Bài 5 - Thống kê Bayes tính toán (Computational Bayesian Statistics)

Thống kê máy tính và ứng dụng
(Computational Statistics and Applications)

Vũ Quốc Hoàng (vqhoang@fit.hcmus.edu.vn)

FIT - HCMUS

Ngày 28 tháng 3 năm 2023

Nội dung

1. Suy diễn Bayes
2. Mô hình nhị thức
3. "Tính" phân phối hậu nghiệm
4. Markov chain Monte Carlo
5. Lập trình xác suất
6. Mô hình tuyến tính tổng quát

Nội dung

Công thức Bayes

Trên không gian mẫu Ω, các biến cố $\{E_1, E_2, ..., E_n\}$ được gọi là một họ đầy đủ nếu

- $E_i \cap E_j = \varnothing$ khi $i \neq j$ (mutually exclusive, loại trừ, "không trùng"),
- $\Omega = \bigcup_{i=1}^{n} E_i$ (exhaust all possibilities, đầy đủ, "không sót").

Công thức Bayes (Bayes' rule)

$$\underbrace{P(E_i|D)}_{\text{posterior}} = \frac{\overbrace{P(D|E_i)}^{\text{likelihood}}\overbrace{P(E_i)}^{\text{prior}}}{\underbrace{P(D)}_{\text{evidence}}} = \frac{P(D|E_i)P(E_i)}{\sum_{j=1}^{n} P(D|E_j)P(E_j)} \propto P(D|E_i)P(E_i).$$

Công thức Bayes hướng dẫn cách "cập nhật xác suất" hay "phân bổ lại niềm tin" (reallocation of **credibility**).

Suy diễn Bayes

Các **dữ liệu** (data) được sinh ra từ một **mô hình xác suất** (probabilistic model). Mô hình được xác định bởi các **tham số** (parameter). Từ các dữ liệu quan sát, **suy diễn** (inference) về mô hình qua tham số. **Suy diễn Bayes** (Bayesian inference)

1. Xác định dữ liệu, mô hình và hàm hợp lý của tham số,
2. "Chọn" phân phối tiên nghiệm phù hợp cho tham số,
3. "Tính" phân phối hậu nghiệm của tham số theo công thức Bayes

$$\underbrace{p(\theta|D)}_{\text{posterior}} = \frac{p(D|\theta)p(\theta)}{p(D)} = \frac{\overbrace{L(\theta|D)}^{\text{likelihood}}\overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(D)}_{\text{evidence}}} \propto L(\theta|D)p(\theta),$$

4. Kiểm tra phân phối hậu nghiệm có sinh dữ liệu phù hợp ("posterior predictive check"). Nếu không, thử mô hình và/hoặc dữ liệu khác,
5. "Dùng" phân phối hậu nghiệm để đưa ra các kết luận.

Suy diễn Bayes - Ví dụ 1

Bài toán. Một nhà máy sản xuất bóng với 4 loại kích thước 1, 2, 3, 4. Đặt 3 quả bóng loại kích thước 2. Nhận được 3 quả bóng với các kích thước: 1.77, 2.23, 2.70. Hỏi nhà máy có sản xuất đúng loại đã đặt?

Suy diễn. Ta mô hình kích thước bóng là biến ngẫu nhiên $X \sim \mathcal{N}(\mu, \sigma^2)$ với tham số μ có thể nhận một trong 4 giá trị

$$\begin{cases} E_1 : \mu = \mu_1 = 1.0, \\ E_2 : \mu = \mu_2 = 2.0, \\ E_3 : \mu = \mu_3 = 3.0, \\ E_4 : \mu = \mu_4 = 4.0. \end{cases}$$

Ta có

$$p(X = x | E_i) = f_{\mathcal{N}(\mu_i, \sigma^2)}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu_i)^2}{2\sigma^2}}, i = 1, ..., 4.$$

Suy diễn Bayes - Ví dụ 1 (tt)

Ta nhận được X_1, X_2, X_3 độc lập và cùng phân phối với X. Cụ thể, ta có dữ liệu

$$D = (X_1 = 1.77) \cap (X_2 = 2.23) \cap (X_3 = 2.70).$$

Hàm hợp lý của tham số μ trên dữ liệu D theo mô hình đã chọn là

$$\begin{aligned} L(\mu_i|D) &= p(D|E_i) = p\left((X_1 = 1.77) \cap (X_2 = 2.23) \cap (X_3 = 2.70)|E_i\right) \\ &= p(X_1 = 1.77|E_i)p(X_2 = 2.23|E_i)p(X_3 = 2.70|E_i) \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^3 e^{\frac{-\left((1.77-\mu_i)^2+(2.23-\mu_i)^2+(2.70-\mu_i)^2\right)}{2\sigma^2}} \\ &\propto e^{\frac{-\left((1.77-\mu_i)^2+(2.23-\mu_i)^2+(2.70-\mu_i)^2\right)}{2\sigma^2}}, i = 1, ..., 4. \end{aligned}$$

"Giả sử" phân phối tiên nghiệm là

$$p(E_i) = \frac{1}{4} \propto 1, i = 1, ..., 4.$$

Suy diễn Bayes - Ví dụ 1 (tt)

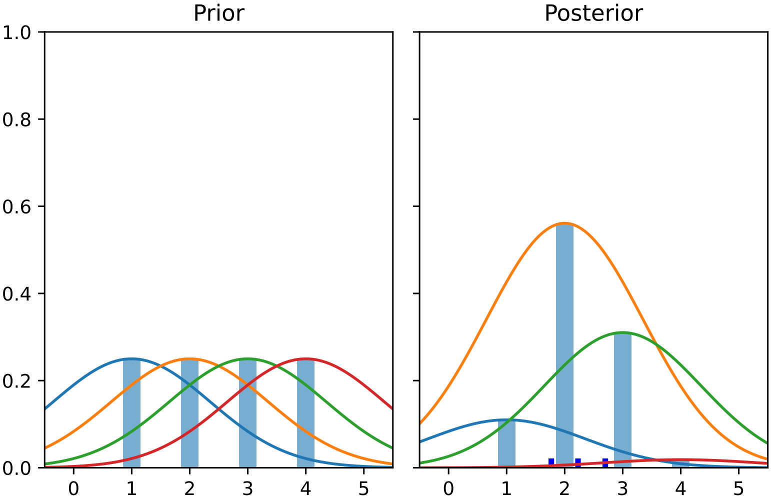
Ta tính phân phối hậu nghiệm theo công thức Bayes

$$p(E_i|D) = \frac{p(D|E_i)p(E_i)}{\sum_{j=1}^{4} p(E_j|D)p(E_j)} = \frac{e^{\frac{-\left((1.77-\mu_i)^2+(2.23-\mu_i)^2+(2.70-\mu_i)^2\right)}{2\sigma^2}}}{\sum_{j=1}^{4} e^{\frac{-\left((1.77-\mu_j)^2+(2.23-\mu_j)^2+(2.70-\mu_j)^2\right)}{2\sigma^2}}}, i = 1, \dots, 4.$$

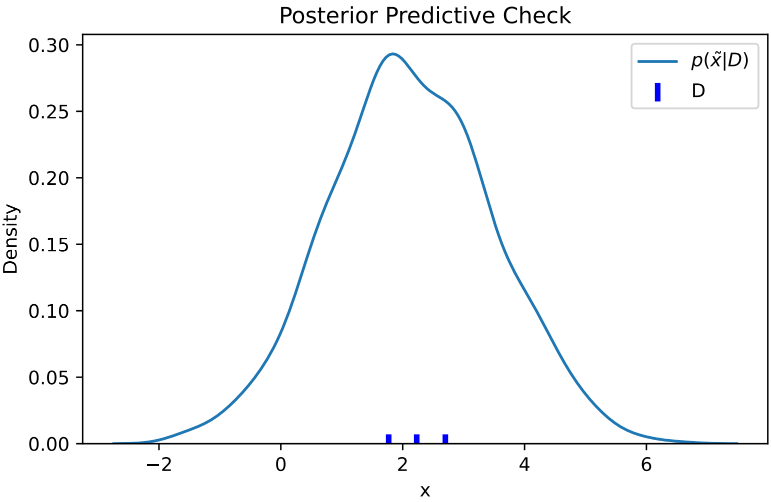
Với một số giá trị của σ^2, tính toán ta có

	$\sigma^2 = 1$	$\sigma^2 = 1.35$	$\sigma^2 = 2$
$P(\mu = 1\|D)$	7%	11%	16%
$P(\mu = 2\|D)$	64%	56%	47%
$P(\mu = 3\|D)$	29%	31%	32%
$P(\mu = 4\|D)$	1%	2%	5%

Suy diễn Bayes - Ví dụ 1 (tt)



Suy diễn Bayes - Ví dụ 1 (tt)



