Pythonで体験するベイズ推論

第6章　事前分布をハッキリさせよう

石田研究室M1 種部俊孝

2017/12/12 (火)

目次

目次

はじめに

本章の内容は、ベイズ推論でいつも激しい議論になるトピック

< 事前分布をどのように選ぶのか？>

本書ではこれまでずっと、事前分布の選び方は気にしてこなかった

しかし、客観性を確保したいとき、特に慎重に選ぶ必要がある

目次

主観的な事前分布と客観的な事前分布

客観的な事前分布

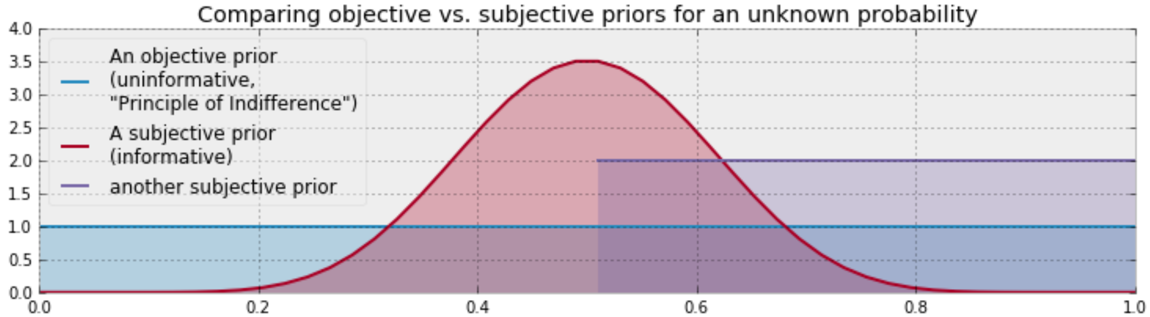
データが事後分布に最も大きな影響を与えることを許す分布

例）未知数が取りうる範囲全体をカバーする一様分布

〇 Uniform(0,1) ← 全ての取りうる値を考慮している
× Uniform(0.5,1) ← 事前知識を使って範囲を限定している

主観的な事前分布

実際に使う人が個人的な意見を事前分布に反映させる



意思決定につぐ意思決定

- 事前分布を客観的にすべきか主観的にすべきかは、解く問題に依存する
  → 科学研究の場合は、結果にバイアスが入らない客観的な事前分布が良い

- 範囲の広い一様分布は客観的な事前分布として適切な場合が多い
  ただし、安易に一様分布を使うと、おかしな結果を招くこともあるので注意が必要

- 主観的でも客観的でも、事前分布の選択はモデリング過程の一部である
  → もし、事後分布が妥当でないと思うなら、今の事前分布を変更するべき

経験ベイズ

ほとんどの推論問題には、ベイズ主義の手法と頻度主義の手法がある

ベイズ手法 → 事前分布を持ち、その事前分布はハイパーパラメータを持つ
頻度主義手法 → 事前分布はない

⬇

経験ベイズは、頻度主義とベイズ推論を組み合わせる

→ ハイパーパラメータの選択には頻度主義の手法を用いて、
ベイズ主義の推論手法を元の問題に適用する

経験ベイズの例

例）σ=5 の正規分布のパラメータ μ を推定したい

μ が取りうる範囲は全ての実数なので、μ の事前分布に正規分布を使用

次に、その事前正規分布のハイパーパラメータ (μ_p, σ_p^2) を決めたい

→ σ_p^2には問題に応じた不確実さを設定する

→ μ_pの選び方には以下の二つの方法がある

1. 経験ベイズでは、経験平均を使う $\mu_p = \frac{1}{N}\sum_{i=0}^{N} X_i$
2. 普通のベイズ推論では、事前知識を使うか、もっと客観的な事前分布を使う

