Chapter 11-3 Deadly Triad

Created	@December 26, 2023 1:49 PM
Tags	

Function approximation

Bootstrapping

Off-policy method

上述三種若同時包含在訓練中，那這個不穩地度跟發散會發生，我們稱之為Deadly Triad，這個危機並不包含control或是GPI，是說control跟GPI分析也比較複雜。這個不穩定是在描述prediction的範疇。然後這個並非起因於環境的不確定性，因為這個也會在planning method下發生，像是已知環境下的動態規劃。

這三個只有兩個出線的話，那這個不穩定性就可以被避免，所以我們可以看看哪一個是比較能去放棄的。

首先我們很清楚，function approximation太重要了，它能夠讓擴展我們解決問題的範圍，我們至少是需要包含許多feature 跟 parameter 的線性 approximation，state aggregation或是nonparametic method對我們來說都太貴或太差了。LSTD處理對於大問題也顯然太貴。

再犧牲算力跟資料的效率使得在沒有bootstrapping的情況是有可能的，犧牲最多的是計算力。MC method 需要能儲存從一開始到目前發生的事的記憶能力(記憶體)直到最後return。這個花費雖然在 serial von Neumann 架構的電腦下沒什麼大問題，但是在其他種是會出問題的。在第十二章，我們結合bootstrap跟eligibility trace，在哪裡生成的資料我們直接就可以拿來用，而且用完就可以丟了。bootstrap 所省下來的溝通及記憶體是極大的。

犧牲的資料效率也是極大的，我們可以在第7張籍第9章反覆地看到某些程度的bootstrapping學的比MC還快的多，第十章的Mountain car control亦然。Bootstrapping通常可以加速學習過程，並在利用狀態特性(state property)的優勢，也可以說是熟識狀態的特點。不過在另一方面，bootstrapping也會在state的表示不全的情況下阻礙學習導致較差的結果(2016 學習Tetris似乎有這樣的問題)，而在state表示不全也會導致偏差(bias)，這是因為漸進的預測有較差的上下限導致，能夠bootstrap是很珍貴的。有時候我們會選擇較長的 n-step bootstrap(或極大的bootstrap

parameter，或是 $\lambda \approx 1$詳情請看第12章)，不過大部分的情況下，bootstrap可以大幅增加效率。所以這個能夠bootstrap需要被保留下來以備不時之需。

最後一個 off-policy method，我們可以犧牲掉它嗎？理論上on-policy method就已經足夠，在任何model free的情況下，我們可以使用的Sarsa取代掉Q-learning。Off-policy method 可以解除目標從target policy的束縛，這個可以被視為一個優勢但是並非必須。不過off-policy在一些情況是必須的像是anticipate use case，但是並沒有收錄在這本書中。