Learning Complex Neural Network Policies with Trajectory Optimization

Sergey Levine SVLEVINE@CS.STANFORD.EDU
Computer Science Department, Stanford University, Stanford, CA 94305 USA

Vladlen Koltun VLADLEN@ADOBE.COM
Adobe Research, San Francisco, CA 94103 USA

【强化学习算法 14】GPS

张楚珩
清华大学 交叉信息院博士在读

这里的GPS不是用来导航的GPS，其全称是Guided Policy Search，是一种局域的model-based RL算法，各种版本基本上都是Levine大神搞出来的。

原文传送门：

【Importance-sampled GPS】Levine, Sergey, and Vladlen Koltun. "Guided policy search." International Conference on Machine Learning. 2013.

【Variational GPS】Levine, Sergey, and Vladlen Koltun. "Variational policy search via trajectory optimization." Advances in Neural Information Processing Systems. 2013.

【Constraint GPS】Levine, Sergey, and Vladlen Koltun. "Learning complex neural network policies with trajectory optimization."International Conference on Machine Learning. 2014.

【Unknown Dynamics GPS】Levine, Sergey, and Pieter Abbeel. "Learning neural network policies with guided policy search under unknown dynamics."Advances in Neural Information Processing Systems. 2014.

这里有好多版本的，暂时这里只讲第三篇工作。

特色：

这里讲了一种model-based的算法，主要思路是先找一条state-action的路径，然后交替地1）优化路径，使得该路径既离现有的控制策略不远，又能够最小化cost；2）优化策略，使得策略离被优化过的路径更近。通过这样trajectory optimization + policy search的方式在bipedal push recovery和walking on uneven terrain问题上取得了较好的实验结果。

分类：

continuous state space、continuous action space、model-based with known model dynamics

过程：

1. 定义优化问题

首先从一个类似SQL算法的优化目标开始

$$\min_{\theta, q(\tau)} D_{\mathrm{KL}}(q(\tau)\|\rho(\tau))$$
$$\text{s.t. } q(\mathbf{x}_1) = p(\mathbf{x}_1),$$
$$q(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t),$$
$$D_{\mathrm{KL}}(q(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|q(\mathbf{x}_t, \mathbf{u}_t)) = 0.$$

首先先明确一下轨迹 τ 是状态 $\mathbf{x}$ 和控制 $\mathbf{u}$ 交替的一个序列。其中 $\rho(\tau) \propto \exp(-l(\tau))$ 是最优轨迹分布，注意到当cost幅值比较大的时候这个最优轨迹更趋近于确定性的轨迹。通过约束，轨迹分布 $q(\tau)$ 就是参数控制的策略 π_θ 下形成的轨迹。之所以不直接对最优轨迹 $\rho(\tau)$ 优化策略，而是要经过中间的 $q(\tau)$，是由于在提供了环境模型的情况下，轨迹更容易被优化。

2. **dual gradient descent求解**

接下来使用dual gradient descent方法来求解上述优化问题，得到算法的大框架

$$\mathcal{L}(\theta, q, \lambda) = D_{\mathrm{KL}}(q(\tau)\|\rho(\tau)) + \sum_{t=1}^{T} \lambda_t D_{\mathrm{KL}}(q(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|q(\mathbf{x}_t, \mathbf{u}_t)).$$

$$\lambda_t \leftarrow \lambda_t + \eta D_{\mathrm{KL}}(q(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|q(\mathbf{x}_t, \mathbf{u}_t)). \quad (2)$$

Algorithm 1 Constrained guided policy search

1: Initialize the trajectories $\{q_1(\tau), \ldots, q_M(\tau)\}$
2: **for** iteration $k = 1$ to K **do**
3: Optimize each $q_i(\tau)$ with respect to $\mathcal{L}(\theta, q_i(\tau), \lambda_i)$
4: Optimize θ with respect to $\sum_{i=1}^{M} \mathcal{L}(\theta, q_i(\tau), \lambda_i)$
5: Update dual variables λ using Equation 2
6: **end for**
7: **return** optimized policy parameters θ

