

Mějme jevy A, B . To není dobrý začátek. Jevy jsou velmi abstraktní, bude proto lepší představit si konkrétní jev A , např. hod kostkou, při němž padne šestka. Nastává s určitou pravděpodobností $p(A)$; např. zde $p(A) = \frac{1}{6}$. Podmíněná pravděpodobnost $p(A|B)$ je pravděpodobnost, že nastane jev A , když nastal jev B ; např. že při hodu padlo číslo větší nebo rovno čtyřem — pak je ovšem $p(B) = \frac{1}{2}$, $p(A|B) = \frac{1}{3}$ a $p(B|A) = 1$. Pravděpodobnost, že nastanou *současně* jevy A i B je zřejmě:

$$p(A \wedge B) = p(A|B)p(B) = p(B|A)p(A);$$

v našem hazardním příkladu vskutku $\frac{1}{3} \cdot \frac{1}{2} = 1 \cdot \frac{1}{6}$. Zároveň odtud plyne vztah mezi podmíněnými pravděpodobnostmi:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}, \quad (1)$$

neboli Bayesova věta, kterou hodláme využít pro následující úlohu, která se netýká hazardu.

Základní úlohou je totiž aproximace přímkou (srov. obr. 3):

$$y = ax + b, \quad (2)$$

kde a označuje směrnici přímky, b nulový bod (kde protíná osu \hat{y}). Měření nezávislé veličiny x_i a závislé y_i , které vždy vykazují nejistoty σ_i (příslušné y_i), tuto rovnici nemohou splňovat přesně, $y_i \neq ax_i + b$ pro $\forall i$. Otázkou je, jak nejlépe volit a, b , a především, jaké jsou jejich nejistoty σ_a, σ_b . V astronomii se s úlohou se setkáváme neustále: při měření rychlosti, atmosférické extinkce, periody v diagramu $O-C$, radiometrického stáří, Hubblovy konstanty, atd. atd.

Ztotožníme-li jev $A \dots a, b$ s parametry, $B \dots x_i, y_i, \sigma_i$ s měřeními, pak (1) přejde na:

$$p(a, b|x_i, y_i, \sigma_i) = \frac{p(x_i, y_i, \sigma_i|a, b)p(a, b)}{p(x_i, y_i, \sigma_i)}, \quad (3)$$

kde $p(x_i, y_i, \sigma_i)$ označuje pravděpodobnost, že při měření obdržíme právě tato měření¹, $p(a, b)$ *a priori* (předchozí) pravděpodobnost parametrů, $p(a, b|x_i, y_i, \sigma_i)$ *a posteriori* (následnou) pravděpodobnost parametrů při daných měřeních a konec konců $p(x_i, y_i, \sigma_i|a, b)$ pravděpodobnost měření při daných parametrech (věrohodnost, angl. likelihood).

Otázkou je, kde je vzít? $p(x_i, y_i, \sigma_i)$ často neznáme, nicméně ji lze získat renormalizací. $p(a, b)$ obvykle známe z literatury, nevíme-li nic, jsme hloupí a volíme

1. rozlišujeme (nedostupný) celý soubor X_i od podmnožiny souboru x_i

konst., $p(a, b|x_i, y_i, \sigma_i)$ vypočítáme z (3). Pro $p(x_i, y_i, \sigma_i|a, b)$ ovšem potřebujeme nějaký věrohodný předpis.

Za předpokladu, že odchylky měření jsou pouze náhodné, se používá rozdělení pravděpodobnosti dle Gaussovy funkce:

$$p(x; \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}, \quad (4)$$

kde x dočasně označuje odchylku (od 0). V dalším budeme potřebovat též výraz $J \equiv -\ln p = \ln \sigma + \ln \sqrt{2\pi} + \frac{x^2}{2\sigma^2}$. V našem případě ztotožníme $x \dots y_i - ax_i - b$, $\sigma \dots \sigma_i$, čímž obdržíme funkcionál:

$$J = \sum_i \left[\frac{(y_i - ax_i - b)^2}{2\sigma_i^2} + \ln \sigma_i + \ln \sqrt{2\pi} \right]. \quad (5)$$

