KYOTO UNIVERSITY

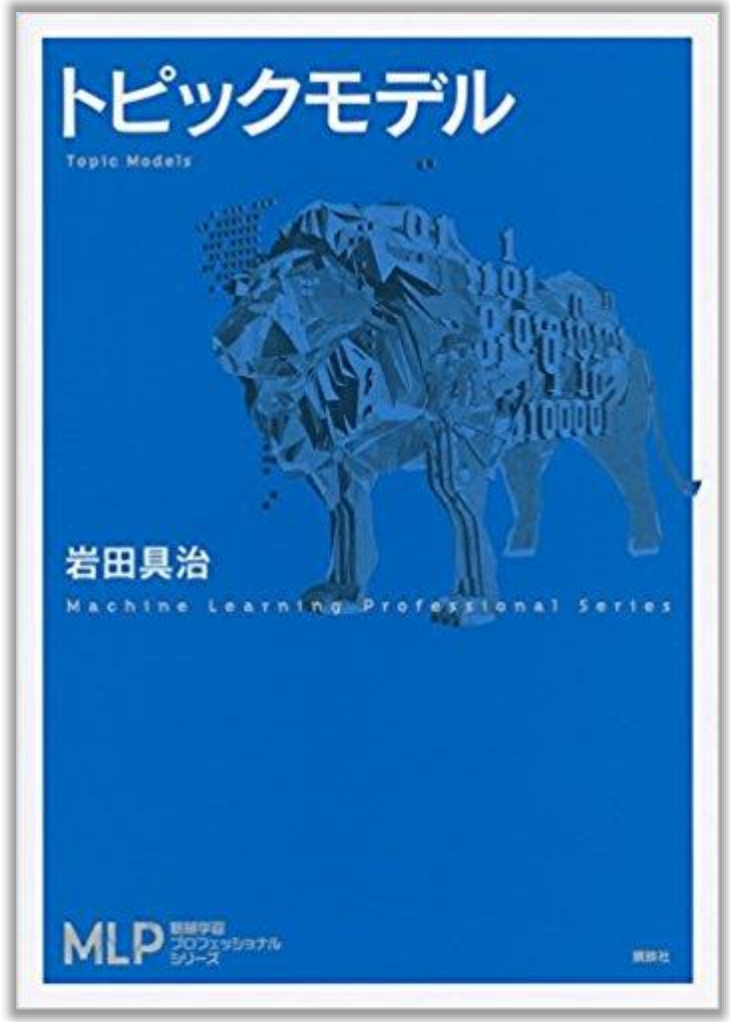
統計的モデリング基礎⑩
～ベイズモデリング～

鹿島久嗣
（情報学科 計算機科学コース）

DEPARTMENT OF INTELLIGENCE SCIENCE
AND TECHNOLOGY

目次：
ベイズモデリング

- ベイズ統計の基礎
  - ベイズの公式、事前分布、事後分布
  - 事後分布による意思決定
  - ナイーブベイズ予測
- ベイズモデリング
  - 離散分布のベイズ推定
  - 階層ベイズモデリング
  - 経験ベイズ法



ベイズ統計：
ベイズの公式によって事前知識と証拠を組み合わせて推論

- 事前知識と観測された証拠から確率を更新する
  - 事前知識：事前分布によって表される、自分が予めもっている、ある事象がどの程度起こりそうかという信念
- 新たな証拠が得られたとき、信念の更新はベイズの公式に基づいて行われる

ベイズ統計で中心的役割を果たすベイズの公式：条件付確率の条件部と帰結部を入れ替える

- 条件付確率$P(\text{rain} \mid \text{cat})$：猫が顔を洗うのを目撃したときに、次の日に雨が降る確率
  - $P(\text{rain} \mid \text{cat}) = \dfrac{P(\text{rain}, \text{cat})}{P(\text{cat})}$
- ベイズの公式：条件付確率の条件部分と帰結部分を入れ替える公式

雨の日の前日に猫が顔を洗っている確率

猫がどうとか関係なく、そもそも雨が降る確率（事前分布）

$$P(\text{rain} \mid \text{cat}) = \frac{P(\text{cat} \mid \text{rain})P(\text{rain})}{P(\text{cat})}$$

雨がどうとか関係なく、そもそも猫が顔を洗う確率

例：
ベイズの公式に基づく事後確率計算

- 事前確率：これまでの経験から雨が降る確率は20%
  - 雨が降らない確率は80%
- 雨がふる日の前日に猫が顔を洗っている確率は80%とすると：
  $P(\text{rain} \mid \text{cat}) = \dfrac{P(\text{cat}|\text{rain})P(\text{rain})}{P(\text{cat})} = \dfrac{0.8\times 0.2}{P(\text{cat})} = \dfrac{0.16}{P(\text{cat})}$ がわかる
- 一方、雨が降らない日の前日に猫が顔を洗っている確率は50%とすると：
  $P(\neg\text{rain} \mid \text{cat}) = \dfrac{P(\text{cat}|\neg\text{rain})P(\neg\text{rain})}{P(\text{cat})} = \dfrac{0.5\times 0.8}{P(\text{cat})} = \dfrac{0.40}{P(\text{cat})}$
- 両者より $P(\text{rain} \mid \text{cat}) = \dfrac{0.16}{0.16+0.40} = 0.29$ （29%）となる

