变分推理方法

最速梯度下降法

$$\underset{x}{\text{minimize}} \ f(x)$$

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

$$\nabla f(x_k) = \underset{v}{\arg\max} \frac{\lim_{\varepsilon \to 0} \frac{f(x_k + \varepsilon v) - f(x_k)}{\varepsilon}}{\sqrt{\langle v, v \rangle}}$$

$$= \underset{v}{\arg\max} \frac{\frac{\delta f}{\delta x}(x_k) \cdot v}{\sqrt{\langle v, v \rangle}}$$

变化最大的方向

$$\frac{\delta f}{\delta x}(x_k) \cdot v \leq \sqrt{\langle v, v \rangle} \sqrt{\frac{\delta f}{\delta x}(x_k) \cdot \frac{\delta f}{\delta x}(x_k)}$$

当且仅当 $v = \frac{\delta f}{\delta x}(x_k) = \nabla f(x_k)$

度量

切空间：$T_x M$ 　局部线性近似

曲线：$x_t : [0,1] \to M$

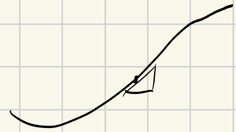
$$\dot{x}_t = v_t$$

曲线长度：$ds^2 = g_x(v, v)$

$$= \langle M_x v, v \rangle$$

练习 1)

$\dot{x}_t = v_t$



曲线 $x(t)$，长度 $\int_0^1 \|\dot{x}(t)\|_2 \, dt$

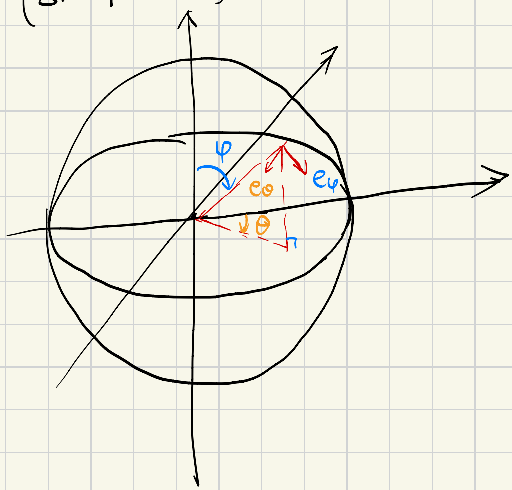
$$= \int_0^1 \sqrt{v_t \cdot v_t} \, dt$$

$$= \int_0^1 \sqrt{g_x(v_t, v_t)} \, dt$$

练习 2) 球面

$(\varphi, \theta) \to (\sin\varphi\cos\theta, \sin\varphi\sin\theta, \cos\varphi)$

切空间



$(\dot{\varphi}, \dot{\theta}) \to \dot{\varphi} e_\varphi + \dot{\theta} e_\theta$

$e_\varphi = (\cos\varphi\cos\theta, \cos\varphi\sin\theta, -\sin\varphi)$

$e_\theta = (-\sin\varphi\sin\theta, \sin\varphi\cos\theta, 0)$

曲线 $\dot{x}_t = v_t = (\dot{\varphi}_t \;\; \dot{\theta}_t)$

欧式空间 距离

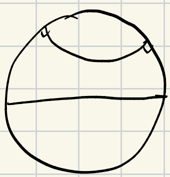
$$g_x(v, v) = \langle \dot{\varphi}\, e_\varphi + \dot{\theta}\, e_\theta, \dot{\varphi}\, e_\varphi + \dot{\theta}\, e_\theta \rangle$$

$$= \dot{\varphi}^2 + \dot{\theta}\, \sin^2\varphi$$

$$= \langle M_x v, v \rangle$$

$$M_x = \begin{bmatrix} 1 & 0 \\ 0 & \sin^2\varphi \end{bmatrix}$$

练习三：庞加莱圆盘（双曲几何）



在边上无限大

内积 $g_x(v, v) = \langle M_x v, v \rangle$

$$M_x = \begin{bmatrix} \frac{4}{(1-x^2-y^2)^2} & \\ & \frac{4}{(1-x^2-y^2)^2} \end{bmatrix}$$

$$(\dot{r}, \dot{\theta}) \to \dot{r}\, e_r + \dot{\theta}\, e_\theta$$

$$e_r = (\cos\theta, \sin\theta) \quad e_\theta = (-r\sin\theta, r\cos\theta)$$

$$g_x(v, v) = (\dot{r}\cos\theta - \dot{\theta}\, r\sin\theta)^2 \frac{4}{(1-r^2)^2}$$

$$+ (\dot{r}\sin\theta + \dot{\theta}\, r\cos\theta)^2 \frac{4}{(1-r^2)^2}$$

$$= \dot{r}^2 \frac{4}{(1-r^2)^2} + \dot{\theta}\, r^2 \frac{4}{(1-r^2)^2}$$

$$= \langle M_x v , v \rangle$$

$$M_x = \begin{bmatrix} \frac{4}{(1-r^2)^2} & \\ & \frac{4r^2}{(1-r^2)^2} \end{bmatrix}$$

