

DarkScope从这里开始

a learner,like Machine Learning.

目录视图

摘要视图

RSS 订阅

个人资料



Dark_Scope

访问: 600457次

积分: 5086

等级:

BLOG 5

排名: 第3473名

原创: 77篇

转载: 2篇

译文: 0篇

评论: 630条

个人介绍

DarkScope, 喜欢机器学习和一些ACM算法//学习ing//求交流, 求指教! =新浪微博 我是Darkscope

文章搜索

博客专栏



机器学习从原理到实践
文章: 5篇
阅读: 115672

文章分类

C++ (3)

ACM (7)

杂七杂八 (9)

ASM (5)

QT相关 (2)

机器学习 (39)

大学杂念集 (6)

C/C++ (1)

机器学习读书笔记 (4)

修身养性治情 (3)

web相关 (4)

nlp (6)

代码 (13)

【专家问答】韦玮: Python基础编程实战专题

【知识库】Swift资源大集合

【公告】博客新皮肤上线啦

CSDN福利第二期

机器学习 cs229学习笔记6(增强学习 reinforcement learning,MDP)

标签: action Action 增强学习 学习 机器学习 笔记

2012-12-05 09:26 11236人阅读 评论(6) 收藏 举报

分类: 机器学习 (38)

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

=====

上周生病再加上课余的一些琐事, 这边的进度就慢下来了, 本篇笔记基于 斯坦福大学公开课cs229 的 lecture16, lecture 17

=====

零: 一些认识

涉及到机器人的操控的时候, 很多事情可能并不是supervised和unsupervised learning能够解决的, 比如说andrew ng之前一直提到的自动控制直升飞机, 另一个例子就是下棋, 有可能很久之前的一步棋就埋下了后面失败的伏笔, 而机器很难去判断一步棋的好坏。这就是增强学习需要解决的问题。

注:这里的Value价值即是很多书上写的Q值, 貌似也有点差别, 在于Q可能是Q(s,a)的, 是给定状态和一个动作之后的V值, 但差异不大。

一: 马尔科夫决策过程 (Markov decision processes)

马尔科夫决策是一个五元组, $(S, A, \{P_{sa}\}, \gamma, R)$, 用一个机器人走地图的例子来说明它们各自的作用

S: 状态集: 就是所有可能出现的状态, 在机器人走地图的例子中就是所有机器人可能出现的位置

A: action, 也就是所有可能的行动。机器人走地图的例子假设机器人只能朝四个方向走, 那么A就是{N, S, E, W}表示四个方向

P: 就是机器人在S状态时采取a行动的概率

γ: 叫做discount factor, 是一个0到1之间的数, 这个数决定了动作先后对于结果的影响度, 在棋盘上的例子来说就是影响了这一步

棋对于最结果的影响有多大可能说起来比较模糊, 通过后面的说明可能会讲得比较清楚。

R: 是一个reward function, 也就是可能是一个: $S \times A \mapsto \mathbb{R}$, 也可能是 $R: S \mapsto \mathbb{R}$, 对应来说就是地图上的权值

=====

有了这样一个决策过程, 那么机器人在地图上活动的过程也可以表现为如下的形式:

趣写算法系列 (2)

文章存档

2015年12月 (1)

2015年07月 (1)

2015年03月 (1)

2015年02月 (1)

2014年05月 (2)

展开

阅读排行

AdaBoost--从原理到实现

(71910)

【面向代码】学习 Deep

(51282)

【面向代码】学习 Deep

(47767)

GBDT(Gradient Boostin

(39560)

RNN以及LSTM的介绍和

(39154)

【面向代码】学习 Deep

(35533)

趣写算法系列之--匈牙利

(30623)

从item-base到svd再到rt

(21181)

新浪微博小爬虫

(20446)

【面向代码】学习 Deep

(20025)

评论排行

【面向代码】学习 Deep

(101)

【面向代码】学习 Deep

(85)

趣写算法系列之--匈牙利

(79)

