Pythonで体験するベイズ推論

2017/10/10

秋山研 M1　林 孝紀

この本でやること

1. ベイズ推論ってなに？
2. Pymcの使い方
3. MCMCの中でなにが行われている？
4. 大数の法則とサンプル数が小さいデータ
5. 損失の定義
6. 事前分布，共役事前分布
7. A/Bテスト

目次

1. 頻度主義とベイズ主義
2. データを用いたベイズ推定（例題）

今回のコード

https://github.com/thtitech/Bayes-Python.git

ベイズの定理

ベイズの定理：

$$P(A \mid X) = \frac{P(X \mid A)\, P(A)}{P(X)}$$

Where: データ: X
事象: A

$P(A)$: 事象Aが成立する確率
$P(X)$: データXが取れる確率
$P(A \mid X)$: データXが与えられた時事象Aが起こる確率
$P(X \mid A)$: 事象Aが成立する時データXが取れる確率

$P(X)$は事象Aによらないため

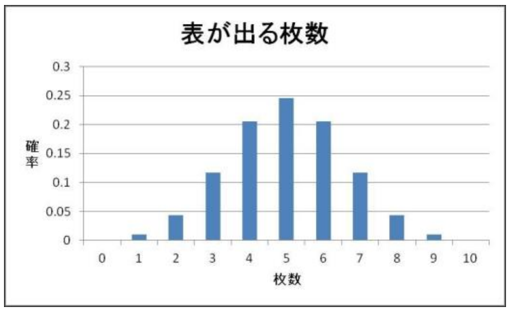
$$P(A \mid X) \propto P(X \mid A)\, P(A)$$