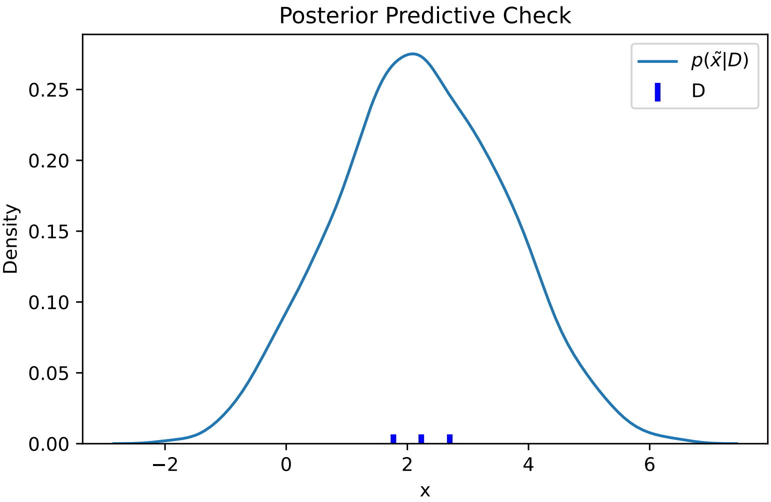
Suy diễn Bayes - Ví dụ 1 - PyMC

```
import pymc as pm
import pytensor

mu = np.array([1.0, 2.0, 3.0, 4.0])
prior = 1/4 * np.ones(len(mu))
x = np.array([1.77, 2.23, 2.70])
sigma2 = 1.35

with pm.Model() as model:
    mu_index_var = pm.Categorical("mu_index", p=prior)
    mu_var = pytensor.shared(mu)[mu_index_var]
    x_var = pm.Normal("x", mu=mu_var, tau=1/sigma2,
                      observed=x)
    trace = pm.sample(return_inferencedata=False)
```

Suy diễn Bayes - Ví dụ 1 - PyMC (tt)

```
mu_index_posterior = trace["mu_index"]
values, counts = np.unique(mu_index_posterior,
                           return_counts=True)
print(mu[values])
# [1. 2. 3. 4.]
print(counts/np.sum(counts))
# [0.10225 0.57525 0.30475 0.01775]

x_post_sample = pm.sample_posterior_predictive(trace,
    model, return_inferencedata=False)["x"].flatten()
sns.kdeplot(x_post_sample)
plt.scatter(x, np.zeros(len(x)))
```

Suy diễn Bayes - Ví dụ 1 - PyMC (tt)

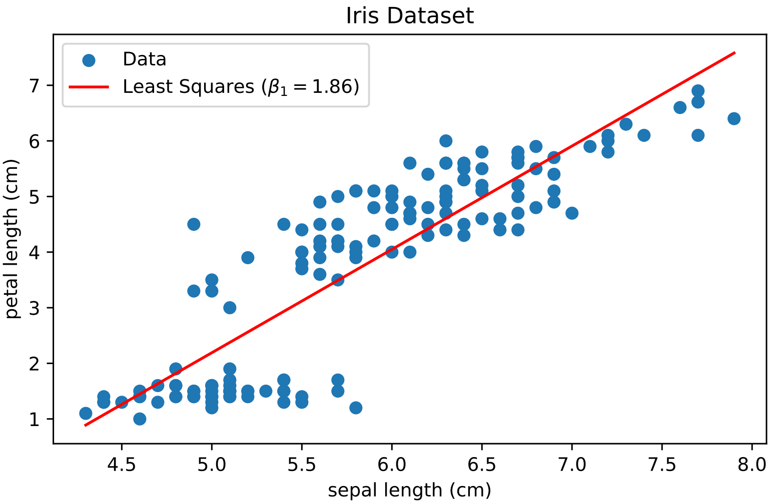


Suy diễn Bayes - Ví dụ 2

Iris Dataset

- Wikipedia: `https://en.wikipedia.org/wiki/Iris_flower_data_set`
- UCI: `https://archive.ics.uci.edu/ml/datasets/iris`
- Scikit-learn: `https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html`

Suy diễn Bayes - Ví dụ 2 (tt)



Suy diễn Bayes - Ví dụ 2 (tt)

Bài toán. Từ dữ liệu Iris, xác định sự phụ thuộc của petal-length vào sepal-length.

Suy diễn. "Từ gợi ý của dữ liệu", ta mô hình sự phụ thuộc của petal-length (y) vào sepal-length (x) như sau

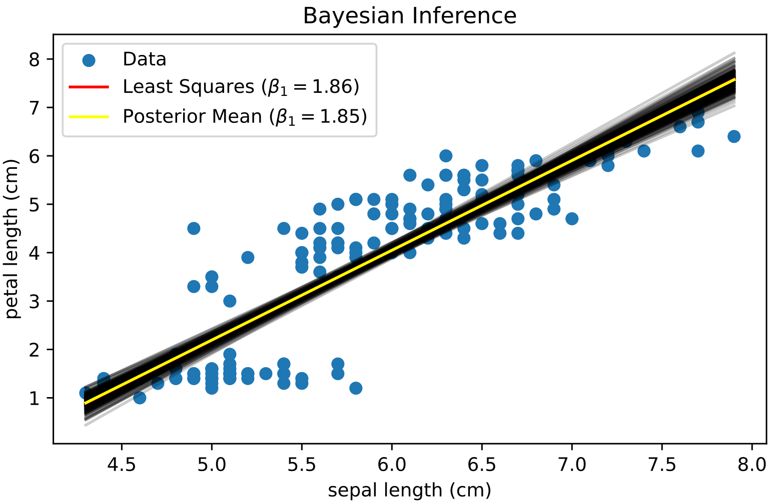
$$
\begin{aligned}
\hat{y} &= \beta_0 + \beta_1 x,\\
y &\sim \mathcal{N}(\hat{y}, \sigma),\\
\beta_0 &\sim \mathcal{N}(0, 10^2),\\
\beta_1 &\sim \mathcal{N}(0, 10^2),\\
\sigma &\sim \mathcal{U}(0, 1000).
\end{aligned}
$$

Dùng PyMC, "tính" phân phối hậu nghiệm, ta có các kết quả sau.

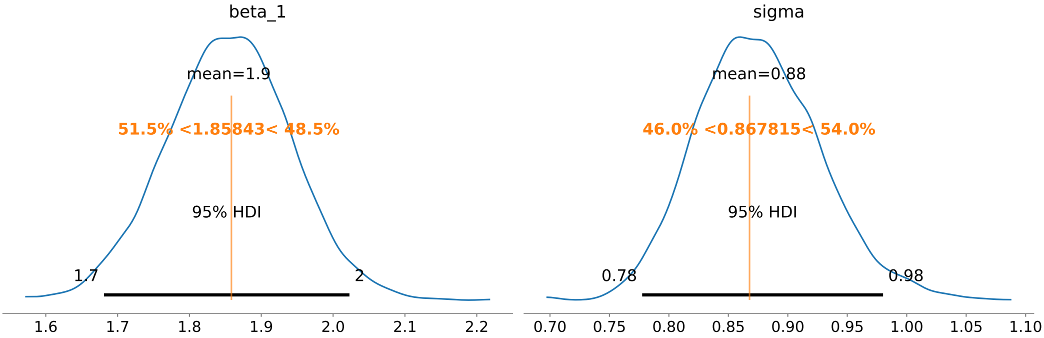
Suy diễn Bayes - Ví dụ 2 (tt)

```python
from sklearn import datasets

iris = datasets.load_iris()
x = iris.data[:, 0]
y = iris.data[:, 2]

with pm.Model() as model:
    beta_0 = pm.Normal("beta_0", mu=0, sigma=10)
    beta_1 = pm.Normal("beta_1", mu=0, sigma=10)
    sigma = pm.Uniform("sigma", lower=0, upper=1000)
    y_hat = beta_0 + beta_1*x
    y_var = pm.Normal("y", mu=y_hat, sigma=sigma,
                      observed=y)
    trace = pm.sample(5000)
```

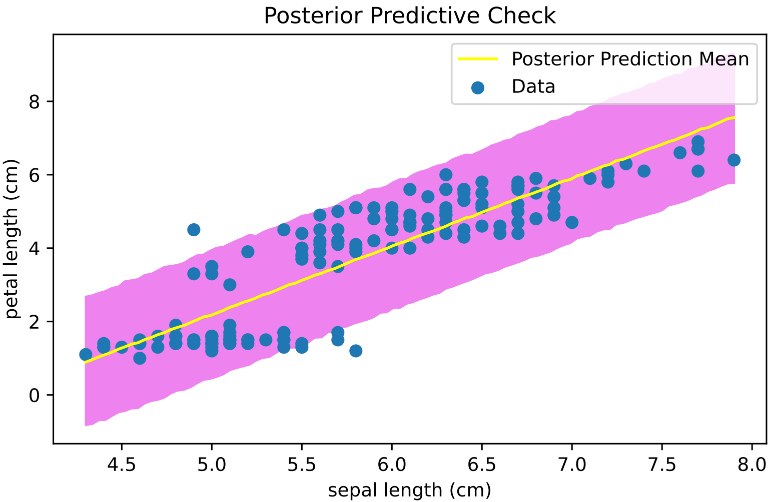
Suy diễn Bayes - Ví dụ 2 (tt)



Suy diễn Bayes - Ví dụ 2 (tt)



Suy diễn Bayes - Ví dụ 2 (tt)



Nội dung

Mô hình nhị thức

Biến ngẫu nhiên rời rạc Y được gọi là kết quả của một **phép thử Bernoulli** (Bernoulli trial) hay có **phân phối Bernoulli** (Bernoulli distribution) với tham số p $(0 \leq p \leq 1)$, kí hiệu $Y \sim \text{Bernoulli}(p)$, nếu

$$\begin{cases} P(Y = 1) = p, \\ P(Y = 0) = 1 - p. \end{cases}$$

$(Y = 1)$ thường được gọi là **"thành công"** (success), $(Y = 0)$ là **"thất bại"** (failure), p là xác suất thành công.