【面向代码】学习 Deep

(50)

新浪微博小爬虫

(44)

从item-base到svd再到rt

(41)

AdaBoost--从原理到实现

(34)

UFLDL练习(Sparse Aut

(30)

RNN以及LSTM的介绍和

(26)

GBDT(Gradient Boostin

(21)

推荐文章

*Android官方开发文档Training系列课程中文版：网络操作之XML解析

*Delta - 轻量级JavaWeb框架使用文档

*Nginx正向代理、负载均衡等功能实现配置

* 浅析ZeroMQ工作原理及其特点

*android源码解析（十九）-->Dialog加载绘制流程

*Spring Boot 实践折腾记（三）：三板斧，Spring Boot下使用Mybatis

最新评论

趣写算法系列之--匈牙利算法qq_28300305: 为啥不是乌索普没妹子 山治表示很受伤

从item-base到svd再到rbm，多yc695653190: 博主你好，看了你的代码获益匪浅，对于RBM for CF请问你手上有C++版本的

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \dots$$

也就是从初始位置开始，选择一个action到达另一个状态，直到到达终状态，因此我们这样来定义这个过程的价值：

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots$$

可以看出越早的决定对价值影响越大，其后则依次因为γ而衰减

其实可以看出，给出一个MDP之后，因为各个元都是定值，所以存在一个最优的策略(policy)，策略即是对于每个状态给出一个action，最优

策略就是在这样的策略下从任意一个初始状态能够以最大的价值到达终状态。策略用π表示。用

$$V^\pi(s) = E [R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots | s_0 = s, \pi].$$

表示在策略π下以s为初始状态所能取得的价值，而通过Bellman equation，上式又等于：

$$V^\pi(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^\pi(s').$$

注意这是一个递归的过程，在知道s的价值函数之前必去知道所有的s'的价值函数。（价值函数指的是Vπ()）

而我们定义最优的策略为π*，最优的价值函数为V*，可以发现这两个东西互为因果，都能互相转化。

二.价值迭代和策略迭代(Value iteration & policy iteration)

//////////价值迭代VI: //////////

1. For each state s, initialize $V(s) := 0$.

2. Repeat until convergence {

For every state, update $V(s) := R(s) + \max_{a \in A} \gamma \sum_{s'} P_{sa}(s') V(s')$.

}

这个过程其实比较简单，因为我们知道R的值，所以通过不断更新V，最后V就是converge到V*，再通过V*就可以得到最优策略π*，通

过V*就可以得到最优策略π*其实就是看所有action中哪个action最后的value值最大即可，此处是通过bellman equation，可以通过解bellman equation得到

所有的V的值，这里有一个动归的方法，注意马尔科夫决策过程中的P其实是指客观存在的概率，比如机器人转弯可能没法精确到一个方向，而不是指在s状态

机器人选择a操作 的概率，刚才没说清楚

在此说明，也就是说：

$$P_{sa}(s') = \frac{\text{\#times took we action } a \text{ in state } s \text{ and got to } s'}{\text{\#times we took action } a \text{ in state } s}$$

是一个客观的统计量。

//////////策略迭代PI//////////

代码么，我找不到。有...

趣写算法系列之-匈牙利算法
s_word_s:@KEYboarderQQ:是啊，通俗易懂

【面向代码】学习 Deep Learnir
高手GKJ: 对连续变量做回归分析，是不是应该将连续变量离散化？毕竟在精度许可范围内是可以的。这样就可以用分类的思...

UFLDL练习(Sparse Autoencode
qq_35020912:@markvq:哦我也看了一下博主在github上的源码，你指的不用标准化是不是不用对原始数据进行归...

UFLDL练习(Sparse Autoencode
qq_35020912:@markvq:Hi, 请问一下‘不用标准化’是什么意思？我写的sparseAutoencoderC...

KNN(k-nearest neighbor algorit
wordless_katherine: 请问博主，在数据挖掘领域里面，对于K-means、Adaboost、EM还有KNN这几种算法，那个相...

