1 ADDITIONAL ISSUES OF AGGREGATION

1.1 OVERVIEW

1 00:00:00,470 –> 00:00:03,550 [Music]
2 00:00:11,480 –> 00:00:13,010 okay
3 00:00:13,010 –> 00:00:18,560 so let's continue here with aggregation
我们继续讲聚合
4 00:00:18,560 –> 00:00:20,820 remember that we define an aggregate
5 00:00:20,820 –> 00:00:22,230 problem a single problem
6 00:00:22,230 –> 00:00:23,880 it's a simpler problem involving
7 00:00:23,880 –> 00:00:26,490 aggregate states
还记得么，我们定义了一个比原系统更简单的有聚合状态的聚合系统
8 00:00:26,490 –> 00:00:28,590 using this disaggregation aggregation probabilities
9 00:00:28,590 –> 00:00:30,689 we can construct transition
10 00:00:30,689 –> 00:00:32,369 probabilities between aggregate States
11 00:00:32,369 –> 00:00:36,000 and transition costs corresponding to
12 00:00:36,000 –> 00:00:38,610 aggregation States and controls
使用聚合概率我们可以构造两个聚合状态间的状态转移概率和聚合状态与控制间的转移成本
13 00:00:38,610 –> 00:00:42,180 we can consider a policy duration algorithm an
14 00:00:42,180 –> 00:00:45,059 exact policy duration algorithm or any
15 00:00:45,059 –> 00:00:47,610 kind of algorithm that solves this problem exactly
16 00:00:47,610 –> 00:00:50,219 it could be also by simulation
我们可以考虑使用精确策略迭代算法或者任何算法求解这个问题的精确解，当然也可以使用仿真来求解
17 00:00:50,219 –> 00:00:56,370 and however when it comes to
18 00:00:56,370 –> 00:00:59,449 policy direction there is a certain certain deficiency
在使用策略迭代时，有一个缺点
19 00:00:59,449 –> 00:01:04,890 it has to do with the
20 00:01:04,890 –> 00:01:10,160 fact that I have to apply control u
21 00:01:10,160 –> 00:01:12,869 the same control u for all the
22 00:01:12,869 –> 00:01:17,000 aggregate for the entire aggregate State
就是我必须对所有聚合状态使用同一个控制 u
23 00:01:17,000 –> 00:01:20,099 so policies for this problem consists of
24 00:01:20,099 –> 00:01:23,130 assignment of a common control to all
25 00:01:23,130 –> 00:01:28,649 the states that correspond to X
所以这个问题的策略需要分配给聚合状态 x 下所有的状态一个通用的控制
26 00:01:28,649 –> 00:01:30,810 it is possible to address this deficiency and
27 00:01:30,810 –> 00:01:33,119 that's the first subject I want to discuss now
这可能导致一些不足，也是我现在要讲的第一个问题

1.2 ALTERNATIVE POLICY ITERATION

28 00:01:33,119 –> 00:01:36,420 so here's an alternative
29 00:01:36,420 –> 00:01:39,959 policy Direction algorithm and it deals
30 00:01:39,959 –> 00:01:41,429 with a **faculty proceeding** policy
31 00:01:41,429 –> 00:01:43,259 direction method uses policy that the
32 00:01:43,259 –> 00:01:45,450 designed policies designed are controlled
33 00:01:45,450 –> 00:01:48,630 which aggregate state
这是一种策略迭代算法的替代品，用来处理策略迭代，策略被设计为控制聚合状态
34 00:01:48,630 –> 00:01:51,989 an alternative is to use policy duration for a combined system
一个替代品是用策略迭代来求解组合系统
35 00:01:51,989 –> 00:01:55,920 not this aggregate system over here
不是这个聚合系统
36 00:01:55,920 –> 00:02:00,119 but rather consider a super system
37 00:02:00,119 –> 00:02:03,149 a bigger system that involves a state's
38 00:02:03,149 –> 00:02:09,419 both the x and also the i
而是一个同时有 x 和 i 的更大的系统
39 00:02:09,419 –> 00:02:12,340 there is a bellman equation associated with the system

