1 LECTURE OUTLINE

1 00:00:16,960 –> 00:00:22,450 okay so let me welcome you to this fifth
2 00:00:22,450 –> 00:00:26,020 out of six lectures
欢迎回来上第五次课
3 00:00:26,020 –> 00:00:28,720 on approximate dynamic programming we spent the first
4 00:00:28,720 –> 00:00:31,300 three lectures on exact dynamic
5 00:00:31,300 –> 00:00:34,570 programming and also an overview of the
6 00:00:34,570 –> 00:00:37,030 general issues in approximate dynamic programming
关于近似动态规划，我们用前三次课讲了精确动态规划并且对近似动态规划的一般性主题进行了综述
7 00:00:37,030 –> 00:00:40,090 in this week starting with
8 00:00:40,090 –> 00:00:42,699 the previous lecture we're going to
9 00:00:42,699 –> 00:00:46,420 focus selectively somewhat deeper on
10 00:00:46,420 –> 00:00:48,760 various aspects of approximate dynamic programming
这周我们开始有选择性地讲一下之前概述中近似动态规划的内容
11 00:00:48,760 –> 00:00:51,430 now we talked about
12 00:00:51,430 –> 00:00:54,160 approximate policy duration based on
13 00:00:54,160 –> 00:00:56,520 projected bellman equations last time
在这里我们要最后一次讲基于投影 bellman 方程的近似策略迭代
14 00:00:56,520 –> 00:01:00,880 however policy duration involves two
15 00:01:00,880 –> 00:01:02,530 aspects there are two parts to each iteration
策略迭代每次迭代都包括两部分
16 00:01:02,530 –> 00:01:05,590 one is evaluating the current
17 00:01:05,590 –> 00:01:07,930 policy and that's what we talked about
18 00:01:07,930 –> 00:01:10,660 last time approximate evaluation of
19 00:01:10,660 –> 00:01:12,340 policies based on projected bellman equations
第一个部分是评价现在的策略，也就是之前我们讲的基于投影 bellman 方程的近似策略评价
20 00:01:12,340 –> 00:01:14,860 and then there's a second part
21 00:01:14,860 –> 00:01:16,570 of iteration that has to do with the
22 00:01:16,570 –> 00:01:19,960 policy improvement process
迭代的第二部分是策略改进
23 00:01:19,960 –> 00:01:23,950 and this involves issues that are tricky and important
这部分包括的内容比较有技巧也很重要
24 00:01:23,950 –> 00:01:27,010 and we're going to discuss those first
我要先讲这个内容 (策略改进)
25 00:01:27,010 –> 00:01:31,000 in particular we talked
26 00:01:31,000 –> 00:01:33,250 about the issue of exploration which we
27 00:01:33,250 –> 00:01:35,650 touched upon last time how do you
28 00:01:35,650 –> 00:01:38,049 introduce exploration into the
29 00:01:38,049 –> 00:01:42,280 approximate policy iteration process
特别地，我要讲讲探索，我们上次讲的内容，如何在近似策略迭代中加入这个东西
30 00:01:42,280 –> 00:01:43,510 and then talk about the issue of oscillations
然后我们要讲讲震荡
31 00:01:43,510 –> 00:01:45,700 approximate policy duration
32 00:01:45,700 –> 00:01:47,409 is not guaranteed to converge to a single policy
近似策略迭代没法保证收敛到一个策略
33 00:01:47,409 –> 00:01:50,740 typically or very often it
34 00:01:50,740 –> 00:01:54,729 will just generate a cycle of policies
35 00:01:54,729 –> 00:01:57,400 many policies perhaps
很多时候只能保证收敛到一个策略的循环，即一个策略的集合
36 00:01:57,400 –> 00:01:59,680 and we want to look at the mechanisms by which these
37 00:01:59,680 –> 00:02:02,470 oscillations occur and see what they can do to us
我们想要看看这个现象的产生机制，震荡什么时候会产生和他们会对我们造成什么影响
38 00:02:02,470 –> 00:02:07,390 after we do that we are going
39 00:02:07,390 –> 00:02:10,119 to discuss an alternative to the project
40 00:02:10,119 –> 00:02:12,730 development equations for approximate
41 00:02:12,730 –> 00:02:14,890 policy evaluation and also for

42 00:02:14,890 –> 00:02:18,310 approximate value duration
43 00:02:18,310 –> 00:02:19,360 which is aggregation
讲过震荡之后，我们会讨论投影方程组进行近似策略评估与近似值迭代的替代方案-聚合
44 00:02:19,360 –> 00:02:21,579 this is a very simple approach it dates
45 00:02:21,579 –> 00:02:22,960 to way back
46 00:02:22,960 –> 00:02:28,120 years and and it is related to the
47 00:02:28,120 –> 00:02:29,770 projected equation approached and we
48 00:02:29,770 –> 00:02:32,370 will discuss this this disconnection
49 00:02:32,370 –> 00:02:35,320 after we describe it give some examples
50 00:02:35,320 –> 00:02:38,380 and we say a few things about how we can
51 00:02:38,380 –> 00:02:43,720 implement them by simulation
这是一个非常简单的方法，可以追溯到很多年以前，并且与投影方程方法相关，讲过聚合之后我会给几个例子然后说一说该怎么用仿真方法来实现聚合

2 DISCOUNTED MDP

52 00:02:43,720 –> 00:02:47,070 okay so we continue to look at finite state
53 00:02:47,070 –> 00:02:49,750 discounted infinite horizon Markov
54 00:02:49,750 –> 00:02:52,540 decision problems
我们接着看一个有限状态折扣无限期马尔科夫决策问题
55 00:02:52,540 –> 00:02:53,890 we have transition probabilities going from different
56 00:02:53,890 –> 00:02:57,430 states which depend on the control we
57 00:02:57,430 –> 00:02:59,380 want to pick controls so as to minimize
58 00:02:59,380 –> 00:03:03,610 the cost of policies and for a given
59 00:03:03,610 –> 00:03:05,410 policy a sequence of functions from state to control
我们有一个状态间跳转的依赖于控制的状态转移概率，我们想要根据状态选择合适的控制来最小化给定策略的成本
60 00:03:05,410 –> 00:03:08,050 we have a cost per
61 00:03:08,050 –> 00:03:13,060 stage and and a long-term cost that's
62 00:03:13,060 –> 00:03:16,870 associated with it for any given
63 00:03:16,870 –> 00:03:22,090 starting state i by taking the
64 00:03:22,090 –> 00:03:23,530 limit of the expected value of the
65 00:03:23,530 –> 00:03:25,900 series gives you the cost of the policy
66 00:03:25,900 –> 00:03:29,620 at state I and because alpha the
67 00:03:29,620 –> 00:03:32,500 discount factor is between is less than
68 00:03:32,500 –> 00:03:34,960 strictly less than one we are guaranteed
69 00:03:34,960 –> 00:03:37,630 that this is well defined and we want to
70 00:03:37,630 –> 00:03:42,870 find pi such that this is minimized
71 00:03:42,870 –> 00:03:45,700 simultaneously for all states
我有一个每阶段成本和与 α 相关的长期成本，对于一个给定的初始状态 i_0，N 趋于 0 的极限的期望值就是这个成本在状态 i_0 时的成本。由于折扣系数 α 严格小于 1，我们可以保证这个成本不发散，我们想要做的就是找到一个策略 π 让这个策略对于所有状态都有最小成本
72 00:03:45,700 –> 00:03:47,260 now we have discussed the special significance
73 00:03:47,260 –> 00:03:49,690 of stationary policies
现在我们要讨论一个特殊情况，平稳策略
74 00:03:49,690 –> 00:03:51,310 stationary policies are the ones where the news do
75 00:03:51,310 –> 00:03:53,290 not change from one stage to the next
平稳策略就是策略 μ 在每一个阶段都不发生变化
76 00:03:53,290 –> 00:03:55,480 we have given various optimality
77 00:03:55,480 –> 00:03:59,350 conditions and all that
这个问题有很多最优性条件
78 00:03:59,350 –> 00:04:00,550 and let me remind you again of the shorthand
79 00:04:00,550 –> 00:04:02,710 notation for the dynamic programming
80 00:04:02,710 –> 00:04:05,500 mapping and also for the evaluation
81 00:04:05,500 –> 00:04:07,810 mapping corresponding to a policy
82 00:04:07,810 –> 00:04:11,380 whereby given any J we generate t news
83 00:04:11,380 –> 00:04:12,880 of J by means of this linear equation
84 00:04:12,880 –> 00:04:16,839 or this nonlinear equation

我再提一次动态规划映射速记符号与某策略的评价映射速记符，给定一个 J，我们通过这个线性方程 (策略评估，最底下的方程) 或者这个非线性方程 (倒数第二个方程) 生成映射 $T_\mu J$

85 00:04:16,839 –> 00:04:21,839 we solve a fix point equation involving t in in in

