**1.Introduction**

奖励函数的设计是解决强化学习问题的关键挑战，对于复杂应用，我们很难手工设计出一个好的奖励函数，（which决定了强化学习的成败），同时，对于存在高风险的环境，我们在训练智能体的过程中无法与环境进行交互和测试策略。

临床决策是一个典型的例子，我们希望从医学专家那里学习策略，但在训练过程中对患者实施策略是完全不切实际和不道德的。此外，我们不仅对演示的策略感兴趣，而且对演示者的偏好和目标（即指导行为的奖励函数）感兴趣。在这种情况下，学习必须完全基于专家演示者记录的轨迹进行，该轨迹仅显示访问的状态和在每种情况下采取的动作。

虽然模仿学习的策略可以匹配演示者的行为，但随着理解动机的增加，可以通过逆向强化学习来实现。(Abbeel & Ng，2004)将潜在奖励假设为特征的线性函数，然后使用前向强化学习相对于奖励的最优策略。(Klein et al., 2011、 Klein et al., 2013 )将该算法扩展为离线策略学习。(Lee et al., 2019)中的DSFN是目前离线最大边际方法中的state-of-art，然而DSFN存在以下问题：（1）使用DQN作为前向强化学习方法，而replay buffer是固定的，因此初始策略需要与学习策略相近，这就需要对初始策略进行预训练；（2）使用神经网络进行特征提取，最终得到的关于特征的奖励函数是不可解释的，这一点尤为重要，因为这与逆强化学习的初衷--得到演示者的目的和偏好相违背。一般脱离了在线条件，对策略进行离线评估，就需要进行warm-start，而warm-start对数据量的要求较高，离线评估时，会遇到外推误差，即选取到了专家序列中没有的动作时，Q值的估计会出现较大的偏差。解决的办法是将动作限制在一定范围内

针对以上问题，我们在(Abbeel & Ng，2004)算法基础上，使用BCQ进行前向策略评估与改进，优点：一是使用的Double-Q学习方法可以对Q估计更加准确；二是在学习Q值的近似的同时，将动作限制在演示的动作集合中；三是只使用演示数据集，不再与环境进行交互，实现了完全离线学习。

我们使用raw state作为特征，来观察学出的奖励是否合理

**2.related work**

在这篇论文中，我们的目的是根据专家数据得到最优策略的同时，恢复隐式指导专家做出决策的奖励函数。我们所使用的框架是NG2004提出的基于最大边距的算法。该算法需要访问环境去优化当前策略，对离线设置进行探索的有。

大多数的IRL算法都需要访问环境去优化策略【Abbeel and Ng, 2004】【Ziebart et al., 2008】【Fu et al., 2017】。【Klein et al., 2011】、【Klein et al., 2012】的缺点；【Lee et al. (2019)】提出使用DSFN来估计特征期望，用TRIL网络暖启动和提取特征，但奖励函数对于这样的特征变得不可解释，而且由于异策略评估会带来冷启动的问题。最近，【14】提出AVRIL使用变分推断来得到奖励函数的一种近似分布，【Daniel Jarrett， 2020】提出EDM使用identifies parameterizations

of the (discriminative) model of a policy with the (generative) energy function for state distributions。但这两种方法都没有显式地学习奖励函数。与我们工作很相近的是【Ioana Bica， 2021】，他们使用反事实推理来model奖励函数，通过估计不同动作的影响，解决了off-policy策略评估的冷启动问题。

据我们所知，我们是第一个提出将BRL方法用于BIRL的策略优化中，以解决异策略优化的冷启动问题。我们的方法成功的得到了可解释的奖励并且无需暖启动。表一中比较了我们的方法和相关的工作

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method |  | 是否  ｏｆｆｌｉｎｅ | 奖励的可解释性 | 不需要ｗａｒｍｓｔａｒｔ |  |
| BC |  | YES | NO | ／ |  |
| AL |  | NO | YES | YES |  |
| EDM |  | YES | NO | YES |  |
| AVRIL |  | YES | NO | YES |  |
| CIRL |  | YES | YES | YES |  |
| DSFN |  | YES | NO | NO |  |
| OURS |  | YES | YES | YES |  |