9ポイント増えたよ！

ベイズ決定：
事後分布をもちいた意思決定

- 予測×効用（結果のうれしさ度合い）によって意思決定

効用（結果のうれしさ度合い）　大きいほど うれしい

実際の天候		ビールの仕入れ量	
		多め	少な目
	☀晴れ	+50	0
	☔雨	-100	-10

- 猫が顔を洗った場合の期待効用 U
  - U(多めに仕入れ) $= -100 \times P(\text{rain} \mid \text{cat}) + 50 \times P(\neg\text{rain} \mid \text{cat}) = -100 \times 0.29 + 50 \times 0.71 = 6.5$　（0.29）
  - U(少な目に仕入れ) $= -10 \times P(\text{rain} \mid \text{cat}) + 0 \times P(\neg\text{rain} \mid \text{cat}) = -10 \times 0.29 + 0 \times 0.71 = -2.9$
  - 多めに仕入れたほうが期待効用が高い

ナイーブベイズ予測：テキスト分類の初等的手法

- ある文書が特定のカテゴリに属する確率を知りたい
  - たとえば、Webページをみて、そのトピックが経済なのか、スポーツなのか、政治なのか、芸能なのか、...を判別するのに使える
  - つまり、事後確率$P(\text{topic} \mid \text{text})$を知りたい
- ベイズの公式により、事後確率は：

$$P(\text{topic} \mid \text{text}) = \frac{P(\text{text} \mid \text{topic})P(\text{topic})}{P(\text{text})}$$

  - $P(\text{topic})$：そのトピックが観測される確率（事前確率）
  - $P(\text{text} \mid \text{topic})$：あるトピックが決まった時に、そのWebページのテキストがつくられる確率

ナイーブベイズ予測：
予測にはテキストの生成確率の計算が必要

- テキストが与えられたときのトピックの事後確率は：

$$P(\text{topic} \mid \text{text}) = \frac{P(\text{text} \mid \text{topic})P(\text{topic})}{\sum_{\text{topic}'} P(\text{text} \mid \text{topic}')P(\text{topic}')}$$

分母$= P(\text{text})$

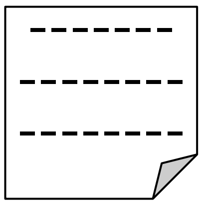
- $P(\text{topic}) = \dfrac{\text{そのトピックの文書数}}{\text{全文書数}}$ で計算できる（最尤推定）

- 一方、$P(\text{text} \mid \text{topic})$ をどのように考えるかは自明ではない
  - トピックが決まった時の文書生成モデルが必要
  - 例えば、マルコフモデル？隠れマルコフ？RNN？
  - もうちょっと簡単な場合を考えてみる…

テキストの発生確率モデル：
単語袋(bag-of-words)モデル

- 文書 **x** を、文書中に出現する単語によって表現

文書



$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix}$$

「選手」の含まれる回数

「政党」の含まれる回数

⋮

「株価」の含まれる回数

bag-of-words 表現

- 単純化のための仮定：各単語の発生は独立とする

$$P(\text{text} \mid \text{topic})$$
$$= P(w_1 \mid \text{topic})^{n_1} P(w_2 \mid \text{topic})^{n_2} \cdots P(w_D \mid \text{topic})^{n_D}$$