客観的なベイズ推論に対して、経験ベイズは半客観的と言われる

経験ベイズの欠点

経験ベイズは観測データを二重にカウントしている

-> データを事前分布と、MCMCでの推論に用いている

⬇

観測データ数が少ない場合、事前分布が結果に影響を与え過ぎてしまう

-> 観測データ数が非常に大きい場合のみ経験ベイズを使った方が良い

経験ベイズはベイズ推論の思想的な部分に抵触する

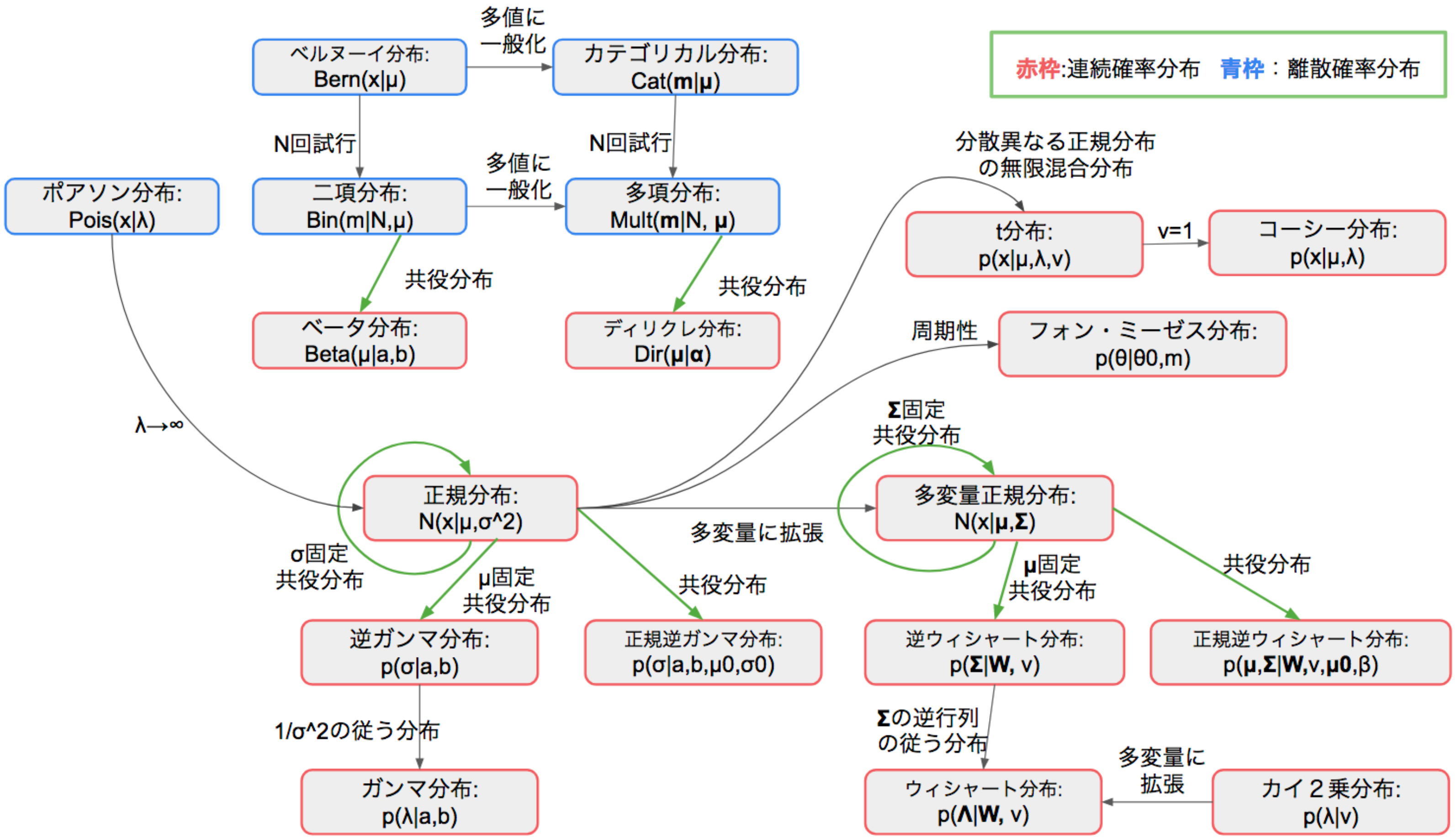
ベイズ推論:　事前分布 ⇨ 観測データ ⇨ 事後分布

経験ベイズ:　観測データ ⇨ 事前分布 ⇨ 観測データ ⇨ 事後分布

理想的には、データを観測する前に全ての事前分布を決定しておくべき

目次

知ってておくべき事前分布



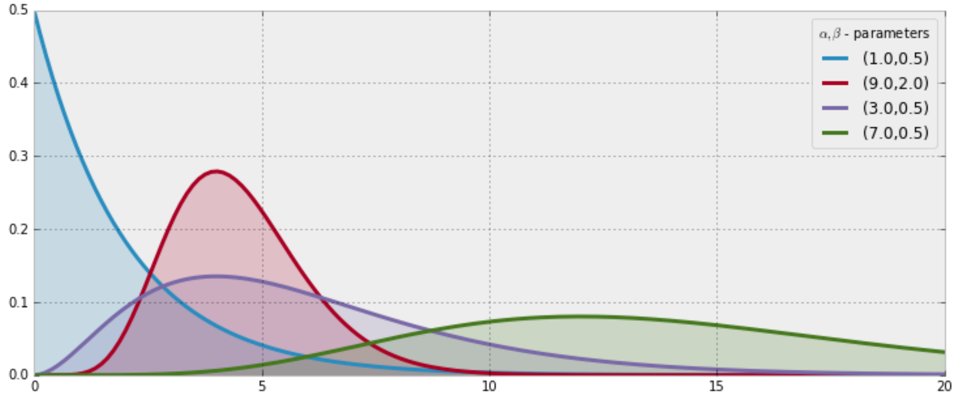
ガンマ分布

ガンマ分布に従う確率変数$X \sim Gamma(\alpha, \beta)$は任意の実数値をとる

ガンマ分布は指数分布の一般系である　　$Exp(\beta) = Gamma(1, \beta)$

パラメータが一つ追加されているため、確率密度関数の自由度が大きくなり、
ユーザが持つ主観的な事前分布の形状をより正確に反映することができる