流形上的最速梯度下降法

$$\frac{\partial f}{\partial x}(x_k) \cdot v \leq \sqrt{\langle M(x) v, v \rangle} \cdot \sqrt{\langle M(x)^{-1} \frac{\partial f}{\partial x}, \frac{\partial f}{\partial x} \rangle}$$

当且仅当 $M(x) v = \frac{\partial f}{\partial x}(x_k)$

能量泛函

$$\frac{\delta E}{\delta \rho}\zeta = \lim_{\varepsilon \to 0} \frac{\int (\rho + \varepsilon \zeta) \log\left(\frac{\rho + \varepsilon \zeta}{\rho^*}\right) - \int \rho \log\left(\frac{\rho}{\rho^*}\right)}{\varepsilon}$$

$$= \lim_{\varepsilon \to 0} \frac{\int \rho \log\left(\frac{\rho + \varepsilon \zeta}{\rho}\right) + \varepsilon \zeta \log\left(\frac{\rho + \varepsilon \zeta}{\rho^*}\right)}{\varepsilon}$$

$$= \lim_{\varepsilon \to 0} \frac{\int \rho \left(\frac{\varepsilon \zeta}{\rho}\right) + \varepsilon \zeta \log\left(\frac{\rho + \varepsilon \zeta}{\rho^*}\right)}{\varepsilon}$$

$$= \int \zeta \log \frac{\rho}{\rho^*} \, d\theta$$

Wasserstein-2度量 (ρ compactly support)

Otto calculus, Wasserstein-2 黎曼度量

考虑: $\inf_{v} \int \frac{\|v\|_2^2}{2} \rho \, d\theta$, $\sigma = -\nabla_\theta \cdot (\rho v)$

$$\inf_{v} \sup_{\psi} \int \frac{\|v\|_2^2}{2} \rho \, d\theta + \int (\sigma + \nabla_\theta (\rho v)) \psi \, d\theta$$

$$= \inf_{v} \sup_{\psi} \int \frac{\|v\|_2^2}{2} \rho \, d\theta + \int \sigma \psi \, d\theta - \int \rho v \cdot \nabla_\theta \psi \, d\theta$$

$$= \sup_{\psi} \inf_{v} \int \rho \left(\frac{v \cdot v}{2} - v \nabla_\theta \psi \right) d\theta + \int \sigma \psi \, d\theta$$

$$= \sup_{\psi} \int \sigma \psi - \rho \frac{\|\nabla_\theta \psi\|^2}{2} \, d\theta \qquad \begin{cases} v = \nabla_\theta \psi \\ \sigma = -\nabla_\theta (\rho \nabla_\theta \psi) \end{cases}$$

由于 $\sigma \in T_\rho \mathcal{P}$ $\sigma = -\nabla_\theta (\rho \nabla_\theta \psi)$, $\nabla_\theta \psi$ 有唯一解

$$= \int -\nabla_\theta (\rho \nabla_\theta \psi) \psi - \rho \frac{\|\nabla_\theta \psi\|^2}{2} \, d\theta$$

$$= \int \frac{\|\nabla_\theta \psi\|_2^2 \rho}{2} \, d\theta$$

因此 $g_\rho^{W_2}(\sigma, \sigma) = \int \|\nabla_\theta \psi\|_2^2 \rho \, d\theta$ $\quad \sigma = -\nabla_\theta \cdot (\rho \nabla_\theta \psi)$

$$= \int \underbrace{-\nabla_\theta \cdot (\rho \nabla_\theta \psi)}_{\sigma} \cdot \psi \, d\theta$$

$$= \langle \psi, z \rangle := \langle M_\rho^{W_2} z, z \rangle$$

$$M^{W_2}(\rho) z = \psi \qquad z = -\nabla_\theta \cdot (\rho \nabla_\theta \psi)$$

$$M^{W_2}(\rho)^{-1} \psi = z = -\nabla_\theta (\rho \nabla_\theta \psi)$$

Wasserstein 梯度流

$$\frac{\partial \rho_t}{\partial t} = \nabla_\theta \cdot (\rho_t \nabla_\theta (\log \rho_t - \log \rho^*))$$

$$= \nabla_\theta \left(\rho_t \left(\frac{\nabla_\theta \rho_t}{\rho_t} - \nabla_\theta [-\Phi_R] \right) \right)$$

$$= \nabla_\theta (\nabla_\theta \rho_t + \rho_t \nabla_\theta \Phi_R)$$

高斯近似的 Wasserstein 梯度流

$$\int \nabla_\theta \cdot f \cdot M = \Sigma \int \partial_i f_i M = \Sigma \int f_i \partial_i M$$

$$= \int (\rho_{a_t} \nabla_\theta \Phi_R + \nabla_\theta \rho_{a_t})_i \left[e_i \cdot (\theta - m_t)^T + (\theta - m_t) e_i^T \right]$$

$$= \int (\rho_{a_t} \nabla_\theta \Phi_R + \nabla_\theta \rho_{a_t}) (\theta - m_t)^T + (\theta - m_t)(\rho_t \nabla_\theta \Phi_R + \nabla_\theta \rho_t)^T$$