关于dual gradient descent方法可以参考这篇帖子：Dual Gradient Descent

这个主算法里面的第3行是在固定 π_θ 和 λ 的情况下优化轨迹 $q(\tau)$，第4行是在固定 $q(\tau)$ 和 λ 的情况下优化策略 π_θ，下面分别介绍如何进行这两步操作。

3. **轨迹优化**

假定轨迹 $q(\tau)$ 是根据线性策略 $q(u_t|x_t) \sim \mathcal{N}(K_t x_t + k_t, A_t)$ 和线性化的动力学 $q(x_{t+1}|x_t, u_t) \sim \mathcal{N}(f_{xt}x_t + f_{ut}u_t, F_t)$ 得到的，其整个轨迹上都是高斯分布，令该轨迹分布的均值为 $\hat{\tau} = (\hat{x}_{1:T}, \hat{u}_{1:T})$ 。说明：用 $\tau_t = \begin{bmatrix} x_t \\ u_t \end{bmatrix}$ 表示拼起来的向量；下标都表示求导；为了和文中一致下标 $\mathbf{x}$ 表示对 τ_t 的求导；有 hat 上标的表示原来的轨迹，没有 hat 上标的表示相对于原来轨迹的扰动（即小量）。

轨迹是由哪些量描述的呢？我们这里说的轨迹是一个轨迹的概率分布，它是一个高斯型的分布，因此决定它的除了均值 $\hat{\tau}_t$ 之外，还有 τ_t 的协方差矩阵

$$\Sigma_t = \begin{bmatrix} \mathbf{S}_t & \mathbf{S}_t\mathbf{K}_t^{\mathrm{T}} \\ \mathbf{K}_t\mathbf{S}_t & \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^{\mathrm{T}} + \mathbf{A}_t \end{bmatrix}.$$

其中 $S_t = Cov[x_t]$ 为定义，其他元素容易求出。

后面的过程比较复杂，我们先要明确一下要我们要干嘛。**我们的目标是把问题化为LQG问题（动力学线性、损失函数二次型），然后利用类似iLQG的最优控制公式求解当前轨迹 $q(\tau)$ 附近的更好的新轨迹 $q_{new}(\tau)$** 。我们这里说的轨迹是由如下参数构成的 $q(\tau) = \{\hat{\tau}, \Sigma(S, A, K)\}$ 。

首先对轨迹进行优化的时候认为 π_θ 和 λ 是固定的，我们可以把损失函数写成只与轨迹有关的形式，通过一些计算之后能够得到（注意，这里省略了对于时间 t 的求和）

$$\begin{aligned}
&\mathcal{L}(q) \approx \sum_{t=1}^{T} \frac{1}{2} \begin{bmatrix} \hat{\mathbf{x}}_t \\ \hat{\mathbf{u}}_t \end{bmatrix}^{\mathrm{T}} \ell_{\mathbf{xu},\mathbf{xu}t} \begin{bmatrix} \hat{\mathbf{x}}_t \\ \hat{\mathbf{u}}_t \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{x}}_t \\ \hat{\mathbf{u}}_t \end{bmatrix}^{\mathrm{T}} \ell_{\mathbf{xu}t} + \\
&\frac{1}{2}\mathrm{tr}\left(\Sigma_t \ell_{\mathbf{xu},\mathbf{xu}t}\right) - \frac{1}{2}\log|\mathbf{A}_t| + \frac{\lambda_t}{2}\log|\mathbf{A}_t| + \\
&\frac{\lambda_t}{2}(\hat{\mathbf{u}}_t - \mu_t^{\pi}(\hat{\mathbf{x}}_t))^{\mathrm{T}}\mathbf{A}_t^{-1}(\hat{\mathbf{u}}_t - \mu_t^{\pi}(\hat{\mathbf{x}}_t)) + \frac{\lambda_t}{2}\mathrm{tr}\left(\mathbf{A}_t^{-1}\Sigma_t^{\pi}\right) + \\
&\frac{\lambda_t}{2}\mathrm{tr}\left(\mathbf{S}_t\left(\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t)\right)^{\mathrm{T}}\mathbf{A}_t^{-1}\left(\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t)\right)\right),
\end{aligned}$$