Dãy biến ngẫu nhiên $Y_1, Y_2, \ldots$ được gọi là một dãy phép thử Bernoulli hay một **quá trình Bernoulli** (Bernoulli process) nếu $Y_1, Y_2, \ldots$ độc lập và cùng phân phối $\text{Bernoulli}(p)$.

Mô hình nhị thức (tt)

Bài toán. Cho dữ liệu của dãy phép thử Bernoulli $D = \{Y_1 = y_1, Y_2 = y_2, ..., Y_n = y_n\}$, "tìm" p.

Suy diễn. Xem tham số $\theta = p$ là biến ngẫu nhiên liên tục, nhận giá trị trong $[0, 1]$.
Với $Y \sim \text{Bernoulli}(\theta)$ có giá trị quan sát $y \in \{0, 1\}$, hàm hợp lý của θ theo y là

$$p(y|\theta) = \theta^y (1-\theta)^{1-y} = \begin{cases} \theta & y = 1, \\ 1-\theta & y = 0. \end{cases}$$

Do đó, với dữ liệu $D = \{y_i\}_{i=1}^n$, hàm hợp lý của θ theo D là

$$p(D|\theta) = \prod_{i=1}^n \theta^{y_i} (1-\theta)^{1-y_i} = \theta^{\sum_{i=1}^n y_i} (1-\theta)^{\sum_{i=1}^n (1-y_i)} = \theta^z (1-\theta)^{n-z},$$

với $z = \sum_{i=1}^n y_i$ là số lần thành công trong n lần quan sát của dữ liệu D.

Mô hình nhị thức (tt)

"Chọn" phân phối tiên nghiệm cho θ là **phân phối Beta** (Beta distribution) với tham số a, b $(a > 0, b > 0), \theta \sim \text{Beta}(a, b)$, ta có

$$p(\theta; a, b) \propto \theta^{a-1}(1-\theta)^{b-1}, 0 \leq \theta \leq 1.$$

(Xem thêm phân phối Beta `https://en.wikipedia.org/wiki/Beta_distribution`.)

Từ công thức Bayes, θ có phân phối hậu nghiệm sau khi quan sát dữ liệu D là

$$p(\theta|D) \propto p(D|\theta)p(\theta; a, b) \propto \theta^{z}(1-\theta)^{n-z}\theta^{a-1}(1-\theta)^{b-1} \propto \theta^{z+a-1}(1-\theta)^{n-z+b-1}.$$

Nhận xét, $(\theta|D) \sim \text{Beta}(a+z, b+n-z)$.
Do đó, phân phối Beta được gọi là **phân phối tiên nghiệm liên hợp** (conjugate prior distribution) của phân phối Bernoulli.

Mô hình nhị thức (tt)

Lưu ý, các **"siêu tham số"** (hyperparameter) a, b của phân phối tiên nghiệm và $\kappa = a + b$ thường được gọi là **pseudo-count** vì

- Trước khi quan sát dữ liệu D, $\theta \sim \text{Beta}(a, b)$, ta tin vào xác suất thành công θ như thể ta đã thấy a lần thành công và b lần thất bại (trong tổng số $\kappa = a + b$ lần),
- Dữ liệu D cho thấy z lần thành công và $n - z$ lần thất bại (trong tổng số n lần),
- Sau khi quan sát dữ liệu D, $(\theta|D) \sim \text{Beta}(a + z, b + n - z)$, ta tin vào xác suất thành công như thể ta đã thấy $a + z$ lần thành công và $b + n - z$ lần thất bại (trong tổng số $\kappa + n$ lần).

Mô hình nhị thức (tt)

Như vậy, có thể nói phân phối hậu nghiệm là "tổng hợp" hay "thỏa hiệp" giữa phân phối tiên nghiệm và hàm hợp lý trên dữ liệu. Chẳng hạn

- Phân phối tiên nghiệm $\theta \sim \text{Beta}(a, b)$ có kỳ vọng
$$E(\theta) = \frac{a}{a+b} = \frac{a}{\kappa},$$
- Hàm hợp lý từ dữ liệu $L(\theta|D) = p(D|\theta) = \theta^z(1-\theta)^{n-z}$ đạt cực đại tại
$$\hat{\theta}_{\text{MLE}} = \frac{z}{n},$$
- Phân phối hậu nghiệm $(\theta|D) \sim \text{Beta}(a+z, b+n-z)$ có kỳ vọng
$$\begin{aligned} E(\theta|D) &= \frac{a+z}{a+z+b+n-z} = \frac{a+z}{\kappa+n} = \frac{\kappa}{\kappa+n}\frac{a}{\kappa} + \frac{n}{\kappa+n}\frac{z}{n} \\ &= \frac{\kappa}{\kappa+n}E(\theta) + \frac{n}{\kappa+n}\hat{\theta}_{\text{MLE}}. \end{aligned}$$

Mô hình nhị thức - Ví dụ

- Dữ liệu: $z = 1, n = 10$ (quan sát thấy 1 lần thành công trong 10 lần), hàm hợp lý

  $$p(D|\theta) = \theta^z(1-\theta)^{n-z} = \theta^1(1-\theta)^9$$

  đạt cực đại tại

  $$\hat{\theta}_{\text{MLE}} = \frac{z}{n} = \frac{1}{10} = 0.1$$

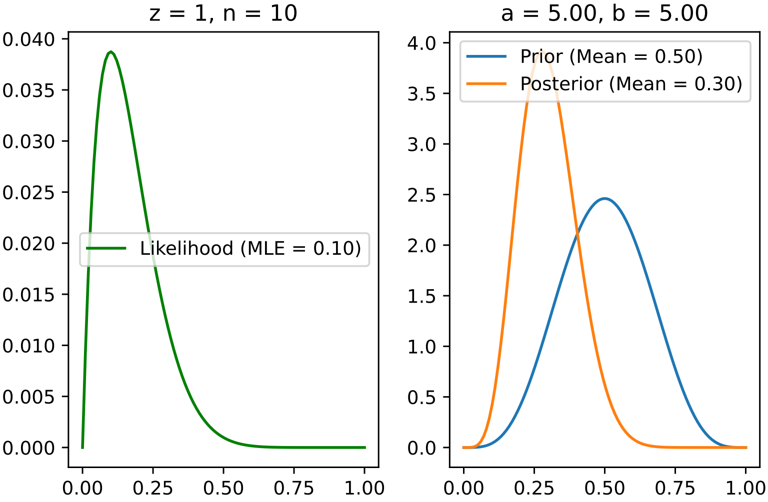
- Phân phối tiên nghiệm: $a = 5, b = 5, \kappa = a + b = 10$ (như thể đã quan sát thấy 5 lần thành công trong 10 lần), $\theta \sim \text{Beta}(a, b) = \text{Beta}(5, 5)$ có

  $$E(\theta) = \frac{a}{\kappa} = \frac{5}{10} = 0.5.$$

- Phân phối hậu nghiệm: $a + z = 6, \kappa + n = 20$ (như thể đã quan sát thấy 6 lần thành công trong 20 lần), $(\theta|D) \sim \text{Beta}(a + z, b + n - z) = \text{Beta}(6, 14)$ có

  $$E(\theta|D) = \frac{a+z}{\kappa+n} = \frac{6}{20} = 0.3 = \frac{\kappa}{\kappa+n}E(\theta) + \frac{n}{\kappa+n}\hat{\theta}_{\text{MLE}} = 0.5E(\theta) + 0.5\hat{\theta}_{\text{MLE}}.$$

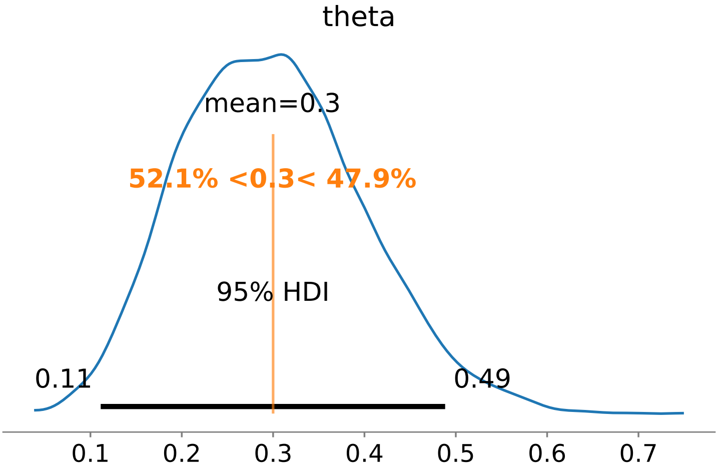
Mô hình nhị thức - Ví dụ (tt)