辞書に含まれるある単語

ナイーブベイズ予測：単語袋モデルに基づくテキストの発生確率の計算

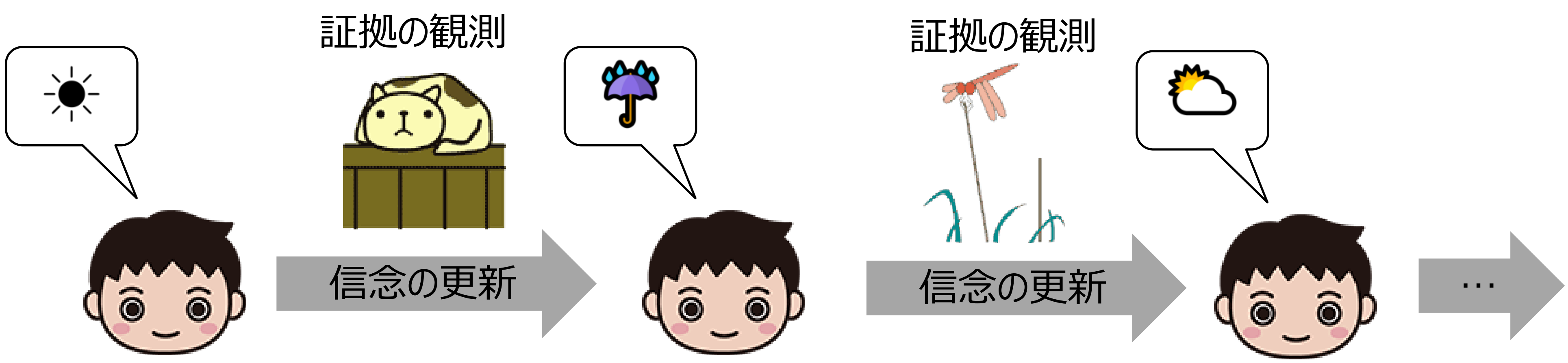
- 計算したいのは $P(\underset{\text{「スポーツ」}}{\text{topic}} \mid \text{text}) \propto P(\text{text} \mid \underset{\text{「スポーツ」}}{\text{topic}}) P(\underset{\text{「スポーツ」}}{\text{topic}})$
- $P(\text{topic}) = \dfrac{\text{そのトピックの文書数}}{\text{全文書数}}$ で推定できる（最尤推定）
- $P(\text{text} \mid \text{topic}) =$
  $P(\underset{\text{「選手」}}{w_1} \mid \underset{\text{「スポーツ」}}{\text{topic}})^{n_1} P(w_2 \mid \text{topic})^{n_2} \cdots P(w_D \mid \text{topic})^{n_D}$
  - $P(\underset{\text{「選手」}}{w_1} \mid \underset{\text{「スポーツ」}}{\text{topic}}) = \dfrac{\text{そのトピック}^{\text{「スポーツ」}}\text{の文書中で}\,w_1^{\text{「選手」}}\,\text{が現れた回数}}{\text{そのトピックの文書中の総単語出現数}}$ と推定（最尤推定）
- すべてのtopicについて$P(\text{text} \mid \text{topic})P(\text{topic})$を計算し正規化する

信念の逐次更新：証拠が得られるごとに信念が更新される

- ベイズの定理は証拠をもとに信念を更新する
- 証拠が新しく得られるたびに信念が更新される：

$$P(\mathrm{rain}) \rightarrow P(\mathrm{rain} \mid \mathrm{cat}) \rightarrow P(\mathrm{rain} \mid \mathrm{cat}, \mathrm{dragonfly}) \rightarrow \cdots$$

  - 証拠間が条件付き独立であるなら証拠の観測順序は関係ない
  - 他の例：テキスト分類で、一単語観測されるごとに予測を更新



ベイズモデリング

離散分布の推定問題：サイコロの各目の出る確率を実際の出目から推定する

- 母集団は離散分布に従うとする
  - $P(X=j)=p_j$ （ただし$\sum_{j\in\{1,2,\ldots,k\}} p_j=1, p_j\geq 0$）
  - たとえば（厳密な）サイコロであれば$P(X=j)=\frac{1}{6}\approx 0.17$
- 標本抽出：
  - サイコロを20回（独立に）振ったところ、
    6 3 5 1 3 1 4 1 2 2 6 1 2 2 5 4 4 4 6 5 が出た

出目	1	2	3	4	5	6
回数	4	4	2	4	3	3

- 母集団のパラメータ（それぞれの目の出る確率）を推定したい

ベイズ的統計モデリングの考え方：最尤推定の尤度の代わりに事後分布を考える

- ベイズ統計では事後分布$P(\text{パラメータ} \mid \text{データ})$を考える
  - 事後分布ではパラメータを確率変数と考える
- 事後分布：

$$P(\text{パラメータ} \mid \text{データ}) = \frac{P(\text{データ} \mid \text{パラメータ})P(\text{パラメータ})}{P(\text{データ})}$$

ベイズの定理

- 対数事後分布：

$$\log P(\text{パラメータ} \mid \text{データ}) = \underbrace{\log P(\text{データ} \mid \text{パラメータ})}_{\text{尤度}} + \underbrace{\log P(\text{パラメータ})}_{\text{事前分布}} + \underbrace{\log P(\text{データ})}_{\text{定数}}$$

離散分布のベイズ推定：
ディリクレ分布を事前分布とする

- 離散分布： $\mathbf{p} = (p_1, p_2, \ldots, p_k), \sum_{j=1}^{k} p_j = 1, p_j \geq 0$
- データ：$\mathbf{n} = (n_1, n_2, \ldots, n_k)$
  - n_j ：各シンボル$j \in \{1,2, \ldots, k\}$の観測数
- パラメータの事後分布 $P(\mathbf{p} \mid \mathbf{n}) = \frac{P(\mathbf{n}|\mathbf{p})P(\mathbf{p})}{P(\mathbf{n})} = \frac{P(\mathbf{n}|\mathbf{p})P(\mathbf{p})}{\int_{\mathbf{p}} P(\mathbf{n}|\mathbf{p})P(\mathbf{p})\mathrm{d}\mathbf{p}}$
- 事前分布$P(\mathbf{p})$をディリクレ分布とする
  - ディリクレ分布は離散分布の共役事前分布
  - 共役事前分布：
事後分布と事前分布の形が同じになるような事前分布