类比 iLQG 算法里面的损失函数 $\mathcal{L} \sim (g + Gx)u + \frac{1}{2}u^T H u$ ，有最优控制 $u_t = k_t + K_t x_t$ 。我们的思路就是针对上面损失函数把相应的项找出来，其方法也很简单就是把上面的损失函数对控制 u_t 求导，可以得到

$$\begin{aligned}
\mathcal{L}_{\mathbf{u}t} &= Q_{\mathbf{u},\mathbf{u}t}\hat{\mathbf{u}}_t + Q_{\mathbf{u},\mathbf{x}t}\hat{\mathbf{x}}_t + Q_{\mathbf{u}t} + \lambda_t\mathbf{A}_t^{-1}(\hat{\mathbf{u}}_t - \mu_t^{\pi}(\hat{\mathbf{x}}_t)) \\
\mathcal{L}_{\mathbf{u},\mathbf{u}t} &= Q_{\mathbf{u},\mathbf{u}t} + \lambda_t\mathbf{A}_t^{-1},
\end{aligned}$$

但是当你求的时候会发现，求出来的形式跟文章一样，但是你式子里面出现单状态损失函数 ℓ（这个说法不是规范的说法，为了方便叙述我自己编的）的时候，文章里面都写的是 Q 。为什么这样呢？因为当你改变了某一处控制函数的时候，它后面相应的轨迹也都发生了改变，因此当某一处控制函数的改变导致了单状态损失函数的时候，应该连带上它对于后续轨迹上损失函数的改变，即用下面的 Q 来替代即可

$$
\begin{aligned}
Q_{\mathbf{x}\mathbf{u}t} &= \ell_{\mathbf{x}\mathbf{u}t} + f_{\mathbf{x}\mathbf{u}}^{\mathrm{T}} \mathcal{L}_{\mathbf{x}t+1} \\
Q_{\mathbf{x}\mathbf{u},\mathbf{x}\mathbf{u}t} &= \ell_{\mathbf{x}\mathbf{u},\mathbf{x}\mathbf{u}t} + f_{\mathbf{x}\mathbf{u}}^{\mathrm{T}} \mathcal{L}_{\mathbf{x},\mathbf{x}t+1} f_{\mathbf{x}\mathbf{u}},
\end{aligned}
$$

接下来，比对 iLQG 的算法公式能够得到新的最优控制扰动

$$
\mathbf{k}_t = -\left(Q_{\mathbf{u},\mathbf{u}t} + \lambda_t \mathbf{A}_t^{-1}\right)^{-1} \left(Q_{\mathbf{u}t} - \lambda_t \mathbf{A}_t^{-1} \mu_t^{\pi}(\hat{\mathbf{x}}_t)\right). \quad (3)
$$

$$
\mathbf{K}_t = -\left(Q_{\mathbf{u},\mathbf{u}t} + \lambda_t \mathbf{A}_t^{-1}\right)^{-1} \left(Q_{\mathbf{u},\mathbf{x}t} - \lambda_t \mathbf{A}_t^{-1} \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t)\right). \quad (4)
$$