86 00:04:21,839 –> 00:04:25,500 to obtain the optimal cost

求解 T 的不动点来找最优成本

87 00:04:25,500 –> 00:04:28,770 and we solved the Optima be the bellman equation

88 00:04:28,770 –> 00:04:30,300 involving tinu

89 00:04:30,300 –> 00:04:32,160 to obtain an evaluation of the policy mu

求解这个 T_μ 的 bellman 方程的不动点对策略 μ 进行评价

3 APPROXIMATE PI

90 00:04:32,160 –> 00:04:37,949 okay so approximate policy

91 00:04:37,949 –> 00:04:40,260 Direction it's the same thing as exact

92 00:04:40,260 –> 00:04:42,510 policy direction except that we

93 00:04:42,510 –> 00:04:45,690 evaluation instead of finding J nu we

94 00:04:45,690 –> 00:04:48,090 find some approximation to that

95 00:04:48,090 –> 00:04:50,490 involving an approximation architecture

96 00:04:50,490 –> 00:04:53,370 with r being the vector of parameters

97 00:04:53,370 –> 00:04:55,760 in the approximation architecture

98 00:04:55,760 –> 00:04:58,889 evaluate approximately new generate and

99 00:04:58,889 –> 00:05:00,870 improve policy by the prop policy

100 00:05:00,870 –> 00:05:03,930 improvement equation and look around in the sweat

这就是策略迭代，除了评价策略计算 J_μ 不一样，剩下的内容与精确策略迭代是一样的。我们找到一种近似结构来近似 J_μ，r 是近似结构的参数向量，近似评价策略 μ，然后用这个评价与策略改进方程进行策略改进

101 00:05:03,930 –> 00:05:08,850 now we focused on linear cost

102 00:05:08,850 –> 00:05:12,320 function approximation involving a

103 00:05:12,320 –> 00:05:17,729 matrix phi ok dimension n the row

104 00:05:17,729 –> 00:05:20,850 dimension and column dimension much

105 00:05:20,850 –> 00:05:23,850 smaller s the columns of this matrix

106 00:05:23,850 –> 00:05:28,310 are our basis functions for a subspace

107 00:05:28,310 –> 00:05:33,780 defined by by vectors like this

现在我们关注这个线性成本近似函数包括一个矩阵 ϕ，ϕ 的行维度是 n，列维度 s 比行维度小得多，矩阵的每一列都是子空间中的基向量，被定义成 $\phi(i)'$

108 00:05:33,780 –> 00:05:36,090 so we are looking to find an approximation of

109 00:05:36,090 –> 00:05:39,750 J mu within the sub space of functions of this form

所以我们想要再子空间中找到一个 Φr 形式的函数对 J_μ 进行近似

110 00:05:39,750 –> 00:05:45,030 if we call the I-th of

111 00:05:45,030 –> 00:05:47,400 phi I prime which is a small vector like

112 00:05:47,400 –> 00:05:51,349 that the policy improvement process

113 00:05:51,349 –> 00:05:54,900 involves this equation so if I give you

114 00:05:54,900 –> 00:05:58,710 r a good r you can get a good mu ok

115 00:05:58,710 –> 00:06:01,740 if I give you a very good art then you

116 00:06:01,740 –> 00:06:04,410 can get a policy that's as good as you

117 00:06:04,410 –> 00:06:09,510 can get it with approximation

如果 $\phi(i)'$ 是一个很小的向量，那么策略改进过程包括这个表达式，如果我给你一个比较好的 r，你可以得到一个比较好的 μ，如果我给你一个特别好的 r，你可以得到和这个 r 一样好的策略，也就是说，近似越好，策略越好

118 00:06:09,510 –> 00:06:13,500 so the issue is how do I get a good R

所以现在的问题就是如何能得到一个比较好的 r

4 EVALUATION BY PROJECTED EQUATIONS

119 00:06:13,500 –> 00:06:17,460 and in the process we discuss we focus on the

120 00:06:17,460 –> 00:06:19,470 right on the evaluation part of the

121 00:06:19,470 –> 00:06:23,729 policy iteration procedure

这里我们主要讨论策略迭代的策略评价部分
122 00:06:23,729 –> 00:06:27,630 instead of finding a fixed point of T mu we find a
123 00:06:27,630 –> 00:06:30,719 fixed point of PI T mu we solve this
124 00:06:30,719 –> 00:06:33,940 equation which is the projected bellman
125 00:06:33,940 –> 00:06:36,340 equation involving a projection
126 00:06:36,340 –> 00:06:40,420 operation Pi which is Euclidean sum of
127 00:06:40,420 –> 00:06:43,570 squares but also weighted weighted by
128 00:06:43,570 –> 00:06:46,240 positive numbers the xi and we can
129 00:06:46,240 –> 00:06:48,340 normalize the size so that they form a
130 00:06:48,340 –> 00:06:50,890 probability distribution
为了取代求 T_μ 的不动点，我们找 ΠT_μ 的不动点，我们求解这个包括以 ξ 为权重的欧几里得范数投影算子 Π 的投影 bellman 方程 (ξ 的值都是正数，可以理解为概率分布)
131 00:06:50,890 –> 00:06:53,680 and in this context we mentioned that given the
132 00:06:53,680 –> 00:06:56,770 policy and assuming this policy gives
133 00:06:56,770 –> 00:06:59,410 you an ergodic Markov chain with a
134 00:06:59,410 –> 00:07:01,180 positive steady state probability
135 00:07:01,180 –> 00:07:04,090 distribution vector then the steady
136 00:07:04,090 –> 00:07:06,940 state distribution weighting the xi
137 00:07:06,940 –> 00:07:10,960 vector corresponding to quick which is
138 00:07:10,960 –> 00:07:12,760 the state the state the the vector of
139 00:07:12,760 –> 00:07:15,220 steady state probabilities
现在你有一个策略，假设这个策略能够以正的平稳状态概率分布遍历马尔科夫链，然后这个平稳状态概率分布作为 Ξ 的权重
140 00:07:15,220 –> 00:07:17,740 when you do projection with that then PI T nu has a
141 00:07:17,740 –> 00:07:21,100 unique solution and is pi T mu is a
142 00:07:21,100 –> 00:07:23,620 contraction has a unique solution and a
143 00:07:23,620 –> 00:07:26,320 lot of other nice things occur
当你对 Ξr 做投影时，映射 ΠT_μ 收缩并有唯一解，有很多好事会发生
144 00:07:26,320 –> 00:07:29,170 such as for example various algorithms become
145 00:07:29,170 –> 00:07:31,210 valid including approximate value iteration
比如很多算法在这个近似值迭代下变得有效
146 00:07:31,210 –> 00:07:37,240 and then the methods of LSPE
147 00:07:37,240 –> 00:07:43,650 LSTD and so on become possible
LSTD 和 LSPE 和其他算法
148 00:07:43,650 –> 00:07:45,610 and the implementation of the solution of this
149 00:07:45,610 –> 00:07:50,110 equation we know that the possibility of
150 00:07:50,110 –> 00:07:52,930 doing it by generating a single long
151 00:07:52,930 –> 00:07:55,960 trajectory using the current policy this
152 00:07:55,960 –> 00:07:59,140 implicitly generates weights that are
153 00:07:59,140 –> 00:08:02,380 the steady state distribution weights and
154 00:08:02,380 –> 00:08:08,130 make the PI T new mapping a contraction
在实现这个算法的时候，跟据当前策略与隐式平稳状态分布权重生成一个很长的轨迹能够保证 ΠT_μ 是一个压缩映射
155 00:08:08,130 –> 00:08:09,310 okay
156 00:08:09,310 –> 00:08:12,220 so that was the gist of last most of
157 00:08:12,220 –> 00:08:14,620 most of what we discussed in the last in the last time
所以上次我们讲的一个很重要的内容是
158 00:08:14,620 –> 00:08:18,640 how do you why the steady
159 00:08:18,640 –> 00:08:21,669 state distribution is important and how
160 00:08:21,669 –> 00:08:24,820 do you do a simulation based solution of
161 00:08:24,820 –> 00:08:28,360 this equation using a single long
162 00:08:28,360 –> 00:08:31,210 trajectory that involves the steady
163 00:08:31,210 –> 00:08:35,669 state probabilities in an average sense
为什么平稳状态分布这么重要，和如何使用仿真从平均意义上在平稳状态概率下获得一个很长的轨迹
164 00:08:35,899 –> 00:08:39,229 there were a number of issues that were
165 00:08:39,229 –> 00:08:44,068 related to this and we're going to focus
166 00:08:44,068 –> 00:08:46,740 on some of them in this lecture

有很多话题与这它相关，我们会在这次课程中关注其中的几个

167 00:08:46,740 –> 00:08:49,170 but let me also remind you that there's a

168 00:08:49,170 –> 00:08:52,279 multi-step option in projected equations

这是一个多步操作的投影方程

169 00:08:52,279 –> 00:08:55,560 whereby instead of solving this equation

170 00:08:55,560 –> 00:08:58,220 you solve an equation involving a

171 00:08:58,220 –> 00:09:03,569 weighted version of T mu

求解这个加权 T_μ 的方程而不是上面那个 Φr 的方程

172 00:09:03,569 –> 00:09:07,230 whereby T mu lambda is the sum of powers of T mu with

173 00:09:07,230 –> 00:09:09,509 lambda between 0 and 1

这个方程里 $T_\mu^{(\lambda)}$ 是 T_μ 的若干次幂的和，λ 大于 0 小于 1

174 00:09:09,509 –> 00:09:13,769 so it's Geometrically weighted sum of powers of

175 00:09:13,769 –> 00:09:19,319 T mu and this mapping has exactly the

176 00:09:19,319 –> 00:09:22,759 same fixed points as T mu okay and

177 00:09:22,759 –> 00:09:27,180 also is a contraction because TMU is a contraction