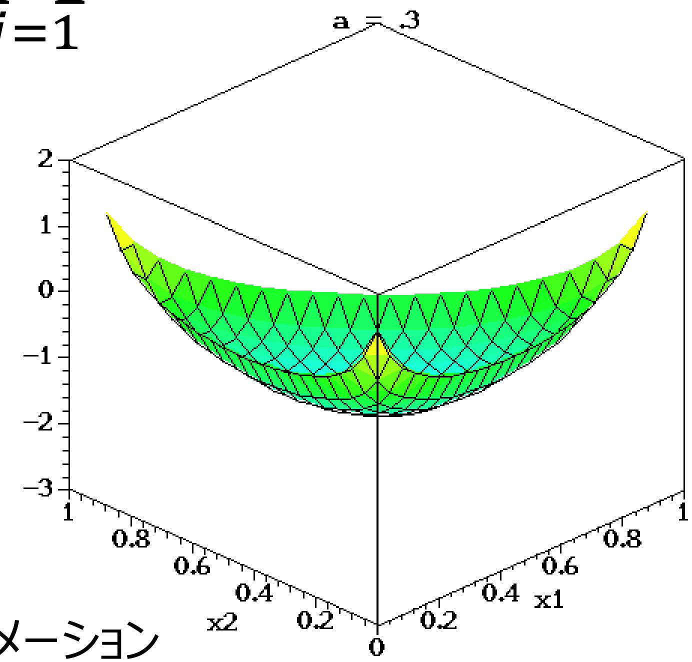
ディリクレ分布：
離散分布の（共役）事前分布

- ディリクレ分布：離散分布$\mathbf{p} = (p_1, p_2, \dots, p_k), p_j \geq 0, \sum_{j=1}^{k} p_j = 1$を生成する確率モデル

ガンマ関数

$$P(p_1, p_2, \dots, p_k) = \frac{\Gamma\left(\sum_{j=1}^{k} \alpha_k\right)}{\prod_{j=1}^{k} \Gamma\left(\alpha_j\right)} \prod_{j=1}^{k} \left(p_j\right)^{\alpha_j - 1}$$

- $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k) \geq 0$は（超）パラメータ
- $\int_{\mathbf{p}} \prod_{j=1}^{k} \left(p_j\right)^{\alpha_j - 1} \mathrm{d}\mathbf{p} = \frac{\prod_{j=1}^{k} \Gamma\left(\alpha_j\right)}{\Gamma\left(\sum_{j=1}^{k} \alpha_k\right)}$



$0.3 \leq \alpha_1 = \alpha_2 = \alpha_3 \leq 2.0$のアニメーション

https://en.wikipedia.org/wiki/Dirichlet_distribution#/media/File:LogDirichletDensity-alpha_0.3_to_alpha_2.0.gif

事後分布の計算：
事前分布と事後分布がともにディリクレ分布になる

- 事後分布：$P(\mathbf{p} \mid \mathbf{n}) = \frac{P(\mathbf{n}|\mathbf{p})P(\mathbf{p})}{P(\mathbf{n})} = \frac{P(\mathbf{n}|\mathbf{p})P(\mathbf{p})}{\int_{\mathbf{p}} P(\mathbf{n}|\mathbf{p})P(\mathbf{p})\mathbf{dp}}$

- $P(\mathbf{n} \mid \mathbf{p})P(\mathbf{p}) = \prod_{j=1}^{k}\left(p_j\right)^{n_j} \frac{\Gamma\left(\sum_{j=1}^{k}\alpha_k\right)}{\prod_{j=1}^{k}\Gamma\left(\alpha_j\right)} \prod_{j=1}^{k}\left(p_j\right)^{\alpha_j-1}$

事前分布がディリクレ分布

- 以上を用いて $P(\mathbf{p} \mid \mathbf{n}) = \frac{\prod_{j=1}^{k}\left(p_j\right)^{n_j+\alpha_j-1}}{\int_{\mathbf{p}} \prod_{j=1}^{k}\left(p_j\right)^{n_j+\alpha_j-1}\mathbf{dp}}$ ※ 正規化したら1にならないといけないので

$$= \frac{\Gamma\left(\sum_{j=1}^{k} n_j + \alpha_j\right)}{\prod_{j=1}^{k}\Gamma\left(n_j + \alpha_j\right)} \prod_{j=1}^{k}\left(p_j\right)^{n_j+\alpha_j-1}$$

事後分布もディリクレ分布

ベイズ予測分布：
事前分布の情報を余すところなく使って予測

- MAP推定では事後分布が最大となるパラメータを**点**推定する

$$\hat{\mathbf{p}} = \mathrm{argmax}_{\mathbf{p}} P(\mathbf{p} \mid \mathbf{n})$$

  - 得られたパラメータを次のシンボルxの予測に用いる

$$P(x \mid \hat{\mathbf{p}}) = \hat{p}_x, x \in \{1, 2, \dots, k\}$$

- ベイズ予測では「事後分布そのもの」を用いて予測する

$$P(x \mid \mathbf{n}) = \int_{\mathbf{p}} P(x \mid \mathbf{p}) P(\mathbf{p} \mid \mathbf{n}) \, \mathrm{d}\mathbf{p}$$