事後確率 事前確率

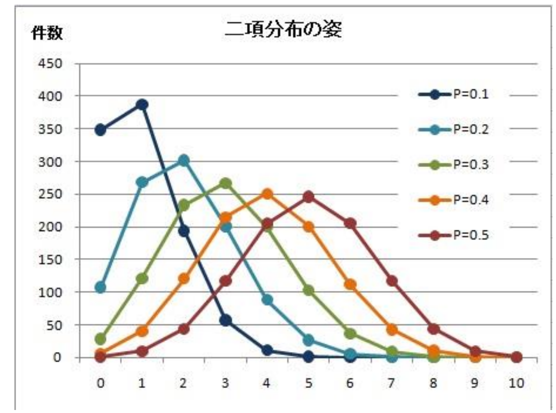
頻度主義とベイズ主義

コイン投げの場合

頻度主義： 表がでる真の確率pがあるはず！！

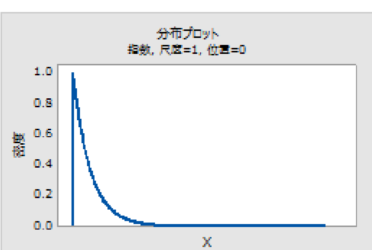


二項分布にフィッティング
（最尤推定）

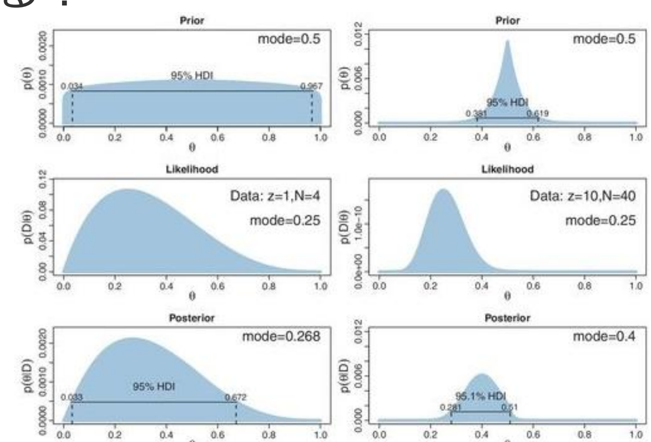


ベイズ主義： 表がでる確率pは確率的に推定できる！

表は出にくい



実験結果から
事後確率を推定

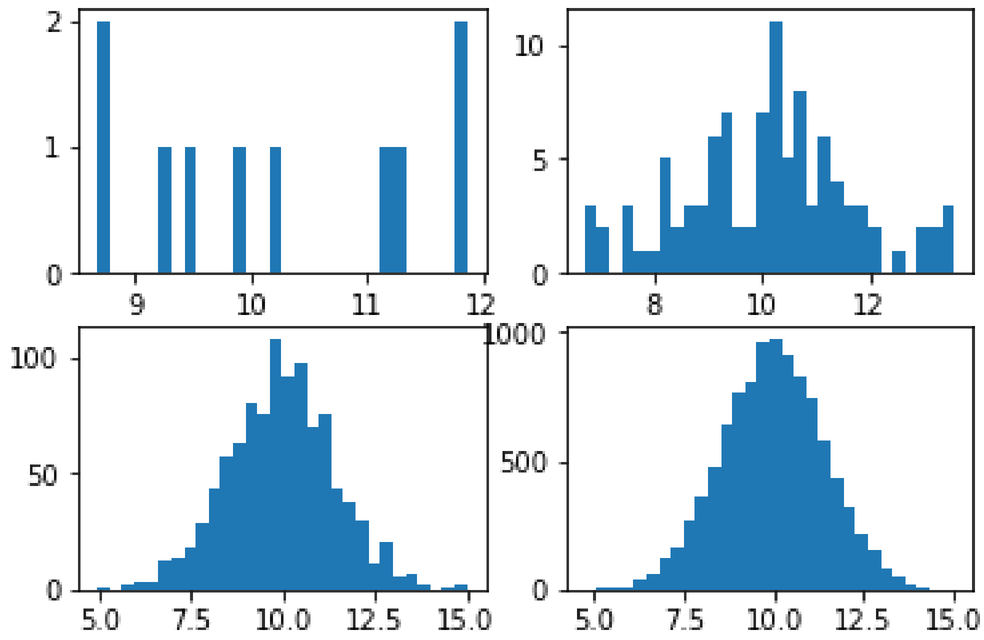


事前分布

事後分布

実際に比べてみた（データセット）

平均10.0, 分散1.0の正規分布から10, 100, 1000, 10000サンプリング



最尤推定（頻度主義）

n個の独立な変数$\{x_0, x_1 \ldots x_{n-1}\}$があった時の正規分布の最尤推定

平均μ, 分散σ^2の正規分布：　$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-(x - \mu)/2\sigma^2)$

尤度：もっともらしさ

確率密度関数$f(x)$に対して尤度$L(\theta)$は以下のように定義される

$$L(\mu, \sigma) = \prod f(x_i) \quad \Rightarrow \quad logL(\mu, \sigma) = \sum \log(f(x_i))$$