所以这是一个 T_μ 的几何加权次幂累加，这个映射能够获得和 T_μ 一样的不动点，而且也是收缩的，因为 T_μ 是收缩的

178 00:09:27,180 –> 00:09:32,120 however has lambda changes

179 00:09:32,120 –> 00:09:36,079 then the projection of this mapping

180 00:09:36,079 –> 00:09:38,370 becomes different of course depends on

181 00:09:38,370 –> 00:09:40,920 lambda its fixed point is different and

182 00:09:40,920 –> 00:09:43,259 also its modulus of contraction is

183 00:09:43,259 –> 00:09:45,720 different as lambda goes to always 1

184 00:09:45,720 –> 00:09:48,800 this becomes a perfect contraction and

185 00:09:48,800 –> 00:09:52,800 this allows you to change this pi from

186 00:09:52,800 –> 00:09:55,529 the steady state distribution and still

187 00:09:55,529 –> 00:09:57,930 have a contraction okay

在 λ 改变的时候，由于投影依赖于 λ，所以投影、不动点和收缩模量都会变化，当 λ 的值趋于 1 的时候，$T_\mu^{(\lambda)}$ 具有很强的收缩性，你通过改变平稳状态分布来改变 Π 的时候，它仍然具有收缩性

188 00:09:57,930 –> 00:09:59,100 so that will come into the process of exploration

189 00:09:59,100 –> 00:10:02,819 exploration becomes better behaved when

190 00:10:02,819 –> 00:10:04,380 you have a lambda greater than zero

191 00:10:04,380 –> 00:10:06,329 because you have better contraction properties

接下来我们会开始讲一点探索，当 λ 的值大于 0 的时候，探索会有更好的表现，因为这时映射具有更好的收缩性

192 00:10:06,329 –> 00:10:10,439 however another major aspect

193 00:10:10,439 –> 00:10:14,279 of the weighted bellman equation lambda

194 00:10:14,279 –> 00:10:18,779 weighted is that its solution depends on

195 00:10:18,779 –> 00:10:21,449 lambda and for lambda equals zero that

196 00:10:21,449 –> 00:10:22,949 case over there

197 00:10:22,949 –> 00:10:25,050 it's something that involves an error

198 00:10:25,050 –> 00:10:29,639 that you see here as lambda changes

199 00:10:29,639 –> 00:10:33,269 towards 1 then this error becomes

200 00:10:33,269 –> 00:10:36,720 smaller in for lambda exactly equal to 1

201 00:10:36,720 –> 00:10:38,610 you get the best possible approximation

202 00:10:38,610 –> 00:10:42,329 error just the direct projection onto

203 00:10:42,329 –> 00:10:43,430 the

204 00:10:43,430 –> 00:10:47,840 approximation subspace

另一个很重要的内容是，λ 权重的 bellman 方程的解依赖于 λ，当 λ 等于 0 时，就是这一页上面的那种情况，这之后被近似值与近似值之间有误差，当 λ 向 1 改变时，误差在变小，当 λ 等于 1 时，误差最小，这就是直接向近似子空间投影的结果

205 00:10:47,840 –> 00:10:50,090 however to solve this projected equation involves more noise

然而在求解这个投影方程 $(\Phi r = \Pi T_\mu^{(\lambda)}(\Phi r))$ 的时候会产生更多噪声

206 00:10:50,090 –> 00:10:53,510 because yes you make simulation

207 00:10:53,510 –> 00:10:55,970 based evaluations of this way that sums

208 00:10:55,970 –> 00:10:58,820 each one of those involves more noise

209 00:10:58,820 –> 00:11:00,980 because it's a mapping that relates far

210 00:11:00,980 –> 00:11:02,000 into the future
因为在使用仿真来进行估值的时候 (multistep option 那一项下面的公式)，累加的每一项都会带有噪声，因为他们都与未来的值有关
211 00:11:02,000 –> 00:11:06,380 so more noisy terms more noise here you
212 00:11:06,380 –> 00:11:08,870 need more samples to counteract the
213 00:11:08,870 –> 00:11:11,750 effects of this noise
所以累加项有噪声，累加结果 (等号左边的项) 会有更大的噪声，所以你需要进行更多次采样来抵消噪声的影响
214 00:11:11,750 –> 00:11:14,650 so that's the typical behavior the typical trade off
这就是典型的权衡
215 00:11:14,650 –> 00:11:18,530 great large lambda smaller approximation
216 00:11:18,530 –> 00:11:21,530 error but we need more samples to
217 00:11:21,530 –> 00:11:23,390 generate an accurate solution so
218 00:11:23,390 –> 00:11:26,420 so-called bias-variance tradeoff okay
为了得到尽量精确的解，λ 越大，近似误差越小，需要的样本就越多，这就是被叫做"bias-variance trade-off"(偏差-方差权衡)
219 00:11:26,420 –> 00:11:27,590 we're not going to come back to this
220 00:11:27,590 –> 00:11:30,970 other than the fact that as lambda
221 00:11:30,970 –> 00:11:34,940 approaches 1 this contraction property is improved
我不会再回来讲了，事实上，λ 的值接近 1，收缩性就会被增强
222 00:11:34,940 –> 00:11:43,310 ok so that's all we did last
223 00:11:43,310 –> 00:11:47,030 time the policy evaluation part and how
224 00:11:47,030 –> 00:11:47,780 to solve it
这就是我们刚刚讲的内容，策略评价和如何实现它

5 EXPLORATION

225 00:11:47,780 –> 00:11:50,780 ok now let's call let's talk about
226 00:11:50,780 –> 00:11:52,990 policy improvement
下面我们来谈一谈策略改进
227 00:11:52,990 –> 00:11:55,880 the first major issue is the issue of exploration
第一个话题是探索
228 00:11:55,880 –> 00:11:58,880 in order to evaluate a
229 00:11:58,880 –> 00:12:01,700 policy mu we need to generate core
230 00:12:01,700 –> 00:12:04,040 samples using that policy there's no way around that
为了评价一个策略，我们需要使用这个策略产生很多样本，
231 00:12:04,040 –> 00:12:07,880 however then if you use a
232 00:12:07,880 –> 00:12:10,790 single long trajectory then this long
233 00:12:10,790 –> 00:12:13,460 trajectory using the policy will tend to
234 00:12:13,460 –> 00:12:15,920 go through states that are preferred
235 00:12:15,920 –> 00:12:18,650 from this policy
如果你使用被评价策略生成长期轨迹，这个轨迹就会趋向于访问它喜欢的状态
236 00:12:18,650 –> 00:12:20,510 you may have for example a part of the big state space
237 00:12:20,510 –> 00:12:26,000 that that that that this policy
238 00:12:26,000 –> 00:12:29,210 naturally tends to and very infrequently
239 00:12:29,210 –> 00:12:34,010 goes off to other parts of the space
你可以举出很多例子，策略趋向于非常频繁地访问大状态空间中的一部分，而其他状态很少访问
240 00:12:34,010 –> 00:12:36,410 so the steady state distribution the xi
241 00:12:36,410 –> 00:12:40,040 of this other state is almost zero if
242 00:12:40,040 –> 00:12:42,080 the Markov chain is not even ergodic
243 00:12:42,080 –> 00:12:44,720 then some xi are going to be exactly zero
所以这些不访问的状态对应的平稳状态分布 xi 中的元素几乎是 0，如果马尔科夫链不是可遍历的，xi 的一部分元素就真的是 0 了
244 00:12:44,720 –> 00:12:48,590 so when you do a least squares fit
245 00:12:48,590 –> 00:12:51,350 or a projected equation fit with this
246 00:12:51,350 –> 00:12:52,310 weight
247 00:12:52,310 –> 00:12:54,860 the states that never are never or very

248 00:12:54,860 –> 00:12:58,010 seldom visited by this policy are going
249 00:12:58,010 –> 00:13:02,600 to be underrepresented and the error
250 00:13:02,600 –> 00:13:06,680 that you will get for those states from
251 00:13:06,680 –> 00:13:08,390 the projected equation is going to be
252 00:13:08,390 –> 00:13:09,320 very large
253 00:13:09,320 –> 00:13:11,720 however these states may be important
254 00:13:11,720 –> 00:13:13,490 perhaps not important for the current
255 00:13:13,490 –> 00:13:15,830 policy but important for other policies
256 00:13:15,830 –> 00:13:18,080 and you would like to include them in
257 00:13:18,080 –> 00:13:22,700 the approximation fairly and so there
258 00:13:22,700 –> 00:13:25,070 may be a serious problem the improved
259 00:13:25,070 –> 00:13:27,320 policy may be much different
260 00:13:27,320 –> 00:13:31,810 this is huge errors because because
261 00:13:31,810 –> 00:13:35,030 policies that mu regards as unimportant
262 00:13:35,030 –> 00:13:39,880 mu bar may make regard as very important

如果你在这个权重下使用最小二乘拟合或者投影方程拟合，从来都不被访问的状态对于这个策略来说就是没有代表性的状态，这些状态导致的投影方程拟合的误差会非常大，这些状态对当前策略不重要，但是对其他策略有可能很重要，所以你希望在做近似的时候同时包括对这些状态的近似，这就会导致对策略改进时会产生非常大的误差，因为对策略 μ 不重要的状态可能对策略 $\bar{\mu}$ 非常重要