【面向代码】学习 Deep Learnir
qq_24237925: 博主，我看到MNIST是28*28的矩阵，按他的代码来应该是先卷积、池化、卷积、池化这样来的，所以第...

【面向代码】学习 Deep Learnir
xjxjxj: 博主有了解convolutional auto-encoder吗，请问代码中的bound是做什么用的...

从item-base到svd再到rbm，多
: 你看下面para怎么用的，para，para 什么的，这就是一个python的字典，里面放了这些东西...

1. Initialize π randomly.

2. Repeat until convergence {
(a) Let $V := V^\pi$.
(b) For each state s , let $\pi(s) := \arg \max_{a \in A} \sum_{s'} P_{sa}(s') V(s')$.
}
- 这次就是通过每次最优 π 来使 π converge到 π^* ， V 到 V^* 。但因为每次都要计算 π 的value值，所以这种算法并不常用
- 这两个算法的区别就是过程的区别，但我感觉本质上差别不大。(andrew说有不一样，至少看起来不一样.....这个待查)

三.连续状态的MDP

之前我们的状态都是离散的，如果状态是连续的，下面将用一个例子来予以说明，这个例子就是inverted pendulum问题

也就是一个铁轨小车上有一个长杆，要用计算机来让它保持平衡(其实就是我们平时玩杆子，放在手上让它一直保持竖直状态)

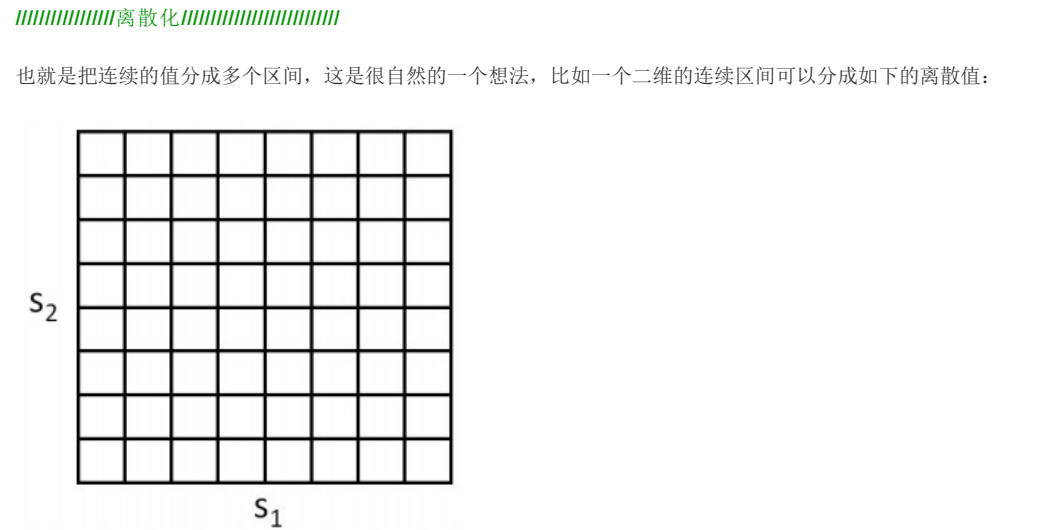
这个问题需要的状态有：都是real的值

x(在铁轨上的位置)

theta(杆的角度)

x'(铁轨上的速度)

thata'(角速度)



