデルから積モデルへ
  - さまざまな制約を取り入れることが可能
  - Deep Learning (RBM)は、積モデルの一例
- さらに進んだモデル
  - 積モデル+潜在変数
  - 系統樹推定、進化モデル、文字列Transducer、...
  - 言語の教師なし学習のフロンティアは無限に広い

# 終わり

---

- Any Questions?

# 参考文献

---

- Kamal Nigam+, “Text Classification from Labeled Unlabeled Documents using EM”, *Machine Learning*, 39(2):103-134, 2000.
- Thomas Minka, “Estimating a Dirichlet distribution”, Technical report, 2000.
- 山本幹雄+, 「混合ディリクレ分布を用いた文脈のモデル化と言語モデルへの応用」, 情報処理学会研究報告2003-SLP-48, 2003.
- Mikio Yamamoto and Kugatsu Sadamitsu, “Dirichlet Mixtures in Text Modeling”, CS Technical Report CS-TR-05-1, University of Tsukuba, 2005.
- Steven L. Scott, “Bayesian Methods for Hidden Markov Models”, *JASA*, 97:337-351, 2002.
- B. Merialdo, “Tagging English text with a probabilistic model”, *Computational Linguistics*, 20(2):155-172, 1994.
- 竹内孔一, 松本裕治, 「隠れマルコフモデルによる日本語形態素解析のパラメータ推定」, 情報処理学会論文誌38(3):500-509, 1997.
- Sjölander+, “Dirichlet Mixtures: A Method for Improved Detection of Weak but Significant Protein Sequence Homology”, *Computing Applications in Biosciences*, 12(4):327-345, 1996.



## 参考文献 (2)

---

- Sharon Goldwater, Thomas L. Griffiths, “A Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging”, ACL 2007.
- Beal, Ghahramani, Rasmussen, “The Infinite Hidden Markov Model”, NIPS 2001.
- Y.W.Teh+, “Hierarchical Dirichlet Processes”, JASA, 101(476):1566-1581, 2006.
- J.van Gael+, “Beam sampling for the infinite hidden Markov model”, ICML 2008.
- O. Cappé, E. Moulines, “Online Expectation-Maximization algorithm for Latent data models”, JRSS(B), 71, 2009.
- P. Liang, D. Klein, “Online EM for Unsupervised Models”, NAACL 2009.
- D. Blei+, “Latent Dirichlet Allocation”, NIPS 2001.
- D. Blei+, “Latent Dirichlet Allocation”, JMLR, 3:993-1022, 2003.
- Issei Sato+, “Deterministic Single-Pass Algorithm for LDA”, NIPS 2010.
- Ivan Titov, Ryan Mcdonald. “A Joint Model of Text and Aspect Ratings for Sentiment Summarization”, ACL 2008.

## 参考文献 (3)

---

- Kobus Barnard and David Forsyth, “Learning the Semantics of Words and Pictures”, ICCV 2001.
- Kobus Barnard+, “Matching Words and Pictures”, JMLR, 3:1107-1135, 2003.
- B. Zhao, L. Fei-Fei, E. Xing, “Image Segmentation with Topic Random Fields”, ECCV 2010.
- Jakob Eisenstein+, “A Latent Variable Model for Geographic Lexical Variation”, EMNLP 2010.
- Hinton, G. E., “Training Products of Experts by Minimizing Contrastive Divergence”, Neural Computation, 14:1771-1800, 2002.
- Peter V. Gehler+, “The Rate Adapting Poisson Model for Information Retrieval and Object Recognition”, ICML 2006.
- R. Salakhutdinov and G. Hinton, “Replicated Softmax: an Undirected Topic Model”, NIPS 2009.
- Yoshua Bengio+, “A Neural Probabilistic Language Model”, JMLR, 3:1137-1155, 2003.
- Andriy Mnih and Geoffrey Hinton, “Three New Graphical Models for Statistical Language Modeling”, ICML 2007.

## 参考文献 (4)

---

- Andriy Mnih and Geoffrey Hinton, “A Scalable Hierarchical Distributed Language Model”, NIPS 2008.
- Andriy Mnih and Yee Whye Teh, “A fast and simple algorithm for training neural probabilistic language model”, ICML 2012.
- Piotr Mirowski+, “Feature-rich Continuous Language Models for Speech Recognition”, IEEE Workshop on Spoken Language Technology, 2010.
- Nicholas Andrews, Jason Eisner, Mark Dredze. “Name Phylogeny: A Generative Model for String Variation”, EMNLP 2012.