263 00:13:39,880 –> 00:13:42,470 so that's the problem of exploration we discussed

这就是我们讨论的关于探索的问题

264 00:13:42,470 –> 00:13:47,530 it and it's a very serious
265 00:13:47,530 –> 00:13:49,520 particularly when the randomness
266 00:13:49,520 –> 00:13:51,680 embodied in the transition probabilities
267 00:13:51,680 –> 00:13:53,930 is relatively small if you have for
268 00:13:53,930 –> 00:13:56,210 example a deterministic system then
269 00:13:56,210 –> 00:13:58,220 there is no natural noise in the system
270 00:13:58,220 –> 00:14:00,440 that makes you wander around and explore
271 00:14:00,440 –> 00:14:02,360 the state expect the state the state
272 00:14:02,360 –> 00:14:06,620 space in order to deal with a problem we
273 00:14:06,620 –> 00:14:08,660 need to change the sampling mechanism
274 00:14:08,660 –> 00:14:11,270 and at the same time modify the
275 00:14:11,270 –> 00:14:15,350 simulation formulas

这是一个非常严重的问题，特别是状态转移的随机性非常小的时候，举个例子，一个没有噪声的确定性系统让你想要在状态附近采样，为了解决这个问题，我们需要一边改变采样机制一边修改仿真函数

276 00:14:15,350 –> 00:14:19,640 in other words we need to solve a different projected
277 00:14:19,640 –> 00:14:24,050 equation that involves projection by bar
278 00:14:24,050 –> 00:14:27,830 with respect to an exploration enhanced normal

换句话说就是我需要解决一个探索增强范数的投影 $\bar{\Pi}$ 的投影方程

279 00:14:27,830 –> 00:14:30,550 in other words change the weights
280 00:14:30,550 –> 00:14:33,950 instead of sigh the natural weights of
281 00:14:33,950 –> 00:14:37,070 the policy use a different weight
282 00:14:37,070 –> 00:14:40,790 distribution Zeta that weighs other
283 00:14:40,790 –> 00:14:45,500 states more fairly

也就是说换一个权重，使用另一个权重分布 ζ 代替原来的权重 ξ，让状态出现的更频繁

284 00:14:45,500 –> 00:14:49,010 so Zeta is more balanced than sigh which is the natural
285 00:14:49,010 –> 00:14:51,140 steady state distribution a Markov chain
286 00:14:51,140 –> 00:14:55,260 of mu

所以 ζ 比平稳状态分布 ξ 的马尔科夫链更具有平衡性，

287 00:14:55,260 –> 00:14:58,220 so what how can we generate this
288 00:14:58,220 –> 00:15:02,010 exploration enhanced weights and how do
289 00:15:02,010 –> 00:15:04,650 we solve this equation now okay that's the issue

所以我们该如何产生这个探索的增强权重和如何求解这个方程 ($\Phi r = \bar{\Pi} T_\mu(\Phi r)$) 是一个很重要的问题

290 00:15:04,650 –> 00:15:10,670 and by the way one more thing
291 00:15:10,670 –> 00:15:14,040 what happens if new is not ergodic then

292 00:15:14,040 –> 00:15:16,830 some states will never be visited so if
293 00:15:16,830 –> 00:15:18,120 we change the steady state distribution
294 00:15:18,120 –> 00:15:20,810 of mu likes xi and use another
295 00:15:20,810 –> 00:15:24,180 distribution that goes across her body
296 00:15:24,180 –> 00:15:27,810 classes recurrent classes then that
297 00:15:27,810 –> 00:15:36,630 addresses this problem as well
顺便说一个事情，如果 mu 不是一个便利性的策略，一些状态永远不会被访问，如果我们改变了策略 μ 的平稳状态分布 ξ，使用其他分布来进行采样，还是会产生同样的问题

6 EXPLORATION MECHANISMS

298 00:15:36,630 –> 00:15:40,140 okay now I'm going to talk about briefly about to
299 00:15:40,140 –> 00:15:44,690 exploration mechanisms
我要简单地讲一讲探索机制
300 00:15:44,690 –> 00:15:47,700 one possibility instead of using a single long
301 00:15:47,700 –> 00:15:51,030 trajectory using the policy that starts
302 00:15:51,030 –> 00:15:54,390 at some selected state and theoretically
303 00:15:54,390 –> 00:15:57,150 visits all other states through the
304 00:15:57,150 –> 00:15:59,790 ergodicity property of the chain instead
305 00:15:59,790 –> 00:16:02,550 of using a single trajectory use
306 00:16:02,550 –> 00:16:05,430 multiple trajectories shorter okay
307 00:16:05,430 –> 00:16:07,500 like ten transitions long five
308 00:16:07,500 –> 00:16:09,840 transitions long one transition long and
309 00:16:09,840 –> 00:16:12,960 we pick the states the initial States of
310 00:16:12,960 –> 00:16:15,180 these trajectories from a rich and a
311 00:16:15,180 –> 00:16:18,540 representative sample so we cover the
312 00:16:18,540 –> 00:16:21,300 states by covering the initial States in
313 00:16:21,300 –> 00:16:23,370 this of these long trajectories and we
314 00:16:23,370 –> 00:16:25,920 use that policy for each one of the short trajectories
一种情况是选择初始状态和能够有理论保证遍历性的马尔科夫链用很多短轨迹代替一条长轨迹。比如十次转移的轨迹，五次转移的轨迹和一次转移的轨迹，这些轨迹都由被评估的策略产生我们选择的初始状态要足够丰富并且具有代表性，这样我们才可以通过选择初始状态来覆盖状态集合
315 00:16:25,920 –> 00:16:29,940 okay there are names
316 00:16:29,940 –> 00:16:31,590 associated with this I'm not going to
317 00:16:31,590 –> 00:16:33,480 get into the details of that but your
318 00:16:33,480 –> 00:16:35,510 textbook has a fairly detailed
319 00:16:35,510 –> 00:16:38,540 discussion also gives references
它们的名字就是这个，我不会讲细节了，但是你的课件有很多细节的讨论，还给出了引用
320 00:16:38,540 –> 00:16:41,550 geometric sampling relates to short
321 00:16:41,550 –> 00:16:43,890 simulation trajectories generated
322 00:16:43,890 –> 00:16:45,830 according to a geometric distribution
短期仿真轨迹根据几何采样产生
323 00:16:45,830 –> 00:16:49,290 where by the end of the trajectory is
324 00:16:49,290 –> 00:16:50,860 determined by a lambda
325 00:16:50,860 –> 00:16:52,660 the parameter there is a positive
326 00:16:52,660 –> 00:16:56,380 probability lambda that that any one
327 00:16:56,380 –> 00:16:58,300 transition will be the end and you have
328 00:16:58,300 –> 00:17:01,290 to restart from another trajectory
轨迹的终点被一个参数 λ 决定，这是一个正数，表示终止概率，每一条轨迹都跟据它终止，然后再开始一条新的轨迹
329 00:17:01,290 –> 00:17:04,810 freeform sampling is a very very
330 00:17:04,810 –> 00:17:06,790 flexible and very very general sort of
331 00:17:06,790 –> 00:17:11,440 sampling which has basically you don't
332 00:17:11,440 –> 00:17:14,230 need to it's much it generalizes geometric sampling
freeform sampling 是一种非常灵活的而且非常一般的采样方法，它比几何采样还要一般
333 00:17:14,230 –> 00:17:17,109 and just about any
334 00:17:17,109 –> 00:17:19,540 kind of sampling mechanism comes under

