Wahrscheinlichkeitstheorie und Statistik

Serie 11 - Lösungen

Aufgabe 11-1 (Lösungsbegriff der Bayesianische Optimierung). Für eine nicht-negative Zufallsvariable L auf einem Wahrscheinlichkeitsraum $(\Theta, \mathcal{A}, \pi_0)$ definieren wir rekursiv die Folge von Wahrscheinlichkeitsmassen

$$\pi_n(d\theta) = \frac{\exp(-L(\theta))\pi_{n-1}(d\theta)}{\int_\Theta \exp(-L(\theta))\pi_{n-1}(d\theta)}, \qquad n = 1, 2, \ldots$$

Zeigen Sie, dass $\lim_{n\to\infty} \pi_n(A) = 0$ für jedes Ereignis $A \in \mathcal{A}$, sodass

$$\exists \delta > 0 : \qquad \operatorname{ess\,inf}_{\theta \in A} L(\theta) \geq \delta + \operatorname{ess\,inf}_{\theta\in\Theta} L(\theta).$$

Die Interpretation dieser Aussage ist, dass sich π_n für grosses n an Minima von L konzentriert.

Lösung: Wir greifen auf die in der Vorlesung bewiesenen Aussage zurück, dass

$$m := \text{ess-inf}_{\theta\in\Theta} L(\theta) = -\lim_{n\to\infty} \frac{1}{n} \log\Big(\int_\Theta \exp(-nL(\theta))\pi_0(d\theta)\Big).$$

Dann gilt auf A fast sicher, dass $L \geq \delta + m$ und daher:

$$\begin{aligned} -\frac{1}{n}\log \pi_n(A) &= -\frac{1}{n}\log \frac{\int_A \exp(-nL(\theta))\pi_0(d\theta)}{\int_\Theta \exp(-nL(\theta))\pi_0(d\theta)} \\ &\geq \delta + m + \frac{1}{n}\log \int_\Theta \exp(-nL(\theta))\pi_0(d\theta) \to \delta. \end{aligned}$$

Aufgabe 11-2 (Bayesianische Optimierung mit quadratischer Verlustfunktion). Berechnen Sie π_n in folgendem Spezialfall von Aufgabe 4:

$$\pi_0 \sim \mathcal{N}(\mu_0, \sigma_0^2), \qquad L(\theta) = \frac{(\theta-\mu_1)^2}{2\sigma_1^2}.$$

Bonusfrage: Konvergiert π_n in einem geeigneten Sinn gegen das globale Minimum μ_1?

Lösung: Wir berechnen

$$\pi_n(d\theta) \propto \exp\left(-\frac{(\theta-\mu_1)^2}{2\sigma_1^2/n}\right)\exp\left(-\frac{(\theta-\mu_0)^2}{2\sigma_0^2}\right) d\theta \propto \exp\left(-\frac{(\theta-\hat\mu_n)^2}{2\hat\sigma_n^2/n}\right) d\theta,$$

wobei man $\hat\mu_n$ und $\hat\sigma_n$ durch Koeffizientenvergleich ermittelt:

$$\hat\mu_n = \frac{\tau_0\mu_0 + n\tau_1\mu_1}{\tau_0 + n\tau_1}, \qquad \hat\sigma_n^2 = 1/(\tau_0 + n\tau_1), \qquad \tau_0 = 1/\sigma_0^2, \qquad \tau_1 = 1/\sigma_1^2.$$

Daher gilt $\pi_n \sim \mathcal{N}(\hat\mu_n, \hat\sigma_n^2)$. Die Wahrscheinlichkeitsmasse π_n konvergieren in Verteilung gegen δ_{μ_1}, denn π_n-verteilte Zufallsvariablen θ_n konvergieren in L^2 gegen μ_1:

$$\begin{aligned}\mathbb{E}\big((\theta_n-\mu_1)^2\big) &= \mathbb{E}\big((\theta_n-\hat{\mu}_n+\hat{\mu}_n-\mu_1)^2\big)\\ &= \underbrace{\mathbb{E}\big((\theta_n-\hat{\mu}_n)^2\big)}_{=\sigma_n^2\to 0}+2\underbrace{\mathbb{E}(\theta_n-\hat{\mu}_n)}_{=0}(\hat{\mu}_n-\mu_1)+\underbrace{(\hat{\mu}_n-\mu_1)^2}_{\to 0}.\end{aligned}$$

MC 11-3 (Bayesianische Optimierung für Maximum-Likelihood Schätzung). Wenden Sie den Algorithmus für Bayesianische Optimierung aus Aufgabe 1 auf das Problem der Maximum-Likelihood Schätzung an. Welche Aussagen sind korrekt? (Mehrere richtige Antworten sind möglich.)

(a) Man muss eine a-priori Verteilung π_0 wählen.

(b) Die Verlustfunktion ist die log-Likelihood $L(\theta)=\log f(x|\theta)$.

(c) Man erhält als π_1 die a-posteriori Verteilung für eine Beobachtung x.