注意到，其中 Q 的计算是包含损失函数在后一步上的导数的，因此该过程需要通过逆序的动态规划来求解。计算 Q 会用到的损失函数计算公式如下。

$$
\begin{aligned}
\mathcal{L}_{\mathbf{x}t} &= Q_{\mathbf{x},\mathbf{x}t}\hat{\mathbf{x}}_t + Q_{\mathbf{x},\mathbf{u}t}(\mathbf{k}_t + \mathbf{K}_t\hat{\mathbf{x}}_t) + \mathbf{K}_t^{\mathrm{T}} Q_{\mathbf{u},\mathbf{x}t}\hat{\mathbf{x}}_t + \\
&\quad \mathbf{K}_t^{\mathrm{T}} Q_{\mathbf{u},\mathbf{u}t}(\mathbf{K}_t\hat{\mathbf{x}}_t + \mathbf{k}_t) + Q_{\mathbf{x}t} + \mathbf{K}_t^{\mathrm{T}} Q_{\mathbf{u}t} + \\
&\quad \lambda_t(\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t))^{\mathrm{T}} \mathbf{A}_t^{-1}(\mathbf{K}_t\hat{\mathbf{x}}_t + \mathbf{k}_t - \mu_t^{\pi}(\hat{\mathbf{x}}_t)) \\
&= Q_{\mathbf{x},\mathbf{u}t}\mathbf{k}_t + \mathbf{K}_t^{\mathrm{T}} Q_{\mathbf{u},\mathbf{u}t}\mathbf{k}_t + Q_{\mathbf{x}t} + \mathbf{K}_t^{\mathrm{T}} Q_{\mathbf{u}t} + \\
&\quad \lambda_t(\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t))^{\mathrm{T}} \mathbf{A}_t^{-1}(\mathbf{k}_t - \mu_t^{\pi}(\hat{\mathbf{x}}_t)) \\
\mathcal{L}_{\mathbf{x},\mathbf{x}t} &= Q_{\mathbf{x},\mathbf{x}t} + Q_{\mathbf{x},\mathbf{u}t}\mathbf{K}_t + \mathbf{K}_t^{\mathrm{T}} Q_{\mathbf{u},\mathbf{x}t} + \mathbf{K}_t^{\mathrm{T}} Q_{\mathbf{u},\mathbf{u}t}\mathbf{K}_t + \\
&\quad \lambda_t(\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t))^{\mathrm{T}} \mathbf{A}_t^{-1}(\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t)), \quad (5)
\end{aligned}
$$

由此我们能够确定新的轨迹的均值 $\hat{\tau}_{new} = \hat{\tau} + \tau$ ，而该新轨迹的协方差矩阵 Σ_{new} 也需要被确定，观察上面关于协方差矩阵的定义，如果我们求出了 S_t 、 A_t 和 K_t ，就可以确定轨迹的协方差矩阵了。

知道新轨迹的均值之后怎么求出相应的 S_t 、 A_t 和 K_t 呢？自然想到的就是对损失函数相对于要求的这些量求导，并且令导数为零。同样，为了避免轨迹协方差变化对于后续轨迹产生的影响，我们需要把单状态损失函数替换为

$$Q_{\mathbf{xu},\mathbf{xu}t} = \ell_{\mathbf{xu},\mathbf{xu}t} + 2f_{\mathbf{xu}t}^{\mathrm{T}}\mathcal{L}_{\mathbf{S}t+1}f_{\mathbf{xu}t}.$$

求导可得

$$\begin{aligned}
\mathcal{L}_{\mathbf{A}t} &= \frac{1}{2}Q_{\mathbf{u},\mathbf{u}t} + \frac{\lambda_t - 1}{2}\mathbf{A}_t^{-1} - \frac{\lambda_t}{2}\mathbf{A}_t^{-1}\mathbf{M}\mathbf{A}_t^{-1} \\
\mathcal{L}_{\mathbf{K}t} &= Q_{\mathbf{u},\mathbf{u}t}\mathbf{K}_t\mathbf{S}_t + Q_{\mathbf{u},\mathbf{x}t}\mathbf{S}_t + \lambda_t\mathbf{A}_t^{-1}\mathbf{K}_t\mathbf{S}_t - \\
&\quad \lambda_t\mathbf{A}_t^{-1}\mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t)\mathbf{S}_t \\
\mathcal{L}_{\mathbf{S}t} &= \frac{1}{2}\left[Q_{\mathbf{x},\mathbf{x}t} + \mathbf{K}_t^{\mathrm{T}}Q_{\mathbf{u},\mathbf{x}t} + Q_{\mathbf{x},\mathbf{u}t}\mathbf{K}_t + \mathbf{K}_t^{\mathrm{T}}Q_{\mathbf{u},\mathbf{u}t}\mathbf{K}_t\right. \\
&\quad \left.+\lambda_t(\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t))^{\mathrm{T}}\mathbf{A}_t^{-1}(\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t))\right], \qquad (6)
\end{aligned}$$