$Gamma(\alpha, \beta)$の密度関数　　$f(x|\alpha, \beta) = \dfrac{\beta^{\alpha} x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$



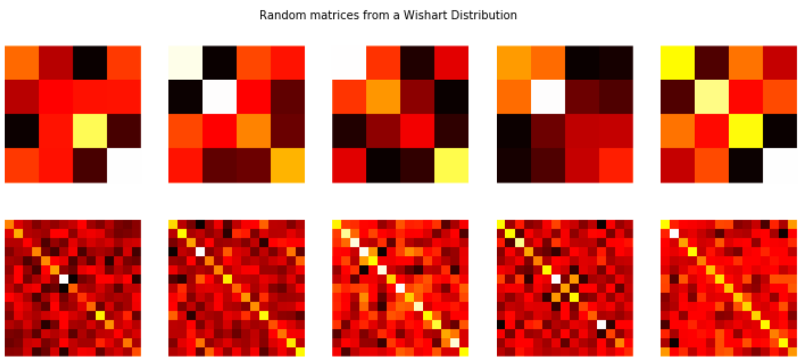
いくつかの異なる
αとβに対するガンマ分布

ウィシャート分布

これまで確率変数は全てスカラーだったが、行列を確率変数としても良い

半正定値行列の分布がウィシャート分布

共分散行列が正定値のため、ウィシャート分布は共分散行列の事前分布に適する



ウィシャート分布は可視化できないため、上図はこの分布からサンプリングした図

共分散行列が対称行列なため、上図の行列も対称となっている

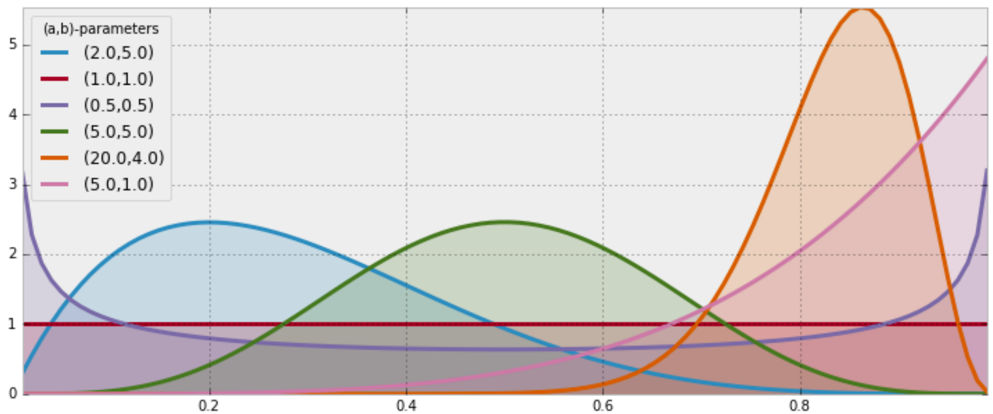
ベータ分布

確率変数Xがパラメータ(α,β)のベータ分布に従うとき、確率密度関数は以下の式

$$f_X(x|\alpha,\beta) = \frac{x^{(\alpha-1)}(1-x)^{(\beta-1)}}{B(\alpha,\beta)} \qquad (B\text{はベータ関数})$$