かけ算はめんどくさい

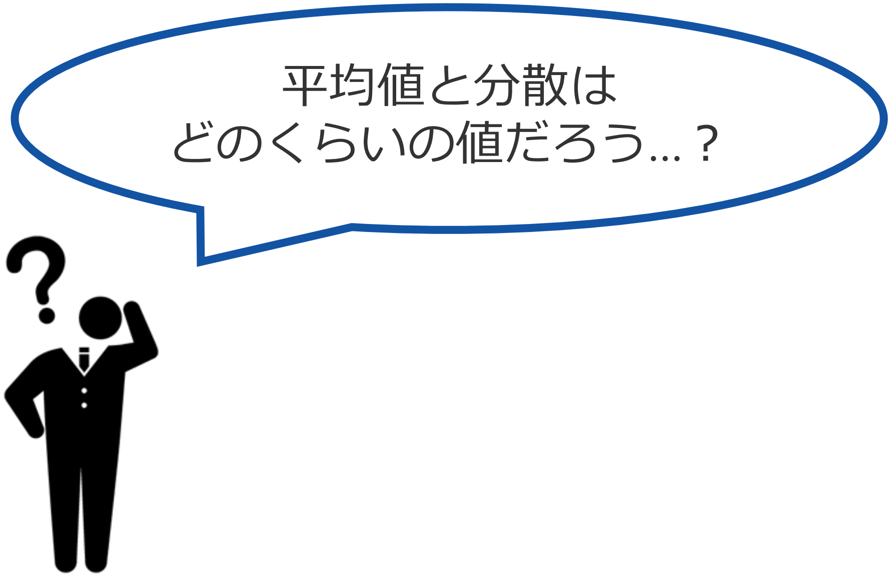
対数尤度を最大化したい！！

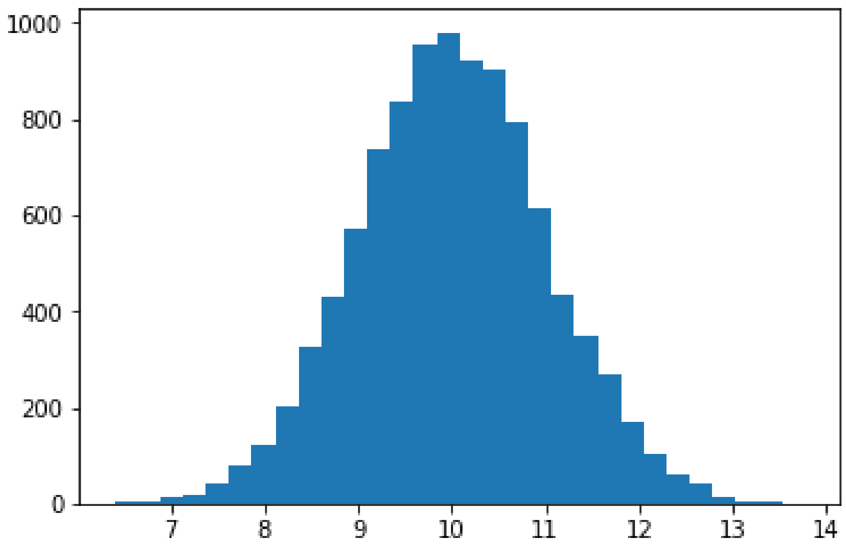
最尤推定（手動）

実際にやってみる（頻度主義）

サンプル数	μ	σ^2	95%信頼区間
10	9.654	0.379	[9.272, 10.036]
100	9.911	0.810	[9.735, 10.088]
1000	10.004	1.006	[9.942, 10.066]
10000	10.009	0.983	[9.990, 10.029]

事前分布の設定（ベイズ主義）





$$\mu \sim Uniform(lower = -100, upper = 100)$$
$$\tau \sim Uniform(lower = 0, upper = 100)$$

$\tau = \frac{1}{\sigma^2}$ は正の値なことに注意

＊実際は主観的な分布を採用することもできる

サンプリング（ベイズ主義）

MCMC (Markov chain Monte Carlo methods):

確率分布**サンプリング**の手法で次のサンプリングする点が
今の点のみから決定される性質をもつ

スタート地点
(ex. $\mu = 0, \tau = 0$
や MAP推定量)
+
事前分布

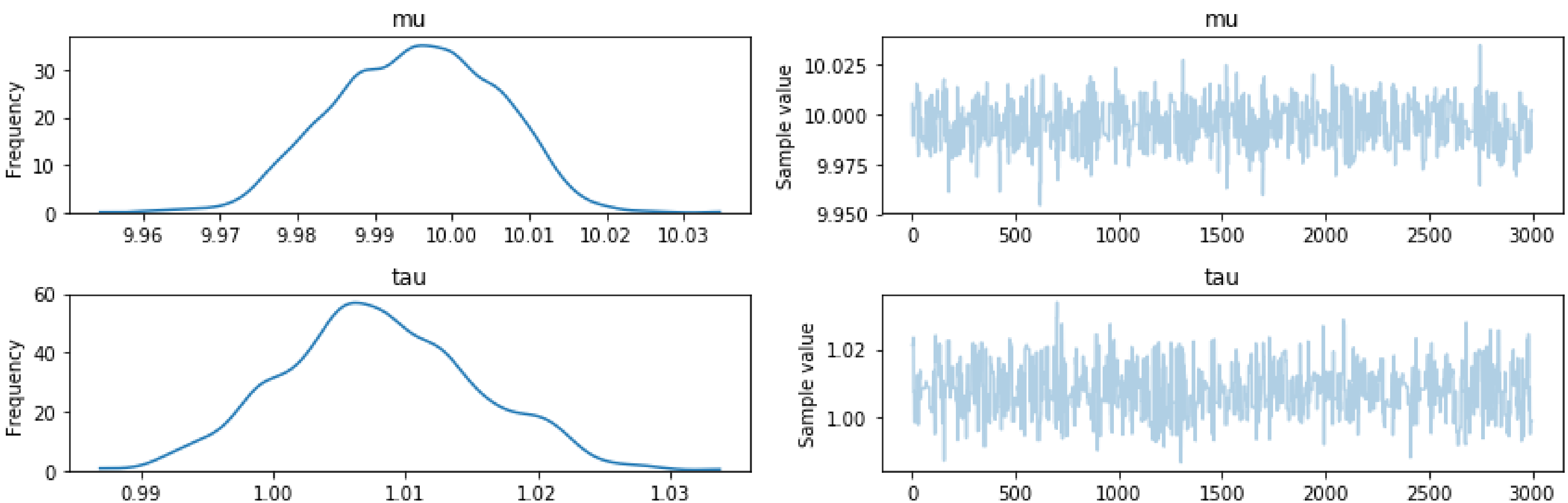
MCMCでサンプリング
(ex. ギブスサンプリング,
メトロポリス法)