注意：YES表示不需要warm-start

**3.Background**

**Preliminaries**

我们使用标准的马尔科夫决策过程（MDP）设置，which定义为一个五元组（S,A，T，r，γ），with状态集合S，动作集合A，转移概率矩阵T（which we don’t know），奖励函数r，折扣因子γ。策略π（a|s）表示在状态s(∈S)处执行动作a(∈A)的概率。价值函数定义为，其中s0∈S0,表示初始状态。状态价值函数定义为:

(1)

衡量在策略π下状态s和动作a的优劣，其中表示策略π下的期望。RL目标是求出一个策略Π\*，使得Q和V的值最大，即。特征期望是一个很重要的概念，是IRL算法中的一个核心，特征期望定义为：the expected discounted accumulated feature，

。(2)其中Φ是特征映射S->[0,1]k,我们为了得到奖励对于原始特征的函数，Φ采用恒等映射，即Φs=s。

我们的设定是R是未知的，且训练过程中不可与环境交互，我们所能得到的只有根据专家策略采样得到专家数据集。,T 表示trajectory

**Imitation Learning**

模仿学习【2】的目标是学习一个策略Π\*，使得和专家策略尽量接近。

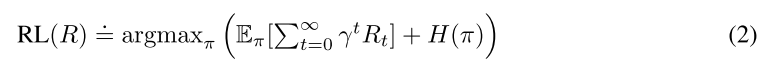
Π\*= （s属于S，）。

其中L代表某种损失函数。BC【5】是一个高效但十分简单的方法，它忽略了状态之间的转移关系这一宝贵信息，仅仅使用监督学习进行分类预测。由于BC未考虑到长序列规划（no-longterm-planning），这导致BC受到compounding errors的影响而漂移到远离专家数据的地方【6】。一旦飘移到这些out-of-distribution的状态，智能体无法做出正确的决策。

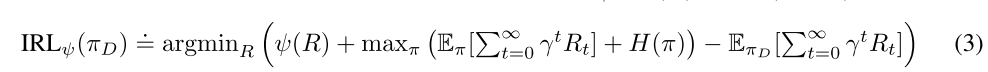
**Inverse Reinforcement Learning**

为了使用状态转移的信息，逆强化学习算法【9】被提出，这种算法不仅能得到最优策略，而且可以recover一个奖励。它使用前向强化学习和逆向强化学习交替运行：

前向强化学习可以表示为：



IRL：



其中，φ（R）表示奖励函数的正则，在【9】中使用的是线性约束。

对抗模仿学习【7】将上面两个过程解释为GAN中的生成器和判别器，证明了结果的收敛性。

这一类的学习方式都是在线学习【注：本文中所提到的在线学习都是指 需要与环境交互，offline或batch都是指完全不与环境交互】，前向的RL需要访问环境来训练。NG【8】提出的算法通过最大化学习策略与专家策略的奖励间距来优化，需要进行交互进行策略评价和优化，这一部分的细节在之后章节。

**Batch IRL**

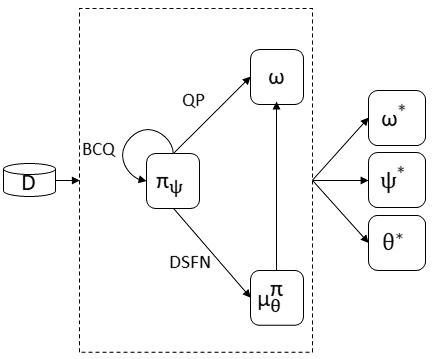
但是，很多时候我们并不能获得与环境的交互。一种简单的方法是学习环境动态的模拟器【10】，但这种方法只适用于低维度的环境，而且需要采集大量的数据。DSFN【11】使用off-policy的方法实施前向RL，但随之带来的问题是要对初始策略进行暖启动，并且要精心设计特征函数。而近年来一些优秀的离线方法EDM【12】，VDICE【13】，AVRIL【14】并没有显式的学习奖励函数。

假设我们，并且假设奖励函数是关于状态动作特征的线性函数，，其中是关于状态s和动作a特征映射。关于特定状态s和动作a的特征期望定义为策略π下，累积折扣“特征访问”的期望：