Minimalizace J vzhledem k a, b zajistí maximalizaci $p(x_i, y_i, \sigma_i|a, b)$. Numericky to můžeme provést např. metodou simplexu (Nelder a Mead 1965).²

Složitější úlohou je případ s nejistotami v obou osách:

$$J = \frac{1}{2} \sum_i \left[\frac{(y_i - ax_i - b)^2}{\sigma_{y_i}^2 + a^2 \sigma_{x_i}^2} + \ln(\sigma_{y_i}^2 + a^2 \sigma_{x_i}^2) + \ln(2\pi) \right],$$

kde minimalizace J vzhledem k a, b dá obecně odlišný výsledek.

Ještě složitější úlohou je případ s podceněnými nejistotami σ_i , obzvlášť když se mezi měřeními vyskytuje odlehlý bod. Užijeme proto faktor f , ovšem v kvadrátu:

$$J = \frac{1}{2} \sum_i \left[\frac{(y_i - ax_i - b)^2}{f^2 \sigma_i^2} + \ln(f^2 \sigma_i^2) + \ln(2\pi) \right] \quad (6)$$

a minimalizaci J provedeme vzhledem k a, b, f , přičemž optimální hodnoty označíme $\bar{a}, \bar{b}, \bar{f}$.

Nezajímá nás však $p(x_i, y_i, \sigma_i|a, b)$, nýbrž $p(a, b|x_i, y_i, \sigma_i)$! Zkusme si to ostatně říci slovně. Proto potřebujeme prior $p(a, b)$. Například se může jednat o výsledek dřívějších měření:

$$p(a; \mu_a, \sigma_a) = \frac{1}{\sigma_a \sqrt{2\pi}} e^{-\frac{(a - \mu_a)^2}{2\sigma_a^2}}, \quad (7)$$

2. metoda nejmenších čtverců, minimalizující $\chi^2 \equiv \sum_i (y_i - ax_i - b)^2 / \sigma_i^2$, poskytuje pro a, b analytické řešení

nebo pouze o meze (přípustné dle příslušných fyzikálních zákonů):

$$p(a) = \begin{cases} \frac{1}{a_{\max} - a_{\min}} & \text{pro } a_{\min} < a < a_{\max}, \\ 0 & \text{jinak.} \end{cases}$$

Obdobně pro b, f .

Zároveň potřebujeme nejistoty parametrů σ_a, σ_b . Nejistota σ_f také ovlivní σ_a, σ_b ! Nezbytné je proto vzorkování funkcionálu $J(a, b, f)$, pro což se používá: (i) metoda Monte Carlo; (ii) metoda Markovových řetězců. O tom, kam se posuneme v a, b nebo f rozhoduje náhodné číslo $x \in \langle 0; 1 \rangle$; je-li menší než určitá pravděpodobnost, $x < p$, dojde k posunu. Pravděpodobnosti p jsou určeny výhradně ze současného stavu (a, b, f) , tzn. že se jedná o řetězec (angl. chain, zkr. MCMC). Pro efektivní vzorkování se užívá zároveň vícero řetězců. Počáteční podmínky se volí v okolí $\bar{a}, \bar{b}, \bar{f}$. Po určité době se statistické rozdělení zaznamenaných hodnot a, b, f již nemění, i když hodnoty oscilují (viz obr. 1). Vytvoří se rovnováha mezi $p(x_i, y_i, \sigma_i | a, b)$ a $p(a, b)$, což je právě hledané $p(a, b | x_i, y_i, \sigma_i)$. Vyřadíme-li počáteční přechodový stav, lze kreslit izočáry, počítat percentily nebo studovat korelace.

Pro naše účely je f *zbytný* parametr (angl. nuisance), nezajímavá hodnota. Proto se provádí součet přes všechny hodnoty f , čímž získáme dvourozměrné rozdělení a, b , nebo i součet přes b , resp. a , čili jednorozměrná rozdělení a , resp. b (obráz. 2). Obvykle se píše se na okraj tabulky a kreslí na okraj obrázku, proto se postupu říká *marginalizace*.