where $\mathbf{M} = \Sigma_t^{\pi} + (\hat{\mathbf{u}}_t - \mu_t^{\pi}(\hat{\mathbf{x}}_t))(\hat{\mathbf{u}}_t - \mu_t^{\pi}(\hat{\mathbf{x}}_t))^{\mathrm{T}} + (\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t))\mathbf{S}_t(\mathbf{K}_t - \mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t))^{\mathrm{T}}$.

零导数为零，可得

$$\mathbf{K}_t = -\left(Q_{\mathbf{u},\mathbf{u}t} + \lambda_t\mathbf{A}_t^{-1}\right)^{-1}\left(Q_{\mathbf{u},\mathbf{x}t} - \lambda_t\mathbf{A}_t^{-1}\mu_{\mathbf{x}t}^{\pi}(\hat{\mathbf{x}}_t)\right). \qquad (7)$$

$$\mathbf{A}_t Q_{\mathbf{u},\mathbf{u}t} \mathbf{A}_t + (\lambda_t - 1)\mathbf{A}_t - \lambda_t \mathbf{M} = 0. \qquad (8)$$

由于 S_t 、 A_t 和 K_t 相互依赖，因此我们需要迭代多次直到它们数值收敛。

4. 策略优化

有了更优的轨迹之后，我们接下来把轨迹固定，来优化相应的策略，这一步的优化相对来讲就比较直接了：写出关于损失函数关于策略参数 θ 的表示，然后使用梯度方法做优化即可，比如SGD或者LBFGS。

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{t=1}^{T} \lambda_t \sum_{i=1}^{N} D_{\mathrm{KL}}(\pi_\theta(\mathbf{u}_t|\mathbf{x}_{ti}) \| q(\mathbf{u}_t|\mathbf{x}_{ti})) \\ &= \sum_{t=1}^{T} \lambda_t \sum_{i=1}^{N} \frac{1}{2} \{ \mathrm{tr}(\Sigma_t^\pi(\mathbf{x}_{ti})\mathbf{A}_t^{-1}) - \log|\Sigma^\pi(\mathbf{x}_{ti})| + \\ &\quad (\mathbf{K}_t\mathbf{x}_{ti} + \mathbf{k}_t - \mu^\pi(\mathbf{x}_{ti}))^{\mathrm{T}} \mathbf{A}_t^{-1} (\mathbf{K}_t\mathbf{x}_{ti} + \mathbf{k}_t - \mu^\pi(\mathbf{x}_{ti})) \} .\end{aligned}$$

算法：

Algorithm 1 Constrained guided policy search

1: Initialize the trajectories $\{q_1(\tau), \ldots, q_M(\tau)\}$
2: **for** iteration $k = 1$ to K **do**
3: Optimize each $q_i(\tau)$ with respect to $\mathcal{L}(\theta, q_i(\tau), \lambda_i)$
4: Optimize θ with respect to $\sum_{i=1}^{M} \mathcal{L}(\theta, q_i(\tau), \lambda_i)$
5: Update dual variables λ using Equation 2
6: **end for**
7: **return** optimized policy parameters θ