335 00:17:19,540 –> 00:17:22,119 freeform sampling
任何形式的采样机制都来源于 freeform sampling
336 00:17:22,119 –> 00:17:24,339 and still you can solve a certain meaningful bellman equation
你可以使用它来求解一个有意义的 bellman 方程
337 00:17:24,339 –> 00:17:29,440 anyway with short trajectories
338 00:17:29,440 –> 00:17:31,720 we can choose the starting stage and
339 00:17:31,720 –> 00:17:34,290 will enhance exploration this way
无论如何，短轨迹采样我们都可以选择初始阶段并且通过这种方式进行探索
340 00:17:34,290 –> 00:17:38,500 however the simulation formulas to solve
341 00:17:38,500 –> 00:17:42,580 this equation with PI bar not Pi the
342 00:17:42,580 –> 00:17:44,320 simulation formulas become a little different okay
然而仿真来求解这个方程 ($\Phi r = \bar{\Pi} T_\mu^{(\lambda)}(\Phi r)$) 的时候会有一点不同
343 00:17:44,320 –> 00:17:48,280 naturally otherwise yeah
344 00:17:48,280 –> 00:17:51,040 it's natural that this would be so and I
345 00:17:51,040 –> 00:17:53,290 don't want to get into the into the
346 00:17:53,290 –> 00:17:56,950 details but the formulas are not
347 00:17:56,950 –> 00:17:59,530 difficult they're just different and the
348 00:17:59,530 –> 00:18:01,510 amount of computation involved in
349 00:18:01,510 –> 00:18:03,340 solving this equation is about the same
350 00:18:03,340 –> 00:18:09,070 as for the regular bellman equation that
351 00:18:09,070 –> 00:18:12,280 does not involve exploration
我不想讲细节了，这个公式算起来不难，只是与正常的动态规划不一样而已，计算量也与 bellman 方程相同，只是常规的 bellman 方程不包括探索
352 00:18:12,280 –> 00:18:15,190 so that's one way balance the weights of the
353 00:18:15,190 –> 00:18:17,350 states by restarting and many different points in space
这是第一种方法，通过多次使用不同的初始状态来平衡状态的权重
354 00:18:17,350 –> 00:18:20,770 the second possibility
355 00:18:20,770 –> 00:18:24,610 is to use a single long trajectory which
356 00:18:24,610 –> 00:18:27,310 however is generated with a different
357 00:18:27,310 –> 00:18:29,920 policy slightly different policy so that
358 00:18:29,920 –> 00:18:32,560 state foil following the policy mu at
359 00:18:32,560 –> 00:18:36,730 every step we occasionally deviate we
360 00:18:36,730 –> 00:18:38,610 deviate perhaps with some probability
361 00:18:38,610 –> 00:18:42,130 using another policy that has broader
362 00:18:42,130 –> 00:18:45,460 exploration properties
第二种方法是使用另一个策略生成一个长轨迹，状态不与策略 μ 相同，我们使用其他策略以某种概率让轨迹偏离当前轨迹来进行更广泛的探索
363 00:18:45,460 –> 00:18:51,600 so that will tend to visit explore more fully the space
这样就可以更完全地探索空间
364 00:18:51,600 –> 00:18:54,610 and this you find the name in the
365 00:18:54,610 –> 00:18:56,740 literature this is the off policy method
在课件中我把它叫做 off-policy 方法
366 00:18:56,740 –> 00:19:00,040 basically here you have two policies one
367 00:19:00,040 –> 00:19:03,399 is the target policy which is the new
368 00:19:03,399 –> 00:19:05,950 that you are currently evaluating and
369 00:19:05,950 –> 00:19:10,089 the other one is the exploration policy
370 00:19:10,089 –> 00:19:13,570 or off policy that tends to take you of
371 00:19:13,570 –> 00:19:16,139 course to visit other parts of the state
你有两个策略，一个是目标策略，也就是你想要评价的策略，另一个是探索策略或者叫 off 策略，你可以用这个策略对状态空间进行探索
372 00:19:16,139 –> 00:19:19,659 and here the modified policy is a
373 00:19:19,659 –> 00:19:22,359 mixture of the target policy in the exploration policy
这里的修正策略是目标策略扩展成探索策略
-
374 00:19:22,359 –> 00:19:24,549 in this copy off
375 00:19:24,549 –> 00:19:26,710 policy approach on policy for the target
376 00:19:26,710 –> 00:19:28,989 off policy for non tank
-

377 00:19:28,989 –> 00:19:33,249 very old method goes back to the early days what of the field
很老的方法了
378 00:19:33,249 –> 00:19:37,029 however it's important to note
379 00:19:37,029 –> 00:19:39,219 that the simulation formulas for the
380 00:19:39,219 –> 00:19:41,739 basic methods have to be modified so
381 00:19:41,739 –> 00:19:44,950 that you solve this equation rather than
382 00:19:44,950 –> 00:19:48,429 equation involving the modified policy
383 00:19:48,429 –> 00:19:50,710 we still want to evaluate the original policy
有一个很重要的事情要说，基本方法的仿真求解必须修改，以便于求解这个方程 ($\Phi r = \bar{\Pi} T_\mu^{(\lambda)}(\Phi r)$) 而不是包含修正策略的方程组，实际上我们想要评价的一直是原始策略而不是修正策略
384 00:19:50,710 –> 00:19:54,849 now this involves ideas from the
385 00:19:54,849 –> 00:19:57,580 theory of importance sampling so that
386 00:19:57,580 –> 00:19:59,619 all the simulation formulas are what
387 00:19:59,619 –> 00:20:04,089 important sampling modified
这个想法涉及到重要性采样的理论，所有的仿真函数都是重要性采样的基础上修改得到的
388 00:20:04,089 –> 00:20:05,889 this is something that you will see only in
389 00:20:05,889 –> 00:20:07,809 recent writings because in the early
390 00:20:07,809 –> 00:20:10,539 days of the field the idea that the
391 00:20:10,539 –> 00:20:15,190 formulas of LSTD and LSPE methods had
392 00:20:15,190 –> 00:20:18,129 to be modified this had not been fully
393 00:20:18,129 –> 00:20:20,999 appreciated
有一些事情提醒请你们，你们只需要看最近的文章就可以了，因为早期的方法，比如 LSTD 和 LSPE 的修改没有被充分认识到
394 00:20:24,470 –> 00:20:28,490 okay now here's another issue
我们来谈另一个话题
395 00:20:28,490 –> 00:20:31,250 suppose we change from PI to PI bar that will
396 00:20:31,250 –> 00:20:33,140 change the contraction properties of
397 00:20:33,140 –> 00:20:37,060 this mapping here however if lambda is
398 00:20:37,060 –> 00:20:41,120 is positive and close to one then the
399 00:20:41,120 –> 00:20:43,370 contraction property of pipe in lambda
400 00:20:43,370 –> 00:20:47,090 is restored and the methods that need
401 00:20:47,090 –> 00:20:51,410 contraction cabott
假设我们把 Π 改成 $\bar{\Pi}$，这个映射的压缩性就会改变，但是如果 λ 是接近 1 的正数，收缩性仍然能够保证
402 00:20:51,410 –> 00:20:53,600 LSTD does not need the contraction of this mapping it's a
403 00:20:53,600 –> 00:20:56,870 matrix inversion method that that has a
404 00:20:56,870 –> 00:20:58,460 solution give you a solution no matter
405 00:20:58,460 –> 00:21:00,740 whether you have a contraction in this mapping or not
LSTD 就不需要映射的收缩性，因为这是一种矩阵求逆的方法，不在乎这个映射有没有收缩性
406 00:21:00,740 –> 00:21:03,440 however the other methods
407 00:21:03,440 –> 00:21:07,340 LSPE lambda and TD lambda require that
408 00:21:07,340 –> 00:21:09,260 this is a contraction in by getting
409 00:21:09,260 –> 00:21:11,120 lambda sufficiently large you have this
410 00:21:11,120 –> 00:21:13,510 property
但是其他方法，比如 LSPE(λ) 和 TD(λ) 就需要这个给定的 λ 足够大保证映射是一个压缩映射

7 POLICY ITERATION ISSUES: OSCILLATIONS

411 00:21:23,020 –> 00:21:26,679 okay so now we want to look at two
412 00:21:26,679 –> 00:21:28,960 issues of policy improvement one is the
413 00:21:28,960 –> 00:21:31,929 exploration issue and that's all I have
414 00:21:31,929 –> 00:21:33,070 to say about that
我们看看策略改进的两个话题，一个是探索，也就是我刚刚讲完的
415 00:21:33,070 –> 00:21:35,740 and now we're going to get into issues
416 00:21:35,740 –> 00:21:38,710 of policy duration that have to do with
417 00:21:38,710 –> 00:21:41,740 the sequence of policies that we generate
现在我要讲一下策略迭代过程中生成的策略

418 00:21:41,740 –> 00:21:45,370 we mentioned that approximants
419 00:21:45,370 –> 00:21:47,110 policy iteration does not terminate but rather
420 00:21:47,110 –> 00:21:50,520 generates in the end a cycle of policies
421 00:21:50,520 –> 00:21:53,620 perhaps many policies
我现在要说的是近似策略迭代不是确定收敛的，而是生成几个策略互相循环
422 00:21:53,620 –> 00:21:58,390 how can we understand this phenomena
我们该如何理解这种现象呢
423 00:21:58,390 –> 00:21:59,770 okay so the certain figure that's a little hard to
424 00:21:59,770 –> 00:22:01,420 understand at first but after you
425 00:22:01,420 –> 00:22:03,130 understand it you get a lot of insight from it
这个图有一点不好理解，但是理解之后就可以知道很多东西
426 00:22:03,130 –> 00:22:07,090 we consider the space of weights okay
427 00:22:07,090 –> 00:22:09,700 so this is a small dimensional
428 00:22:09,700 –> 00:22:14,740 space of the vectors r okay
我们看到的是权重空间，也就是低维度向量 r 的空间
429 00:22:14,740 –> 00:22:17,710 and we form a partition of that space which we call the greedy partition
我们把这个空间分开，叫他们贪婪划分 (greedy partition)
430 00:22:17,710 –> 00:22:21,460 every policy
431 00:22:21,460 –> 00:22:24,210 has a subset in this partition
每一个策略都对应一个划分的子集合
432 00:22:24,210 –> 00:22:27,160 and r of mu is the set of parameter vectors
$R_m u$ 是参数向量的集合
433 00:22:27,160 –> 00:22:31,260 for which mu is greedy with respect to
434 00:22:31,260 –> 00:22:36,100 the cost corresponding to r
对于与 r 相关的成本来说，这个策略 μ 是一个贪心策略
435 00:22:36,100 –> 00:22:41,050 each r gives you a certain J tilde cost
每一个 r 都对应一个成本 $\tilde{J}$
436 00:22:41,050 –> 00:22:43,840 and if you minimize in bellman equation you get a policy mu
如果你最小化这个 bellman 方程，就可以得到一个策略 μ
437 00:22:43,840 –> 00:22:47,530 the set of all R for which
438 00:22:47,530 –> 00:22:52,960 you get mu is called R's of mu
给定一个策略 μ，所有的 r 都可以被记为 R_μ
439 00:22:52,960 –> 00:22:56,890 okay so in the gist of this is that R_μ is
440 00:22:56,890 –> 00:22:59,470 the set of all r such as if we use an
441 00:22:59,470 –> 00:23:02,740 R in it then the next improved policy is
442 00:23:02,740 –> 00:23:08,290 going to be new
如果我们集合 R_μ 中选一个 r，那么经过策略改进获得的策略就是 μ
443 00:23:08,290 –> 00:23:10,600 okay so we have the space R and there is a partition
所以我们有一个空间 R 与 R 的划分
444 00:23:10,600 –> 00:23:13,480 each policy has a set associated with it
每一个策略都对应划分中的一个集合
445 00:23:13,480 –> 00:23:17,290 if I pick my r within that set then my next
446 00:23:17,290 –> 00:23:22,210 policy will be the one corresponding to
447 00:23:22,210 –> 00:23:24,630 the set
如果我在某一个划分中选择一个 r，那么我能得到的策略就是这个 r 对应的那个策略
448 00:23:25,759 –> 00:23:30,200 okay now notice something else also
来看一些其他东西
449 00:23:30,200 –> 00:23:33,419 suppose that policy evaluation is exact
450 00:23:33,419 –> 00:23:36,720 so for every policy that you evaluate
451 00:23:36,720 –> 00:23:39,749 you obtain a certain weight vector r_μ
452 00:23:39,749 –> 00:23:41,820 for every mu design r of
453 00:23:41,820 –> 00:23:46,440 mu so I can plot these sets R of mu
454 00:23:46,440 –> 00:23:48,899 and I can plot also the point r of mu in here
假设策略评价是准确的，所以对于每一个策略评价你都能获得一个确定的权重向量 r_μ，所以我可以这么画集合 R_μ 和权重向量 r_μ
455 00:23:48,899 –> 00:23:52,950 now here's how approximate
456 00:23:52,950 –> 00:23:55,730 policy iteration is going to work