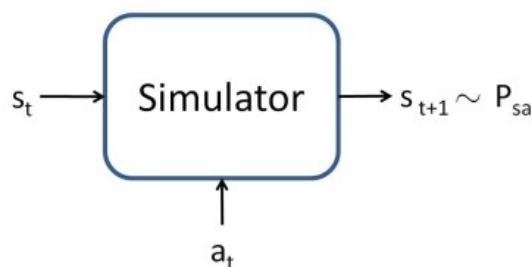
但是这样做的效果并不好，因为用一个离散的去表示连续空间毕竟是有限的离散值。

离散值不好的另一个原因是因为curse of dimension(维度诅咒)，因为连续值离散值后会有多个离散值，这样如果维度很大就会造成有非常多状态

从而使需要更多计算，这是随着dimension以指数增长的

////////////////simulator方法////////////////

也就是说假设我们有一个simulator，输入一个状态s和一个操作a可以输出下一个状态，并且下一个状态是服从MDP中的概率 P_{sa} 的分布，即：



这样我们就把状态变成连续的了，但是如何得到这样一个simulator呢？

①：根据客观事实

比如说上面的inverted pendulum问题，action就是作用在小车上的水平力，根据物理上的知识，完全可以解出这个加速度对状态的影响

也就是算出该力对于小车的水平加速度和杆的角加速度，再去一个比较小的时间间隔，就可以得到S(t+1)了

②：学习一个simulator

这个部分，首先你可以自己尝试控制小车，得到一系列的数据，假设力是线性的或者非线性的，将S(t+1)看作关于S(t)和a(t)的一个函数

得到这些数据之后，你可以通过一个supervised learning来得到这个函数，其实就是得到了simulator了。

比如我们假设这是一个线性的函数：

$$s_{t+1} = As_t + Ba_t,$$

在inverted pendulum问题中，A就是一个4*4的矩阵，B就是一个4维向量，再加上一点噪音，就变成了：其中噪音服从 $\epsilon_t \sim \mathcal{N}(0, \Sigma)$ 。

$$s_{t+1} = As_t + Ba_t + \epsilon_t,$$

我们的任务就是要学习到A和B

(这里只是假设线性的，更具体的，如果我们假设是非线性的，比如说加一个feature是速度和角速度的乘积，或者平方，或者其他，上式还可以写作：)

$$s_{t+1} = A\phi_s(s_t) + B\phi_a(a_t)$$

这样就是非线性的了，我们的任务就是得到A和B，用一个supervised learning分别拟合每个参数就可以了

四.连续状态中得Value(Q)函数

这里介绍了一个fitted value(Q) iteration的算法

在之前我们的value iteration算法中，我们有：

$$V(s) := R(s) + \gamma \max_a \int_{s'} P_{sa}(s') V(s') ds' \quad (6)$$

$$= R(s) + \gamma \max_a E_{s' \sim P_{sa}} [V(s')] \quad (7)$$

这里使用了期望的定义而转化。fitted value(Q) iteration算法的主要思想就是用一个参数去逼近右边的这个式子

也就是说：令

$$V(s) = \theta^T \phi(s).$$

其中 ϕ 是一些基于s的参数，我们需要去得到系数 θ 的值，先给出算法步骤再一步步解释吧：

```

1. Randomly sample  $m$  states  $s^{(1)}, s^{(2)}, \dots, s^{(m)} \in S$ .
2. Initialize  $\theta := 0$ .
3. Repeat {
    For  $i = 1, \dots, m$  {
        For each action  $a \in A$  {
            Sample  $s'_1, \dots, s'_k \sim P_{s^{(i)}a}$  (using a model of the MDP).
            Set  $q(a) = \frac{1}{k} \sum_{j=1}^k R(s^{(i)}) + \gamma V(s'_j)$ 
            // Hence,  $q(a)$  is an estimate of  $R(s^{(i)}) + \gamma E_{s' \sim P_{s^{(i)}a}}[V(s')]$ .
        }
        Set  $y^{(i)} = \max_a q(a)$ .
        // Hence,  $y^{(i)}$  is an estimate of  $R(s^{(i)}) + \gamma \max_a E_{s' \sim P_{s^{(i)}a}}[V(s')]$ .
    }
    // In the original value iteration algorithm (over discrete states)
    // we updated the value function according to  $V(s^{(i)}) := y^{(i)}$ .
    // In this algorithm, we want  $V(s^{(i)}) \approx y^{(i)}$ , which we'll achieve
    // using supervised learning (linear regression).
    Set  $\theta := \arg \min_{\theta} \frac{1}{2} \sum_{i=1}^m (\theta^T \phi(s^{(i)}) - y^{(i)})^2$ 
}