ベータ関数に従う確率変数は0から1までの実数値を取るため、モデリングに適する

ベータ分布は一様分布の一般化である



いくつかの異なる
αとβに対するベータ分布

パラメータが(1,1)の
場合に一様分布となる

目次

6.1 はじめに

6.2 主観的な事前分布と客観的な事前分布

6.3 知っておくべき事前分布

6.4 例題：ベイズ多腕バンディット

例題：ベイズ多腕バンディット

設定

・カジノで10台のスロットマシン（多腕バンディット）の前にやってきた

・アーム（レバー）を引いたら当たりが出る確率は、バンディットによって違う
（どのバンディットも当たりの賞金は同じで、確率だけが違うと仮定）

・一回につき一つのバンディットを選んでアームを引く

目標

自分の勝ち分を最大化する戦略を立てる

→ 最も良いバンディットがどれなのかを、できるだけ早く見抜く

多腕バンディット問題の応用

ネット広告

企業は、訪問者に見せる広告のパターンをいくつか持っている
しかし、どの広告戦略が売り上げを最大化するのかはわからない

生態学

動物は限られたエネルギーで行動しなければならないが、どの行動が有益かはわからない
動物たちはどうやって最適正を定義し、最大化しているのだろう

金融

リターン特性が時間的に変化する状況で、どのストックオプションのリターンが最も高いだろうか？

臨床試験

研究者たちは最も効果の高い治療法を見つけたいが、多数の治療を行うため、失敗は最小限に抑えたい

心理学

罰則と報酬は行動に影響をどのように与えるのか？
また、それを人間はどのように学習しているのか？

ベイズバンディットによる解法

最適解を得るのは難しく、何十年にもわたって手法が開発されている

ここでは、ベイズバンディットによる解法を示す

ベイズバンディットのアルゴリズム

ベイズバンディットでは、各バンディットで勝つ確率の事前分布を仮定することから始める
この例題では、その確率が全く分からないとして、0から1の一様分布を適用

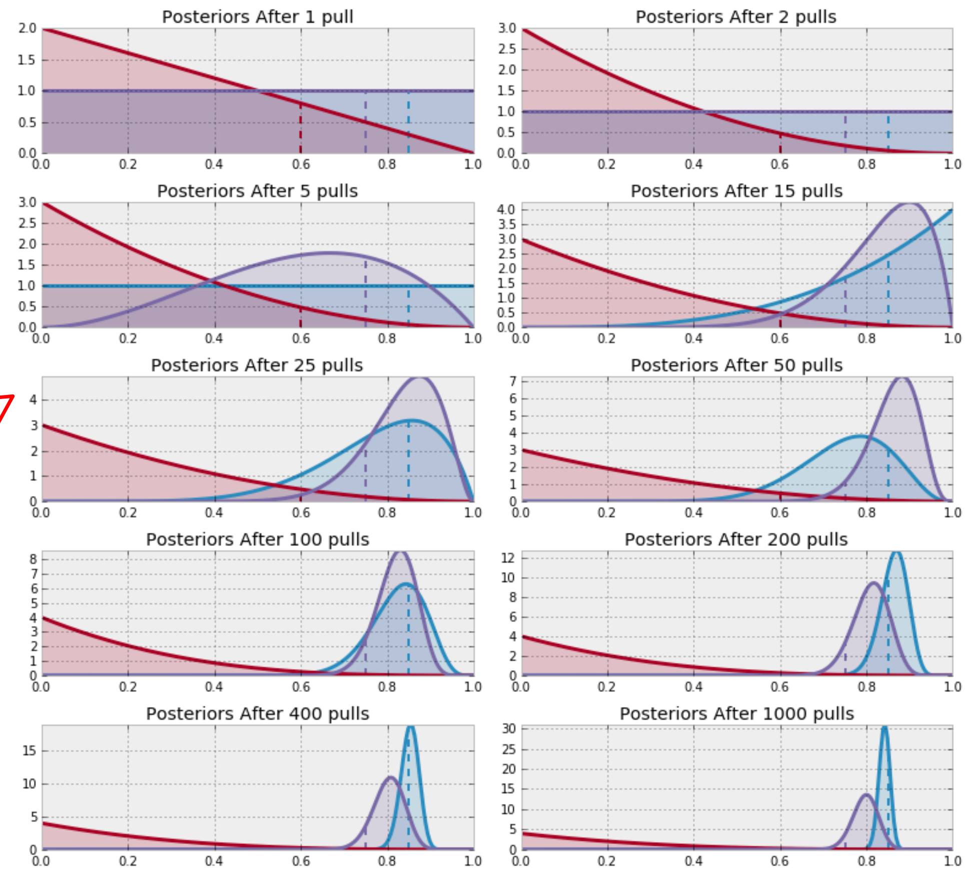
1. 各バンディットbの事前分布からランダムな結果X_bをサンプリングする
2. 最も高い値のサンプルを出したバンディットBを選択する $(B = \arg\max X_b)$
3. バンディットBの結果を観測し、バンディットBの事前分布を更新する
4. 1に戻る