这就是近似策略迭代工作的方式

457 00:23:55,730 –> 00:24:00,989 suppose that I have the current policy

458 00:24:00,989 –> 00:24:05,149 mu K and I evaluate this policy

459 00:24:05,149 –> 00:24:07,440 according to a projected equation

460 00:24:07,440 –> 00:24:09,210 there's a unique fixed point there's a

461 00:24:09,210 –> 00:24:13,499 unique r okay now now this r use of

462 00:24:13,499 –> 00:24:16,700 K is going to fall into subset r mu

463 00:24:16,700 –> 00:24:20,639 and let's say and the set within it

464 00:24:20,639 –> 00:24:22,980 false is going to correspond to the next

465 00:24:22,980 –> 00:24:28,049 policy if I have argue here the next

466 00:24:28,049 –> 00:24:30,450 policy is going to correspond to the set

467 00:24:30,450 –> 00:24:34,200 within which it falls so mu K plus 1 is

468 00:24:34,200 –> 00:24:36,869 going to be the next policy but new K

469 00:24:36,869 –> 00:24:41,759 plus 1 is evaluated using some weight

470 00:24:41,759 –> 00:24:45,419 vector and the next point in the policy

471 00:24:45,419 –> 00:24:46,859 direction process is going to be this

472 00:24:46,859 –> 00:24:50,669 one this vector falls within some set

473 00:24:50,669 –> 00:24:52,200 and it's going to be the set

474 00:24:52,200 –> 00:24:55,559 corresponding to the next policy and now

475 00:24:55,559 –> 00:24:58,619 we are going to move here to the set R K

476 00:24:58,619 –> 00:25:00,539 nu plus 1 is going to fall into a set

477 00:25:00,539 –> 00:25:03,149 correspond to still another policy which

478 00:25:03,149 –> 00:25:06,119 I'm going to go back and at some point

479 00:25:06,119 –> 00:25:10,139 because these sets are finite there is a

480 00:25:10,139 –> 00:25:12,539 finite number of them because there's

481 00:25:12,539 –> 00:25:14,659 only a finite number of policies

482 00:25:14,659 –> 00:25:17,279 eventually there is no alternative that

483 00:25:17,279 –> 00:25:20,399 you will close a cycle and this is the

484 00:25:20,399 –> 00:25:23,519 cycle generated by the approximate

485 00:25:23,519 –> 00:25:26,039 policy duration with the exact policy

486 00:25:26,039 –> 00:25:28,999 evaluation

假设现在有一个策略 μ_k，然后我通过求解一个投影方程获得的不动点，也就是 r_k，现在这个 r_k 落在了子集 R_μ^{k+1} 中，我们说下一个策略是 μ_{k+1}，我们再对策略 μ_{k+1} 进行评估可以得到新的参数向量 $r_{\mu^{k+1}}$，可以看到 $r_{\mu^{k+1}}$ 落在了子集 R_μ^{k+2} 中，即新策略是 μ^{k+2}，对 μ^{k+2} 进行评价，得到 $r_{\mu^{k+2}}$ 落在了子集 R_μ^{k+3} 中，对 μ^{k+3} 进行评价，得到 $r_{\mu^{k+3}}$，此时 r 落在了子集 R_μ^k 中，策略又回来了，不停地进行策略迭代，就一直这么循环下去。由于参数集合是有限的 (策略是有限个的)，近似策略迭代迭代就会一直在这些集合中循环下去。