由于 $\int \nabla_\theta \rho_{a_t} (\theta - m_t)^T = \int \rho_{a_t} C_t^{-1} (\theta - m_t)(\theta - m_t)^T = I$

$$\int \rho_{a_t} \nabla_\theta \Phi_R (\theta - m_t)^T = \int \nabla_\theta \Phi_R (C_t \nabla_\theta \rho_{a_t})^T$$

$$= -\int \nabla_\theta \nabla_\theta \Phi_R \, \rho_{a_t} \, d\theta \, C_t$$

收敛性：

$$\partial_t KL[\rho_t \| \rho^*] = \frac{\partial}{\partial t} \int \rho_t \log \frac{\rho_t}{\rho^*} d\theta$$

$$= \int \dot{\rho}_t \log \frac{\rho_t}{\rho^*} + \underbrace{\rho_t \frac{1}{\rho_t} \dot{\rho}_t \, d\theta}_{\dot{1} = 0}$$

$$= \int \nabla_\theta \cdot (\rho_t \nabla_\theta \log \rho_t - \log \rho_*) \log \left(\frac{\rho_t}{\rho^*}\right) d\theta$$

$$= -\int \rho_t \left\| \nabla_\theta \log \frac{\rho_t}{\rho^*} \right\|_2^2 d\theta$$

log-sobolev

$$\leq -2\alpha \, KL[\rho_t \| \rho^*]$$

Gronwall 引理

$$\partial_t \left(e^{2\alpha t} KL[\rho_t \| \rho^*]\right) \leq 0$$

$$e^{2\alpha t} KL[\rho_t \| \rho^*] \leq KL[\rho_0 \| \rho^*]$$

Fisher - Rao metric

$$\int_{\tilde{\theta}\in A} T_\# \rho(\tilde{\theta})\, d\tilde{\theta} = \int_{T\theta\in A} \rho(\theta)\, d\theta$$

$$= \int_{T\theta\in A} T_\# \rho(T\theta)\, |T'(\theta)|\, d\theta$$

$$\rho(\theta) = T_\# \rho(T\theta)\, |T'(\theta)|$$

$$\sigma(\theta) = T_\# \sigma(T\theta)\, |T'(\theta)|$$

$$\int \frac{T_\# \sigma(\tilde{\theta}) \cdot T_\# \sigma(\tilde{\theta})}{T_\# \rho(\tilde{\theta})}\, d\tilde{\theta} \qquad \tilde{\theta} = T\theta$$

$$= \int \frac{\sigma(\theta)/|T'(\theta)| \cdot \sigma(\theta)/|T'(\theta)|}{\rho(\theta)/|T'(\theta)|}\, |T'(\theta)|\, d\theta$$

$$= \int \frac{\sigma(\theta) \cdot \sigma(\theta)}{\rho(\theta)}\, d\theta$$

对偶梯度

$$\frac{d}{dt}\left(\int \rho_t \log\frac{\rho_t}{\rho^*} + \int \rho^* \log\frac{\rho^*}{\rho_t}\right)$$

$$c = -KL[\rho_t \,\|\, \rho^*]$$

$$\dot{\rho}_t = \rho_t \log\frac{\rho^*}{\rho_t} - \rho_t c$$

$$= \int \dot{\rho}_t \log\frac{\rho_t}{\rho^*} - \int \rho^* \frac{1}{\rho_t}\dot{\rho}_t$$

$$= -\int \rho_t \left(\log\frac{\rho_t}{\rho^*}\right)^2 - \int \rho_t \log\frac{\rho_t}{\rho^*}\cdot c - \int \rho^* \log\frac{\rho^*}{\rho_t} + \int \rho^* c$$

$$= \underbrace{-\int \rho_t \left(\log\frac{\rho_t}{\rho^*}\right)^2 + \left(\int \rho_t \log\frac{\rho_t}{\rho^*}\right)^2}_{\leq 0} - KL[\rho^* \,\|\, \rho_t] - KL[\rho_t \,\|\, \rho^*]$$

$$\leq -KL[\rho^* \,\|\, \rho_t] - KL[\rho_t \,\|\, \rho^*]$$

生灭过程

θ 以 $1-e^{-|\alpha_t| dt}$ 的概率 复制 / 湮灭

当 $\alpha_t > 0$ / $\alpha_t < 0$.

$$\int_{\theta\in A} \rho_{t+\Delta t}(\theta)\, d\theta - \int_{\theta\in A} \rho_t(\theta)\, d\theta$$

$$= \int_{\substack{\alpha_t(\theta)>0\\ \theta\in A}} \rho_t(\theta)\left(1-e^{-|\alpha_t| dt}\right) - \int_{\substack{\alpha_t(\theta)<0\\ \theta\in A}} \rho_t(\theta)\left(1-e^{-|\alpha_t| dt}\right)$$

$$= \int \rho_t(\theta)\, \mathrm{sign}(\alpha_t(\theta))\left(|\alpha_t|\, dt + o(dt)\right)$$

$$\frac{\partial \rho_t}{\partial t} = \rho_t \, \partial_t$$

参数化变分推理

$$\int \rho_{a_t}\left[(-\Phi_R - \log \rho_{a_t}) - \underbrace{E_{\rho_{a_t}}[-\Phi_R - \log \rho_{a_t}]}\right]\theta \, d\theta$$

