Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models

Kurtland Chua
University of California, Berkeley
kchua@berkeley.edu

Roberto Calandra
University of California, Berkeley
roberto.calandra@berkeley.edu

Rowan McAllister
University of California, Berkeley
rmcallister@berkeley.edu

Sergey Levine
University of California, Berkeley
svlevine@eecs.berkeley.edu

【强化学习算法 12】PETS

张楚珩
清华大学 交叉信息院博士在读

并不是宠物的意思，其实它是 PE+TS，即ensembles of probabilistic model + trajectory optimization。

原文传送门：

Chatzilygeroudis K, Vassiliades V, Stulp F, et al. A survey on policy search algorithms for learning robot controllers in a handful of trials[J]. arXiv preprint arXiv:1807.02303, 2018.

特色：

这是一个model-based的方法。一般来说，model-based的好处是由于其对环境的动力学特性（dynamics，即 $p(s_{t+1}|s_t,a_t)$ ）进行建模，其sample efficiency更好，在样本很少的情况下学习的更好。但是一般来说其渐近表现不如model-free的算法好，即收敛之后的性能。这篇文章认为其主要原因是其采用的model和planning的capacity不够，因此组合使用了所有方法里面capacity最大的方法，形成了该算法。该算法mujoco上面使用更少的样本，达到了类似SAC和PPO的渐近性能。

背景：

model-based算法有两个关键的问题，一个是建立什么样的模型，一个是怎样使用模型去做控制。

模型的选择有1）nonparametric类方法，比如Bayesian nonparametric model；2）local models，比如guided policy search，这个方法主要是反复的找更好的轨迹，并且把策略朝着该轨迹上拟合，之后会在专栏里面讲；3）parametric models，比如使用神经网络来拟合，这种方法又分为deterministic模型和stochastic模型，deterministic计算更简单但是在样本少的时候会更容易overfit。

本文主要考虑使用神经网络拟合的第3类模型。这类模型有几个可能的形式：

1. Deterministic（D）：神经网络接受 s_t 和 a_t 输入，直接输出 a_{t+1} ，有了数据集 $\mathcal{D}=\{((s_n,a_n),s_{n+1})\}$ 之后，模型的拟合就是一个有监督学习的问题了，可以很容易解决。其损失函数为 $loss=\sum\|s_{n+1}-f_\theta(s_n,a_n)\|$ ；
2. Probabilistic（P）：这里的神经网络接受 s_t 和 a_t 输入，输出 a_{t+1} 的一个分布，常见的是高斯分布 $f=Pr(s_{n+1}|s_n,a_n)\sim N(\mu_\theta(s_n,a_n),\Sigma_\theta(s_n,a_n))$ 。其损失函数为 $loss=-\sum\log f_\theta(s_{n+1}|s_n,a_n)$ ；
3. Deterministic Ensemble（DE）：这里的模型就是一个神经网络的ensemble，最后的预测值是ensemble的平均 $f=\frac{1}{B}\sum_{b=1}^{B}f_{\theta_b}$ ，其中每个神经网络是通过bootstrap的方式训练得到的；
4. Probabilistic Ensemble（PE）：和上述类似，只不过是P模型的ensemble。

上述模型的主要区别就是能够刻画不确定性的种类。文中提出了两种不确定性：一种是aleartoric uncertainty，即环境本身的不确定性；另一种是epistemic uncertainty，它是由于样本较少，估计不充分引起的不确定性。P相对于D来说（PE相对于DE来说）增加了刻画第一种不确定性的能力；ensemble的加入增加了刻画第二种不确定性的能力。

使用模型来做控制的方法主要有1）policy based method，即利用模型去探索并找到一个好的policy $\pi: s_t \to a_t$ ；2）model predictive control（MPC），这种方法不去寻求一个依赖于当前状态的策略，而是每次遇到一个新的选择的时候，都基于模型去逐步预测、模拟、做选择。个人感觉两者的区别就是learning和planning的区别。

本文主要考虑使用的是第2类方法，这类模型有几个可能的形式：

1. Deterministic（E）：在逐步的模拟中，每一步都只使用均值，而忽略其他的分布信息；
2. Parametric类方法：即每一步都把分布去参数化，用一些比如Gaussian或者Mixture Gaussian模型去去拟合。这类方法包括Moment Matching（MM）方法和Distribution Sampling（DS）方法；
3. Particle类方法：即演化一组模特卡洛模拟的粒子，把这群粒子的分布当做选择一序列action之后的分布。当使用ensemble类模型的时候，根据每一步是否更换ensemble内不同的模型，可以分为TS1（每一步都更换不同的模型）和TS ∞ （每个粒子从头到尾只使用一个模型），其中TS表示trajectory sampling。

