KYOTO UNIVERSITY

統計的モデリング基礎⑪
～ベイズモデリング（つづき）＋α～

鹿島久嗣
（情報学科 計算機科学コース）

DEPARTMENT OF INTELLIGENCE SCIENCE AND TECHNOLOGY

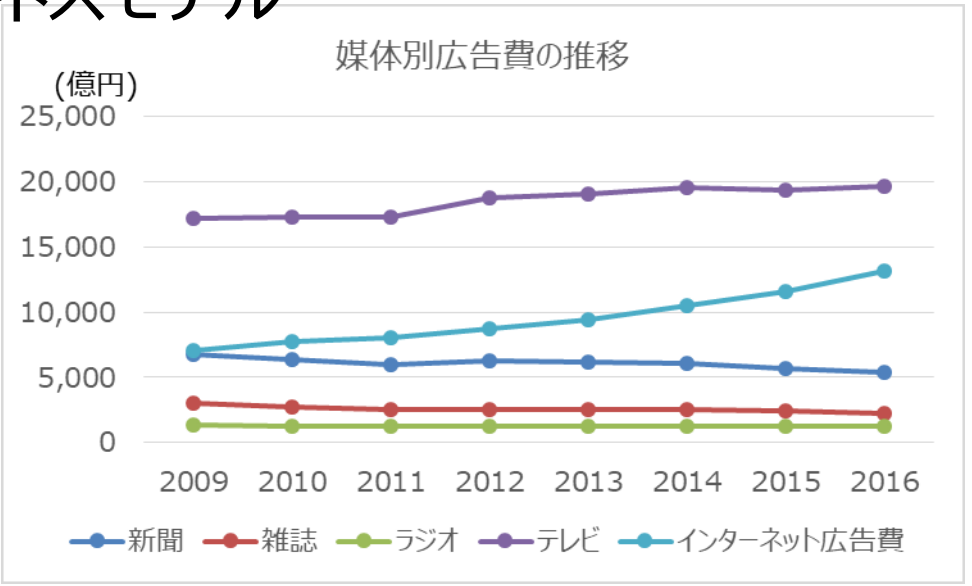
目次：
ベイズモデリング（つづき）と他の話題（異常検知）

- ベイズモデリングの応用例：
  - 広告配信最適化問題
  - バンディット問題
  - トンプソン抽出
- 異常検知

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（← コレを考えてみる）
  - 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
- 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

- 最尤推定では$\hat{\theta}_i = \frac{h_i}{n_i}$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？
- 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

?

$=$

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

?

$>$

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

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
  - θ_iの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：