  - あらゆるパラメータのモデルの予測を事後確率で重みづけて予測

- パラメータを点推定するのでなく**事後分布ごっそり**使う

ベイズ予測分布：事前分布による重みづけ予測

- ベイズ予測

パラメータのとりうるあらゆる可能性を重みづけて用いる

$$P(x \mid \mathbf{n}) = \int_{\mathbf{p}} P(x \mid \mathbf{p}) P(\mathbf{p} \mid \mathbf{n})\, \mathrm{d}\mathbf{p}$$

パラメータのとりうるあらゆる可能性	そのパラメータをもつモデルの予測確率	重み（パラメータの事後確率）
$\mathbf{p} = (0,0,1)$	$P(x\|(0,0,1))$	$\times P\big((0,0,1)\big\|\mathbf{n}\big)$
$\mathbf{p} = (0,0.5,0.5)$	$P(x\|(0,0.5,0.5))$	$\times P\big((0,0.5,0.5)\big\|\mathbf{n}\big)$
$\mathbf{p} = (0.33,0.33,0.34)$	$P(x\|(0.33,0.33,0.34))$	$\times P\big((0.33,0.33,0.34)\big\|\mathbf{n}\big)$
$\vdots$	$\vdots$	$\vdots$

最尤推定・MAP推定は、ある特定のパラメータの値を決定（点推定）して用いる

離散分布のベイズ予測分布：MAP推定同様に加算平滑化が導かれる

$$\int_{\mathbf{p}} \prod_{j=1}^{k} (p_j)^{\alpha_j - 1} \, \mathrm{d}\mathbf{p} = \frac{\prod_{j=1}^{k} \Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{k} \alpha_k)}$$

$$P(x \mid \mathbf{n}) = \int_{\mathbf{p}} P(x \mid \mathbf{p}) P(\mathbf{p} \mid \mathbf{n}) \, \mathrm{d}\mathbf{p}$$

$$= \int_{\mathbf{p}} p_x \frac{\Gamma\left(\sum_{j=1}^{k} n_j + \alpha_j\right)}{\prod_{j=1}^{k} \Gamma\left(n_j + \alpha_j\right)} \prod_{j=1}^{k} \left(p_j\right)^{n_j + \alpha_j - 1} \mathrm{d}\mathbf{p}$$

$$= \frac{\Gamma\left(\sum_{j=1}^{k} n_j + \alpha_j\right)}{\prod_{j=1}^{k} \Gamma\left(n_j + \alpha_j\right)} \int_{\mathbf{p}} \left(p_x\right)^{n_x + 1 + \alpha_x - 1} \prod_{j \neq x} \left(p_j\right)^{n_j + \alpha_j - 1} \mathrm{d}\mathbf{p}$$

$$= \frac{\Gamma\left(\sum_{j=1}^{k} n_j + \alpha_j\right)}{\prod_{j=1}^{k} \Gamma\left(n_j + \alpha_j\right)} \frac{\Gamma(n_x + 1 + \alpha_x) \prod_{j \neq x} \Gamma\left(n_j + \alpha_j\right)}{\Gamma\left(1 + \sum_{j=1}^{k} n_j + \alpha_j\right)}$$

$$= \frac{n_x + \alpha_x}{\sum_{j=1}^{k} n_j + \alpha_j}$$

$$\Gamma(x+1) = x\Gamma(x)$$

この例ではMAP推定と結論がほぼ変わらず、ベイズ予測のありがたみはないが…

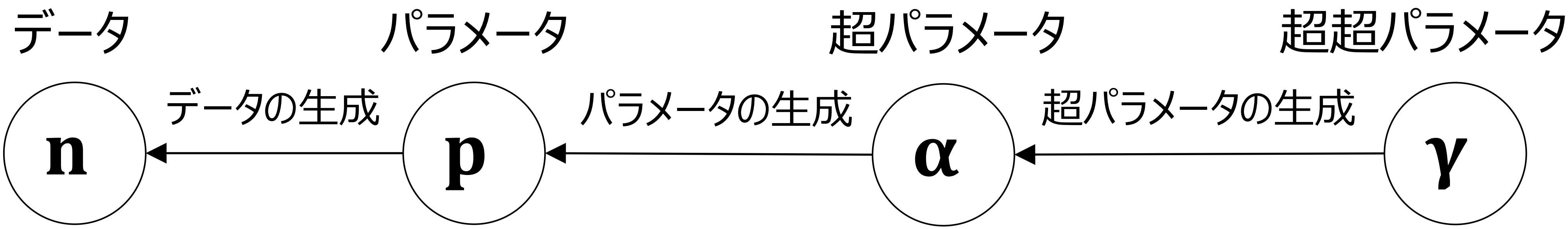
階層ベイズモデル：
事前分布の事前分布（の事前分布の･･･）を考える

- 事前分布$P(\mathbf{p} \mid \boldsymbol{\alpha})$も（超）パラメータ$\boldsymbol{\alpha}$をもつ
  → どのように決めたらよいか