ベイズバンディットによる解法

負けを出したバンディットを選択する確率を残しておかなければならない

→ 負けを出したものが今後Bになる可能性はいつまでもゼロにならない

（繰り返すことでその可能性は減る）



各バンディットの事後分布の進行状況

初期の事前分布 $Beta(\alpha=1,\beta=1)$

事後分布 (観測結果X)

$Beta(\alpha=1+X,\beta=1+1-X)$

観測した当たりの割合が、真の確率からどの程度離れているのかが評価指標

良さを測る

戦略がどれだけうまく機能しているかを測る指標が必要

ベストな戦略: 勝ちの確率が最も大きいものだけを引き続ける

→ 最良のバンディット(確率w_{opt})を引き続けた場合と比較してどれだけ良いか？

戦略の全リグレットを定義

=最良の戦略をT回引いた場合のリターンと別の戦略でT回引いた場合のリターンとの差

$$R_T = \sum_{i=1}^{T}\left(w_{opt} - w_{B(i)}\right) = Tw_{opt} - \sum_{i=1}^{T} w_{B(i)}$$

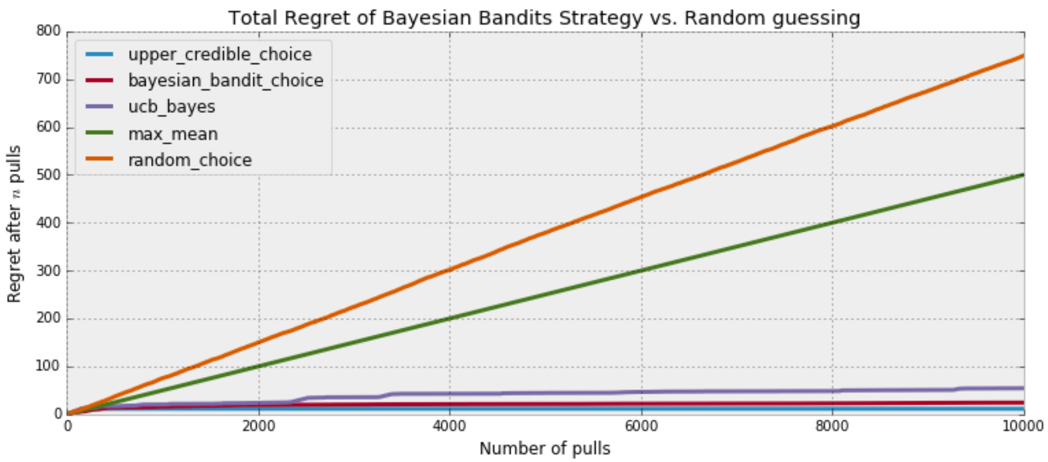
$w_{B(i)}$はi回目に選ばれたバンディットの当たりの確率

理想的には最良のバンディットを学習($w_{B(i)} = w_{opt}$)すれば、戦略の全リグレットは小さくなる

良さを測る

以下の戦略の全リグレットを調べる

1. ランダム : ランダムに選択したバンディットのアームを引く
2. 最大ベイズ信用区間上限 : 推定された確率の95%信用区間の上限が最も大きいバンディットを引く
3. ベイズ-UCBアルゴリズム :「スコア」が最も大きいバンディットを引く
   ここでスコアとは、事後分布の分位点を動的に更新したもの
4. 事後分布平均 : 事後分布の平均が最も大きいバンディットを引く
5. 最大観測割合 : 現在までに観測された、当たりの割合が最も大きいバンディットを引く