```

算法步骤其实很简单，最主要的其实就是他的思想：

在对于action的那个循环里，我们尝试得到这个action所对应的 $R(s^{(i)}) + \gamma E_{s' \sim P_{s^{(i)}a}}[V(s')]$ ，记作 $q(a)$

这里的 $q(a)$ 都是对应第 i 个样例的情况

然后 $i=1 \dots m$ 的那个循环是得到是最优的action对应的Value值，记作 $y(i)$ ，然后用 $y(i)$ 拿去做supervised learning，大概就是这样思路

至于reward函数就比较简单了，比如说在inverted pendulum问题中，杆子比较直立就是给高reward，这个可以很直观地从状态得到衡量奖励的方法

关闭



我们就可以去算我们的policy了：

$$\arg \max_a E_{s' \sim P_{sa}}[V(s')]$$

模型

去其实是针对一个非确定性的模型，即一个动作可能到达多个状态，有 P 在影响到达哪个状态

中，其实是一个简化的问题，得到的样例简化了，计算也简化了

状态和一个动作，只能到达另一个状态，而不是多个，特例就不细讲了

上一篇 突如其来写个小总结

下一篇 大学杂念集 算法之道

我的同类文章

机器学习（38）

• RNN以及LSTM的介绍和公... 2015-07-25

阅读 39119

• GBDT(Gradient Boosting ... 2014-05-03

阅读 39539

• CNN(Convolutional Neural... 2013-12-03

阅读 11008

• KNN(k-nearest neighbor a... 2013-11-13

阅读 6209

• 理解机器学习算法的一点心得 2014-05-10

阅读 6198

• 从item-base到svd再到rbm... 2013-12-14

阅读 21171


• SVM--从“原理”到实现 2013-11-23

阅读 10575

• AdaBoost--从原理到实现 2013-11-03

阅读 71889

参考知识库



算法与数据结构知识库

1038 关注 | 2080 收录

猜你在找

- Spark 1.x大数据平台

深入浅出Unity3D——第一篇

Spark零基础入门（4）：Scala 类和对象

实战进阶学习Unity3d游戏开发

微信公众平台开发入门
- Stanford 机器学习笔记 Week6 Machine Learning

Stanford 机器学习笔记 Week6 Advice for Applying Machine Learning Week6 学习笔记之机器学习系统设计

斯坦福大学公开课 机器学习课程Andrew Ng15无监督学

CS229 lecture16强化学习-马尔科夫决策过程MDP




查看评论

4楼 [chengdu2013](#) 2016-04-24 16:03发表




问lz一个问题，在fitted value iteration 中，首先不知道value函数，在算法中又要计算q(a)，q(a)计算需要用V(s')，请问这个矛盾怎么解决的？

Re: [Dark_Scope](#) 2016-04-25 10:22发表




回复chengdu2013：没记错的话这本来就是一个迭代的过程，一开始两个值可能都不对，慢慢迭代到最后计算出来的

3楼 [Romanticone](#) 2015-03-02 20:38发表




请问后面那个例子是哪里的啊？

2楼 [ssiaw12345](#) 2014-01-22 11:54发表




lz有看过这门课的公开课视频吧，在这篇讲义之后的内容就没讲义了，比如讲LQR，斯坦福的网上也没，lz有什么学习资料吗？

Re: [Dark_Scope](#) 2014-01-22 12:24发表



回复ssiaw12345：增强学习确实用得很少，所以后来都没怎么了解了，你可以在谷歌学术里面搜一下相关的paper看

1楼 [TNTDoctor](#) 2013-02-21 05:20发表



写得不错，lz挺用心得

您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题
- Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
- VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
- BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
- Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
- FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
- Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
- Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 