总体的特征期望定义为对初始状态集合的期望：

**4.Method**

我们使用的是NG2004提出的最大间距学徒学习框架，在此基础上，我们将交互式的前向RL修改为离线学习，并且使用神经网络拟合特征期望。



模型框架如图1，我们使用神经网络来近似特征期望，以及动作价值函数Q；输入为专家数据集，输出为特征期望参数、最优策略以及奖励函数参数。

算法1描述了我们提出的BCAL，BCAL迭代地更新奖励函数参数来最小化专家特征期望与当前策略得到的特征期望的距离。当前策略的特征期望由FEN估计，该方法是一种off-policy的评价方法，在4.1中详述。相比于DSFN使用DQN方法固定buffer优化当前策略，我们使用batch-constraint的方法。我们的方法不再需要初始策略pai0与专家策略相近，并且不需要额外的网络来学习一个合适的特征表示。在4.2中详述。

**Algorithm 1** (Batch, Max-Margin) BCAL

1:**Input:** Dataset D,convergence threshold e

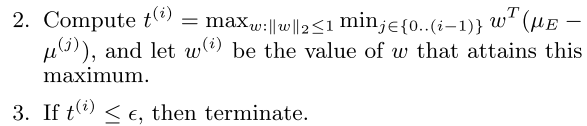
2:Random Initial:FEN weights ,reward weights w, Q network weights,

3:计算mu-e

4: for i = 0 : n do

5: Evaluate µθπ(i) using FEN

6: optimize w(i) by solving max-margin QP





这一部分用OQ；

5: train FEN use pai(i+1)

End for



**Algorithm 2** FEN

1:**Input:** Dataset D,convergence threshold e

2:Random Initial:FEN weights ,reward weights w, Q network weights,

3:计算mu-e

4: for i = 0 : n do

5: Evaluate µθπ(i) using FEN

6: optimize w(i) by solving max-margin QP

**Algorithm 3** BCQ

1:**Input:** Dataset D,R参数

2:Random Initial: Q network weights,

3:

4: for i = 0 : n do

5: Evaluate µθπ(i) using FEN

6: optimize w(i) by solving max-margin QP

可分为三个部分：

1. 根据当前策略，使用DSFN估计特征期望；
2. 求解二次优化QP或者使用投影法优化奖励函数参数；
3. 用BCQ进行策略评估以及策略改进得到

下面是这三个部分的细节：

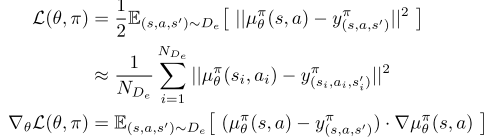
1. DSFN估计特征期望

Similar to (Lee et al., 2019)

我们想要学习一个神经网络来近似特征期望，使得，实际上是动作价值函数Q的一个分量,由（1）（2）两式可得：，因此优化Q网络的所有方法适用于特征期望网络的训练，例如MC方法或时域差分误差TD方法。我们使用了DQN【15】中同样的设定，使用两个网络，其中一个目标网络用来求目标值，给定π，φ，贝尔曼目标值如下：



其中a’是由当前策略Π产生的，并随着策略的改变而改变。我们使用L2误差训练网络：

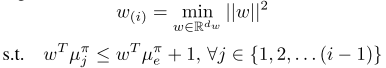


在离线设定下，我们只能使用交叉验证来判断学习是否收敛，当损失小于设定的固定阈值δ时训练结束。

1. 计算奖励函数参数

这部分参考自(Abbeel & Ng，2004)。

通过二次规划QP求解最大间隔问题：



为了简化计算，我们采用了投影法求解：



这部分计算在和上的正交投影。



表示专家特征期望和学习到的特征期望的距离，通常整个算法的结束条件为减小到某一阈值。

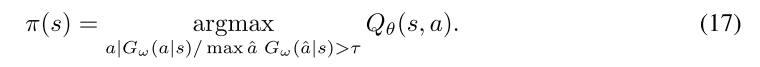
1. 用BRL改进策略

这部分参考自(Fujimoto et al, 2019)

