ベイズ統計学入門

梅崎直也

2020 年 10 月 9 日

目次

第 1 章

ベイズモデリングの考え方

1.1 はじめに

統計モデリングとベイズ推論を行う上で、まずは以下の三つの概念について理解したい。

1. モデル
2. パラメータ
3. データ

与えられたデータを説明するためのモデルを立て、与えられたデータを用いてパラメータを推定し、未知のデータを予測したり現象の起こる原理を説明する。これが基本的な流れである。

データはある種のランダムさを持って得られたものであると考える。データの発生の仕方をうまく説明したい。また推定されたパラメータに基づいて、将来的に得られるであろうデータを予測したい。

データの発生の仕方を説明するための枠組みをモデルと呼ぶ。基本的にはモデルをどう設定するか、よいモデルを作れるかが分析者のやること。このデータならこのモデルのように固定して考えるのではなく、与えられたデータに対していくつものモデルを試してその中でよいモデルを選ぶことが重要である。統計モデリングでは、確率分布の組み合わせでモデルを記述する。

パラメータとはモデルを構成する確率分布を決めるための数値などをいう。多くの名前のついた確率分布はパラメータをもつ。パラメータは分析者が決めるのではなく、与えられたデータと分析者が決めたモデル（とパラメータの計算方法）から数学的に計算できる。

例えば正規分布であれば平均 μ と分散 σ^2 をパラメータにもち、確率密度関数は

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

二項分布であれば n と p がパラメータで確率質量関数は

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

ポアソン分布であれば λ がパラメータで確率質量関数は

$$f(x) = \frac{\lambda^x}{x!} e^{-\lambda}$$

統計モデリングにおいては、パラメータを推定することが未知のデータを予測したり母集団について調べたりする上での一つの手がかりである。統計モデリングでは確率や確率分布といった概念が基礎にある。特にベイズ統計の枠組みでは、データだけではなくパラメータに対しても確率分布を考えるという方法をとる。ここは混乱しやすいポイントなので注意が必要である。

まずは上の統計モデリングの三つの概念について、特にパラメータの推定について理解するため、次のような簡単な問題を考えよう。

問題 1.1.1. 手元にコインがある。すでに 10 回投げて、表表裏表裏表表裏表表、という結果を観測している。次にコインを投げた結果表が出るか、裏が出るかを予測せよ。

モデルとして、コインはある一定の確率 p で表が出て、$1-p$ で裏が出るとする。特に、それ以前の結果には左右されずに次の結果が決まっている。

ここにすでに分析者の意思が反映されていることに注意しよう。現実の現象が全くこの通りの原理を持っているということではなく、あくまでこういうモデルを設定するとどうなるかを調べる。例えば表や裏以外の結果が出るというモデルや、確率が前回の結果に依存して変動するモデルなどを考える、ということもありうる。

この p が今回設定したモデルに付随するパラメータである。また 10 回投げた結果が、表表裏表裏表表裏表表、というのが現在のデータである。このデータを用いて p を推定する、というのがパラメータの推定。推定されたパラメータを用いて、次に表が出るか裏が出るかを予測することができる。

では、どのようにパラメータ p を推定することができるだろうか？例えば

1. 10 回中 7 回なので $p = 0.7$ と推定
2. コインだから表裏半々ぐらい出るはずなので $p = 0.5$ と推定
3. ちょっと表が出やすくて $p = 0.6$ と推定

など、色々考え方がある。これについて、最尤推定の考え方とベイズ推論の考え方を以下で紹介する。

いずれの場合でも**尤度関数**が中心的な役割を担っていることに注意しよう。

1.2　尤度関数と最尤推定

パラメータ推定の方法としてもっともよく用いられるのが最尤推定と呼ばれる方法である。これは尤度関数という関数を計算し、尤度関数を最大にするようなパラメータを推定値とするもの。先ほどのコインの例で一つ目の推測 $p = 0.7$ は、最尤法に基づいた結論と同じものになる。最尤法の考え方を理解するために、次のような実験を行うことにしよう。

以下は R で乱数を生成するプログラムである。それぞれ確率が $p = 0.7, 0.6, 0.5$ のコインを 10 回投げて表の出る回数を `x, y, z` としている。