$$= \int \rho_{a_t}[-\Phi_R - \log \rho_{a_t}](\theta - m_t) \, d\theta$$

$$= -\int C_t \nabla_\theta \rho_{a_t} (-\Phi_R - \log \rho_{a_t}) \, d\theta$$

$$= -C_t \int \rho_a (\nabla_\theta \Phi_R + \nabla_\theta \log \rho_{a_t}) \, d\theta$$

$$= -C_t \, E \, \nabla_\theta \Phi_R$$

$$\int \rho_{a_t}\left[(-\Phi_R - \log \rho_{a_t}) - E_{\rho_{a_t}}[-\Phi_R - \log \rho_{a_t}]\right](\theta - m_t)(\theta - m_t)^T d\theta$$

$$= \int \rho_{a_t}[-\Phi_R - \log \rho_{a_t}]\left[(\theta - m_t)(\theta - m_t)^T - C_t\right] d\theta$$

$$= C_t \int [-\Phi_R - \log \rho_{a_t}] \nabla_\theta \nabla_\theta \rho_{a_t} \, d\theta \, C_t$$

$$= C_t \int \nabla_\theta \nabla_\theta [-\Phi_R - \log \rho_{a_t}] \rho_{a_t} \, d\theta \, C_t$$

$$= C_t - C_t \, E \, \nabla_\theta \nabla_\theta \Phi_R \, C_t$$

当 Φ_R 强凸， KL 散度在 W_2 下是强凸，

Gaussian space， W_2 ， BW_2 ， KL 散度也是强凸

因为 G_A G_B 测地线全是高斯

自然梯度下降法

$$\mathcal{P} = \{ \rho : \int \rho = 1 \} \qquad T_\rho \mathcal{P} = \{ \sigma : \int \sigma = 0 \}$$

$$\mathcal{P} = \{ a \in R^{N_a} \} \qquad T_{\rho_a} \mathcal{P} = \{ v : \in R^{N_a} \}$$

$$\rho_a \to \rho_{a+\varepsilon v} = \rho_a + \varepsilon \sigma$$

$$\sigma = \lim_{\varepsilon \to 0} \frac{\rho_{a+\varepsilon v} - \rho_a}{\varepsilon} = \nabla_a \rho_a \cdot v$$

$$g^{FR}_{\rho_a} (\nabla_a \rho_a \cdot v_1 , \nabla_a \rho_a \cdot v_2)$$

$$= \int \frac{(\nabla_a \rho_a \cdot v_1)^T \cdot (\nabla_a \rho_a \cdot v_2)}{\rho_a} d\theta$$

$$= v_1^T \int \frac{\nabla_a \rho_a \, \nabla_a \rho_a}{\rho_a} d\theta \, v_2$$

$$KL[\rho_{a+da} \| \rho_a] = \int \rho_{a+da} \log \frac{\rho_{a+da}}{\rho_a}$$

$$\rho_{a+da} = \rho_a + \nabla_a \rho_a \cdot da + \frac{1}{2} da^T \nabla_a^2 \rho_a \, da$$

$$\log \frac{\rho_{a+da}}{\rho_a} = \frac{\nabla_a \rho_a \cdot da}{\rho_a} + \frac{\frac{1}{2} da^T \nabla_a^2 \rho_a \, da}{\rho_a} - \frac{1}{2}\left(\frac{\nabla_a \rho_a \cdot da}{\rho_a}\right)^2$$

$$\log 1 + x = x - \frac{x^2}{2}$$

代入

$$= \int \nabla_a \rho_a \cdot da + \frac{1}{2} da^T \nabla_a^2 \rho_a \, da + \frac{(\nabla_a \rho_a \cdot da)^2}{\rho_a} d\theta$$

$$- \frac{1}{2}\left(\frac{\nabla_a \rho_a \cdot da}{\rho_a}\right)^2 \rho_a$$

$$= \frac{1}{2} da^T \int \frac{\nabla_a \rho_a \cdot \nabla_a \rho_a}{\rho_a} d\theta \; da$$

1. 指数分布族

对于指数分布族

$$\rho(\theta; a) = h(\theta)\exp\{T(\theta)\cdot a - A(a)\}$$
$$0 = \int \nabla_a \rho(\theta; a) d\theta = \mathbb{E}_\rho[T(\theta) - \nabla_a A(a)]$$

Here we introduce another parameter vector λ. Secondly, the Fisher information matrix FIM is

$$\begin{aligned}
FIM(a) &= \mathbb{E}_\rho[\nabla_a \log\rho(\theta;a)^T \nabla_a \log\rho(\theta;a)] \\
&= \int \nabla_a\rho(\theta;a)\big[T(\theta) - \nabla_a A(a)\big]d\theta \\
&= \int \rho(\theta;a)\nabla_a\nabla_a A(a) d\theta \quad \text{Using } \nabla_a\int\rho(\theta;a)\big[T(\theta)-\nabla_a A(a)\big]d\theta = 0 \\
&= \nabla_a\nabla_a A(a)
\end{aligned}$$

2. 高斯变分推理

对于高斯分布

$$\rho_a = \frac{1}{(2\pi)^{N_\theta/2}\sqrt{|C|}}\exp\{-\frac{1}{2}(\theta-m)^T C^{-1}(\theta-m)\}$$