487 00:25:30,579 –> 00:25:36,529 okay so genetically assuming that you do

488 00:25:36,529 –> 00:25:39,559 the policy evaluation exactly so for a

489 00:25:39,559 –> 00:25:41,869 given new there's a unique weight vector

490 00:25:41,869 –> 00:25:44,419 associated with it there is going to be

491 00:25:44,419 –> 00:25:46,159 a cycle like this that's going to be closed

一般地说，假设策略评估是精确的，对于一个给定的策略 mu，得到的权重向量 r 会在这个闭环中一直循环下去

492 00:25:46,159 –> 00:25:49,099 the algorithm ends up repeating

493 00:25:49,099 –> 00:25:52,639 some cycle of policies with r mu k

494 00:25:52,639 –> 00:25:56,029 belonging to R capital R mu k plus 1 nu

495 00:25:56,029 –> 00:25:59,059 K R mu k plus 1 belonging to the set of

496 00:25:59,059 –> 00:26:01,820 Nu K plus 2 and so on

497 00:26:01,820 –> 00:26:05,329 and somewhere a cycling is going to be closed

算法在这种循环中执行几次之后就会停止了

498 00:26:05,329 –> 00:26:09,799 now you might ask is it necessary

499 00:26:09,799 –> 00:26:12,289 that we have a cycle can this process terminate

你可能会问我们一定要这样循环迭代吗，这个过程能不能终止呢

500 00:26:12,289 –> 00:26:16,339 sure if r of mu falls

501 00:26:16,339 –> 00:26:18,829 within the set R of mu then you have

502 00:26:18,829 –> 00:26:23,899 convergence in one step

如果 r_μ 在集合 R_μ 中，则迭代一步就收敛了

503 00:26:23,899 –> 00:26:28,719 okay it's only when you have this movement around the
504 00:26:29,289 –> 00:26:33,559 query a little R does not correspond to
505 00:26:33,559 –> 00:26:37,969 to look to to capital R that you get an
506 00:26:37,969 –> 00:26:40,299 oscillation
只有得到的 r 和集合 R 不一样的时候才会循环，也就是产生震荡
507 00:26:47,910 –> 00:26:51,270 so in terms of this figure the typical
508 00:26:51,270 –> 00:26:53,280 trajectory of approximate policy duration
这个图介绍了典型的策略迭代过程
509 00:26:53,280 –> 00:26:55,549 is you start with some policy
510 00:26:55,549 –> 00:26:59,039 you move into some new set corresponding
511 00:26:59,039 –> 00:27:01,590 to the policy then you go to a new
512 00:27:01,590 –> 00:27:05,309 weight vector a new set corresponding to
513 00:27:05,309 –> 00:27:07,669 that weight back to that weight vector
514 00:27:07,669 –> 00:27:10,110 involving a new policy and so on
你从某些策略开始，获得了一个权重向量，然后得到了这个向量对应的策略，继续计算权重向量，策略，权重向量，这么持续下去
515 00:27:10,110 –> 00:27:12,360 so you meet between policies like that and then
516 00:27:12,360 –> 00:27:14,820 you come to some point where you just go on a cycle
所以你在这些策略之间不停地跳转，构成了一个循环
517 00:27:14,820 –> 00:27:19,770 now you hope that this cycle
518 00:27:19,770 –> 00:27:23,039 is a good cycle that all the policies
519 00:27:23,039 –> 00:27:26,190 involved in this cycle are reasonably good policies
现在你希望最终得到的策略是一个好策略，循环中所有的策略都有理由认为是一个好策略
520 00:27:26,190 –> 00:27:30,030 practice with many
521 00:27:30,030 –> 00:27:30,690 problems
522 00:27:30,690 –> 00:27:33,510 indeed indicates that that the cycles
523 00:27:33,510 –> 00:27:38,220 occur in good parts of the space rather
524 00:27:38,220 –> 00:27:40,590 than bad parts of the space
实际解决了很多问题显示循环确实在好策略之间进行而不是在坏策略之间进行
525 00:27:40,590 –> 00:27:44,100 however that's ambitious examples where you get
526 00:27:44,100 –> 00:27:47,600 oscillations within very bad cycles
但是这里有一个例子说明你可以在一些非常坏的策略之间循环
527 00:27:47,600 –> 00:27:49,799 particularly when the problem is of small dimension
特别是这个问题的维度还很低
528 00:27:49,799 –> 00:27:52,140 there are very simple
529 00:27:52,140 –> 00:27:54,000 examples to move in two states three
530 00:27:54,000 –> 00:27:58,169 states okay very unusual kind of but
531 00:27:58,169 –> 00:28:01,380 very very very very revealing kinds of
532 00:28:01,380 –> 00:28:03,740 examples
这是一个非常简单只有三个状态，但是很不寻常能够给人启示的例子
(someone asking questions
这个循环中的策略都离最优策略比较远
533 00:28:07,690 –> 00:28:12,080 now suppose that are there questions
534 00:28:12,080 –> 00:28:14,480 about this figure I know it's a hard
535 00:28:14,480 –> 00:28:16,460 figure to understand I always have to
536 00:28:16,460 –> 00:28:18,710 look at it again and again to figure it
537 00:28:18,710 –> 00:28:29,780 out yes please yes the question is what
538 00:28:29,780 –> 00:28:31,430 about the quality of this cycle is it
539 00:28:31,430 –> 00:28:32,960 possible that all the policies in this
540 00:28:32,960 –> 00:28:35,600 cycle are far from optimal definitely so
541 00:28:35,600 –> 00:28:39,260 yes definitely so you can compare it to
542 00:28:39,260 –> 00:28:41,720 if you can make an analogy that's
543 00:28:41,720 –> 00:28:43,220 somewhat superficial but perhaps
544 00:28:43,220 –> 00:28:46,210 somewhat revealing with local minima
545 00:28:46,210 –> 00:28:50,990 local minima in in optimization where
546 00:28:50,990 –> 00:28:52,850 the method gets attracted some local
547 00:28:52,850 –> 00:28:55,490 minimum that could be very suboptimal
548 00:28:55,490 –> 00:28:58,550 very far from optimal the same thing can

549 00:28:58,550 –> 00:29:01,610 happen here it's a process of like local minimum
2-th question

非常坏的情况是在一个好策略和坏策略之间震荡。这个算法没法保证近似，如果愿意的话可以做一些其他计算给出置信度，你甚至没法判断这个结果有多好，因为谁都不知道最优值是多少。

550 00:29:01,610 –> 00:29:08,800 okay here's another yes please
551 00:29:13,480 –> 00:29:23,539 the arrow of the function this
552 00:29:23,539 –> 00:29:26,470 oscillation maybe
553 00:29:31,350 –> 00:29:36,880 yes yes the question is is it necessary
554 00:29:36,880 –> 00:29:39,730 that I get a bad oscillation definitely
555 00:29:39,730 –> 00:29:41,320 you might get converges to an exact
556 00:29:41,320 –> 00:29:47,110 policy in in very few directions and it
557 00:29:47,110 –> 00:29:51,640 is also possible that this oscillation
558 00:29:51,640 –> 00:29:54,940 can happen but all the policies here are
559 00:29:54,940 –> 00:29:57,130 close to optimal actually this happens
560 00:29:57,130 –> 00:30:07,240 quite often in general it will oscillate
561 00:30:07,240 –> 00:30:10,870 and depending on how you do the policy
562 00:30:10,870 –> 00:30:14,580 evaluation it makes typically oscillate
563 00:30:14,580 –> 00:30:17,260 typically yes there is no convergence
564 00:30:17,260 –> 00:30:19,600 guarantee of this algorithm there's only
565 00:30:19,600 –> 00:30:23,430 a fourth that you will get a good cycle
566 00:30:23,430 –> 00:30:26,050 there's also even worse it's also even
567 00:30:26,050 –> 00:30:28,480 worse suppose you get a cycle and you
568 00:30:28,480 –> 00:30:30,190 look at your computational results and
569 00:30:30,190 –> 00:30:31,690 you see well there's got to be a cycle
570 00:30:31,690 –> 00:30:33,340 because I get this policy that's good
571 00:30:33,340 –> 00:30:35,350 but I get this other policy that's very
572 00:30:35,350 –> 00:30:37,480 bad and I keep oscillating between good
573 00:30:37,480 –> 00:30:40,930 and bad policies how do I know that what
574 00:30:40,930 –> 00:30:44,440 looks to you as good policy is indeed
575 00:30:44,440 –> 00:30:49,060 good okay how do I know that there's not
576 00:30:49,060 –> 00:30:52,860 another policy that's much much better
577 00:30:55,770 –> 00:30:58,450 it cannot be guaranteed in general
578 00:30:58,450 –> 00:31:02,230 except for the generic error bound that
579 00:31:02,230 –> 00:31:05,410 I gave you earlier which is so loose to
580 00:31:05,410 –> 00:31:09,280 be in useless in practice convergence
581 00:31:09,280 –> 00:31:13,570 cannot be guaranteed and and you can
582 00:31:13,570 –> 00:31:16,090 gain some confidence about the results
583 00:31:16,090 –> 00:31:18,180 by doing some additional computations
584 00:31:18,180 –> 00:31:23,140 but still there will be some doubt they
585 00:31:23,140 –> 00:31:25,180 may be some doubt as to whether the
586 00:31:25,180 –> 00:31:27,100 results of the computation are good or
587 00:31:27,100 –> 00:31:29,200 bad because you don't know where the
588 00:31:29,200 –> 00:31:32,010 optimum is
asking completed)
589 00:31:33,900 –> 00:31:37,600 okay there's some even more weird things happening here
有一些非常奇怪的事情将要发生了
590 00:31:37,600 –> 00:31:44,049 if we have a certain
591 00:31:44,049 –> 00:31:48,220 policy mu k let's say the typical
592 00:31:48,220 –> 00:31:52,059 policy exactly I'm sorry solving the
593 00:31:52,059 –> 00:31:53,799 projected bellman equation will give you
594 00:31:53,799 –> 00:31:57,250 a unique weight vector however if you
595 00:31:57,250 –> 00:31:59,559 use an optimistic method whereby you
596 00:31:59,559 –> 00:32:01,390 solve this bellman equation only
597 00:32:01,390 –> 00:32:03,190 approximately then you're going to get
598 00:32:03,190 –> 00:32:05,559 something that has some error all you
599 00:32:05,559 –> 00:32:08,230 can say is that from arm UK you can go
600 00:32:08,230 –> 00:32:10,929 towards this point but not necessarily add

如果我有一个策略 μ^k，求解 bellman 方程组可以得到一个唯一的权重向量，如果你使用一种乐观方法 (optimistic method)，你只能得到一个近似的向量 r，这存在一些误差，然后你可以说 r 从初始点到了点 r_{μ^k}，但是并不是一定能到达这个点 $(r_{\mu^{k+1}})$

MORE ON OSCILLATIONS/CHATTERING

601 00:32:10,929 –> 00:32:13,809 it so a different kind of figure will work then

这是另一个类型的工作状况的图

602 00:32:13,809 –> 00:32:19,120 whereby let's say which

603 00:32:19,120 –> 00:32:22,809 we are at we have policy mu one and

604 00:32:22,809 –> 00:32:24,549 then we generate policy mu two buy

605 00:32:24,549 –> 00:32:27,429 policy improvement and from here we go

606 00:32:27,429 –> 00:32:30,280 part of the way towards this but we

607 00:32:30,280 –> 00:32:33,460 don't quite reach it because we use an optimistic algorithm

我们现在在 μ_1，通过策略改进策略变成了 μ_2，在图上从这个点到这个点，但是由于我们使用乐观算法进行策略改进，我们并不是准确地到达新策略的，

608 00:32:33,460 –> 00:32:36,580 and then from here

609 00:32:36,580 –> 00:32:39,250 we go towards here but we don't quite reach it

然后通过策略改进到达一个新的点，同样不能准确到达

610 00:32:39,250 –> 00:32:41,950 so what's happening is the

611 00:32:41,950 –> 00:32:45,130 magnitude of the oscillation is is

612 00:32:45,130 –> 00:32:49,600 becomes reduced it becomes less if you

613 00:32:49,600 –> 00:32:53,320 make the we we evaluation progressively

614 00:32:53,320 –> 00:32:55,480 more and more optimistic then you can

615 00:32:55,480 –> 00:32:58,299 get smaller and smaller oscillation

所以如果你在进行策略评价的时候越来越乐观，震荡幅度就会逐渐变得越来越小

616 00:32:58,299 –> 00:33:00,210 and in fact you can get convergence

617 00:33:00,210 –> 00:33:03,460 convergence of the weight vectors and

618 00:33:03,460 –> 00:33:06,549 then the strange thing is that we can

619 00:33:06,549 –> 00:33:09,640 have convergence of the R vectors as you

620 00:33:09,640 –> 00:33:12,090 are having an oscillation of policies

一件很奇怪的事情就是在你的策略是一个震荡策略的时候可以收敛到一个固定的权重向量 r

621 00:33:12,090 –> 00:33:14,289 it's just that you convert that this

622 00:33:14,289 –> 00:33:17,710 joint this junction point in the greedy

623 00:33:17,710 –> 00:33:20,409 partition between policies and you

624 00:33:20,409 –> 00:33:22,929 generate a cycle of policies but the r

625 00:33:22,929 –> 00:33:25,270 vectors seem to converge

也就是在使用贪心划分策略时你可以产生一个策略循环但是权重向量 r 最终会收敛

626 00:33:25,270 –> 00:33:27,070 so you look at the r vectors and you may be very