上から
ランダム
最大観測割合
ベイズ-UCBアルゴリズム
ベイズバンディット戦略
最大ベイズ信用区間上限

良さを測る

1回のシミュレーションの結果では偶然に運が良かった可能性もある

期待リグレットを定義

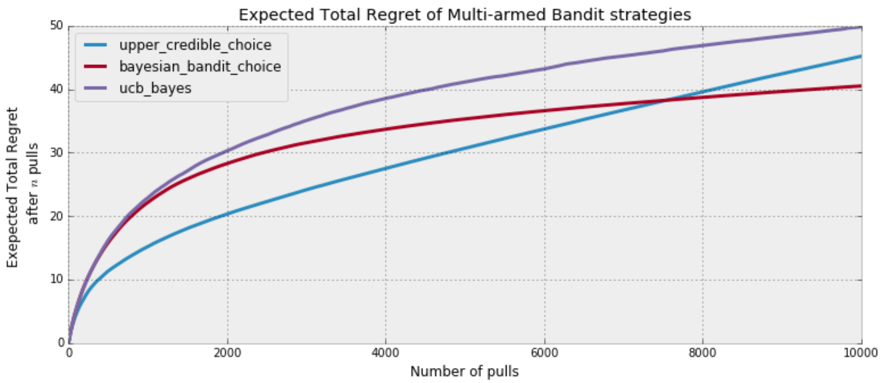
＝ 可能な全てのシナリオの全てのリグレットの期待値で評価

$$\bar{R}_T = E[R_T]$$

どのような準最適な戦略でも、期待リグレットは対数的に増加することが示されている

$$E[R_T] = \Omega(\log(T))$$

対数的に増加するリグレットを持つ戦略は、どれも多腕バンディット問題が解けると言える

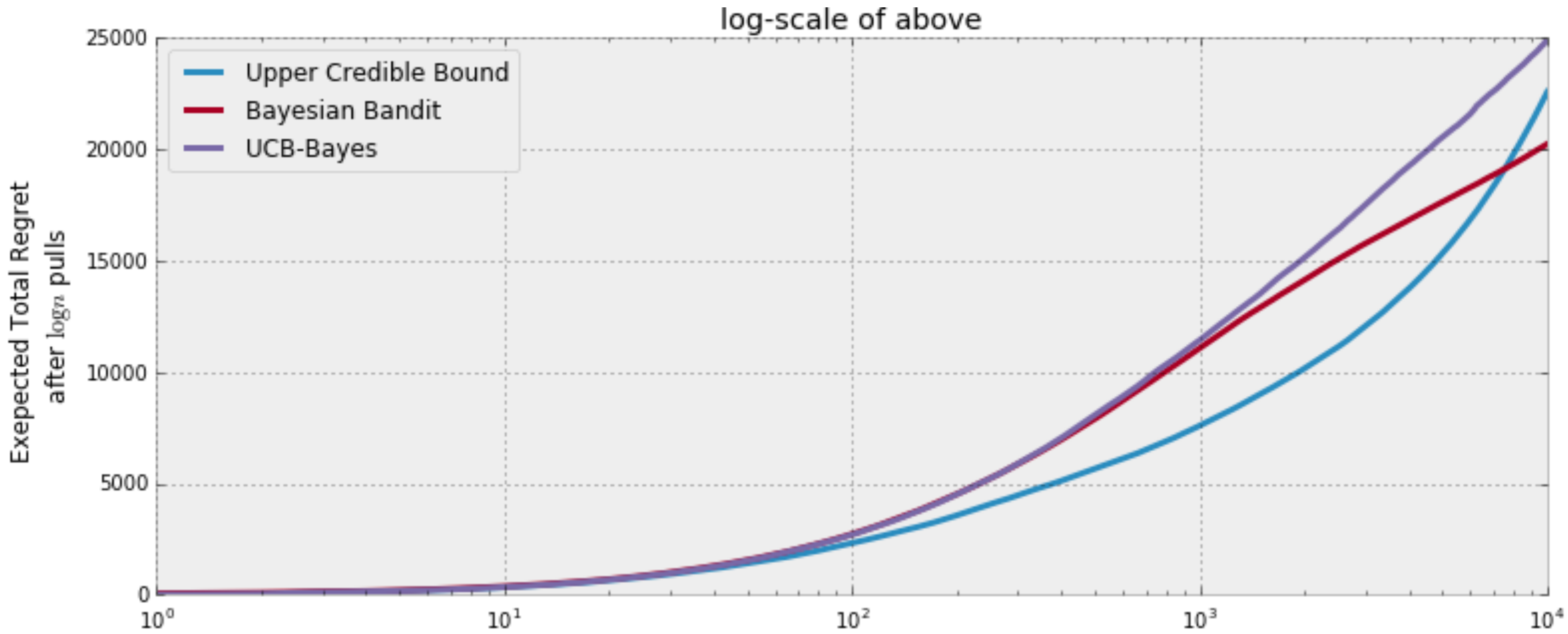


大数の法則を使うと、同じ実験を何回も実行することで、ベイズバンディット戦略の期待リグレットを近似することができる

上から、最大ベイズ信用区間上限、
ベイズ-UCBアルゴリズム、
ベイズバンディット戦略

良さを測る

先ほどの図を大数スケールで示したものが以下の図



アルゴリズムの拡張

ベイズバンディットアルゴリズムは単純なため、拡張も簡単である

・もし「最小」確率のバンディットを選びたいなら、単純に$B = \arg\min X_b$を選択する

・学習率の追加
時間経過によって環境が変化すると仮定する
標準的なベイズバンディットならば、自分自身で行動を更新する
このアルゴリズムを拡張して、更新時に学習率をかけた項を加えることで、
変化する環境にもっと速く追従するように学習することができる

・階層的アルゴリズム
小さなバンディットアルゴリズムの上に、さらにバンディットアルゴリズムを積み重ねられる

それぞれが少しずつ違うN個のベイズバンディットモデルがあるとする
これらN個の下位ベイズバンディットの上に、別のベイズバンディットを作る
上位バンディットは下位バンディットを一つ選択し、選択された下位バンディットはどのバンディットを
引くのかをその内部で決定する