我们有

$$\begin{aligned}
&\nabla_m\rho_a = \rho_a C^{-1}(\theta - m) = -\nabla_\theta\rho_a \\
&\nabla_C\rho_a = \rho_a\Big(-\frac{1}{2}C^{-1} + \frac{1}{2}C^{-1}(\theta-m)(\theta-m)^T C^{-1}\Big) = \frac{1}{2}\nabla_\theta\nabla_\theta\rho_a
\end{aligned}$$

我们用了

$$\nabla_C|C| = |C|C^{-T} \qquad \nabla_C x^T C^{-1} x = -C^{-1}xx^T C^{-1}$$

对于高斯变分推理，我们有

$$\begin{aligned}
\nabla_m KL[\rho_a\|\rho^*] &= \int \nabla_m\rho_a(\log(\rho_a) + \Phi_R)d\theta \\
&= -\int \nabla_\theta\rho_a(\log(\rho_a) + \Phi_R)d\theta \\
&= \int \rho_a\nabla_\theta(\log(\rho_a) + \Phi_R)d\theta \\
&= \int \nabla_\theta\rho_a + \rho_a\nabla_\theta\Phi_R d\theta \\
&= \int \rho_a\nabla_\theta\Phi_R d\theta \\
&= \mathbb{E}_{\rho_a}[\nabla_\theta\Phi_R] \\
\nabla_C KL[\rho_a\|\rho^*] &= \int \nabla_C\rho_a(\log(\rho_a) + \Phi_R)d\theta \\
&= \int \frac{1}{2}\nabla_\theta\nabla_\theta\rho_a(\log(\rho_a) + \Phi_R)d\theta \\
&= \int \frac{1}{2}\rho_a\nabla_\theta\nabla_\theta(\log(\rho_a) + \Phi_R)d\theta \\
&= \int \frac{1}{2}\rho_a(-C) + \frac{1}{2}\rho_a\nabla_\theta\nabla_\theta\Phi_R d\theta \\
&= -\frac{1}{2}C + \frac{1}{2}\mathbb{E}_{\rho_a}[\nabla_\theta\nabla_\theta\Phi_R]
\end{aligned}$$

3. 自然梯度下降方法

我们首先推导高斯的Fisher信息矩阵 我们有

$$\begin{aligned}
&\int \frac{\nabla_m\rho_a\nabla_m\rho_a^T}{\rho_a}d\theta = \int \rho_a C^{-1}(\theta-m)(\theta-m)^T C^{-1}d\theta d\theta = C^{-1} \\
&\int \frac{\nabla_m\rho_a\nabla_C\rho_a^T}{\rho_a}d\theta = \int \rho_a C^{-1}(\theta-m)\otimes\Big(-\frac{1}{2}C^{-1} - \frac{1}{2}C^{-1}(\theta-m)(\theta-m)^T C^{-1}\Big)d\theta = 0
\end{aligned}$$

对于协方差项，我们定义

$$\begin{aligned}
X :=& \frac{1}{4}\int \rho_a\Big(C^{-1}(\theta-m)(\theta-m)^T C^{-1} - C^{-1}\Big)\otimes\Big(C^{-1}(\theta-m)(\theta-m)^T C^{-1} - C^{-1}\Big)d\theta \\
=& \frac{1}{4}\int \mathcal{N}(y;0,I)C^{-1/2}\Big(yy^T - I\Big)C^{-1/2}\otimes C^{-1/2}\Big(yy^T - I\Big)C^{-1/2}dy, \text{ where } y = C^{-1/2}(\theta - m).
\end{aligned}$$

它满足

$$\begin{aligned}
X[ij,lm] =& \frac{1}{4}\sum_{r,s,p,q} C^{-1/2}[i,r]C^{-1/2}[j,s]C^{-1/2}[l,p]C^{-1/2}[m,q]\int(y_r y_s - \delta_{r,s})(y_p y_q - \delta_{p,q})\mathcal{N}(y;0,I)dy \\
=& \frac{1}{4}\sum_{r,s,p,q} C^{-1/2}[i,r]C^{-1/2}[j,s]C^{-1/2}[l,p]C^{-1/2}[m,q]\big(\delta_{r,p}\delta_{s,q} + \delta_{r,q}\delta_{s,p}\big) \\
=& \frac{1}{4}\Big(C^{-1}[i,l]C^{-1}[j,m] + C^{-1}[i,m]C^{-1}[j,l]\Big).
\end{aligned}$$

因此

$$\begin{aligned}
(XY)_{ij} &= \sum_{l,m} X[ij,lm]Y[l,m] \\
&= \frac{1}{4}\sum_{l,m}\Big(C^{-1}[i,l]Y[l,m]C^{-1}[j,m] + C^{-1}[i,m]Y[l,m]C^{-1}[j,l]\Big) \\
&= \frac{1}{4}(C^{-1}YC^{-1} + C^{-1}Y^T C^{-1})_{ij}.
\end{aligned}$$