627 00:33:27,070 –> 00:33:29,620 happy ok my algorithm is converging but

628 00:33:29,620 –> 00:33:31,960 in fact it may not be converging it may

629 00:33:31,960 –> 00:33:34,240 be oscillating and very widely so between policies

所以在关注向量 r 的时候你可能会非常开心，算法收敛了。但是实际上没有收敛，他可能还在大幅度地在策略间震荡

630 00:33:34,240 –> 00:33:39,820 convergence in the

631 00:33:39,820 –> 00:33:42,610 space of r but divergence or

632 00:33:42,610 –> 00:33:45,130 oscillation in the space of MU

r 空间收敛了，但是在策略空间上发散

633 00:33:45,130 –> 00:33:48,070 it's a very difficult phenomenon to

634 00:33:48,070 –> 00:33:51,430 analyze but you can witness it with very

635 00:33:51,430 –> 00:33:52,540 simple examples

这种现象非常难以分析，但是很容易用简单的例子验证

636 00:33:52,540 –> 00:33:55,120 you know like three state examples four

637 00:33:55,120 –> 00:33:57,280 state examples and you can get two state exams

你知道的，那种两个三个或者四个状态的例子

638 00:33:57,280 –> 00:34:01,900 you can get convergence of r two

639 00:34:01,900 –> 00:34:04,690 points that are meaningless okay they're

640 00:34:04,690 –> 00:34:06,550 just Junction points in some strange

641 00:34:06,550 –> 00:34:09,370 diagram that you can never compute you

642 00:34:09,370 –> 00:34:13,389 can never understand

你可以得到一个收敛的 r，但是这没有意义，在一些奇怪的图中这个联合点 (收敛的点) 没有办法计算，而且根本没法理解为什么会产生这种状况

643 00:34:13,389 –> 00:34:17,040 okay so what why is this happening and how can we fix it

那么这种现象是如何产生的，我们该如何补救呢

644 00:34:17,040 –> 00:34:20,219 mathematically speaking fundamentally

645 00:34:20,219 –> 00:34:22,929 oscillations are due to the lack of

646 00:34:22,929 –> 00:34:26,429 monotonicity of the projection operator

从数学上说，本质上震荡是由于投影算子不具有单调性导致的

647 00:34:26,429 –> 00:34:29,770 by monotonicity I mean that if I have

648 00:34:29,770 –> 00:34:33,429 two functions J and J Prime when I

649 00:34:33,429 –> 00:34:35,889 project them on the approximation

650 00:34:35,889 –> 00:34:38,830 subspace their order may be may be switched around

关于单调性，我想说的是如果我有两个函数 J 和 J'，当我在近似子空间内对他们进行投影的时候，结果的关系会与原关系相反

651 00:34:38,830 –> 00:34:43,210 now remember that we

652 00:34:43,210 –> 00:34:47,710 have PI T T is monotone a very

653 00:34:47,710 –> 00:34:49,480 fundamental property in dynamic

654 00:34:49,480 –> 00:34:52,360 programming once you compose it with the

655 00:34:52,360 –> 00:34:54,429 projection operator you lose the monotonicity property

还记得吧，ΠT 和 T 是单调的，这是动态规划中很基本的性质了，如果你把他们与投影算子结合，就会失去单调性

656 00:34:54,429 –> 00:34:57,400 and all the proofs

657 00:34:57,400 –> 00:34:59,470 that I gave earlier for exact dynamic

658 00:34:59,470 –> 00:35:01,000 programming which are based on model

659 00:35:01,000 –> 00:35:03,270 monotonicity do not hold anymore

我们之前在基于模型的精确动态规划中进行的所有单调性的证明在这里都失效了

660 00:35:03,270 –> 00:35:06,310 mathematically that's the problem

这就是问题所在

661 00:35:06,310 –> 00:35:11,050 lack of one monotonicity of this operator

算子不再具有单调性了

662 00:35:11,050 –> 00:35:14,530 if you were to change this operator and use

663 00:35:14,530 –> 00:35:17,440 adapt a different operator say some W

664 00:35:17,440 –> 00:35:20,500 operator that is monotone and also this

665 00:35:20,500 –> 00:35:22,600 is a contraction then you would not have

666 00:35:22,600 –> 00:35:24,970 these oscillations and you will have

667 00:35:24,970 –> 00:35:28,930 converges to a single policy

如果你换了一个具有单调性的算子 W，同时 WT_μ 是收缩的，你的算法就不会震荡而且可以收敛到一个确定的策略

668 00:35:28,930 –> 00:35:30,700 this is a little difficult to explain but not

669 00:35:30,700 –> 00:35:33,790 difficult but will take us too much time

670 00:35:33,790 –> 00:35:35,620 to explain you may explore it on your

671 00:35:35,620 –> 00:35:37,990 own this is the mathematical reason for

672 00:35:37,990 –> 00:35:40,510 all these oscillations

这是一个不太难解释但是会花费很长时间去解释的问题，你可以尝试自己解决它，从震荡的数学角度上进行解释

673 00:35:40,510 –> 00:35:43,320 however there's an important special case for which

674 00:35:43,320 –> 00:35:46,900 the evaluation equation involves a

675 00:35:46,900 –> 00:35:50,710 mapping W that is both monotone and the

676 00:35:50,710 –> 00:35:54,180 contraction and this is the next method aggregation

现在又要给很重要的特殊例子，评价方程包括了单调而且收缩的映射 W 的方法，聚合法

677 00:35:54,180 –> 00:35:56,750 in aggregation

678 00:35:56,750 –> 00:36:00,109 we do have similarities but a mapping

679 00:36:00,109 –> 00:36:03,140 that is monitored and avoids this oscillations

聚合方法有这样的特性，映射单调并且可以避免震荡

680 00:36:03,140 –> 00:36:12,020 so let's leave this thought

681 00:36:12,020 –> 00:36:13,970 at this point and then we'll come back

682 00:36:13,970 –> 00:36:16,550 after a break to look at aggregation in

683 00:36:16,550 –> 00:36:21,760 why this is happening

我们要休息一会，一会来讲一下聚合和为什么聚合会导致这种现象出现
(someone asking questions
问题 1：如果探索不足是如何影响到结果的，和如何判断需要多少探索能够避免这些问题
回答：没法判断到底多少探索能行，很难的问题，凭感觉
684 00:36:23,710 –> 00:36:26,710 and any questionsyes
685 00:36:38,550 –> 00:36:46,110 okay the question is if we have
686 00:36:46,110 –> 00:36:50,310 inadequate exploration how damaging can
687 00:36:50,310 –> 00:36:54,330 this be and how much exploration do we
688 00:36:54,330 –> 00:36:58,020 need to avoid the problems the answer is
689 00:36:58,020 –> 00:37:00,690 that it's very difficult to give
690 00:37:00,690 –> 00:37:02,700 anything to say anything quantitative
691 00:37:02,700 –> 00:37:06,150 about this it's mostly a matter of trial
692 00:37:06,150 –> 00:37:10,080 and error that's that that's the only way I can answer
问题 2：如何知道探索不足
回答：看结果，如果收敛到了一个很奇怪的策略，就是探索不足了
693 00:37:10,080 –> 00:37:21,890 any other questions yes
694 00:37:26,580 –> 00:37:28,170 okay how do we know that we have
695 00:37:28,170 –> 00:37:31,109 inadequate exploration well basically by
696 00:37:31,109 –> 00:37:34,290 looking at the results if you see that
697 00:37:34,290 –> 00:37:38,670 the policies are changed widely and you
698 00:37:38,670 –> 00:37:41,900 get strange policies by your process
699 00:37:41,900 –> 00:37:45,810 because basically you're using cost
700 00:37:45,810 –> 00:37:48,300 functions that are way off and balanced
701 00:37:48,300 –> 00:37:51,510 in favor of some states and relative to
702 00:37:51,510 –> 00:37:54,119 others then that's sort of a an
703 00:37:54,119 –> 00:37:56,280 indication that your exploration problem
704 00:37:56,280 –> 00:37:58,080 is that you have an exploration prop
705 00:37:58,080 –> 00:38:00,210 usually that's not hard to tell because
706 00:38:00,210 –> 00:38:01,920 the algorithm just completely breaks
707 00:38:01,920 –> 00:38:03,840 down if you have an adequate exploration
708 00:38:03,840 –> 00:38:05,400 so if you see that you're getting
709 00:38:05,400 –> 00:38:07,740 garbage that's that's the first place
710 00:38:07,740 –> 00:38:09,810 that you look it's my exploration
711 00:38:09,810 –> 00:38:12,380 sufficient
712 00:38:17,530 –> 00:38:19,700 okay so let's take a break for ten
713 00:38:19,700 –> 00:38:20,990 minutes and then we'll come back to look
714 00:38:20,990 –> 00:00:00,000 into aggregation
asking completed)