1 摘要

基于模型的RL允许一个agent发现好的policy

2 算法

深度PILCO和PILCO的不同在于深度PILCO使用能够扩展到[1] 深度神经网络。和PILCO类似，我们的策略搜索算法交替进行拟合动力学模型和评估policy，改善policy三步。
代替PILCO的高斯过程以深度网络是个相当复杂的尝试，因为我们希望我们的dynamics模型能够保持概率的本性，捕捉输出和输入不确定性。

2.1 输出不确定性

首先，我们要求输出不确定性，对于PILCO的数据有效性是关键的。然而，简单的NN模型不能表示输出的模型不确定性，因此不能捕捉我们对于隐藏系统的dynamics的无知。为了结局这个问题，我们使用贝叶斯神经网络。
在低数据设定下，BNN利用基于NN权值的后验分布来表示模型不确定性 。然而，BNN的真实后验复杂而棘手。我们的近似方法是使用variational inference，当我们发现一个分布在可处理的家族，这个家族最小化和真实后验的KL距离。[6]表明dropout能够解释为variational 贝叶斯近似，在这里近似的分布是两个小方差的高斯分布，其中一个高斯分布的均值固定为0.权值的不确定性导致预测的不确定性，通过使用蒙特卡洛集成来边缘化近似的后验。这相当于[2]常规的dropout程序仅仅在测试的时候dropout，从我们的dynamics模型中给出输出不确定性。
这个方法也提供小数据NN的洞悉。[6]表明网络的权值腐蚀可以被参数化作为数据大小，dropout概和观察噪声的函数。再加上适应性的学习率最优化技巧，要求微调的参数的数量可以忽略不计。

2.2 输入不确定性

第二个难点是处理输入不确定性。为了在不确定性的dynamics下计划，PILCO通过dynamics模型分析性地传送了状态分布。为了达到这一点，动力学模型必须从给定的时间步骤传递不确定性的动力学输出作为下一时刻的不确定的输入。这种处理方法不能用在NN上。
为了从动力学模型中得到一个分布，我们借助于particle方法（算法2）。这涉及到从输入分布中采样一系列particles，将这些particles通过BNN模型传播（和采样不确定的输出），今儿产生particles的输出分布。
这个方法在过去未能成功。原因在于最优化表面局部最优值的丰富性阻止了BFGS最优化方法。[7]认为这可能是由于使用了有限数量的particles以及确定性的最优化。为了避免这些问题，我们在每个最优化步骤中随机采样了一系列的particles，给出我们对于这个目标函数的无偏估计。然后使用随机最优化方法Adam代替BFGS。
我们发现在每个时间步骤中拟合高斯分布到输出的状态分布，正像PILCO所做的，是很重要的一点。这个瞬间匹配避免了dynamics模型的多峰问题。
用宽广的高斯模型拟合多峰分布导致了目标函数对许多高代价的高斯跨度状态取平均。通过强制一个单峰的拟合，算法惩罚了能够导致预测状态分支的policy，

[1] scale to,扩展到，按比例排列
[2] amount to,相当于，总计为

而分支这经常是失去控制的一个先兆。 这也可以看做是当分支发生时，光滑期望代价的梯度，简化了控制器的最优化。（这个之后解释，以实验为例）我们假设这是在PILCO中的一个重要的建模选择，并且在我们的实验中评估这个假设。

2.3 从dynamics模型中采样函数

不像PILCO，我们的方法允许从dynamics模型中采样个别的函数[3]以及在整个实验中遵守一个函数。这是因为BNN的重复性应用可以被视为一个简单的贝叶斯RNN（RNN，输入仅仅在第一个时间步骤中给出）。贝叶斯RNN的近似推断通过对dynamics模型采样一次，然后在所有的时间步骤中使用相同的权值。利用dropout，这个在整个rollout通过在所有的时间步骤中采样和固定dropout mask来完成。PILCO没有考虑相邻状态转移之间模型不确定性的时间关联，导致PILCO低估了未来时间步骤的状态不确定性。
另一个审视我们作为贝叶斯RNN的dynamics模型的结果是，模型可以容易地扩展到更加有趣的RNN比如贝叶斯LSTM，捕捉在状态之间的长期依赖。在这篇文章中，我们把模型限制到马尔可夫系统，一个简单的贝叶斯RNN模型足够在给定一个单独输入状态的条件下预测一个单独的输出。
图5（下面仔细描述）表明实验拥有固定的控制器，为cartpole swing-up任务采样动力学模型。这些通过从最初的分布中采样particles，在整个实验过程中为每个particle采样和固定一个dropout mask来生成。

3 实验建立

[3]individual,个人的，个别的，独特的