我们用了

由于

$$\frac{\mathrm{d}C_t^{-1}}{\mathrm{d}t} = -C_t^{-1}\frac{\mathrm{d}C_t}{\mathrm{d}t}C_t^{-1}$$

自然梯度下降法可以写成

$$\begin{aligned}
\frac{\mathrm{d}m_t}{\mathrm{d}t} &= C_t {C^*}^{-1}(m^* - m_t), \\
\frac{\mathrm{d}C_t^{-1}}{\mathrm{d}t} &= -C_t^{-1} + {C^*}^{-1}.
\end{aligned}$$

对于协方差，我们有

$$C_t^{-1} = (1 - e^{-t}){C^*}^{-1} + e^{-t}C_0^{-1}.$$

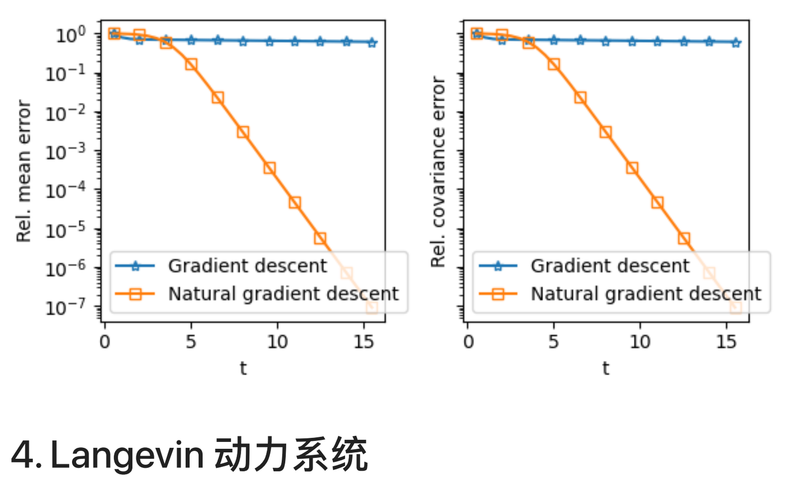
对于期望，我们有

$$m^* - m_t = e^{-t}C_t C_0^{-1}(m^* - m_0).$$

计算它的导数，我们能得到

$$\begin{aligned}
\frac{\mathrm{d}m_t}{\mathrm{d}t} &= -e^{-t}C_t C_0^{-1}(m_0 - m^*) + e^{-t}\frac{\mathrm{d}C_t}{\mathrm{d}t}C_0^{-1}(m_0 - m^*) \\
&= -e^{-t}C_t C_0^{-1}(m_0 - m^*) + e^{-t}(C_t - C_t {C^*}^{-1} C_t)C_0^{-1}(m_0 - m^*) \\
&= C_t {C^*}^{-1} e^{-t}C_t C_0^{-1}(m^* - m_0) \\
&= C_t {C^*}^{-1}(m^* - m_t).
\end{aligned}$$

In [1]:
```
using LinearAlgebra
using PyPlot
function expectation(method_type::String, m_oo, C_oo, m, C)
    return C_oo\(m - m_oo), inv(C_oo)
    if method_type == "gradient_descent"
        # under-determined case
        return -C_oo\(m - m_oo), 1/2.0*(inv(C) - inv(C_oo))

    elseif method_type == "natural_gradient_descent"
        # over-determined case
        return -(m - m_oo), C - C*(C_oo\C)

    elseif method_type == "natural_gradient_descent_Cinv"
        # over-determined case
        return -(m - m_oo), -inv(C) + C_oo

    elseif method_type == "wasserstein_gradient_descent"
        # over-determined case
        return -C_oo\(m - m_oo), 2I - C_oo\C - C/C_oo

    else
        error("Problem type : ", problem_type, " has not implemented!")
    end
end

function Continuous_Dynamics(method_type::String, m_oo, C_oo, m_0, C_0, Δt, N_t)

    N_θ = length(m_0)
    m = zeros(N_t+1, N_θ)
    C = zeros(N_t+1, N_θ, N_θ)

    m[1, :] = m_0
    C[1, :, :] = C_0

    for i = 1:N_t
        E∇Φ, E∇∇Φ= expectation(method_type, m_oo, C_oo, m[i, :], C[i, :, :])
        if method_type == "gradient_descent"
            m[i+1, :]    = m[i, :] - E∇Φ * Δt
            C[i+1, :, :] = C[i, :, :] + 1/2.0*(inv(C[i, :, :]) - E∇∇Φ) * Δt
        elseif method_type == "natural_gradient_descent"
            m[i+1, :]    = m[i, :] - C[i, :, :]*E∇Φ * Δt
            C[i+1, :, :] = inv( inv(C[i, :, :]) + (E∇∇Φ - inv(C[i, :, :])) * Δt)
        else
            error("Problem type : ", problem_type, " has not implemented!")
        end
    end

    return m, C
end


Δt, N_t = 5e-1, 30
fig, ax = PyPlot.subplots(ncols=2, sharex=true, sharey=true, figsize=(6,3))


m_oo = [1; 1]
C_oo = [100.0 0; 0 1.0]

m_0 = [0.0; 0.0]
C_0 = [1.0 0; 0 1.0]

m_gd, C_gd   = Continuous_Dynamics("gradient_descent", m_oo, C_oo, m_0, C_0, Δt, N_t)
m_ngd, C_ngd = Continuous_Dynamics("natural_gradient_descent", m_oo, C_oo, m_0, C_0, Δt, N_t)

e_gd, e_ngd  = zeros(N_t+1, 2),  zeros(N_t+1, 2)


for i = 1:N_t+1

    e_gd[i, 1]  = ( norm(m_gd[i,:] - m_oo) / norm(m_oo) )
    e_gd[i, 2]  = ( norm(C_gd[i,:,:] - C_oo) / norm(C_oo) )

    e_ngd[i, 1] = ( norm(m_ngd[i,:] - m_oo) / norm(m_oo) )
    e_ngd[i, 2] = ( norm(C_ngd[i,:,:] - C_oo) / norm(C_oo) )
end

ts = Array(1:N_t+1)*Δt
ax[1].semilogy(ts, e_gd[:, 1],  "-*", fillstyle="none", markevery=div(N_t, 10), label="Gradient descent")
ax[1].semilogy(ts, e_ngd[:, 2], "-s", fillstyle="none", markevery=div(N_t, 10), label="Natural gradient descent")
ax[2].semilogy(ts, e_gd[:, 1],  "-*", fillstyle="none", markevery=div(N_t, 10), label="Gradient descent")
ax[2].semilogy(ts, e_ngd[:, 2], "-s", fillstyle="none", markevery=div(N_t, 10), label="Natural gradient descent")

ax[1].legend()
ax[2].legend()
ax[1].set_ylabel("Rel. mean error")
ax[2].set_ylabel("Rel. covariance error")
ax[1].set_xlabel("t")
ax[2].set_xlabel("t")
fig.tight_layout()
fig.savefig("Gaussian-VI.pdf")
```