R
```
x <- rbinom(n = 1, size = 10, prob = 0.7)
x
y <- rbinom(n = 1, size = 10, prob = 0.6)
y
z <- rbinom(n = 1, size = 10, prob = 0.5)
z
```

上は単に一度そのようなデータをえたということにすぎない。しかし、これを仮想的にデータを繰り返し得るとどのようになるかを考えることが、統計的な（あるいは数学的に）考える上での重要な視点である。

以下ではデータを 100 回取り直したらどうなるか、を実験している。

R
```
x1 <- rbinom(n = 100, size = 10, prob = 0.8)
x1
x2 <- rbinom(n = 100, size = 10, prob = 0.7)
x2
x3 <- rbinom(n = 100, size = 10, prob = 0.6)
x3
x4 <- rbinom(n = 100, size = 10, prob = 0.5)
x4
par(mfrow=c(2,2))
hist(x1, breaks = 0:10)
hist(x2, breaks = 0:10)
hist(x3, breaks = 0:10)
hist(x4, breaks = 0:10)
```

このようにパラメータ p の値によって、表の出る回数にばらつきがある。この表のでる回数の割合をパラメータ p に対する尤度という。

今回のデータでは、表の出た回数が 7 回なので、パラメータ p ごとに表が 7 回出る確率を計算し、それを $L(p)$ と表す。これを尤度関数という。

今回のデータに基づいて尤度関数を計算すると

$$L(p) = p^7(1-p)^3$$

となる。

最尤法では尤度関数が最大となる p をパラメータの推定値とする。これのグラフをかくと、

R
```
f <- function(p){120*p^7*(1-p)^3}
plot(f, 0, 1)
```

であり、$p = 0.7$ の時に最大となることがわかる。

1.2.1　尤度関数

上ですでに出てきたように、パラメータを入力すると現在観測しているデータが発生する確率を計算する関数を**尤度関数**という。この関数はモデルとデータが与えられるごとに定まるもの。

つまり $L(\theta)$ を尤度関数とすれば、パラメータ θ に対して所与のデータが発生する確率（密度）が $L(\theta)$ である。

ある動作の成功率 p を推定したい。10 回実験を行い、7 回成功している。このデータを元に、二項分布モデルによってパラメータ p に関する尤度関数を求めると、

$$\binom{10}{7} p^7 (1-p)^3$$

である。

あるイベントが一日に発生する回数を推定したい。過去三日間の結果はそれぞれ $5, 3, 4$ 件であった。これに相関がないと仮定して、ポアソン分布モデルによってパラメータ λ に関する尤度関数を求めると

$$\frac{\lambda^5}{5!} e^{-\lambda} \frac{\lambda^3}{3!} e^{-\lambda} \frac{\lambda^4}{4!} e^{-\lambda} = \frac{1}{5!3!4!} \lambda^{12} e^{-3\lambda}$$

となる。

得られたデータ $x_1, \ldots, x_N$ から、その集団の x についての母平均を推定したい。正規分布モデルで μ, σ に関する尤度関数を計算すると

$$L(\mu, \sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x_i - \mu)^2}{2\sigma^2})$$

となる。

このように、分析者が設定したモデルに基づいて、与えられたデータから尤度関数を決定する。この尤度関数を最大にするようなパラメータを求め、それによりデータを予測する方法が最尤推定である。

最尤推定をどのように行なっているか実験してみよう。ランダムにコインを取ってきて、そのコインの表の出る確率を推定するという実験を行う。

ランダムに選んだコインの表の出る確率を θ とする。これは一様分布から乱数生成する。θ の中身はこの時点では見ないことにしよう。（現実的にもこれを知ることはできない。これをデータから推定したい）

この θ を元にして、コインを n 回振り、表の出た回数を x とする。ここも二項分布から乱数生成する。この結果は見てもよい。（現実的にもこれはデータとして実際に得られる数値である。）

このデータとモデルを元にして尤度関数を計算し、そのグラフをプロットする。最尤推定の結果は x/n になることが数学的に証明できる。

n を変えてみると、推定結果が当たりやすくなる。

R
```
theta <- runif(1)

n <- 10
x <- rbinom(n=1, size=n, prob=theta)
x

f <- function(p){choose(n,x) * p^x * (1-p)^(n-x)}
plot(f, 0, 1)

theta
```