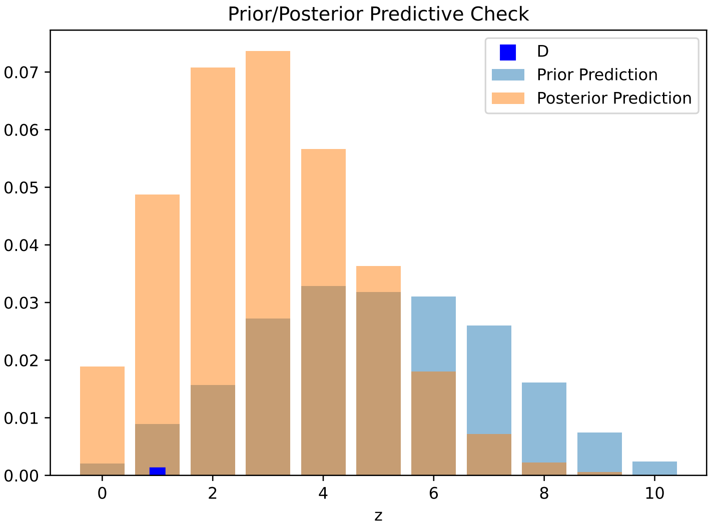
Mô hình nhị thức - Ví dụ - PyMC

```
with pm.Model() as model:
    theta = pm.Beta("theta", alpha=a, beta=b)
    z_var = pm.Binomial("z", p=theta, n=n, observed=z)
    trace = pm.sample(5000)
    prior_pred = pm.sample_prior_predictive(1000,
       return_inferencedata=False)
    posterior_pred = pm.sample_posterior_predictive(trace,
       return_inferencedata=False)

    az.plot_posterior(trace, var_names=["theta"],
             hdi_prob=0.95, ref_val=posterior_dist.mean())
```

Mô hình nhị thức - Ví dụ - PyMC (tt)



Mô hình nhị thức - Ví dụ - PyMC (tt)



Nội dung

“Tính” phân phối hậu nghiệm

Các cách “tính” phân phối hậu nghiệm

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}.$$

- Chọn $p(\theta)$ là phân phối **tiên nghiệm liên hợp** (conjugate prior) cho hàm hợp lý $p(D|\theta)$ để phân phối hậu nghiệm $p(\theta|D)$ có dạng như phân phối tiên nghiệm.
- “Rời rạc hóa” hay **“xấp xỉ lưới”** (grid approximation) θ

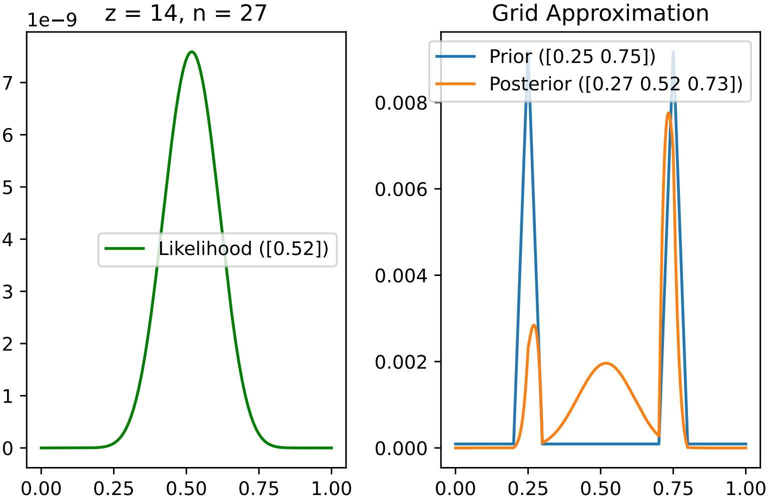
$$p(D) = \sum_{\theta*} p(D|\theta^*)p(\theta^*).$$

- Dùng các phương pháp **lấy mẫu ngẫu nhiên** (randomly sampling), tức là các phương pháp sinh số ngẫu nhiên, đặc biệt là các phương pháp **Markov chain Monte Carlo** (MCMC) để sinh các giá trị θ từ phân phối $p(\theta|D)$ với số lượng đủ nhiều.
- (và các cách khác)

Phân phối tiên nghiệm liên hợp

- `https://en.wikipedia.org/wiki/Conjugate_prior`.
- *Ví dụ:* trong mô hình nhị thức, khi hàm hợp lý là phân phối Bernoulli (hay phân phối nhị thức, nhị thức âm), ta đã chọn phân phối tiên nghiệm cho tham số $\theta = p$ (xác suất thành công) là phân phối Beta để phân phối hậu nghiệm cũng là phân phối Beta.

“Xấp xỉ lưới” - Ví dụ



Lấy mẫu loại bỏ - Ví dụ

Xét một bài toán suy diễn Bayes đơn giản: ta muốn suy diễn về $X \sim \text{Exp}(1)$ bằng cách sử dụng một quan sát y từ $Y \sim \mathcal{N}(0, X)$. Ta cần tìm phân phối hậu nghiệm $(X|Y = y)$.

Trước hết, phân phối tiên nghiệm của X, $X \sim \text{Exp}(1)$, có hàm mật độ

$$p(x) = e^{-x}\mathbb{I}_{[0,\infty)}(x).$$

Hàm hợp lý của X khi biết $Y \sim \mathcal{N}(0, X)$ nhận giá trị y là

$$p(y|x) = \frac{1}{\sqrt{2\pi x}} e^{-y^2/(2x)}.$$

Từ đó, dùng công thức Bayes, phân phối hậu nghiệm $(X|Y = y)$ có hàm mật độ

$$p(x|y) \propto p(y|x)p(x)$$

Lấy mẫu loại bỏ - Ví dụ (tt)

Đặt

$$f(x) = \frac{1}{\sqrt{x}} e^{-y^2/(2x)-x} \mathbb{I}_{[0,\infty)}(x) \propto p(y|x)p(x) \propto p(x|y),$$

ta có thể dùng phương pháp lấy mẫu loại bỏ để sinh mẫu từ phân phối hậu nghiệm $(X|Y=y)$. Lưu ý, phương pháp lấy mẫu loại bỏ không cần f phải được chuẩn hóa.

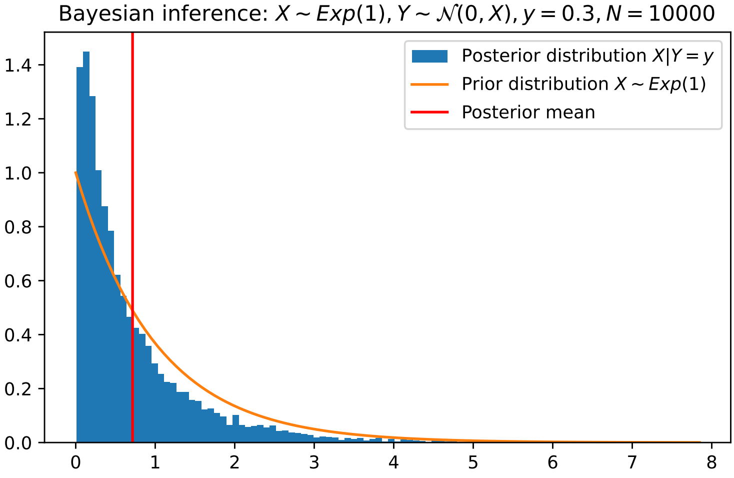
Cụ thể, dùng thuật toán lấy mẫu loại bỏ theo khuôn với phân phối đề cử là Exp(1) và hằng số

$$c = \frac{1}{|y|} e^{-1/2},$$

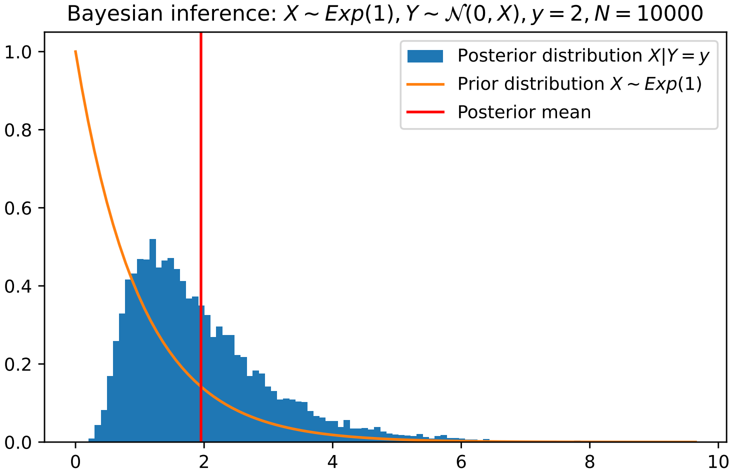
ta sinh mẫu $X_1, ..., X_N$ iid theo phân phối hậu nghiệm $(X|Y=y)$, vẽ histogram và tính

$$E(X|Y=y) \approx \frac{1}{N} \sum_{j=1}^{N} X_j = \bar{X}, \quad Var(X|Y=y) \approx \frac{1}{N-1} \sum_{j=1}^{N} (X_j - \bar{X})^2.$$