（尤度）*（事前分布）
を小さくする

事後分布
（パラメータの分布）

＊この過程で最初の方のサンプルを捨てる,
何個かおきに記録するなどのことをする

実際にやってみる（ベイズ主義）

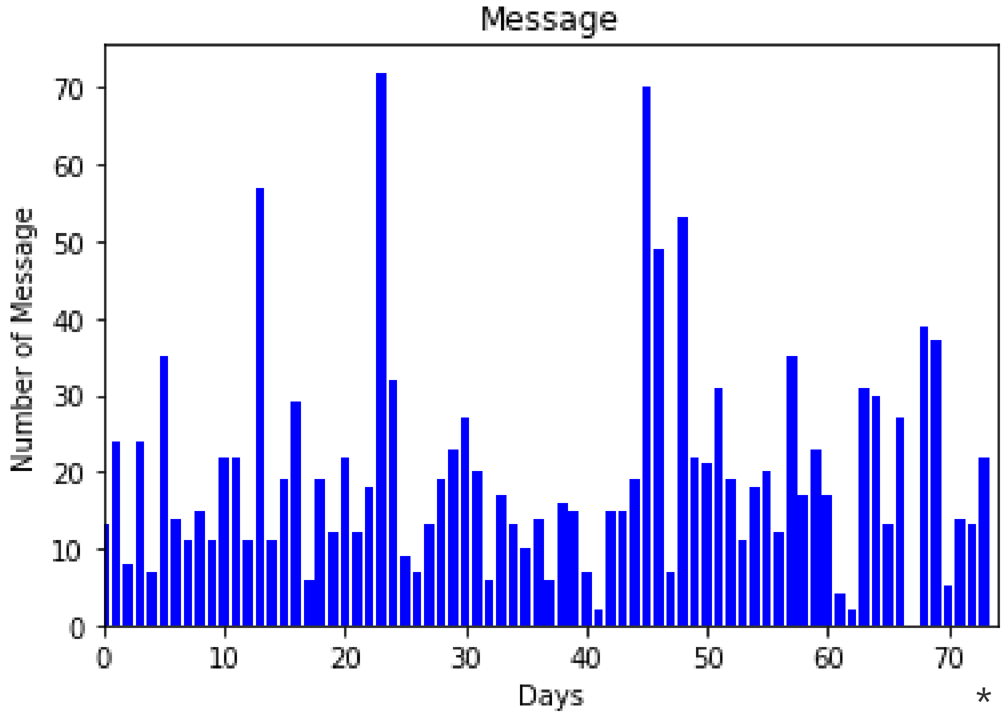


＊サンプリング数10000のデータの結果

目次

メール受信数の推定

ある日から観測したメールの受信数



* https://git.io/vXTVC

仮説：ある日をさかいにメールの受信数が増えているのではないか？

モデリング（事前分布の設定）

Step 1.　モデリングする分布の設定

メールの受信数は自然数を取るため，ポアソン分布が妥当

$$f(X=k)=\frac{\lambda^k e^{-\lambda}}{k!}, E[X]=V[X]=\lambda$$

Step 2.　事前分布の設定

- 受信メール数が増えた日 τ
  $\tau \sim$
- τ日までのポアソン分布の母数λ_1
  $\lambda_1 \sim$
- τ日以降のポアソン分布の母数λ_2
  $\lambda_2 \sim$