这是这个系统的 bellman 方程
40 00:02:12,340 –> 00:02:15,970 in wood for this system I can
41 00:02:15,970 –> 00:02:18,670 assign controls u to each I rather than to each X
在这个系统中，我可以把控制 u 分配给每一个 i 而不是每一个 x
42 00:02:18,670 –> 00:02:22,420 that's the a large system or combined system
这就是一个大系统或者叫组合系统
43 00:02:22,420 –> 00:02:26,470 now bellman equations
44 00:02:26,470 –> 00:02:32,830 involve equations for X equations for AI
45 00:02:32,830 –> 00:02:37,030 equations for J and related to equations for y
这个 bellman 方程包括 x，i，j 和 y
46 00:02:37,030 –> 00:02:40,750 so here are the equations here's the bellman
这就是 bellman 方程
47 00:02:40,750 –> 00:02:45,310 equation I'm sorry we're
48 00:02:45,310 –> 00:02:48,670 having a problem with the audio I don't
49 00:02:48,670 –> 00:02:51,420 know what I have done wrong
不知道话筒出什么问题了
50 00:03:25,470 –> 00:03:28,800 okay so the optimal costs of the
51 00:03:28,800 –> 00:03:32,130 aggregate states are related
52 00:03:32,130 –> 00:03:34,770 using these probabilities to the optimal
53 00:03:34,770 –> 00:03:38,220 cost of states I
这个聚合状态的最优成本与 i 的最优成本与相应的概率有关
54 00:03:38,220 –> 00:03:41,030 the states I satisfy this bellman equation here
状态 i 满足 bellman 方程
55 00:03:41,030 –> 00:03:44,640 with a control chosen within the set use of I
控制在集合 $U(i)$ 中
56 00:03:44,640 –> 00:03:51,510 and the we the cost of state J are
57 00:03:51,510 –> 00:03:53,910 related to our stash of Y according to this equation
状态 j 的成本与 y 和相应的方程有关
58 00:03:53,910 –> 00:03:56,610 so this is a set of
59 00:03:56,610 –> 00:04:00,390 equations with unknowns both the J
60 00:04:00,390 –> 00:04:04,230 tildes and the J ones and the r stars
61 00:04:04,230 –> 00:04:07,400 or a much much larger dimensional setting
这是一个方程集合，未知数包括 $\tilde{J}$，J_1 和 R^* 这是一个很高维度的方程
62 00:04:07,400 –> 00:04:11,940 it is possible to use again
63 00:04:11,940 –> 00:04:15,480 exact policy duration and value
64 00:04:15,480 –> 00:04:21,959 iteration and the simulation is process
65 00:04:21,959 –> 00:04:24,420 is similar involving a simulator of the
66 00:04:24,420 –> 00:04:26,670 system but the computations are more complicated
可以用精确策略迭代，值迭代和仿真求解，但是计算量特别大
67 00:04:26,670 –> 00:04:30,210 so so there is this
68 00:04:30,210 –> 00:04:32,130 possibility and their approximate value
69 00:04:32,130 –> 00:04:33,660 iteration and policy the duration
70 00:04:33,660 –> 00:04:35,940 methods associated with it
所以可以使用相关的近似值迭代和近似策略迭代求解
71 00:04:35,940 –> 00:04:38,160 you'll find this in the literature and salvage it
72 00:04:38,160 –> 00:04:44,310 what kind of method
你可以从文章中找到他们

1.3 RELATION OF AGGREGATION/PROJECTION

73 00:04:44,310 –> 00:04:47,160 okay the second issue on the touch upon is the relation
74 00:04:47,160 –> 00:04:49,290 between aggregation and the projection approaches
第二个话题是聚合与投影之间的关系
75 00:04:49,290 –> 00:04:52,560 the projected equation
76 00:04:52,560 –> 00:04:56,610 approach involves linear weighting of
77 00:04:56,610 –> 00:04:59,490 features combined with projection to
78 00:04:59,490 –> 00:05:03,840 solve this equation either with the T
79 00:05:03,840 –> 00:05:05,910 mapping or the team new mapping in the