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

事前分布

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

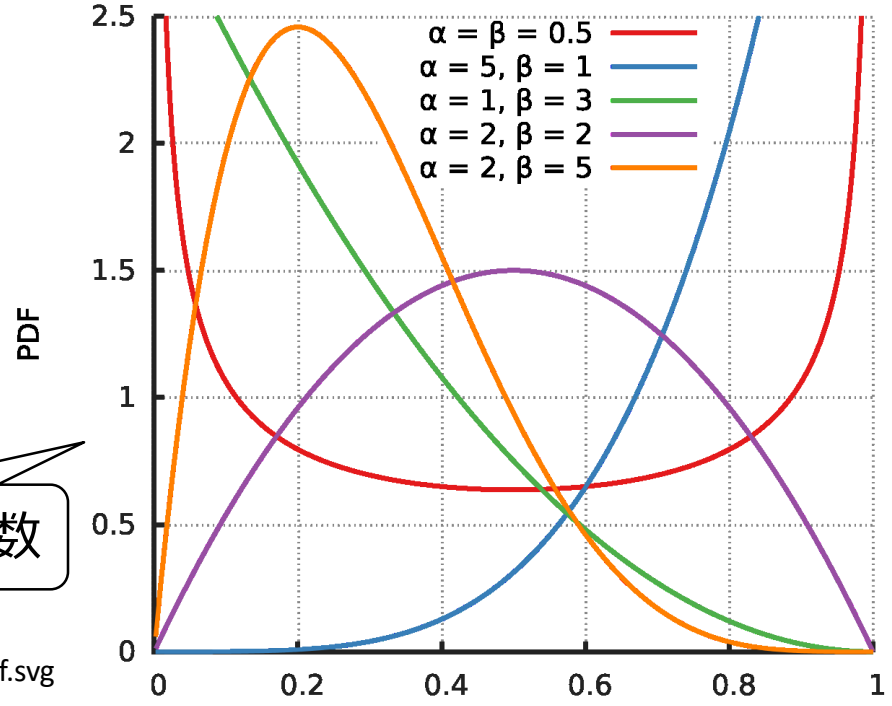
- 事前分布をベータ分布とする：

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

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）



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

- 事後分布はベータ分布：

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

- 導出：

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

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

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

トンプソン抽出：
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の当たり確率が最も高い確率」をもとめ、これに従ってマシンを選択する
- 「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：

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

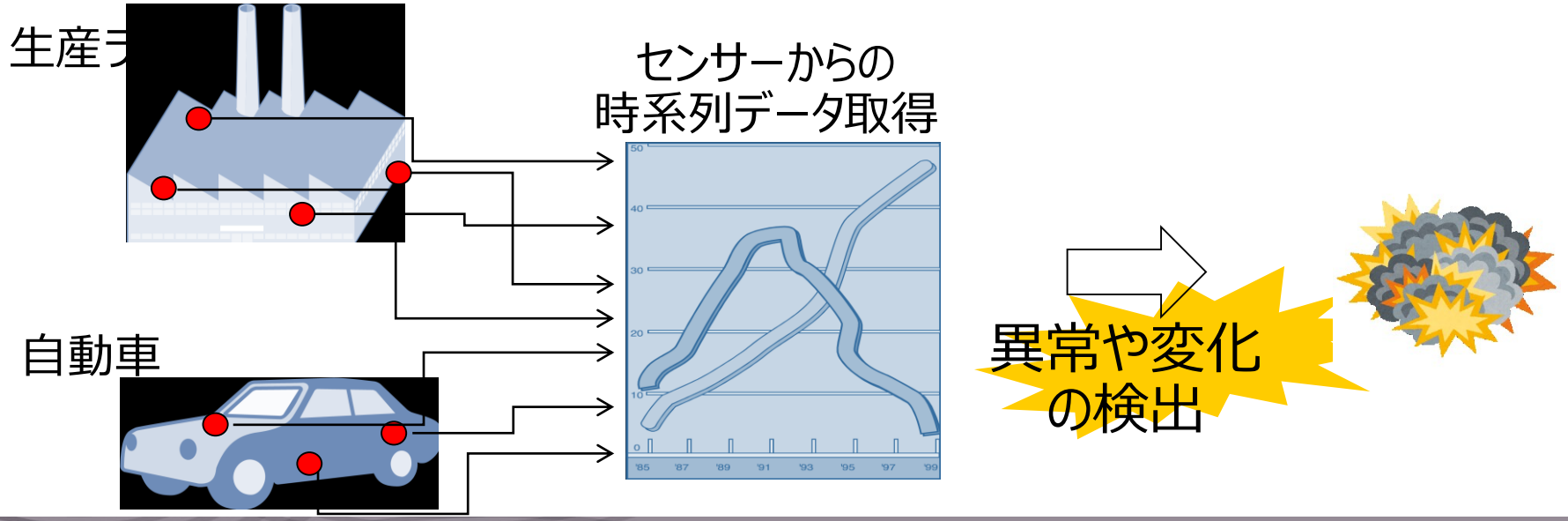
- トンプソン抽出：サンプリングで解決する

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$\{\theta_i\}_{i=1,\ldots,m}$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
3. 結果に基づきθ_iの事後分布を更新

異常検知

異常検知：
システムの問題を事前に察知し対策することでコスト削減

- 大規模で複雑なシステムの停止は一旦起こると大損失：
  - 生産ラインの故障、コンピュータのウィルス感染、不正侵入
- 対象システムに設置されたセンサーからの取得データを分析し、異常を早期検知したい



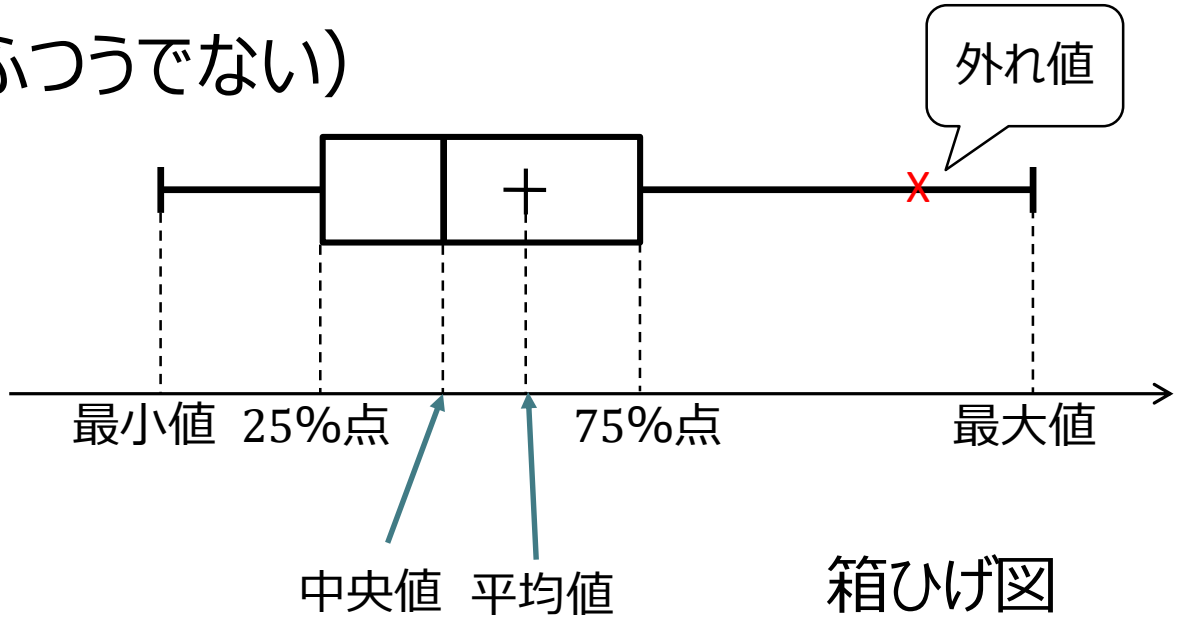
異常検知技術：
「普通でない」データのパターンをみつけだす技術

- 異常： データの中に含まれる「普通でない」パターン
  - 対象のシステムのなんらかの異常を原因として表れる
    - クレジットカード不正使用・システム侵入・テロ・システムダウン…
  - あるいは対象システムの状態変化
    - 新規ニューストピック出現・システムの設定変更・環境変化…
- これらをデータの中から発見し報告するのが異常検知

異常検知の基本的な考え方：稀な値を異常と考える

- 単純な場合として1変数（例えば温度）を対象とした異常検知を考える
- 通常の値の範囲をとらえる（温度は通常20～50°Cの間）
- そこから逸脱した値を検出し、異常として報告する

（80°Cはふつうでない）



箱ひげ図

統計的な異常検知：
モデルからの生成確率が小さいデータを異常と考える

- 正常なデータ$x_1, x_2, \ldots, x_n$が与えられる
- 正常なデータからモデル$p(x)$を推定する
  - たとえば最尤推定によって（正常時モデルの）パラメータ推定値を得る
- 検証対象のデータx^{NEW}に対して、モデルが与える確率$p\left(x^{\mathrm{NEW}}\right)$を算出する
- これがある閾値τより低ければ（$p\left(x^{\mathrm{NEW}}\right) < \tau$）異常として報告する