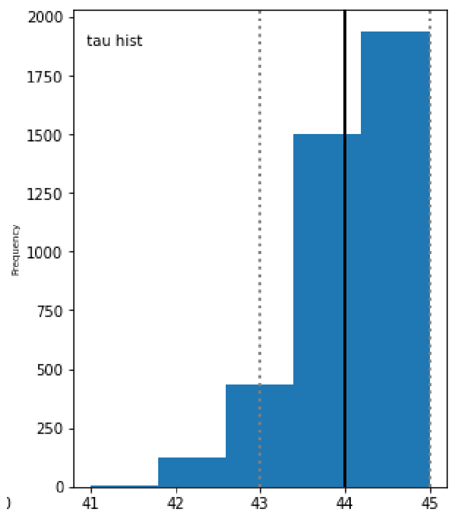
モデリング（コードに反映）

ここに事前分布を定義

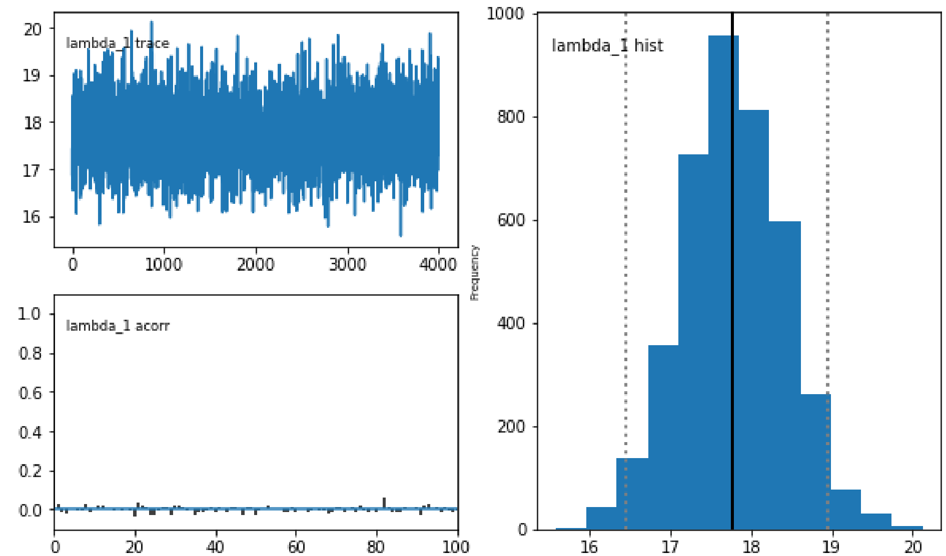
```
alpha = 1 / data.mean()

# 事前分布は指数分布
#lambda_1 = pm.Exponential("lambda_1", alpha)
#lambda_2 = pm.Exponential("lambda_2", alpha)

# 事前分布を一様分布にしてみる
lambda_1 = pm.Uniform("lambda_1", upper = (data.mean() * 2), lower = 0)
lambda_2 = pm.Uniform("lambda_2", upper = (data.mean() * 2), lower = 0)

# 変化した日にちtau
tau = pm.DiscreteUniform("tau", lower = 0, upper = N)

# ポアソン分布のパラメータ
# tau日より前はlambda_1, あとはlambda_2
lam = pm.Lambda("lam", lambda tau = tau, lambda_1 = lambda_1,
                lambda_2 = lambda_2: np.array([(lambda_1 if (i < tau) else lambda_2)for i in range(N)]))
# 観測結果の指定
observe = pm.Poisson("obs", lam, value = data, observed = True)

model = pm.Model([observe, lambda_1, lambda_2, tau])

mcmc = pm.MCMC()
# 50000回サンプリングして10000個捨てる, 10個づつ記録する
mcmc.sample(50000, 10000, thin=10)
```

モデリング（結果）

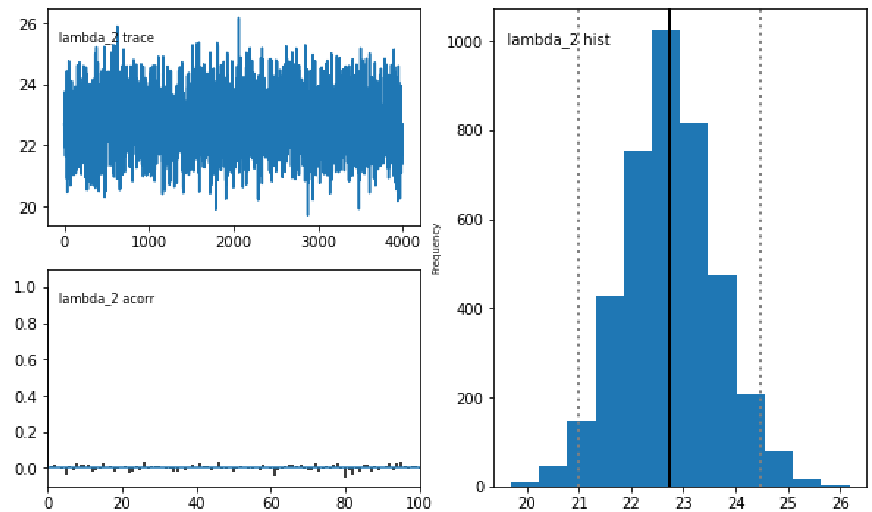
τ



λ_1



λ_2



考察

- λ_1, λ_2 はそれぞれ18, 23くらいにありそう
- τ は45または44になりそう

次回は

1. サンプリング結果から$\lambda_1 < \lambda_2$ となっている割合を出す
2. 今回の変化点が2箇所あると仮定してモデリング
3. メッセージ受信増加率の期待値を求める
4. 色々な事前分布を試してみる