Lấy mẫu loại bỏ - Ví dụ (tt)



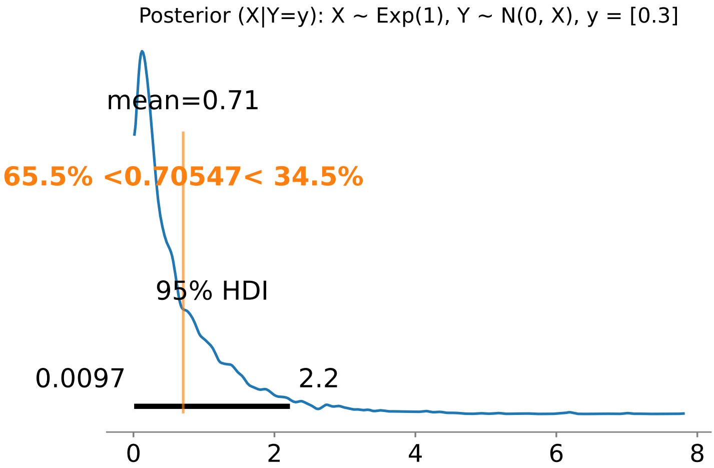
Lấy mẫu loại bỏ - Ví dụ (tt)



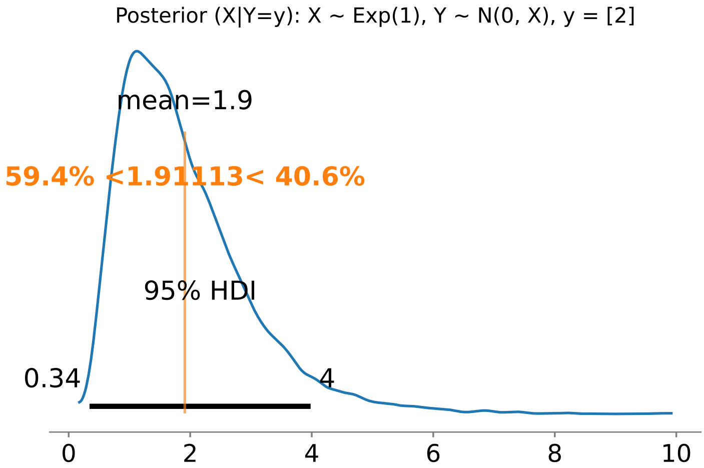
Lấy mẫu loại bỏ - Ví dụ - PyMC

```
#y = [0.3]
#y = [2]
y = [0.3, 2, 1.5, 2.3]

with pm.Model() as model:
    x = pm.Exponential("x", lam=1)
    y_var = pm.Normal("y", mu=0, tau=1/x, observed=y)
    trace = pm.sample(5000, return_inferencedata=False)
    posterior_pred = pm.sample_posterior_predictive(trace)
    az.plot_posterior(trace, var_names=["x"],
                hdi_prob=0.95, ref_val=np.mean(trace["x"]))
```

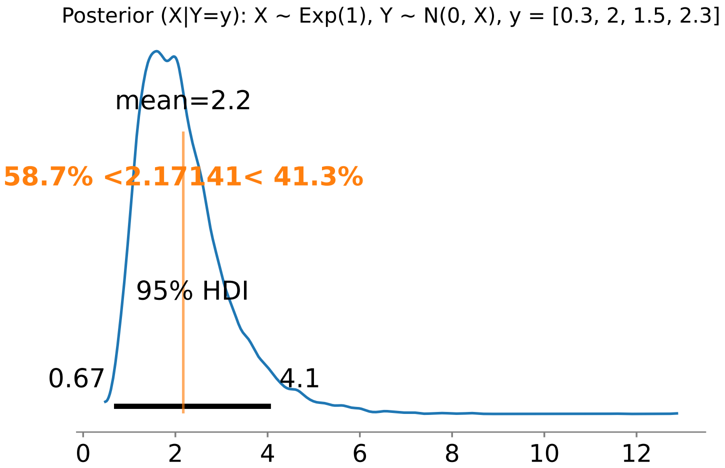
Lấy mẫu loại bỏ - Ví dụ - PyMC (tt)



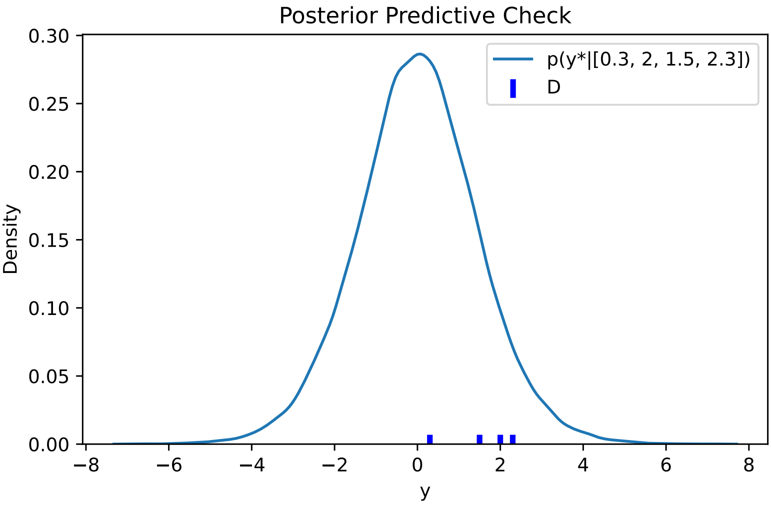
Lấy mẫu loại bỏ - Ví dụ - PyMC (tt)



Lấy mẫu loại bỏ - Ví dụ - PyMC (tt)



Lấy mẫu loại bỏ - Ví dụ - PyMC (tt)



Nội dung

Markov chain Monte Carlo

- Giả sử ta tính được phân phối tiên nghiệm $p(\theta)$ và hàm hợp lý $p(D|\theta)$ "đến một hệ số", tức là tính được $f(\theta) \propto p(\theta)p(D|\theta) \propto p(\theta|D)$, ta có thể dùng các phương pháp lấy mẫu ngẫu nhiên để sinh các giá trị đại diện cho phân phối hậu nghiệm $p(\theta|D)$.
- Các phương pháp sinh số ngẫu nhiên ta đã học (như lấy mẫu loại bỏ) sinh dãy $\theta_1, \theta_2, \ldots \overset{\text{iid}}{\sim} \frac{1}{Z_f} f(\theta)$ với Z_f là hằng số chuẩn hóa. Trong nhiều trường hợp (như khi $\theta \in \mathbb{R}^d$ với d lớn), các phương pháp này thường kém hiệu quả.
- Các phương pháp **Markov Chain Monte Carlo** (MCMC) sinh **xích Markov** (Markov chain) $\theta_1, \theta_2, \ldots$ có **phân phối dừng** (stationary distribution) là $\frac{1}{Z_f} f(\theta)$, thường dễ dàng và hiệu quả hơn.
- Chính nhờ các thuật toán MCMC (và các thư viện phần mềm chạy trên phần cứng mạnh), ta có thể dùng suy diễn Bayes để phân tích các dữ liệu thực tế.

Thuật toán Metropolis-Hastings

Thuật toán MH. (Metropolis-Hastings method)

Input:

- hàm f với giá trị trong khoảng $[0, \infty)$ (non-normalised target density),
- hàm mật độ xác suất chuyển $g(x'|x)$ (proposal density),
- X_0 với $f(X_0) > 0$.

Output: xích Markov $X_1, X_2, \ldots$ có phân phối dừng là $\frac{1}{Z_f} f(x)$. Đặt

$$\alpha(x'|x) = \min\left(1, \frac{f(x')g(x|x')}{f(x)g(x'|x)}\right).$$

for $n = 1, 2, 3, \ldots$ **do**
    sinh $X' \sim g(.|X_{n-1})$ #sinh đề cử
    if $\mathcal{U}(0, 1) \leq \alpha(X'|X_{n-1})$ **then**
        $X_n \leftarrow X'$ #X' được chấp nhận với xác suất $\alpha(X'|X_{n-1})$
    else
        $X_n \leftarrow X_{n-1}$
end

Thuật toán Metropolis-Hastings - Ví dụ

Giả sử ta cần lấy mẫu cho hàm mật độ chưa chuẩn hóa $f(x) = 2^{-|x|}\mathbb{I}_{\mathbb{Z}}(x)$. Lưu ý

$$Z_f = \sum_{x \in \mathbb{Z}} f(x) = \left(... + \frac{1}{4} + \frac{1}{2} + 1 + \frac{1}{2} + \frac{1}{4} + ... \right) = \left(1 + 2\sum_{x=1}^{\infty} 2^{-x} \right) = 3$$

nhưng ta không cần tính Z_f.