Batch　RL又称作offlineＲＬ，它的目标是如何在不与环境交互的情况下最大化累积奖励，这类问题中的奖励一般是已知的。我们在２中得到奖励后，使用BRL方法优化当前的策略。

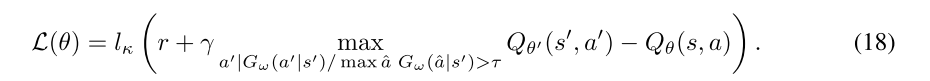
该方法解决的主要问题是在离线设置下产生的外推误差（分布漂移）。外推误差即，针对专家演示集中未出现的状态动作对，Q价值网络产生严重的错误估计。BCQ的解决办法是在最大化奖励的同时，对选择的动作进行限制。

我们使用的限制参考了【BCQ】，计算每个离散动作的概率，在概率高于设定阈值中的动作集中选择Q值最大的。

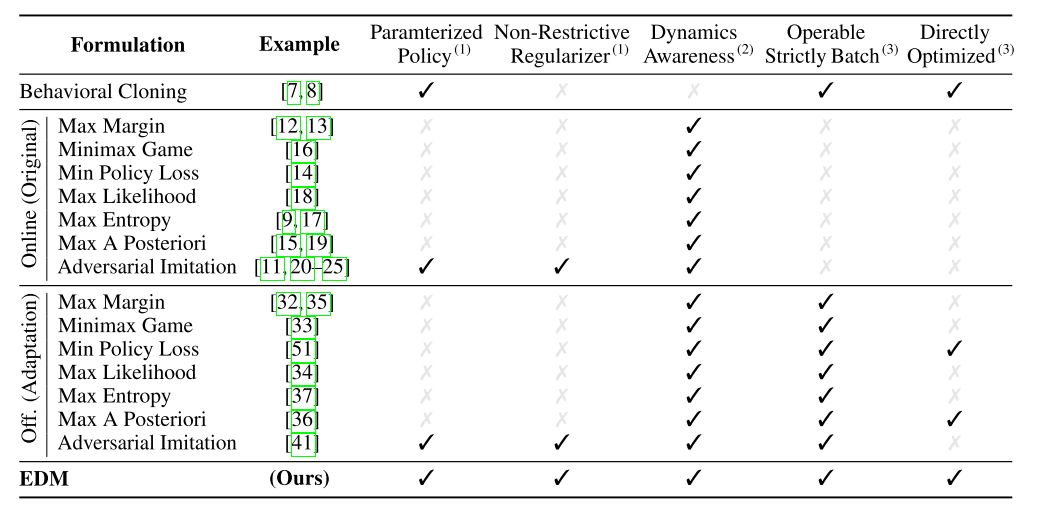


其中Gｗ该成ｐａｉｂ为，用BC方法训练这个ｐａｉｂ。由于动作离散，使用交叉熵损失训练

该算法也是在DQN的基础上进行的修改，使用TDerror指导训练，因此Q网络的训练损失如下：



其中ｌｋ是Huberｌｏｓｓ。当ｔａｏ＝０时该算法为Q学习，当tao=1时算法为完全的BC学习。



**实验**

为了验证我们方法的有效性，我们在gym环境上进行了实验。我们比较了我们的方法和DSFN，LSTD和BC

To evaluate the effectiveness of Batch-Apprenticeship learning in a high-dimensional setting, we focus on MuJoCo environments in OpenAI gym (Todorov et al., 2012; Brockman et al., 2016). We compare our method with DSFN，behavioral cloning method (BC).

**Training Details** For BCQ model, we first trained a TRIL network for warm start. We used a 70-30 training- validation split and following [Duan et al., 2016], included a Gaussian output layer that learns the means and standard deviations for the transition prediction — necessary to learn the uncertainty in our highly stochastic clinical domain exper- iment (Section 7). Further training details in terms of TRIL, DSFN model architecture and hyperparameters are provided in the appendix (Table 4). The IRL update was computed with the max-margin algorithm [Abbeel and Ng, 2004](Algorithm 1).

Exact implementation and experimental details are provided in the Supplementary Material.

Conclusion

在未来，要讨论非线性函数的情况。

如何将离线强化学习这一大类算法与IRL结合起来也相当有前景