Algorithm 2 Trajectory optimization iteration

Compute $f_{\mathbf{xu}t}$, $\mu^{\pi}_{\mathbf{x}t}$, $\ell_{\mathbf{xu}t}$, $\ell_{\mathbf{xu},\mathbf{xu}t}$ around $\hat{\tau}$
for $t = T$ to 1 **do**
  Compute $\mathbf{K}_t$ and $\mathbf{k}_t$ using Equations 3 and 4
  Compute $\mathcal{L}_{\mathbf{x}t}$ and $\mathcal{L}_{\mathbf{x},\mathbf{x}t}$ using Equation 5
end for
Initialize $\alpha \leftarrow 1$
repeat
  Obtain new trajectory $\hat{\tau}'$ using $\mathbf{u}_t = \alpha\mathbf{k}_t + \mathbf{K}_t\mathbf{x}_t$
  Decrease step size α
until $\mathcal{L}(\hat{\tau}', \mathbf{K}, \mathbf{A}) > \mathcal{L}(\hat{\tau}, \mathbf{K}, \mathbf{A})$
Compute $f_{\mathbf{xu}t}$, $\mu^{\pi}_{\mathbf{x}t}$, $\ell_{\mathbf{xu}t}$, $\ell_{\mathbf{xu},\mathbf{xu}t}$ around $\hat{\tau}'$
repeat
  Compute $\mathbf{S}_t$ using current $\mathbf{A}_t$ and $\mathbf{K}_t$
  for $t = T$ to 1 **do**
    repeat
      Compute $\mathbf{K}_t$ using Equation 7
      Compute $\mathbf{A}_t$ by solving CARE in Equation 8
    until $\mathbf{A}_t$ and $\mathbf{K}_t$ converge (about 5 iterations)
    Compute $\mathcal{L}_{\mathbf{S}t}$ using Equation 6
  end for
until all $\mathbf{S}_t$ and $\mathbf{A}_t$ converge (about 2 iterations)
return new mean $\hat{\tau}'$ and covariance terms $\mathbf{A}_t, \mathbf{K}_t$

其中算法1是主要的算法，算法2是其中第3行的具体算法。其中值得说明的是，第7-10行在最优开环控制上面缩小，以找到确定能改善损失函数的开环控制，其原因是由于环境是非线性的，而我们在局域假设了线性，因此如果开环控制变化的太多会导致假设不成立，形成的损失函数甚至不降反升。另外，第10行的大于符号应该是写错了，应该是小于符号，这里希望做的是最小化损失函数。

从绿框公式到红框公式的推导？

$$
\begin{aligned}
D_{KL}(q(\tau)||\rho(\tau)) &= \sum_{t=1}^{T} D_{KL}(q(x_t,u_t)||\rho(x_t,u_t)) \\
&= \sum_{t=1}^{T} \int q(x_t,u_t)[\log q(x_t,u_t) - \log(\rho(x_t,u_t))]dx_t du_t \\
&= \sum_{t=1}^{T} \mathcal{H}(q(x_t,u_t)) + \mathbb{E}_{q(x_t,u_t)}[l(x_t,u_t)]
\end{aligned}
$$

注意每一步的熵函数其实并不独立，后面的会依赖前面的，所以上面这样写其实不规范，不过我们在运算的过程中记住这一点就好了，同时，高斯分布的熵函数为

```
In[26]:= Assuming[A > 0, ∫_-∞^∞ 1/Sqrt[2 π A] Exp[-(x - μ)^2/(2 A)] (Log[1/Sqrt[2 π A]] - (x - μ)^2/(2 A)) dx]
Out[26]= 1/2 (-1 - Log[2 A π])
```

有

$$
\begin{aligned}
\mathcal{H}(q(x_t,u_t)) &= \int q(x_t,u_t)\log q(x_t,u_t)dx_t du_t \\
&= \int q(x_t,u_t)\log q(x_t|x_{t-1},u_{t-1})dx_t du_t + \int q(x_t,u_t)\log q(u_t|x_t)dx_t du_t \\
&= \int q(x_t)\log q(x_t|x_{t-1},u_{t-1})dx_t + \int q(u_t)\log q(u_t|x_t)du_t \\
&= -1-\log 2\pi - \frac{1}{2}|A_t| - \frac{1}{2}|F_t| \\
&\sim -\frac{1}{2}|A_t|
\end{aligned}
$$