Sử dụng thuật toán MH, ta dễ dàng sinh xích Markov có phân phối dừng là $\frac{1}{Z_f} f(x)$. Chẳng hạn, dùng hàm xác suất đề cử

$$g(X' = x + 1 | X = x) = g(X' = x - 1 | X = x) = \frac{1}{2}, x \in \mathbb{Z},$$

ta có hàm xác suất chấp nhận là

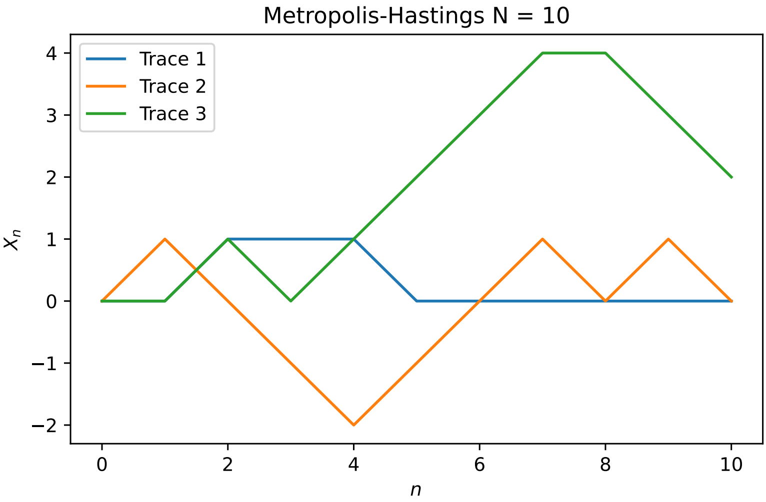
$$\alpha(x'|x) = \min \left(1, \frac{f(x')g(x|x')}{f(x)g(x'|x)} \right) = \min \left(1, \frac{2^{-|x'|}g(x|x')}{2^{-|x|}g(x'|x)} \right) = \begin{cases} 2^{|x|-|x'|} & |x'| > |x| \\ 1 & \text{khác} \end{cases}.$$

Thuật toán Metropolis-Hastings - Ví dụ (tt)

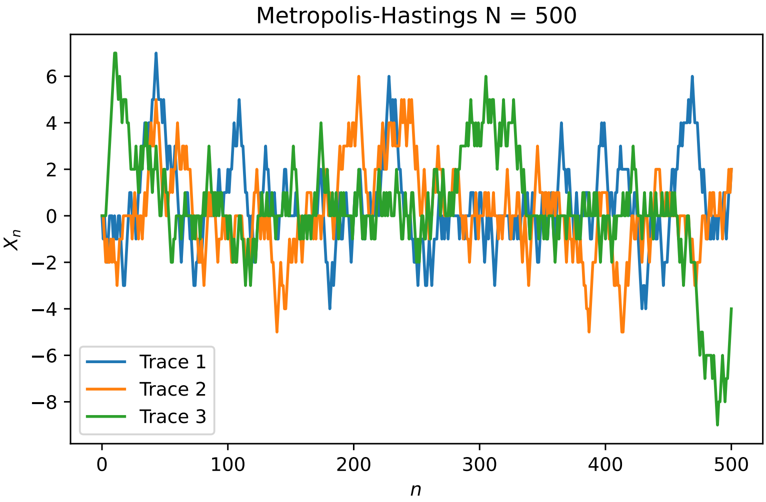
Dùng hàm xác suất đề cử ở trên, chọn $X_0 = 0$, ta có thuật toán MH sinh xích Markov có phân phối dừng là hàm mật độ chưa chuẩn hóa $f(x) = 2^{-|x|}\mathbb{I}_{\mathbb{Z}}(x)$.

for $n = 1, 2, 3, \ldots$ **do**
    sinh $\epsilon \sim \mathcal{U}\{-1, 1\}$
    $X' \leftarrow X_{n-1} + \epsilon$
    sinh $U \sim \mathcal{U}(0, 1)$
    if $U \leq 2^{|X_{n-1}|-|X'|}$ **then**
        $X_n \leftarrow X'$
    else
        $X_n \leftarrow X_{n-1}$
end

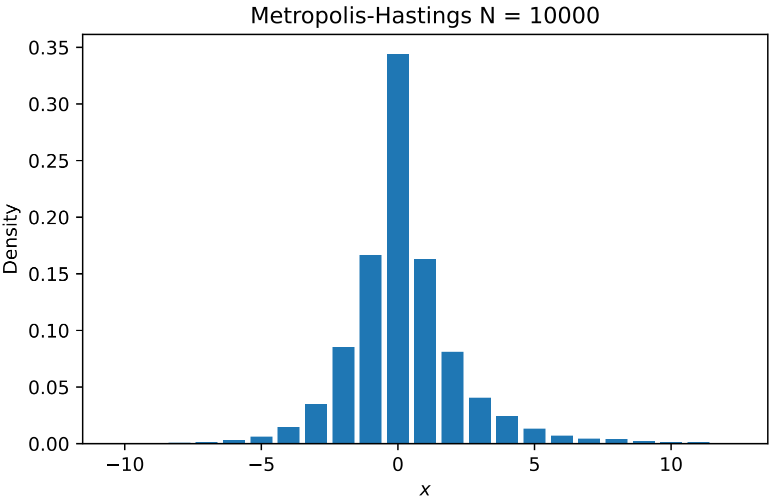
Thuật toán Metropolis-Hastings - Ví dụ (tt)



Thuật toán Metropolis-Hastings - Ví dụ (tt)



Thuật toán Metropolis-Hastings - Ví dụ (tt)



Thuật toán Metropolis

Trường hợp hàm mật độ xác suất đề cử đối xứng

$$g(x'|x) = g(x|x'), \forall x, x'$$

ta có hàm xác suất chấp nhận là

$$\alpha(x'|x) = \min\left(1, \frac{f(x')g(x|x')}{f(x)g(x'|x)}\right) = \min\left(1, \frac{f(x')}{f(x)}\right).$$

Khi đó, thuật toán Metropolis-Hastings được gọi là **thuật toán Metropolis**. Đặc biệt, khi đề cử có dạng

$$X' = X_{n-1} + \epsilon$$

với ϵ có phân phối đối xứng (tức là ϵ có cùng phân phối như $-\epsilon$) thì thuật toán được gọi là **bước ngẫu nhiên Metropolis** (random walk Metropolis).

Bước ngẫu nhiên Metropolis - Ví dụ

Trong trường hợp rời rạc, như Ví dụ trước, "độ dời" hay "bước" ϵ thường được dùng là $\mathcal{U}\{-1, 1\}$, tức là

$$P(\epsilon = -1) = P(\epsilon = 1) = \frac{1}{2}.$$

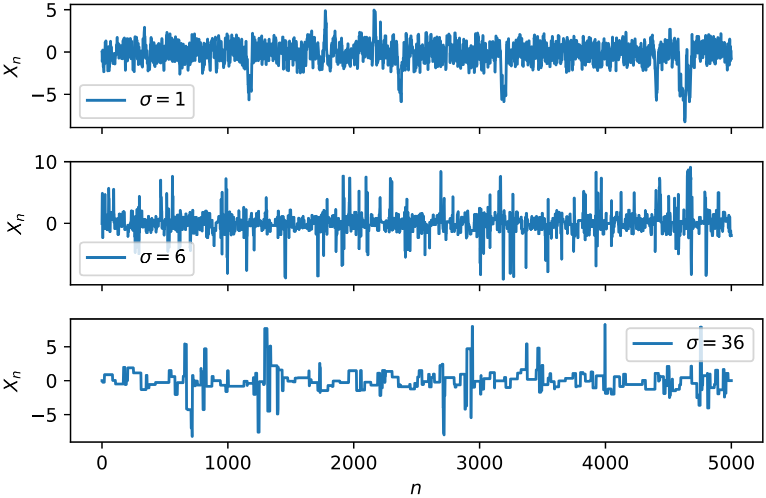
Trong trường hợp liên tục, độ dời thường được dùng là $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Trong đó σ^2 phải được chọn phù hợp để việc lấy mẫu hiệu quả.

