Inferenţă Bayesiană
Cursul 3

Programare şi modelare probabilistă - anul III

Facultatea de Informatică, UAIC

e-mail: adrian.zalinescu@uaic.ro

web: https://sites.google.com/view/fiicoursepmp/home

23 Octombrie 2023

Plan

Statistică şi modele

Ştiinţa statisticii are ca obiect:

- colectarea,
- organizarea,
- analizarea şi
- interpretarea datelor.

Ea se împarte în două mari direcţii:

- **statistică descriptivă** – despre sumare numerice (*medie*, *mod*, *deviaţie standard*, *cvartile*, etc);
- **statistică inferenţială** – pentru a face afirmaţii dincolo de datele curente: *înţelegerea* unui anumit fenomen, *predicţii* pentru viitor sau *alegerea între mai multe explicaţii* privind aceleaşi observaţii.

- Software-ul modern, precum PyMC, permite definirea şi rezolvarea modelelor statistice într-un mod relativ simplu.
- Multe din aceste modele erau de nerezovat acum câţiva ani, sau necesitau un nivel ridicat de înţelegere, atât matematică, cât şi tehnică.

Lucrul cu datele

- Datele sunt un ingredient esenţial în statistică (şi ştiinţa datelor).
- Datele vin din surse diverse, cum ar fi experimente, simulări pe computer, sondaje şi observaţii din teren.
- Dacă noi suntem cei responsabil cu generarea sau adunarea datelor, o idee bună este să ne întrebăm mai întâi la ce întrebări vom dori răspuns şi care sunt metodele ce vor fi utilizate ulterior.
- Ca regulă generală, putem presupune că procesele care au generat datele sunt stochastice (aleatoare), deoarece:
  - există dificultăţi tehnice ce implică adaugare de zgomot sau care ne împiedică să facem măsuratori de o precizie arbitrar de mică;
  - există limitări conceptuale care ascund detaliile.
- Din această cauză, datele trebuie întotdeauna analizate în contextul modelelor, incluzând pe cele mentale şi pe cele formale.
- Vom presupune că datele au fost deja strânse.

Modelare Bayesiană

- Modelele sunt descrieri simplificate ale unui anumit sistem sau proces.
- Aceste descrieri sunt în mod deliberat concepute astfel încât să capteze cele mai relevante aspecte ale sistemului.
- Detaliile minore nu sunt incluse în model.

Modelarea Bayesiană foloseşte următorii paşi:

- Folosind datele şi câteva presupuneri despre cum au fost generate aceste date, concepem un *model* prin combinarea elementelor de bază, anume *distribuţiile de probabilitate*.
- Folosim *formula lui Bayes* pentru a adăga datele modelului nostru şi a deduce consecinţele logice ale combinării datelor cu presupunerile făcute.
- Analizăm modelul verificând diverse criterii, incluzând datele, expertiza noastră asupra subiectului şi câteodată, comparându-l cu alte modele.

Distribuţii

Convenţii:

- $X, Y, Z, \ldots$: variabile aleatoare (v.a.)
- x, y, z: valori posibile ale v.a;
- x poate fi un vector: $x = (x_1, \ldots, x_n)$.

Exemplu de generare a valorilor unei variabile aleatoare $X \sim \mathcal{N}(\mu, \sigma)$:

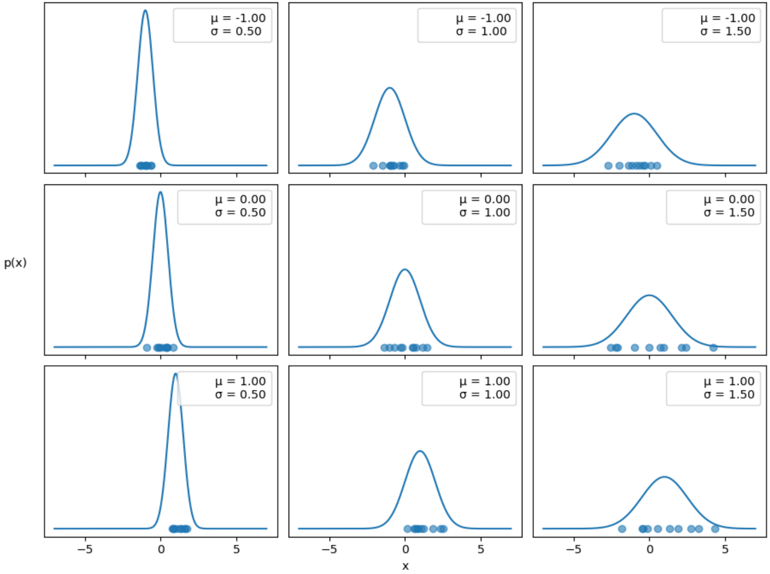
```
from scipy import stats
mu = 0.
sigma = 1.
X = stats.norm(mu, sigma)
x = X.rvs(3)
print(x)
```

Obs. 95% dintre valori vor fi în intervalul $[-1.96, 1.96]$

Exemplu de densităţi pentru distribuţii normale (cu diverşi parametri):

```
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats

mu_params = [-1, 0, 1]
sd_params = [0.5, 1, 1.5]
x = np.linspace(-7, 7, 200)
_, ax = plt.subplots(len(mu_params), len(sd_params), sharex=True,
sharey=True, figsize=(9, 7), constrained_layout=True)
for i in range(3):
    for j in range(3):
        mu = mu_params[i]
        sd = sd_params[j]
        X = stats.norm(mu, sd)
        y = X.pdf(x)
        z = X.rvs(10)
        ax[i,j].plot(x, y)
        ax[i,j].plot([], label="$\mu$ = {:3.2f}\n$\sigma$ = {:3.2f}".format(mu,sd), alpha=0)
        ax[i,j].scatter(z, np.zeros(10), alpha=0.6)
        ax[i,j].legend(loc=1)
ax[2,1].set_xlabel('x')
ax[1,0].set_ylabel('p(x)', rotation=0, labelpad=20)
ax[1,0].set_yticks([])
plt.show()
```


μ = -1.00
σ = 0.50
μ = -1.00
σ = 1.00
μ = -1.00
σ = 1.50
μ = 0.00
σ = 0.50
μ = 0.00
σ = 1.00
μ = 0.00
σ = 1.50
μ = 1.00
σ = 0.50
μ = 1.00
σ = 1.00
μ = 1.00
σ = 1.50
p(x)
−5
0
5
x

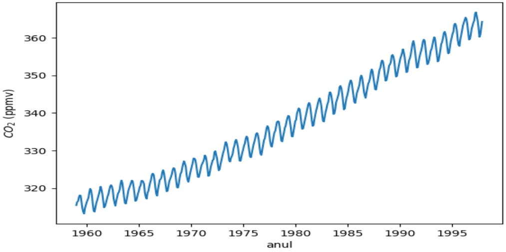
Variabile aleatoare independente

- Multe modele presupun că v.a. sunt generate din aceeaşi distribuţie şi în mod independent una de cealaltă.
- Spunem că ele sunt *identic şi independent distribuite* (*i.i.d.*).

Exemplu de variabilă aleatoare non-i.i.d.:

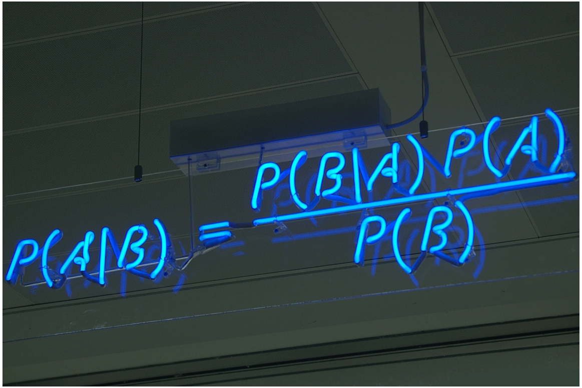
O *serie temporală*, unde o dependenţă temporală trebuie luată în calcul:

```
import numpy as np
import matplotlib.pyplot as plt
data = np.genfromtxt('./Data/mauna_loa_CO2.csv', delimiter=',')
plt.plot(data[:,0], data[:,1])
plt.xlabel('anul')
plt.ylabel('$CO_2$ (ppmv)')
plt.show()
```



record of atmospheric CO2 measurements from 1959 to 1997, `http://cdiac.esd.ornl.gov`

Formula lui Bayes

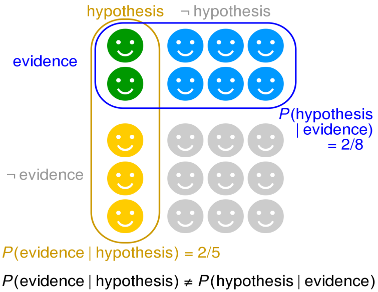


- θ: *parametrul* de care depind *ipotezele* făcute;
- y: *datele*.

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

- $p(\theta)$: probabilitatea a priori;
- $p(y|\theta)$: verosimilitatea (sau *plauzibilitatea*);
- $p(\theta|y)$: probabilitatea a posteriori;
- $p(y)$: verosimilitatea marginală.

Intuitiv, $$p(ipoteză|date) = \frac{p(date|ipoteză)p(ipoteză)}{p(date)}$$



Sursă: `https://en.wikipedia.org/wiki/Prosecutor's_fallacy`

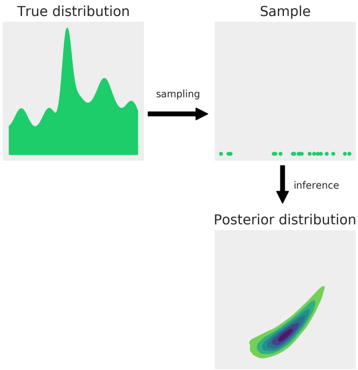
Aici, ipoteza reprezintă posibilitatea ca acuzatul să fie vinovat, în timp ce probele se referă la un test pozitiv, cum ar fi potrivirea de ADN sau de grupă de sânge.

Bayes: $$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

- Distribuţia (probabilitatea) *a priori*, $p(\theta)$, reprezintă ceea ce credem despre valoarea parametrului θ *înainte* de a vedea datele. Dacă nu ştim nimic, putem considera o distribuţie *plată* (*uniformă*).
- *Verosimilitatea*, $p(y|\theta)$ este modul în care introducem datele în analiză; este o expresie a plauzibilităţii datelor în funcţie de parametrul θ.
- Combinaţia între cele două, distribuţia a priori şi verosimilitatea, se numeşte *model*.

- Distribuţia (probabilitatea) *a posteriori*, $p(\theta|y)$, este rezultatul analizei Bayesiene şi reflectă tot ce ştim despre o anumită problemă (o dată ce avem datele şi modelul).
- Conceptual, putem gândi probabilitatea a posteriori drept o actualizare a probabilităţii a priori în lumina evidenţelor (datelor).
- Ultimul termen, *verosimilitatea marginală*, $p(y)$ este din punct de vedere formal probabilitatea de a observa datele ponderată de toate valorile posibile pe care parametrul θ le poate lua (conform distribuţiei a priori).
- De multe ori, aceasta este neglijată, întrucât formula lui Bayes implică

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



Sumio Watanabe, *Mathematical Theory of Bayesian Statistics*, 2018

Exemplu de inferenţă (uni-parametrică)

Problema aruncării unei monede (sau modelul beta-binomial)**:**

- Se aruncă o monedă de un număr de ori şi se înregistrează de câte ori avem stemă (sau ban);
- Bazat pe aceste date, ne punem problema dacă moneda este măsluită? Sau cât de măsluită este?
- Este un model simplu, ce se poate rezolva şi calcula cu uşurinţă.
- Parametrul θ va reprezenta probabilitatea necunoscută de a obţine stemă.
- Pentru a reprezenta numărul de steme dintr-un total de N aruncări, vom utiliza variabila y.

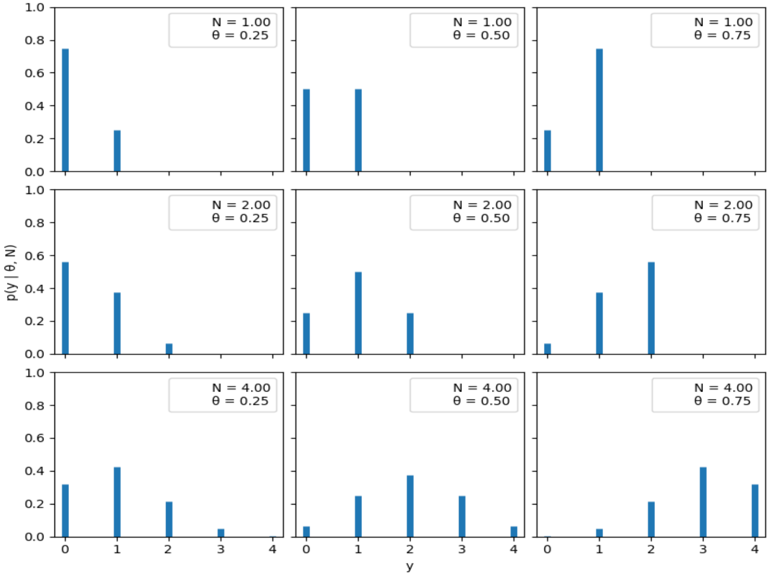
Alegerea verosimilităţii

- Aruncarea de N ori a unei monede generează N v.a. i.i.d., fiecare distribuită *Bernoulli*, de parametru θ.
- Astfel, suma acestora, care dă numărul de steme, este distribuită *binomial*, $y|\theta \sim \mathrm{Bin}(N, \theta)$:

$$p(y|\theta) = C_N^y\, \theta^y (1-\theta)^{N-y} = \frac{N!}{y!(N-y)!}\theta^y (1-\theta)^{N-y}.$$

Exemple de distribuţii binomiale:

```
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
n_params = [1, 2, 4]  # Number of trials
p_params = [0.25, 0.5, 0.75]  # Probability of success
x = np.arange(0, max(n_params)+1)
f,ax = plt.subplots(len(n_params), len(p_params), sharex=True, sharey=True,
                    figsize=(8, 7), constrained_layout=True)
for i in range(len(n_params)):
    for j in range(len(p_params)):
        n = n_params[i]
        p = p_params[j]
        y = stats.binom(n=n, p=p).pmf(x)
        ax[i,j].vlines(x, 0, y, colors='C0', lw=5)
        ax[i,j].set_ylim(0, 1)
        ax[i,j].plot(0, 0, label="N = {:3.2f}\nθ = {:3.2f}".format(n,p), alpha=0)
        ax[i,j].legend()
ax[2,1].set_xlabel('y')
ax[1,0].set_ylabel('p(y | θ)')
ax[0,0].set_xticks(x)
plt.show()
```

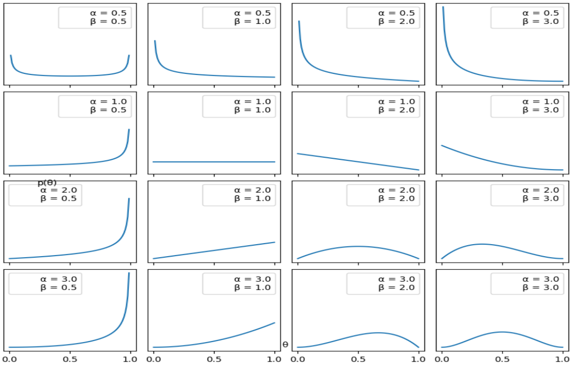

N = 1.00
θ = 0.25
N = 1.00
θ = 0.50
N = 1.00
θ = 0.75
N = 2.00
θ = 0.25
N = 2.00
θ = 0.50
N = 2.00
θ = 0.75
N = 4.00
θ = 0.25
N = 4.00
θ = 0.50
N = 4.00
θ = 0.75
p(y | θ, N)
y
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4

Alegerea probabiltăţii a priori

Vom folosi o distribuţie de tip *Beta*, a cărei densitate este dată de

$$p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1},\ \theta \in [0,1],$$

unde $\alpha, \beta > 0$ şi Γ este funcţia *Gamma*. Folosind un script asemănător pentru cel cu distribuţia normală, dăm câteva grafice ale acestei densităţi, variind α şi β.



În acest exemplu, în afară de forma ei, distribuţia Beta este folosită şi din cauză că este *conjugata a priori* a distribuţiei binomiale, adică:

- folosită cu o verosimilitate binomială, obţinem o distribuţie a posteriori de acelaşi tip, i.e. distribuţie Beta.