- 超パラメータの事前分布$P(\boldsymbol{\alpha})$を考える
  - あるいは、データにあわせてチューニングする（後述）

- 事前分布のパラメータもまた事前分布をもつとすると
  モデルが階層化される

この辺になると
なんかもうどうでも
よくなってくる



経験ベイズ推定：
周辺尤度の最大化による超パラメータの推定

- 超パラメータを推定したい
- 周辺尤度：$P(\mathbf{n} \mid \boldsymbol{\alpha}) = \int_{\mathbf{p}} P(\mathbf{n} \mid \mathbf{p}) P(\mathbf{p} \mid \boldsymbol{\alpha})\, \mathrm{d}\mathbf{p}$ を考える
  - パラメータ$\mathbf{p}$が積分消去（周辺化）されていることに注意
- 経験ベイズ法：
  周辺尤度を最大化する超パラメータ$\boldsymbol{\alpha}$を推定値とする

$$\hat{\boldsymbol{\alpha}} = \mathrm{argmax}_{\boldsymbol{\alpha}} P(\mathbf{n} \mid \boldsymbol{\alpha})$$

- MAP推定では超パラメータは推定できないことに注意する
  （正則化パラメータのときと同じ理由）

離散分布の経験ベイズ推定：
対数周辺尤度を勾配法によって最大化する

- 周辺尤度：$P(\mathbf{n} \mid \boldsymbol{\alpha}) = \int_{\mathbf{p}} P(\mathbf{n} \mid \mathbf{p}) P(\mathbf{p} \mid \boldsymbol{\alpha})\,\mathrm{d}\mathbf{p}$

- 離散分布の周辺尤度（数値的に最大化する必要あり）：

$$P(\mathbf{n} \mid \boldsymbol{\alpha}) = \int_{\mathbf{p}} \prod_{j=1}^{k} \left(p_j\right)^{n_j} \frac{\Gamma\left(\sum_{j=1}^{k} \alpha_k\right)}{\prod_{j=1}^{k} \Gamma\left(\alpha_j\right)} \prod_{j=1}^{k} \left(p_j\right)^{\alpha_j - 1} \mathrm{d}\mathbf{p}$$

$$= \frac{\Gamma\left(\sum_{j=1}^{k} \alpha_k\right)}{\prod_{j=1}^{k} \Gamma\left(\alpha_j\right)} \int_{\mathbf{p}} \prod_{j=1}^{k} \left(p_j\right)^{n_j + \alpha_j - 1} \mathrm{d}\mathbf{p} = \frac{\Gamma\left(\sum_{j=1}^{k} \alpha_k\right)}{\prod_{j=1}^{k} \Gamma\left(\alpha_j\right)} \frac{\prod_{j=1}^{k} \Gamma\left(n_j + \alpha_j\right)}{\Gamma\left(\sum_{j=1}^{k} n_j + \alpha_j\right)}$$

- 勾配法による$\boldsymbol{\alpha}$の最適化：

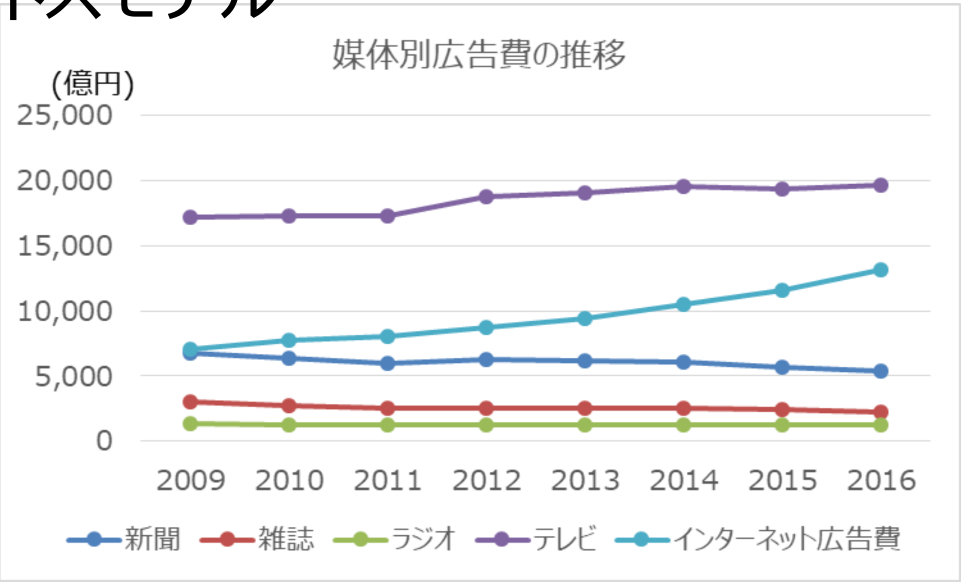
$$\boldsymbol{\alpha}^{\mathrm{NEW}} = \boldsymbol{\alpha} + \eta \boldsymbol{\nabla}_{\boldsymbol{\alpha}} \log P(\mathbf{n} \mid \boldsymbol{\alpha})$$

  - ディガンマ関数（対数ガンマ関数の微分）が必要でやや面倒

ベイズモデリングの応用

インターネット広告：
あらゆる場所に潜むオンライン広告

- インターネット上は広告だらけ
  - Webページ上のバナー広告
  - 検索エンジンのリスティング広告
- 多くのWeb系企業の主要なビジネスモデル
  - 広告表示・クリック・成約に応じた課金
  - アフィリエイト