Ví dụ: dùng thuật toán bước ngẫu nhiên Metropolis lấy mẫu từ phân phối có hàm mật độ

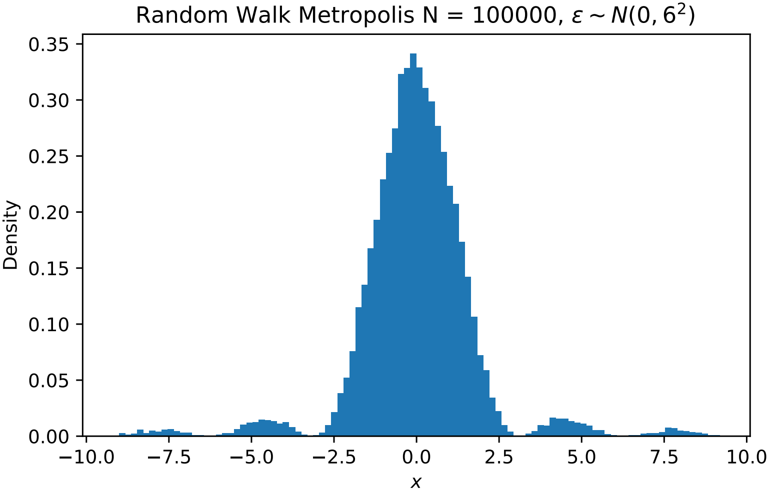
$$f(x) \propto \frac{\sin^2 x}{x^2} \mathbb{I}_{[-3\pi, 3\pi]}(x)$$

với độ dời $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Bước ngẫu nhiên Metropolis - Ví dụ (tt)



Bước ngẫu nhiên Metropolis - Ví dụ (tt)



Lấy mẫu Gibbs

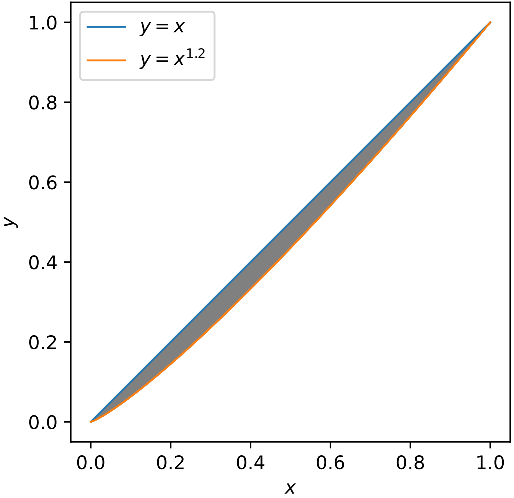
Thuật toán Gibbs. (Gibbs sampling)
Input:

- phân phối $f_{X_1,X_2,...,X_d}$,
- $X^{(0)} = \left(X_1^{(0)}, X_2^{(0)}, ..., X_d^{(0)}\right)$.

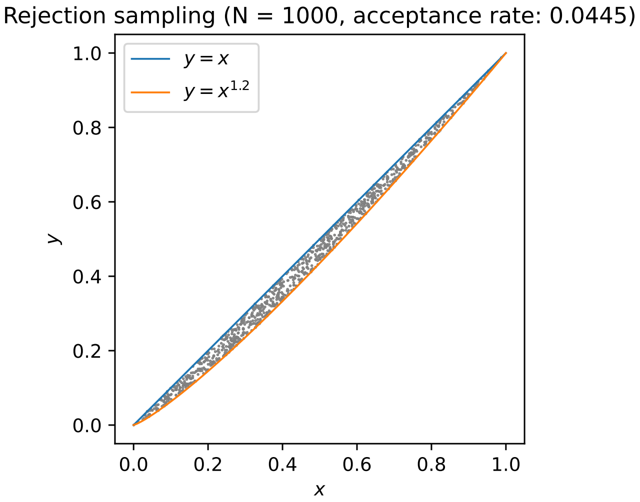
Output: xích Markov $X^{(1)}, X^{(2)}, ...$ có phân phối dừng là f.

for $n = 1, 2, 3, ...$ **do**
  for $i = 1, 2, ..., d$ **do**
    sinh $X_i^{(n)} \sim f_{X_i|\neg X_i}\left(.\middle|X_1^{(n)}, ..., X_{i-1}^{(n)}, X_{i+1}^{(n-1)}, ..., X_d^{(n-1)}\right)$
  end for
end for

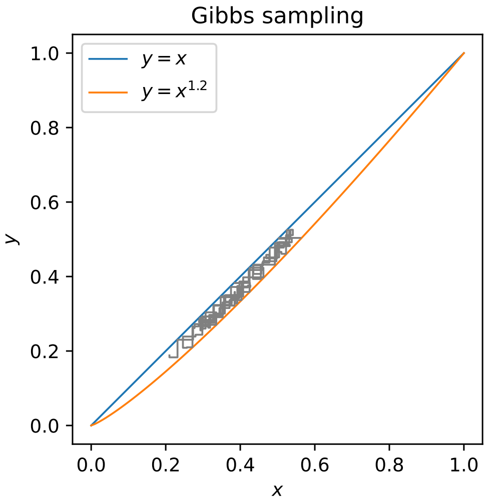
Lấy mẫu Gibbs - Ví dụ



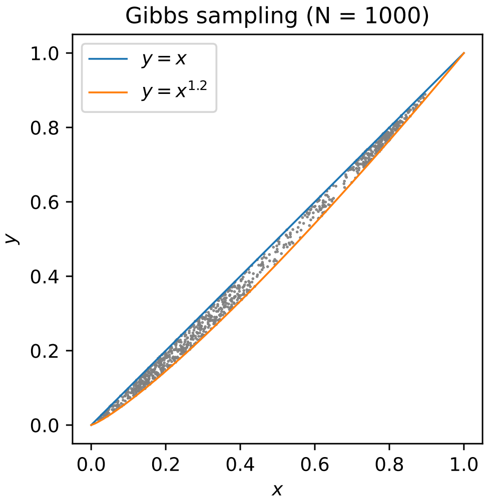
Lấy mẫu Gibbs - Ví dụ (tt)



Lấy mẫu Gibbs - Ví dụ (tt)



Lấy mẫu Gibbs - Ví dụ (tt)



Nội dung

Lập trình xác suất

Lập trình xác suất (Probabilistic Programming)

- `https://en.wikipedia.org/wiki/Probabilistic_programming`.
- Stan, PyMC, TensorFlow Probability (TFP), Pyro, ...

PyMC

- `https://www.pymc.io/welcome.html`
- Cameron Davidson-Pilon. *Bayesian Methods for Hackers*. Addision-Wesley, 2016. (`https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers`)
- Osvaldo Martin. *Bayesian Analysis with Python*. Packt Publishing, 2018.
- Osvaldo A. Martin, Ravin Kumar and Junpeng Lao. *Bayesian Modeling and Computation in Python*. CRC Press, 2022.

Nội dung

Mô hình tuyến tính tổng quát

- Ta muốn xác định sự phụ thuộc của một biến, gọi là **biến phụ thuộc** (dependent/predicted/response/output variable), vào một số biến khác, gọi là các **biến độc lập** (independent/predictor/explanatory/input variables).
- Các biến có thể được đo lường theo các **thang đo** (scale) khác nhau: metric, count, ordinal, nominal/categorical.
- Mô hình **hồi qui tuyến tính đơn** (simple linear regression)

$$y \sim \mathcal{N}(\beta_0 + \beta_1 x, \sigma^2),$$

  trong đó, y là metric response và x là metric explanatory.
- **Mô hình tuyến tính tổng quát** (generalized linear model - GLM)

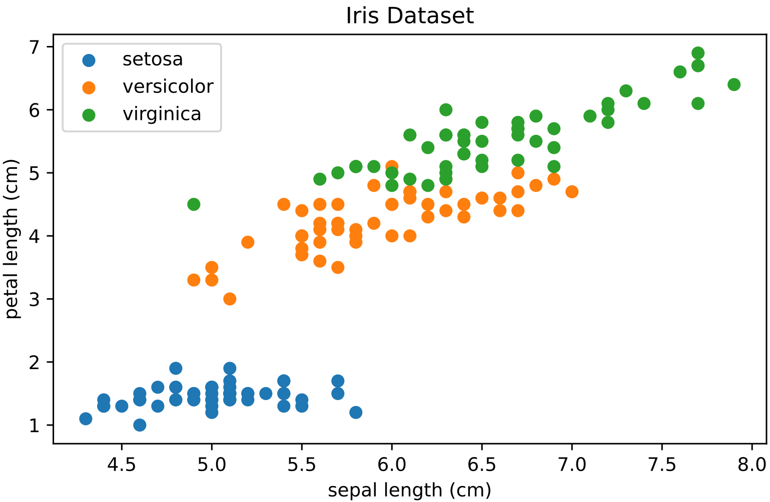
$$\begin{aligned} \text{lin}(x) &= \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k, \\ \mu &= f\left(\text{lin}(x), [\text{các tham số}]\right), \\ y &\sim \text{pdf}(\mu, [\text{các tham số}]). \end{aligned}$$

Mô hình tuyến tính tổng quát (tt)