4. Langevin 动力系统

考虑随机动力系统

$$d\theta_t = f(\theta_t, t)dt + \sqrt{g}dW_t$$

它对应的 描述 $\theta_t \sim \rho_t$ 的 Fokker-Planck 方程是

$$\partial_t\rho_t = -\nabla\big(f(x,t)\rho_t\big) + \frac{1}{2}\nabla\cdot\nabla\cdot(g\rho_t)$$

使用 Itô's 公式，我们有

$$\begin{aligned}
dh(\theta_t) &= \nabla h(\theta_t)^T(f(\theta_t,t)dt + \sqrt{g}dW_t) + \frac{1}{2}g\nabla^2 h(\theta_t)dt \\
\partial_t\mathbb{E}[h(\theta_t)] &= \mathbb{E}\Big[f(\theta_t,t)\nabla h(\theta_t) + \frac{g}{2}\nabla^2 h(\theta_t)\Big] \\
\int h(\theta)\partial_t\rho_t &= \int \rho_t f(\theta,t)\nabla h(\theta) + \frac{g}{2}\rho_t\nabla^2 h(\theta)d\theta \\
&= \int h(\theta)\nabla(-\rho_t f(\theta,t)) + h(\theta)\frac{1}{2}\nabla\cdot\nabla\cdot(g\rho_t)d\theta
\end{aligned}$$

我们可以取

$$f = -\nabla_\theta\Phi_R \qquad g = 2$$

5. 仿射不变性

给定可逆仿射变换 $\tilde{\theta} = \mathcal{T}\theta = A\theta + b$,

$$\begin{aligned}
&\tilde{\rho}(\tilde{\theta}) = \rho(\theta)|A^{-1}| \qquad \tilde{\rho}_t(\tilde{\theta}) = \rho_t(\theta)|A^{-1}| \qquad \tilde{C}_t = AC_tA^T \\
&\nabla_{\tilde{\theta}}f(\tilde{\theta}) = A^{-T}\nabla_\theta f(\theta) \qquad \nabla_{\tilde{\theta}}\cdot v(\tilde{\theta}) = \nabla_\theta\cdot(A^{-1}v(\theta))
\end{aligned}$$