1.3 ベイズ推論

ではベイズ推論の枠組みでは、上の問題に対してどのような回答をするか？まず一つの大きなポイントは、パラメータ p をある決まった値として推定するのではなく、ある程度幅を持った分布として推定することにある。言い方を変えると、最尤推定では尤度関数のある一つの値の様子しか見ないのに対し、ベイズ推論では尤度関数全体の情報を利用する。

分布にすると何が違うかを説明する。例えば与えられたデータが

1. 2 回投げて 1 回表
2. 100 回投げて 50 回表

というふた通りの可能性を考えよう。この時、最尤法でパラメータを推定するとどちらも $p = 0.5$ となる。しかし、このように推定してしまうと投げた回数が異なるという情報を捨てることになる。

しかし例えば p の分布が以下の二つでどのように違うか検討せよ。

R
```
f1 <- function(p){choose(2,1) * p^2 * (1-p)^2}
f2 <- function(p){choose(100,50) * p^50 * (1-p)^50}
par(mfrow=c(1,2))
plot(f1, 0, 1)
plot(f2, 0, 1)
```

これがパラメータの予測を値ではなく分布として行うことによる違いの一つである。

またこの先に説明していくように

1. パラメータの事前分布を設定（正則化との関係、データに大きく左右されない、特にサンプルサイズが小さい時の影響）
2. 重ね合わせとしての予測分布

という違いが現れ、パラメータを点推定した場合と異なる予測を与える。

最尤推定の場合に $L(\theta)$ が $\theta = \hat{\theta}$ で最大値をとり、その $\hat{\theta}$ を用いて x の分布を記述して

いた。x の分布はパラメータ θ を使ってかけるので、x が従う分布は $p(x \mid \hat{\theta})$ である。

一方で $\hat{\theta}$ という単一の値だけでなく尤度関数 $L(\theta)$ 全体を使うためには、事前分布と事後分布という概念を導入し、それにより予測分布を計算することが必要になる。このことについて詳しくみていこう。

1.3.1　事前分布と事後分布

パラメータの分布を推測する方法として、以下のようなものを採用する。

1. まずパラメータの事前分布 $p(\theta)$ を設定する。
2. 尤度関数 $L(\theta) = p(x \mid \theta)$ を計算する。
3. この二つの積を正規化することで $p(\theta \mid x)$ を計算する。

正規化とは、$p(x \mid \theta)p(\theta)$ に比例する関数として確率分布を定める操作で、

$$\int p(x \mid \theta)p(\theta)d\theta$$

で割ることで

$$p(\theta \mid x) = \frac{1}{\int p(x \mid \theta)p(\theta)} p(x \mid \theta)p(\theta)$$

によって得られる分布 $p(\theta \mid x)$ をパラメータの事後分布という。観測を通して事前分布を事後分布に取り替えることを、パラメータ更新、ベイズ更新などという。

1.3.2　予測分布

得られた事後分布 $p(\theta \mid x)$ に基づいてデータの予測を行う。点推定の場合、モデルの分布のパラメータを単に推定した値にすることでデータの予測を行う。

しかし、推定を分布で行なった場合にはその情報を全て活かすには事後分布に沿って重ね合わせを行う必要がある。

事後分布 $p(\theta \mid x)$ に対して、予測分布 $q(y)$ は

$$q(y) = \int p(y \mid \theta)p(\theta \mid x)d\theta$$

と計算できる。このとき、あらかじめ設定したモデルの分布とは異なるものが一般には得られることに注意しよう。例えば正規分布モデルを用いたとして、x についての予測分布は正規分布になるとは限らない。

1.4　線形回帰モデル

線形回帰分析を上で述べたような統計モデリングの視点から見直してみよう。データが $(x_1, y_1), \ldots, (x_N, y_N)$ と与えられたとしよう。線形回帰は $y = a + bx$ という式に当てはめるという考え方だが、最小二乗法を確率分布で表現すると、y が平均 $a + bx$ で分散 σ^2 の正規分布に従うというモデルである。