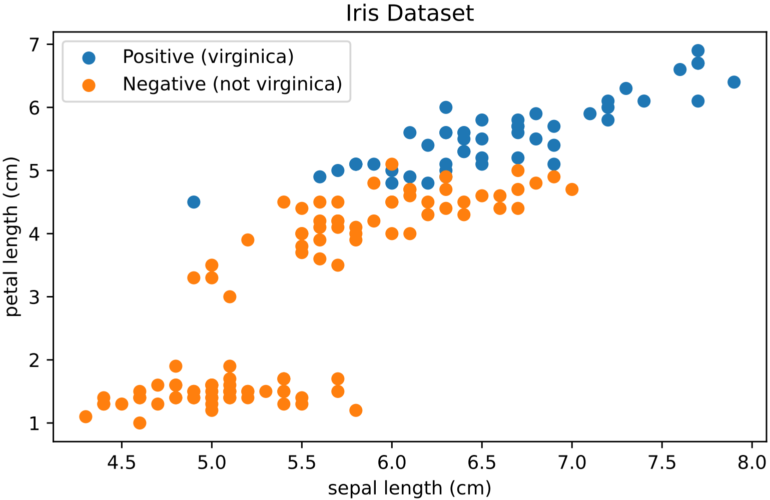
Scale type of predicted y	Typical noise distribution $y \sim \text{pdf}(\mu, [\text{parameters}])$	Typical inverse link function $\mu = f(\text{lin}(x), [\text{parameters}])$
Metric	$y \sim \text{normal}(\mu, \sigma)$	$\mu = \text{lin}(x)$
Dichotomous	$y \sim \text{bernoulli}(\mu)$	$\mu = \text{logistic}(\text{lin}(x))$
Nominal	$y \sim \text{categorical}(\ldots, \mu_k, \ldots)$	$\mu_k = \dfrac{\exp(\text{lin}_k(x))}{\sum_c \exp(\text{lin}_c(x))}$
Ordinal	$y \sim \text{categorical}(\ldots, \mu_k, \ldots)$	$\mu_k = \Phi((\theta_k - \text{lin}(x))/\sigma) - \Phi((\theta_{k-1} - \text{lin}(x))/\sigma)$
Count	$y \sim \text{poisson}(\mu)$	$\mu = \exp(\text{lin}(x))$

(John K. Kruschke. *Doing Bayesian Data Analysis – A Tutorial with R, JAGS, and Stan*.)

Ví dụ - Hồi qui logistic



Ví dụ - Hồi qui logistic (tt)



Ví dụ - Hồi qui logistic (tt)

Bài toán. Từ dữ liệu Iris, xác định sự phụ thuộc của class vào petal-length và sepal-length, cụ thể, phân biệt nhóm virginica với 2 nhóm còn lại.

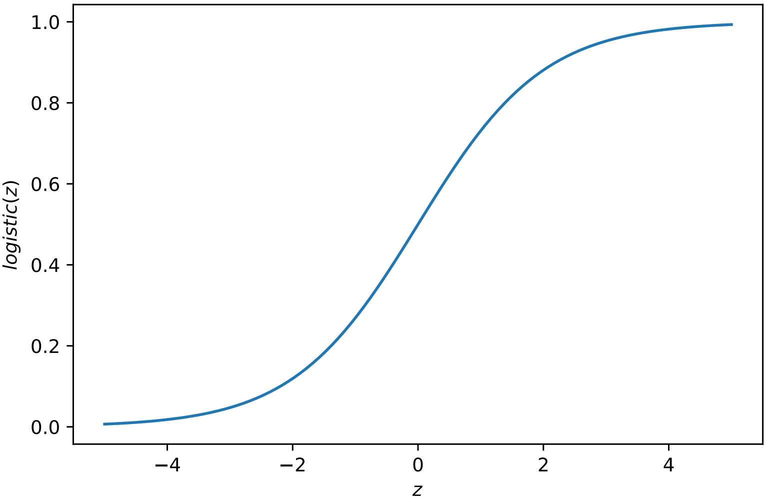
Suy diễn. "Từ gợi ý của dữ liệu", ta mô hình sự phụ thuộc của class (y, nhận giá trị 1 nếu là virginica và 0 nếu không) vào sepal-length (x_1) và petal-length (x_2) như sau

$$\begin{aligned}
\mu &= \text{logistic}(\beta_0 + \beta_1 x_1 + \beta_2 x_2),\\
y &\sim \text{Bernoulli}(\mu),\\
\beta_j &\sim \mathcal{N}(M_j, S_j^2), j = 0, 1, 2.
\end{aligned}$$

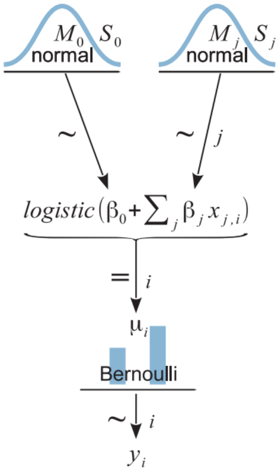
với

$$\text{logistic}(z) = \frac{1}{1 + e^{-z}}.$$

Ví dụ - Hồi qui logistic (tt)



Ví dụ - Hồi qui logistic (tt)

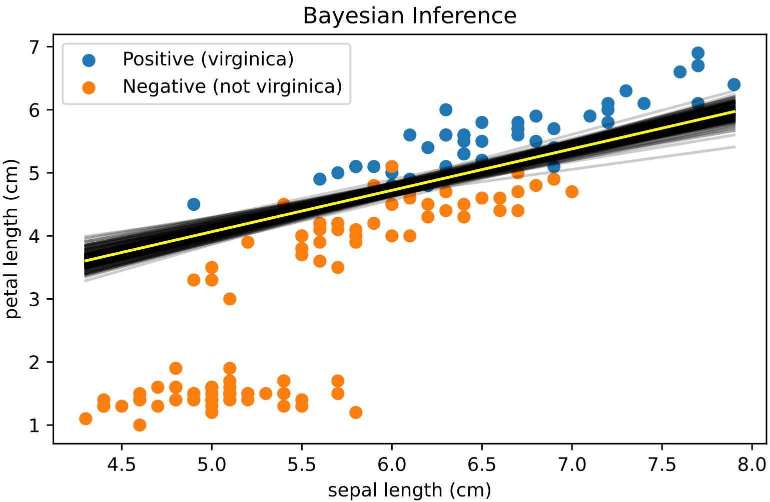


(John K. Kruschke. *Doing Bayesian Data Analysis – A Tutorial with R, JAGS, and Stan*.)

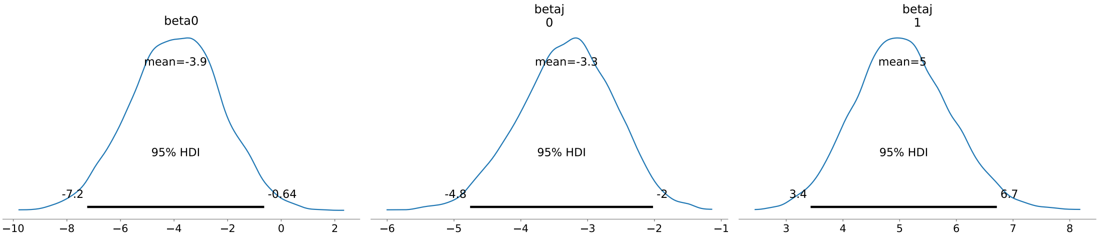
Ví dụ - Hồi qui logistic (tt)

```
with pm.Model() as model:
    beta0 = pm.Normal("beta0", mu=0, sd=2)
    betaj = pm.Normal("betaj", mu=0, sd=2, shape=X.shape
        [1])
    p = pm.invlogit(beta0 + pm.math.dot(X, betaj))
    y_var = pm.Bernoulli("y", p, observed=y)
    trace = pm.sample(5000, return_inferencedata=False)

pm.model_to_graphviz(model)
```

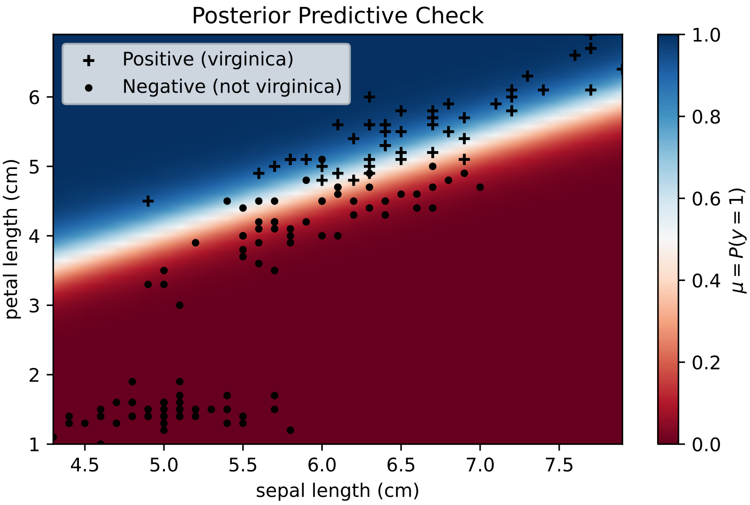
Ví dụ - Hồi qui logistic (tt)



Ví dụ - Hồi qui logistic (tt)



Ví dụ - Hồi qui logistic (tt)



Tài liệu tham khảo

Chapter 2, 6-9, 15, 21. John K. Kruschke. *Doing Bayesian Data Analysis – A Tutorial with R, JAGS, and Stan*. Elsevier, 2015.

Chapter 4. Jochen Voss. *An Introduction to Statistical Computing - A Simulation-based Approach*. John Wiley & Sons, 2014.