電通「日本の広告費」http://www.dentsu.co.jp/knowledge/ad_cost/2011/media.htmlより作成

ネット広告配信の最適化：
適切な広告を 適切な人に届ける

- ネット広告の特徴：
  - 細かい配信調整ができる
    - ページ閲覧ごとに別の広告を提示できる
  - 効果が測定しやすい
    - 誰が広告をみて商品を買ったかがデータとして取得できる
- 細かい配信最適化によって広告の効果を最大化する
  - どの広告を提示するべきか（← この問題を考えてみる）
  - 誰に広告を提示すべきか
  - どこに広告を出すべきか

提示する広告選択の最適化：
広告効果を推定して効果の高い広告に集中投下

- 広告配信の最適化問題：
  - 100種類の広告から100万回配信する
  - 広告がクリックされる回数を最大化したい

- 各広告の効果（クリック率）は未知
  - 実際に広告を配信してみないと広告の効果はわからない
- 推定と利用のトレードオフ

1. 推定：すべての広告をまんべんなく配信し、クリック率を推定
2. 利用：クリック率が大きいものを集中的に配信

の両者をバランスよく行う必要がある

バンディット問題：
広告配信最適化問題のモデル

- バンディット問題：
  - 当たりの確率が不明な複数のスロットマシンがある
  - 1回につきひとつのスロットマシンを選んでプレイ
  - 当たり回数を最大化したい
- 広告配信最適化問題は
バンディット問題として定式化できる
- ポイント：
推定と利用のバランスをいかにとるか

http://en.wikipedia.org/wiki/File:Las_Vegas_slot_machines.jpg

バンディット問題のモデル化：各スロットマシンが未知の当たり確率をパラメータとしてもつ

- m台のスロットマシンがある
- スロットマシンiの当たり確率はθ_i（未知）とする
  - つまり、スロットマシンの当たりをベルヌイ分布としてモデル化する
- データ：スロットマシンiをこれまでにn_i回プレイしてh_i回当たった
- 次にどのマシンをプレイすべきかを決定する問題

スロットマシン1
当たり確率θ_1

スロットマシン2
当たり確率θ_2

・・・

スロットマシンm
当たり確率θ_m

バンディット問題の問題：推定精度のばらつき

- 当たり確率の最尤推定値は $\hat{\theta}_i = \frac{h_i}{n_i}$ なので、$\hat{\theta}_i$が最大のスロットマシンをプレイすればよいだろうか？
- でも、当たり確率の推定値が同じでも、過去のプレイ回数が多い（少ない）ほど推定値の信頼度は高い（低い）はず…

$\overset{?}{=}$

$\frac{1}{2} = 50\%$　　$\frac{5}{10} = 50\%$

$\overset{?}{>}$

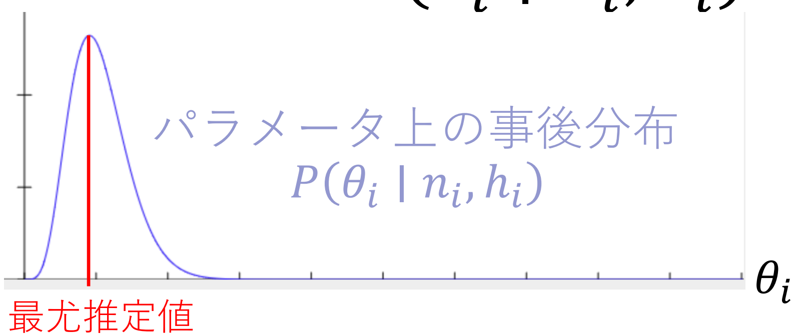
$\frac{2}{3} = 66\%$　　$\frac{10}{20} = 50\%$

当たり確率のベイズモデリング：推定精度を考慮した当たり確率の推定

- 当たり確率の推定値のばらつきをベイズモデリングによって考慮する
- 当たり確率の事後分布は$P(\theta_i \mid n_i, h_i)$
  - マシンiをこれまでにn_i回プレイしてh_i回当たったという状況
  - 事後分布の広がり具合が推定の曖昧さに対応
- 例によってベイズの定理を用いると：

事前分布

$$P(\theta_i \mid n_i, h_i) = \frac{P(n_i, h_i \mid \theta_i)P(\theta_i)}{\int_0^1 P(n_i, h_i \mid \theta_i)P(\theta_i)\mathrm{d}\theta_i}$$



当たり確率の事前分布：
ベータ分布はベルヌイ分布の共役事前分布

- 事前分布をベータ分布とする：

$$P(\theta_i) = \frac{1}{B(\alpha,\beta)}(\theta_i)^{\alpha-1}(1-\theta_i)^{\beta-1}$$