上位バンディットモデルは下位バンディットモデルが正しかったかどうかによって、自分自身を更新する

アルゴリズムの拡張

- バンディットAの当たりの賞金y_aを、確率分布$f_{y_a}(y)$に従う確率変数に拡張する
  この問題は「最大期待値を持つバンディットを見つける」と言い換えられる

これまでの例は、f_{y_a}は確率p_aのベルヌーイ分布に従う確率変数で、
そのバンディットの期待値はp_aに等しかった

そのため、勝つ確率を最大化しようとしているように見えた

もし当たった結果の確率変数がベルヌーイ分布に従わず、非負なら、分布を平行移動するだけで達成できる

アルゴリズムは以前と変わらず以下のようである

1. 各バンディットbの事前分布からランダムな結果X_bをサンプリングする
2. 最も高い値のサンプルを出したバンディットBを選択する ($B = \arg\max X_b$)
3. バンディットBを引いた結果$R \sim f_{yb}$を観測し、バンディットBの事前分布を更新する
4. 1に戻る

問題は、X_bをサンプリングするステップ
→ fは任意の分布であり、事後分布がどのような分布になるのかわからず、サンプリングが難しいこともある

アルゴリズムの拡張

ベイズバンディットアルゴリズムはコメントシステムにも拡張されている

第4章の問題

全投票数に対するupvote数の割合のベイズ下限をもとにしたランキングアルゴリズム
→トップのコメントが古いコメントに偏る
→ 本当は良いかもしれないコメントが、新しいというだけで下の方に表示されてしまう

ベイズバンディットアルゴリズムを用いた改善

それぞれのコメントをバンディットとみなす

選択する回数は投票された回数に等しく、当たりの回数はupvoteの数に等しいとする
→ この事後分布は$Beta(1+U, 1+D)$になる

ユーザがページを訪問すると、サンプルを各バンディット（コメント）からサンプリングする

コメントはそれが持つサンプル値に応じてランキングされる

アルゴリズムの拡張

試しに、ベイズバンディットアルゴリズムで35種類の異なるバンディットを学習してみた結果が以下のようである