このとき、尤度関数は

$$L(a,b,\sigma^2)=\prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(y_i-(a+bx_i))^2}{2\sigma^2})$$

である。

この式において、尤度関数を最大にするためには結局

$$\frac{(y_i-(a+bx_i))^2}{2\sigma^2})$$

を最小にする必要があり、これがまさに最小二乗法そのものである。つまり、最尤推定で回帰係数を行った結果が最小二乗法による係数そのものである。σ^2 も同様に推定することで、与えられた説明変数 x の値に対して y が正規分布として予測できる。

ではベイズ的に行うと何が得られるか？ a,b について分布が得られることになるが、線形回帰でパラメータの信頼区間や検定について扱ったと思う。これと全く同様ではないものの、このような情報がおおよそパラメータ a,b の分布である。これらを総合することで、与えられた説明変数 x の値に対して y が正規分布の重ね合わせとして予測できる。

なお、上のように考えると正規分布以外の分布を使ってもよいのではないか？と思うかもしれないが、実際それは正しい。例えば外れ値に対する頑健性を得るためにコーシー分布や t 分布を用いるという話を聞いたことがあるかもしれない。

1.5 階層モデル

ベイズ推論を用いるモデルの例としてよく取り上げられるのが階層モデルである。ここでは階層モデルの簡単な例を紹介しよう。

いくつかの都道府県で小学生の身長の変化と給食の関係について調べる。給食の種類が二種類 A,B とあり、各都道府県はこのいずれかの給食を使っているとしよう。

ある一学年の小学生の平均身長と、その小学生の一年後の平均身長をデータとして、この給食の種類により影響があるかを調べる。

シンプルなモデルとして、都道府県の平均身長が正規分布に従うと仮定し、一年目の平均身長 $y=\beta_1$ と二年目の平均身長を $y=\beta_1+\beta_2+cz$ とする。ここで z が給食のタイプを表すダミー変数であり、β_2 が一年での成長分のベースを表す。

これを階層化したものが以下のモデルである。階層化というのは、各都道府県ごとに β_1,β_2 が異なる、つまり都道府県ごとの個体差 $r_1[i],r_2[i]$ を導入する。また、この r_1,r_2 がそれぞれ同一の正規分布に従うと仮定する。

stan

```
data{
  int N; //県の数
  real height1[N]; //1 年目の平均
  real height2[N]; //2 年目の平均
  int X[N]; //給食タイプ
  int N_beta; //係数の数
  real sd1[N]; //1 年目の標準偏差
  real sd2[N]; //2 年目の標準偏差
}

parameters{
  real beta[N_beta];
  real<lower=0> sigma1;
  real<lower=0> sigma2;
  real r1[N];
  real r2[N];
}

transformed parameters{
  real mu1[N];
  real mu2[N];
  for(i in 1:N){
    mu1[i] = beta[1] + r1[i];
    mu2[i] = beta[1] + r1[i] + beta[2] + r2[i] + beta[3] * X[i];
  }
}

model{
  for(i in 1:N){
    height1[i] ~ normal(mu1[i], sd1[i]);
    height2[i] ~ normal(mu2[i], sd2[i]);
  }
  for(i in 1:N_beta){
    beta[i] ~ uniform(-1.0e+4, 1.0e+4);
  }
  for(i in 1:N){
    r1[i] ~ normal(0, sigma1);
    r2[i] ~ normal(0, sigma2);
  }
  sigma1 ~ uniform(0, 1.0e+4);
  sigma2 ~ uniform(0, 1.0e+4);
}
```

R

```
data <- read.csv("Chap1.1.csv", fileEncoding="cp932")
data
X <- abs(as.numeric(data$給食)-2)
d <- list(N=10,
          height1=data$height1,
          height2=data$height2,
          X=X[1:10],
          N_beta=3,
          sd1=data$sd1,
          sd2=data$sd2)

library(rstan)
fit <- stan(file='lunch.stan',data=d,
            iter=1000,chains=4)
fit

stan_trace(fit,pars="beta")
stan_hist(fit,pars="beta")

beta <- rstan::extract(fit)$beta
beta[1:100,]
sum(beta[,3]>0)/length(beta[,3])
```