实验过程：

文中的PETS算法就是使用了PE模型和TS的planning方法进行组合得到的。注意到TS方法其实是给定一组actions $a_{t:t+T}$ 然后使用学到的模型来评价这组actions的。那么应该怎样生成一组随机的actions呢？一种方法是均匀随机采样，但是对于高纬度的任务来说，不可能均匀随机采样到一组连续的较好actions。所以这里用了有指向性的随机采样，即使用CEM方法对于哪些action较好有个大致的方向，然后基于它来采样。

算法：

Algorithm 1 Our model-based MPC algorithm '*PETS*':

Initialize data $\mathbb{D}$ with a random controller for one trial.
for Trial $k = 1$ to K **do**
  Train a PE dynamics model $\widetilde{f}$ given $\mathbb{D}$.
  for Time $t = 0$ to TaskHorizon **do**
    for Actions sampled $\boldsymbol{a}_{t:t+T} \sim \text{CEM}(\cdot)$, 1 to NSamples **do**
      Propagate state particles $\boldsymbol{s}_\tau^p$ using TS and $\widetilde{f}|\{\mathbb{D}, \boldsymbol{a}_{t:t+T}\}$.
      Evaluate actions as $\sum_{\tau=t}^{t+T} \frac{1}{P} \sum_{p=1}^{P} r(\boldsymbol{s}_\tau^p, \boldsymbol{a}_\tau)$
      Update CEM$(\cdot)$ distribution.
    Execute first action $\boldsymbol{a}_t^*$ (only) from optimal actions $\boldsymbol{a}_{t:t+T}^*$.
    Record outcome: $\mathbb{D} \leftarrow \mathbb{D} \cup \{\boldsymbol{s}_t, \boldsymbol{a}_t^*, \boldsymbol{s}_{t+1}\}$.

Planning的时候 $r(s,a)$ 是怎么得到的？

是认为建模规定的reward/cost。

参看了github上面的代码，对于不同的gym任务，代码硬编码了不同的 $r(s,a)=r(s)+r(a)$ ，具体的每个任务不同的reward参看其dmbrl/config文件夹下面的文件，使用的地方在dmbrl/controllers/MPC.py里面的self.obs_cost_fn和self.ac_cost_fn。

我们感觉这也是这篇文章最为诟病的地方了，只对transition建模，而reward就全靠人为定义，这样使得该算法的适用性很差。

如何理解PE既能抓住aleatoric uncertainty又能抓住epistemic uncertainty？

先看图

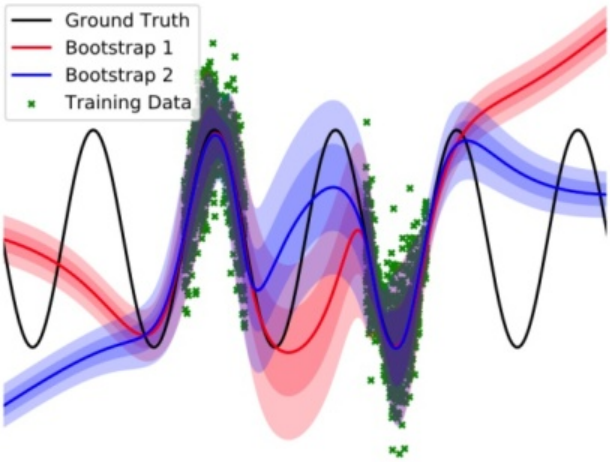

Figure A.5: Our probabilistic ensemble (PE) dynamics model: an ensemble of two bootstraps (for visual clarity, we normally use five bootstraps), each a probabilistic neural network that captures aleatoric uncertainty (in this case: observation noise). Note the bootstraps agree near data, but tend to disagree far from data. Such bootstrap disagreement represents our model's epistemic uncertainty.

比如要拟合图中的绿色点，ensemble里面两个不同的模型（红和蓝）使用了bootstrap的不同数据点拟合出了不同的模型。**各个模型variance的mean反映所刻画数据本身的不确定性，即aleatoric uncertainty；而各个模型mean的variance反映的是由于数据较少形成的不确定性，即epistemic uncertainty。**可以这么理解，如果数据足够多，不管用哪部分数据，拟合出来的模型差距不应该太大，因此模型拟合结果的方差不大，即mean的variance小。

编辑于 2018-10-01

强化学习 (Reinforcement Learning)

文章被以下专栏收录

强化学习前沿
读呀读paper

进入专栏