其中随后一步是因为，求导的损失函数是要对于轨迹参数求导的，无关的常数项就可以不写了，有

$$
\begin{aligned}
\mathcal{L}(q) = \sum_{t=1}^{T} E_{q(\mathbf{x}_t,\mathbf{u}_t)}[\ell(\mathbf{x}_t,\mathbf{u}_t)] - \frac{1}{2}\log|\mathbf{A}_t| + \\
\lambda_t E_{q(\mathbf{x}_t)}[D_{\mathrm{KL}}(\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|q(\mathbf{u}_t|\mathbf{x}_t))].
\end{aligned}
$$

注意到 $\mathbf{E}_q[\tau_t] = \hat{\tau}_t$ ，有

$$
\begin{aligned}
\mathbf{E}_{q(x_t,u_t)}[l(x_t,u_t)] &= \mathbf{E}_{q(x_t,u_t)}[l(0,0) + l_{xut}^T\tau_t + \frac{1}{2}\tau_t^T l_{xuxut}\tau_t] \\
&= l(0,0) + l_{xut}^T\hat{\tau}_t + \frac{1}{2}\mathbf{E}_{q(x_t,u_t)}[(\tau_t-\hat{\tau}_t+\hat{\tau}_t)^T l_{xuxut}(\tau_t-\hat{\tau}_t+\hat{\tau}_t)] \\
&= l(0,0) + l_{xut}^T\hat{\tau}_t + \frac{1}{2}\mathbf{E}_{q(x_t,u_t)}[(\tau_t-\hat{\tau}_t)^T l_{xuxut}(\tau_t-\hat{\tau}_t)] + \frac{1}{2}\mathbf{E}_{q(x_t,u_t)}[(\tau_t-\hat{\tau}_t)^T l_{xuxut}\hat{\tau}_t] \\
&\quad + \frac{1}{2}\mathbf{E}_{q(x_t,u_t)}[\hat{\tau}_t^T l_{xuxut}(\tau_t-\hat{\tau}_t)] + \frac{1}{2}\mathbf{E}_{q(x_t,u_t)}[\hat{\tau}_t^T l_{xuxut}\hat{\tau}_t] \\
&= l(0,0) + l_{xut}^T\hat{\tau}_t + \frac{1}{2}tr(\Sigma_t l_{xuxut}) + 0 + 0 + \frac{1}{2}\hat{\tau}_t^T l_{xuxut}\hat{\tau}_t \\
&\sim l_{xut}^T\hat{\tau}_t + \frac{1}{2}tr(\Sigma_t l_{xuxut}) + \frac{1}{2}\hat{\tau}_t^T l_{xuxut}\hat{\tau}_t
\end{aligned}
$$

其中最后一个等号的成立可以参看这个课件的第六页。

同时注意到高斯函数的交叉熵

```
In[25]:= Assuming[A1 > 0 && A2 > 0, Integrate[1/Sqrt[2 π A1] Exp[-(x - μ1)^2/(2 A1)] (Log[1/Sqrt[2 π A2]] - (x - μ2)^2/(2 A2)), {x, -∞, ∞}]]
Out[25]= -(A1 + (μ1 - μ2)^2 + A2 Log[2 A2 π])/(2 A2)
```