Prakticky postup naprogramujeme v Pythonu [1]:

```
#!/usr/bin/env python

"""
bayes.py
Aproximace primkou, vc. vypoctu nejistot (Monte Carlo Markov Chain; MCMC)
"""

import numpy as np
import scipy.optimize
import matplotlib.pyplot as plt
import emcee
import corner

def minus_ln_likelihood(params, x, y, sigma):
    """Verohodnost -ln p(x_i, y_i, sigma_i | a, b, f)"""
    a, b, f = params
    model = a*x + b
    s2 = (f*sigma)**2
    J = 0.5*np.sum((y-model)**2/s2 + np.log(s2) + np.log(2.0*np.pi))
    print "J = ", J
    return J
```

```

def minus_ln_prior(params):
    """Prior -ln p(a,b,f)"""
    a, b, f = params
    if 0.1 < a < 2.0 and 0.0 < b < 2.0 and 0.5 < f < 5.0:
        return 0.0
    else:
        return -np.inf

def ln_posterior(params, x, y, sigma):
    """Posterior ln p(a,b,f|x_i,y_i,sigma_i)"""
    tmp = minus_ln_prior(params)
    if not np.isfinite(tmp):
        return -np.inf
    else:
        return -(minus_ln_prior(params) + minus_ln_likelihood(params, x, y, sigma))

def main():
    """Nacteni dat, maximalizace verohodnosti, vypocet MCMC, rohovy obrazek, ..."""
    x, y, sigma = np.loadtxt("xy.dat", usecols=[0,1,2], unpack=True)

    a = 1.0
    b = 0.0
    f = 1.0
    result = scipy.optimize.minimize(lambda *args: minus_ln_likelihood(*args), \
        [a,b,f], args=(x,y,sigma), method='Nelder-Mead', tol=1.0e-4)
    a, b, f = result.x
    print result

    ndim = 3
    walkers = 32
    position = [ result.x + 1.0e-4*np.random.rand(ndim) for i in range(walkers) ]
    sampler = emcee.EnsembleSampler(walkers, ndim, ln_posterior, args=(x,y,sigma))
    sampler.run_mcmc(position, 500)

    fig = plt.figure()
    plt.subplot(311)
    plt.plot(sampler.chain[:, :, 0].transpose())
    plt.xlabel("krok")
    plt.ylabel("$a$")
    plt.subplot(312)
    plt.plot(sampler.chain[:, :, 1].transpose())
    plt.ylabel("$b$")
    plt.subplot(313)
    plt.plot(sampler.chain[:, :, 2].transpose())
    plt.ylabel("$f$")
    plt.savefig("chain.png")

    samples = sampler.chain[:, 100:, :].reshape((-1, ndim))

    fig = corner.corner(samples, labels=["$a$", "$b$", "$f$"], truths=[a,b,f])
    fig.savefig("corner.png")

    fig = plt.figure()

```

```

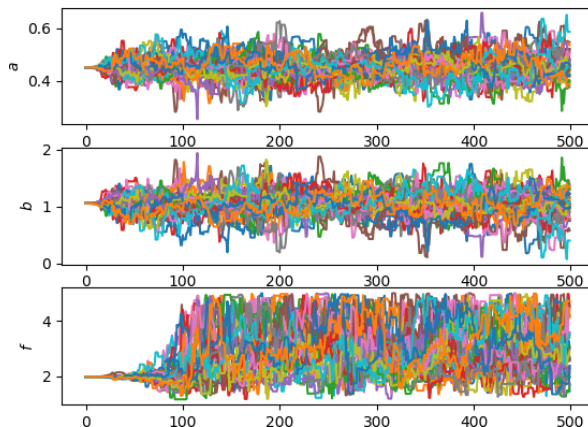
for a_tmp, b_tmp, f in samples[np.random.randint(len(samples), size=100)]:
    plt.plot(x, a_tmp*x + b_tmp, color='k', alpha=0.1)
plt.plot(x, a*x + b, color='r')
plt.errorbar(x, y, yerr=sigma, fmt='k+', ecolor='y', capsize=2.0)
plt.xlabel("$x$")
plt.ylabel("$y$")
plt.savefig("xy.png")

a, b, f = map(lambda x: (x[1], x[2]-x[1], x[0]-x[1]), \
    zip(*np.percentile(samples, [16,50,84], axis=0)))
print "a = ",
a print "b = ", b
print "f = ", f

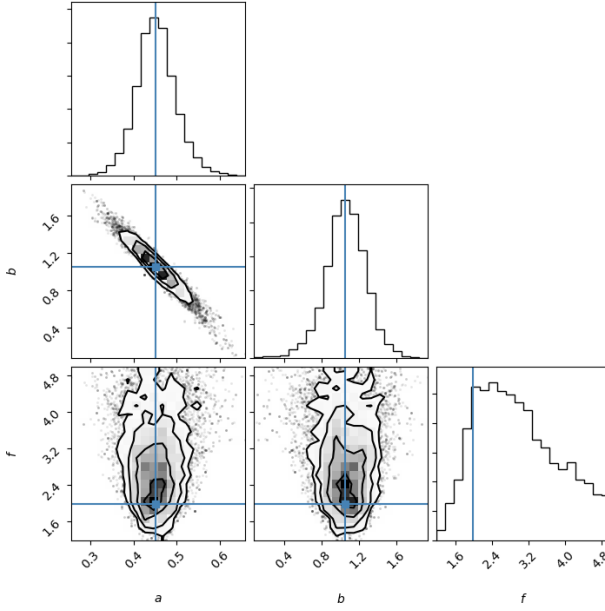
if __name__ == "__main__":
    main()