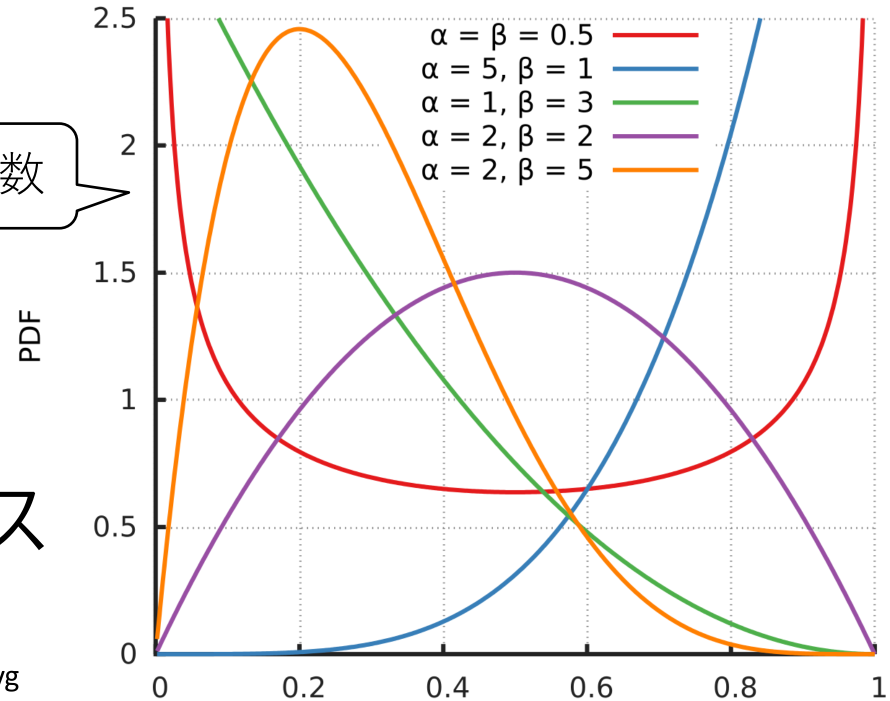
ディリクレ分布で
$k=2$の場合に相当

- ベータ関数 $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$
- 分散は$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

（α,βが大きくなると小さくなる）

- ディリクレ分布の特殊（2値）ケース

確率密度関数



https://en.wikipedia.org/wiki/Beta_distribution#/media/File:Beta_distribution_pdf.svg

当たり確率の事後分布：事後分布もベータ分布になる

- 事後分布はベータ分布になる（共役事前分布）：

$$P(\theta_i \mid n_i, h_i) = \frac{1}{B(h_i+\alpha, n_i-h_i+\beta)}(\theta_i)^{h_i+\alpha-1}(1-\theta_i)^{n_i-h_i+\beta-1}$$

- 導出：

$$P(\theta_i \mid n_i, h_i) \propto P(n_i, h_i \mid \theta_i)P(\theta_i)$$

$$= (\theta_i)^{h_i}(1-\theta_i)^{n_i-h_i}\frac{1}{B(\alpha,\beta)}(\theta_i)^{\alpha-1}(1-\theta_i)^{\beta-1}$$

$$= \frac{1}{B(\alpha,\beta)}(\theta_i)^{h_i+\alpha-1}(1-\theta_i)^{n_i-h_i+\beta-1}$$

改めて$\frac{1}{B(h_i+\alpha, n_i-h_i+\beta)}$で正規化する

つぎのマシンの決定：
事後分布から最も当たり確率が高そうなマシンをプレイ

- 戦略：
  「マシンiの当たり確率θ_iが全マシンの中で最も高い確率」π_i
  に従ってマシンを選択する：

$$\pi_i = \Pr\left[\, \theta_i \geq \theta_j, \forall j \neq i \mid \{n_i, h_i\}_{i=1,\ldots,m} \,\right]$$

- しかし、π_i を解析的に直接評価することは困難

- サンプリングで解決：
  実際に事後分布$P(\theta_i \mid n_i, h_i)$を使って$\theta_i$（$i = 1,2, \ldots, m$）を
  サンプリングしてみて、その中で一番大きいθ_iを採用する

トンプソン抽出：
サンプリングによって次のマシンを選択する

- 「スロットマシンiの値が最も高い確率」は直接評価が困難：

$$\Pr\left[P(\theta_i \mid n_i, h_i) > P\left(\theta_j \mid n_j, h_j\right), \forall j \neq i\right]$$

- トンプソン抽出：サンプリングで解決する

1. 各マシンの当たり確率の事後分布に基づいて当たり確率$\{\theta_i\}_{i=1,\dots,m}$をサンプリングする
2. θ_iが一番大きいスロットマシンを選ぶ
   ※ここまでがマシンの決定に関する部分
3. 結果に基づきθ_iの事後分布を更新
   ※n_i, h_iが更新されることに伴って事後分布が変わり次の決定に影響する

まとめ：
ベイズモデリング

- ベイズ統計
  - 事前分布＋データ（証拠）＝事後分布
- ベイズモデリング
  - ディリクレ分布を共役事前分布とした離散分布のベイズ推定
  - 階層ベイズモデリングによるモデルの抽象化
  - 経験ベイズ法による超パラメータの推定
- 広告配信最適化問題への応用：
  - バンディット問題としての定式化
  - ベイズ事後分布によって、当たり確率の推定のばらつきを考慮
  - トンプソン抽出による、情報の獲得と利用をバランスした、効率的な決定