Demonstraţie:

$$\text{Bayes:} \qquad p(\theta|y) \propto p(y|\theta)p(\theta);$$

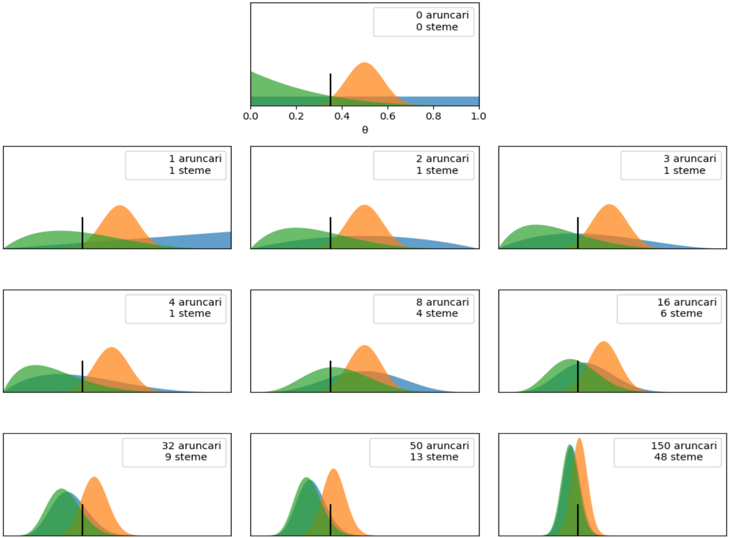
Dar

$$\begin{aligned} p(y|\theta)p(\theta) &= \frac{N!}{y!(N-y)!}\theta^y(1-\theta)^{N-y}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &\propto \theta^y(1-\theta)^{N-y}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &= \theta^{y+\alpha-1}(1-\theta)^{N-y+\beta-1}, \end{aligned}$$

deci $p(\theta|y)$ este o distribuţie de tip Beta, cu parametrii $y+\alpha$ şi $N-y+\beta$.

Calculul şi reprezentarea grafică a distribuţiei a posteriori

```
plt.figure(figsize=(10, 8))
n_trials = [0, 1, 2, 3, 4, 8, 16, 32, 50, 150]
data = [0, 1, 1, 1, 1, 4, 6, 9, 13, 48]
theta_real = 0.35
beta_params = [(1, 1), (20, 20), (1, 4)]
dist = stats.beta
x = np.linspace(0, 1, 200)
for idx, N in enumerate(n_trials):
    if idx == 0:
        plt.subplot(4, 3, 2)
        plt.xlabel('θ')
    else:
        plt.subplot(4, 3, idx+3)
        plt.xticks([])
    y = data[idx]
    for (a_prior, b_prior) in beta_params:
        p_theta_given_y = dist.pdf(x, a_prior + y, b_prior + N - y)
        plt.fill_between(x, 0, p_theta_given_y, alpha=0.7)
    plt.axvline(theta_real, ymax=0.3, color='k')
    plt.plot(0, 0, label=f'{N:4d} aruncari\n{y:4d} steme', alpha=0)
    plt.xlim(0, 1)
    plt.ylim(0, 12)
    plt.legend()
    plt.yticks([])
plt.tight_layout()
plt.show()
```



Primul subgrafic arată distribuţia a priori, iar următoarele distribuţiile a posteriori corespunzătoare, pentru diferite valori ale datelor.

Observaţii:

- Rezultatul analizei Bayesiene este distribuţia a posteriori, nu doar o singură valoare a parametrului θ.
- Cea mai probabilă valoare a lui θ se numeşte *modul* distribuţiei (corespunzătoare vârfului graficului).
- Alungirea distribuţiei a posteriori e proporţională cu gradul de incertitudine asupra parametrului.
- Suntem mai încrezători în rezultatul analizei atunci când avem o cantitate mare de date; acest lucru se reflectă şi în graficul de mai sus – alungirea distribuţiei e din ce în ce mai mică pe măsură ce N creşte.
  - Pentru $N \to \infty$, distribuţiile a posteriori tind să se confunde, indiferent de distribuţiile a priori cu care am pornit.

Alegerea distribuţiei a priori (din nou):

- distribuţii a priori *non-informative*, cunoscute drept distribuţii *plate* (*vagi*, *difuze*): au cel mai mic impact asupra analizei.
- distribuţii a priori *slab-informative* [Gelman, McElreath, Kruschke]: se folosesc informaţiile cunoscute despre parametru. De exemplu,
  - ia valori pozitive;
  - ne aşteptăm ca valorile să aparţină unui anumit interval (domeniu).

  În abordarea frecvenţială, o metodă des întâlnită este de a estima parametrii prin *verosimilitatea maximă*:

$$\hat{\theta} = \theta_{MLE} = \operatorname{argmax}_{\theta} p(y|\theta).$$

  - în exemplul precedent, $\hat{\theta} = y/N$.
  - acest caz coincide cu alegerea modului distribuţiei a posteriori atunci când alegem o distribuţie a priori uniformă (plată).

Descrierea modelului

Pentru modelul anterior, o posibilitate de a-l descrie succint este:

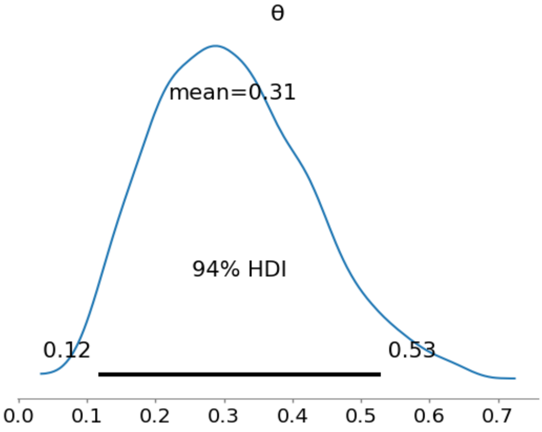
$$\begin{aligned} \theta &\sim \text{Beta}(\alpha, \beta) \\ y &\sim \text{Bin}(n = N, p = \theta) \end{aligned}$$

- La primul nivel, avem distribuţia a priori, $p(\theta)$;
- la al doilea, verosimilitatea, $p(y|\theta)$.

Rezumatul distribuţiei a posteriori

- Conţine informaţia esenţială obţinută din distribuţia a posteriori în urma procesului de inferenţă.
- Este rezumatul consecinţelor logice ale modelului şi datelor.
- O practică uzuală este de a oferi următoarele date despre distribuţia a posteriori:
  - *media* (*mediana*, *modul*): ca măsură a *tendinţei centrale*;
  - *deviaţia standard* (*cvartile*): ca măsură a *dispersiei*.
- O altă unealtă standard pentru măsurarea dispersiei este *intervalul HDI* (*Highest Density Interval*):
  - cel mai mic interval care conţine o anumită porţiune a densităţii distribuţiei a posteriori;
  - des întâlnite sunt intervale 94%HDI sau 50%HDI.

```python
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
import arviz as az
np.random.seed(1)
az.plot_posterior({'θ':stats.beta.rvs(5, 11, size=1000)})
plt.show()
```



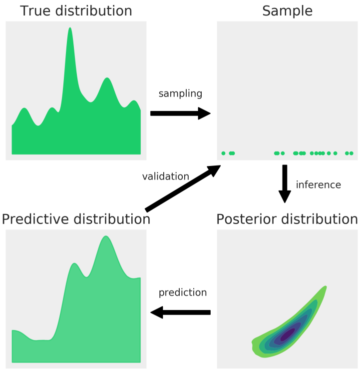
Predicţii bazate pe date
Odată ce distribuţia a posteriori a fost generată pe baza analizei Bayesiene, o proprietate foarte utilă este utilizarea acestei pentru a face/genera predicţii ale sistemului:

- Distribuţia *predictivă a posteriori* este dată de:

$$p(\hat{y}|y) = \int p(\hat{y}|\theta)p(\theta|y)d\theta$$

- Această integrală se aproximează printr-un proces în doi paşi:
  - se generează o valoare θ conform distribuţiei a posteriori, $p(\cdot|y)$;
  - se *hrăneşte* (*feed*) verosimilitatea $p(\cdot|\theta)$ cu această valoare θ, generând o valoare prezisă $\tilde{y}$.
- Pe lângă generarea de predicţii, acest procedeu serveşte şi la analiza critică a modelului, prin compararea setului de date y cu setul de date prezise $\tilde{y}$.
- Metoda de mai sus se mai numeşte *verificare predictivă a posteriori* (*ulterioară*).

Rezumat:



Sumio Watanabe, *Mathematical Theory of Bayesian Statistics*, 2018