```

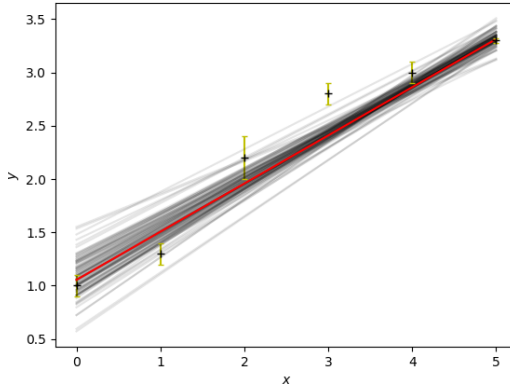
- [1] FOREMAN-MACKEY, D. AJ. *Example: Fitting model to data*. [online] [cit. 2018-12-03]. <http://dfm.io/emcee/current/user/line/>.
- [2] NELDER, J. A., MEAD, R. *A simplex method for function minimization*. Comput. J., **7**, s. 308–313, 1965.
- [3] SHAVER, B. *A zero-math introduction to Markov chain Monte Carlo methods*. [online] [cit. 2018-12-03]. <https://towardsdatascience.com/a-zero-math-introduction-to-markov-chain-monte-carlo-methods-dcba889e0c50>.



Obr. 1 — Určování nejistot parametrů a, b, f pro aproximaci přímkou $y_i = ax_i + b$ pomocí Markovových řetězců, které vzorkují a posteriorní pravděpodobnost. Prvních 100 kroků bude následně vynecháno, neboť až poté je dosaženo rovnovážného stavu.



Obr. 2 — Rohový obrázek pro marginalizované pravděpodobnosti $p(a)$, $p(b)$, $p(f)$ na diagonále (tzn. jednorozměrné histogramy), a $p(a, b)$, $p(a, f)$, $p(b, f)$ pod diagonálou (dvourozměrné grafy s izočarami odpovídajícími četnosti). Zřejmá je negativní korelace mezi a , b . Nejistoty σ_a , σ_b , σ_f se pak určují jako percentily. Modře jsou vyznačeny parametry a , b , f pro maximální věrohodnost $p(x_i, y_i, \sigma_i | a, b)$.



Obr. 3 — Data x_i, y_i a jejich nejistoty σ_i , spolu s aproximací přímkou $y = ax + b$ (červeně) a stovkou přímek náhodně vybraných z Markovova řetězce, což naznačuje nejistoty σ_a, σ_b . Výsledné hodnoty parametrů jsou: $a = 0,450^{+0,044}_{-0,046}$, $b = 1,061^{+0,217}_{-0,198}$, $f = 2,90^{+1,09}_{-0,85}$. Je-li přímkový model vůbec platný, pak jsou nejistoty měření značně podceněné.