有

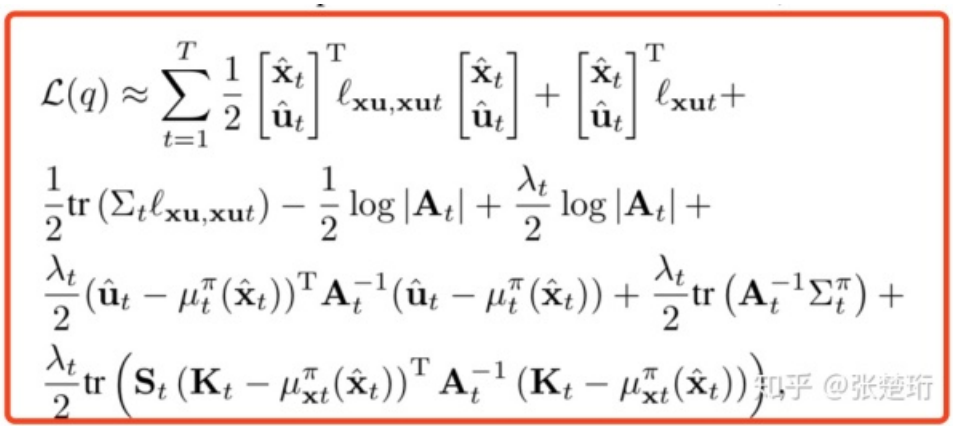
$$
\begin{aligned}
&\mathbf{E}_{q(x_t)}[D_{KL}(\pi_\theta(u_t|x_t)||q(u_t|x_t))] \\
=&\mathbf{E}_{q(x_t)}[\int \pi_\theta(u_t|x_t)\log\pi_\theta(u_t|x_t)du_t] - \mathbf{E}_{q(x_t)}[\int \pi_\theta(u_t|x_t)\log q(u_t|x_t)du_t] \\
\sim&0 + \frac{1}{2}tr(A_t^{-1}\Sigma_t^\pi) + \frac{1}{2}\log|A_t| + \frac{1}{2}\mathbf{E}_{q(x_t)}[(u_t-\mu_t^\pi(x_t))^T A_t^{-1}(u_t-\mu_t^\pi(x_t))] \\
=&\frac{1}{2}tr(A_t^{-1}\Sigma_t^\pi) + \frac{1}{2}\log|A_t| + tr(S_t(K_t-\mu_{xt}^\pi)^T A_t^{-1}(K_t-\mu_{xt}^\pi)) + (\hat{u}_t-\mu_\pi(\hat{x}_t))^T A_t^{-1}(\hat{u}_t-\mu_\pi(\hat{x}_t))
\end{aligned}
$$

其中约等于符号是因为前一项是完全关于策略的，与我们这里要优化的轨迹无关，因此不理会这一项，其中比较复杂的最后一项细写下来是

$$
\begin{aligned}
&\mathbf{E}_{q(x_t)}[(K_t x_t-\mu_t^\pi(x_t))^T A_t^{-1}(K_t x_t-\mu_t^\pi(x_t))] \\
=&\mathbf{E}_{q(x_t)}[(K_t x_t - K_t\hat{x}_t + \hat{u}_t - \mu_\pi(\hat{x}_t) + \mu_\pi(\hat{x}_t) - \mu_\pi(x_t))^T A_t^{-1}(\cdots)] \\
=&\mathbf{E}_{q(x_t)}[((K_t-\mu_{xt}^\pi)(x_t-\hat{x}_t) + (\hat{u}_t-\mu_\pi(\hat{x}_t)))^T A_t^{-1}(\cdots)] \\
=&\mathbf{E}_{q(x_t)}[((K_t-\mu_{xt}^\pi)(x_t-\hat{x}_t))^T A_t^{-1}(\cdots)] + 0 + 0 + \mathbf{E}_{q(x_t)}[(\hat{u}_t-\mu_\pi(\hat{x}_t))^T A_t^{-1}(\cdots)] \\
=&tr(S_t(K_t-\mu_{xt}^\pi)^T A_t^{-1}(K_t-\mu_{xt}^\pi)) + (\hat{u}_t-\mu_\pi(\hat{x}_t))^T A_t^{-1}(\hat{u}_t-\mu_\pi(\hat{x}_t))
\end{aligned}
$$

其中轨迹的控制和策略的控制都有直流项的，但是由于后面求协方差并不影响，为了简便，这里就不写直流项了。

最后拼起来有



推导过程参考了【强化学习大讲坛】强化学习进阶 第九讲 引导策略搜索。

发布于 2018-10-01

机器学习 算法 强化学习 (Reinforcement Learning)

文章被以下专栏收录