(d) Man erhält als π_2 die a-posteriori Verteilung, indem man eine neue Beobachtung x unter dem vorherigen Prior π_1 berücksichtigt.

Lösung: Nur (c) ist falsch.

(a) Richtig. Bayesianische Optimierung verlangt die Wahl eines Wahrscheinlichkeitsmasses π_0 und dieses kann als a-priori Verteilung interpretiert werden.

(b) Falsch. Korrekt wäre die negative log-Likelihood $L(\theta)=-\log f(x|\theta)$.

(c) Richtig. Bayesianische Optimierung liefert die a-posteriori Verteilung

$$\pi_1(d\theta)=\exp(-L(\theta))\pi_0(d\theta)=f(x|\theta)\pi_0(d\theta).$$

(d) Richtig. Das Wahrscheinlichkeitsmass π_2 ist

$$\pi_2(d\theta)=\exp(-2L(\theta))\pi_0(d\theta)=f(x|\theta)^2\pi_0(d\theta).$$

Dies entspricht einer zweimaligen unabhängigen Beobachtung derselben Daten x.

MC 11-4 (Gradientenfluss). Der Gradientenfluss $\mathrm{Fl}_t(x_0)$ einer Funktion f mit Anfangswert x_0 ist definiert als die Lösung der Differentialgleichung

$$\frac{d}{dt}\mathrm{Fl}_t(x_0)=-\nabla f\big(\mathrm{Fl}_t(x_0)\big),\qquad x(0)=x_0.$$

Welche Aussagen gelten im Limes $t\to\infty$?

(a) Für $f(x)=\exp(-x)$ konvergiert $\mathrm{Fl}_t(0)$ gegen einen Minimierer von f.

(b) Für $f(x)=\exp(-x)$ konvergiert $f\big(\mathrm{Fl}_t(0)\big)$ gegen das Infimum von f.

(c) Für $f(x)=x^2/2$ konvergiert $\mathrm{Fl}_t(1)$ gegen einen Minimierer von f.

(d) Für $f(x)=x^2/2$ konvergiert $f\big(\mathrm{Fl}_t(1)\big)$ gegen das Infimum von f.

Lösung:

(a) Falsch. Der Gradientenfluss $\mathrm{Fl}_t(0) = \log(1+t)$ divergiert gegen ∞ und f hat keinen Minimierer.

(b) Richtig. Die Verlustfunktion $f\big(\mathrm{Fl}_t(0)\big) = 1/(1+t)$ konvergiert entlang des Gradientenflusses gegen das Infimum 0 von f.

(c) Richtig. Der Gradientenfluss $\mathrm{Fl}_t(1) = e^{-t}$ konvergiert gegen das Minimum 0 von f.

(d) Richtig. Die Verlustfunktion $f\big(\mathrm{Fl}_t(1)\big) = e^{-2t}/2$ konvergiert gegen das Infimum 0 von f.

Aufgabe 11-5 (Langevin Dynamik). Implementieren Sie die diskrete Langevin-Dynamik

$$\theta_{n+1} = \theta_n - \eta \nabla L(\theta_n) + \sqrt{\eta}\sigma\epsilon_n, \qquad \epsilon_n \sim \mathcal{N}\big(\left(\begin{smallmatrix}0\\0\end{smallmatrix}\right), \left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)\big),$$

mit unabhängigen Störungen (ϵ_n), um folgende Verlustfunktion Bayesianisch zu minimieren:

$$L : \mathbb{R}^2 \to \mathbb{R}, \qquad L(x,y) = (x^2-1)^2 + y^2.$$

Simulieren Sie mehrere Lösungen θ_{500} mit Anfangswert $\theta_0 = \left(\begin{smallmatrix}0\\0\end{smallmatrix}\right)$, Schrittweite $\eta = 0.1$ und Standardabweichung $\sigma = 0.1$. Plotten Sie die Lösungen gemeinsam mit den Kontouren der Verlustfunktion.

```
import numpy as np
import matplotlib.pyplot as plt

def loss(x, y):
    return (x**2 - 1)**2 + y**2

def grad_loss(x, y):
    dL_dx = 4 * x * (x**2 - 1)
    dL_dy = 2 * y
    return np.array([dL_dx, dL_dy])

def langevin_dynamics(x0, eta=0.1, sigma=0.1, steps=500):
    x = np.array(x0, dtype=float)
    for _ in range(steps):
        grad = grad_loss(x[0], x[1])
        noise = np.random.normal(0, 1, size=2)
        x -= eta * grad + np.sqrt(eta) * sigma * noise
    return x

particles = [langevin_dynamics([0, 0]) for _ in range(50)]

plt.figure(figsize=(10, 8))
X, Y = np.meshgrid(np.linspace(-2, 2, 400), np.linspace(-2, 2, 400))
Z = loss(X, Y)
contours = plt.contour(X, Y, Z, levels=50, cmap='viridis')
plt.colorbar(contours)
```

```
for particle in particles:
    plt.scatter(particle[0], particle[1], color='red', s=10)
plt.show()
```