带入Kalman-Waserstein梯度流

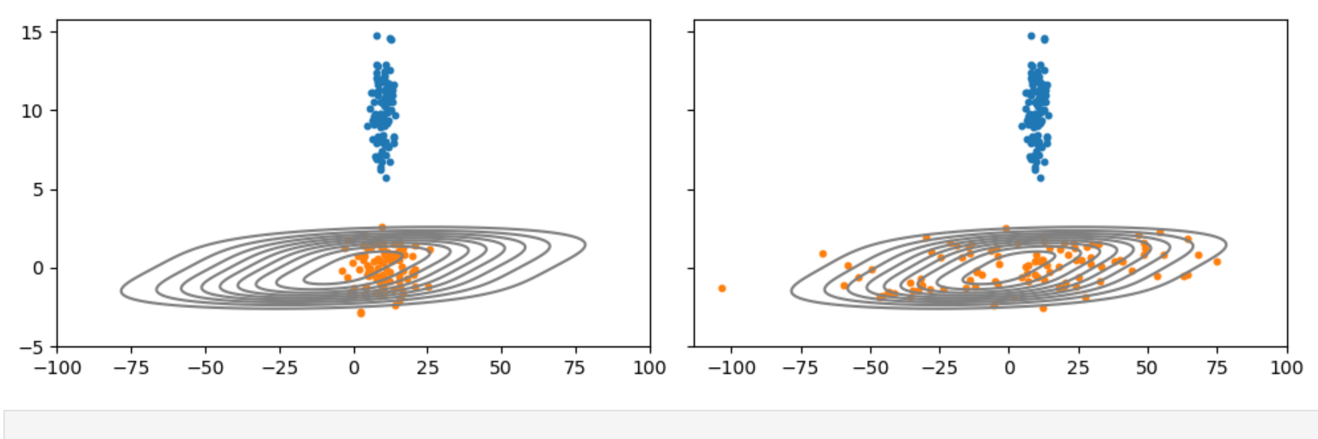
$$\begin{aligned}
&\frac{\partial\tilde{\rho}_t}{\partial t} = \nabla_{\tilde{\theta}}\cdot\Big[\tilde{\rho}_t\tilde{C}_t(\nabla_{\tilde{\theta}}\log\tilde{\rho}^* - \nabla_{\tilde{\theta}}\log\tilde{\rho}_t)\Big] \\
\Longleftrightarrow &\frac{\partial\rho_t}{\partial t}|A^{-1}| = \nabla_{\tilde{\theta}}\cdot\Big[\rho_t AC_tA^T(\nabla_{\tilde{\theta}}\log\tilde{\rho}^* - \nabla_{\tilde{\theta}}\log\tilde{\rho}_t)\Big]|A^{-1}| \\
\Longleftrightarrow &\frac{\partial\rho_t}{\partial t}|A^{-1}| = \nabla_\theta\cdot\Big[\rho_t C_t(\nabla_\theta\log\tilde{\rho}^* - \nabla_\theta\log\tilde{\rho}_t)\Big]|A^{-1}| \\
\Longleftrightarrow &\frac{\partial\rho_t}{\partial t} = \nabla_\theta\cdot\Big[\rho_t C_t(\nabla_\theta\log\rho^* - \nabla_\theta\log\rho_t)\Big]
\end{aligned}$$

6. 练习

In [2]:
```
using PyPlot, Random, Statistics, Distributions, ForwardDiff, LinearAlgebra

Random.seed!(42)

function Phi_R_Convex(θ)
    ϵ = 0.1
    θ₁, θ₂ = θ
    return ( (ϵ*θ₁ - θ₂)^2 + θ₂^4)/20
end


ΦR(θ) = Phi_R_Convex(θ)
function ∇ΦR(θ)
    return ΦR(θ), ForwardDiff.gradient(ΦR, θ)
end


function update_ensemble(θ, Δt, func_neg_log_rho::Function,  preconditioner::Bool)
    N_θ, N_ens = size(θ)

    Φ_R, ∇Φ_R = zeros(N_ens), zeros(N_θ, N_ens)
    for i = 1:N_ens
        Φ_R[i], ∇Φ_R[:,i] = func_neg_log_rho(θ[:, i])
    end

    θ_mean = mean(θ, dims=2)
    θθ_cov = ((θ .- θ_mean) * (θ .- θ_mean)') / N_ens
    Prec = (preconditioner ? θθ_cov : 1)
    σ = (preconditioner ? cholesky(Hermitian(θθ_cov)).L : 1)

    noise = rand(Normal(0, 1), (N_θ, N_ens))
    θ = θ - Δt*Prec*∇Φ_R + sqrt(2*Δt)*(σ*noise)

    return θ
end
```

Out[2]:
```
update_ensemble (generic function with 1 method)
```

In [3]:
```
N_θ = 2
m_0 = [10.0; 10.0]
σ0 = 2
C_0 = [σ0^2  0.0; 0.0  σ0^2]



Lx , Ux = -100.0,  100.0
Ly , Uy = -5.0,  5.0
N = 500
X = zeros(N, N)
Y = zeros(N, N)
ρ = zeros(N, N)
for i = 1:N
    for j = 1:N
        X[i, j], Y[i, j] = Lx + (Ux - Lx) * (i-1)/(N-1), Ly + (Uy - Ly) * (j-1)/(N-1)
        ρ[i, j] = Phi_R_Convex([X[i, j]; Y[i, j]])
    end
end
Z = sum(ρ)
ρ .= exp.(-ρ)/Z

fig, ax = PyPlot.subplots(ncols=2, nrows=1, sharex="col", sharey=true, figsize=(10,3))
ax[1].contour(X, Y, ρ, 10, colors="grey")
ax[2].contour(X, Y, ρ, 10, colors="grey")


N_ens = 100
θ0 = Array(rand(MvNormal(m_0, C_0), N_ens))
θ = zeros(N_θ, N_ens)
Δt = 0.01
N_t = 2000



for preconditioner = 1:2
    θ .= θ0
    for i = 1:N_t
        θ = update_ensemble(θ, Δt, ∇ΦR,  preconditioner == 2)
    end

    ax[preconditioner].scatter(θ0[1, :], θ0[2, :], s = 10)
    ax[preconditioner].scatter(θ[1, :], θ[2, :], s = 10)


end

fig.tight_layout()
fig.savefig("GF.png")
```



In [ ]: