

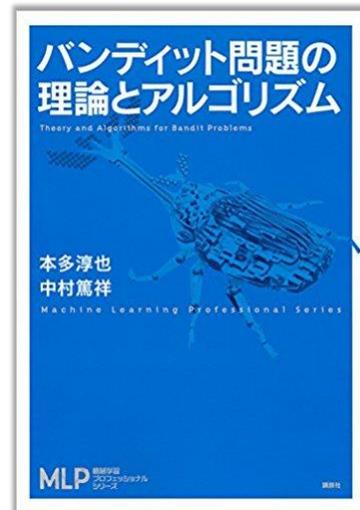
統計的モデリング基礎⑪ ～ベイズモデリング（つづき）～

鹿島久嗣
(情報学科 計算機科学コース)

目次：

ベイズモデリング（つづき）

- 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
- 補遺：
 - 統計的モデリングふりかえり
 - 異常検知
 - その他のトピック



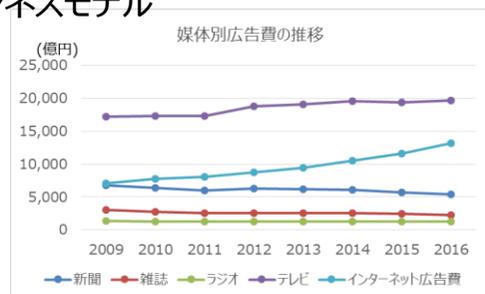
ベイズモデリングの応用

3

KYOTO UNIVERSITY

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

4

KYOTO UNIVERSITY

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

5

KYOTO UNIVERSITY

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



6

KYOTO UNIVERSITY

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

7

KYOTO UNIVERSITY

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

...



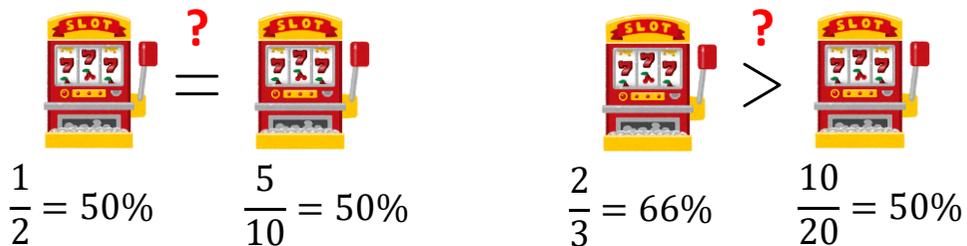
スロットマシン m
当たり確率 θ_m

8

KYOTO UNIVERSITY

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



9

KYOTO UNIVERSITY

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 $P(\theta_i | n_i, h_i)$
 - 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
- 例によってベイズの定理を用いると：

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

10

KYOTO UNIVERSITY

当たり確率の事後分布：
事前分布と事後分布がともにベータ分布

ディリクレ分布で
 $k = 2$ の場合に相当

- 事前分布をベータ分布とする：

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

- 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)

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

- $$P(\theta_i | n_i, h_i) \propto P(n_i, h_i | \theta_i) P(\theta_i)$$

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

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

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：

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

- トンプソン抽出：サンプリングで解決する

- 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
- θ_i が一番大きいスロットマシンを選ぶ
- 結果に基づき θ_i の事後分布を更新