80 00:05:05,910 –> 00:05:09,590 case of single policy
投影方程包括线性加权特征投影的组合来求解这个方程，方程中算子可能是 T，也可能是 T_μ
81 00:05:09,590 –> 00:05:12,870 the aggregation equation looks like this again it
82 00:05:12,870 –> 00:05:16,710 involves linear weighting of features
聚合方程是这样的，它也包括线性加权特征
83 00:05:16,710 –> 00:05:18,540 now the features the the weights have a
84 00:05:18,540 –> 00:05:20,400 different meaning they are related to
85 00:05:20,400 –> 00:05:23,550 the optimal solution or the policy costs
86 00:05:23,550 –> 00:05:26,810 associated with the aggregate problem
这个权重就有不同的含义了，它与聚合问题的最优解或者策略成本有关
87 00:05:27,380 –> 00:05:30,720 there is a reflection that capital Phi
88 00:05:30,720 –> 00:05:37,110 is is is is is a feature it's a feature
89 00:05:37,110 –> 00:05:41,340 matrix basis a matrix of basis functions
这里有一个反射，Φ 是一个矩阵形式的基函数
90 00:05:41,340 –> 00:05:43,500 but the basis functions here are defined
91 00:05:43,500 –> 00:05:45,090 in terms of probabilities they are not general
但是这个基函数被定义成概率的形式，并不是一般意义上的基函数
92 00:05:45,090 –> 00:05:50,550 and finally there is the mapping
93 00:05:50,550 –> 00:05:54,780 that multiplies T here it is Phi D the
94 00:05:54,780 –> 00:05:56,910 aggregation disaggregation probabilities
95 00:05:56,910 –> 00:06:00,180 here is a general weighted Euclidean projection
最后，乘以 T 的是聚合/分解概率 Φ，而这里是一个一般性的加权欧几里得投影
96 00:06:00,180 –> 00:06:06,150 now if the product of Phi and
97 00:06:06,150 –> 00:06:10,080 D is a projection matrix a weighted
98 00:06:10,080 –> 00:06:12,330 projection matrix with respect to some
99 00:06:12,330 –> 00:06:15,330 way that you can Euclidean norm
这里的 ΦD 是一个带有欧几里得范数的加权投影矩阵
100 00:06:15,330 –> 00:06:18,330 then the aggregation approach is a special case
101 00:06:18,330 –> 00:06:21,690 of the projection approach
但是聚合方法只是投影方法的一种特例
102 00:06:21,690 –> 00:06:27,770 with just a particular instance of projection okay
投影的一种特殊形式
103 00:06:27,770 –> 00:06:31,800 so what kinds of aggregations are are
104 00:06:31,800 –> 00:06:34,470 equivalent to some form of projection
那么什么样的聚合与投影等价呢
105 00:06:34,470 –> 00:06:36,270 it turns out that hard aggregations a special case
硬聚合是一种特例
106 00:06:36,270 –> 00:06:39,270 in hard aggregation where
107 00:06:39,270 –> 00:06:42,540 phi consists of zeros and ones then
108 00:06:42,540 –> 00:06:44,760 this matrix here can be verified to be a
109 00:06:44,760 –> 00:06:47,310 projection with respect to weights
110 00:06:47,310 –> 00:06:50,640 xi I which are proportional to the
111 00:06:50,640 –> 00:06:54,390 disaggregation probabilities
在硬聚合中 Φ 包括 0 和 1，这个矩阵 (ΦD) 可以是与分解概率成比例的关于权重 ξ_i 的投影
112 00:06:54,390 –> 00:06:56,280 if you assume this aggregation probabilities
113 00:06:56,280 –> 00:06:58,800 that are you that are positive across each aggregate state
如果你假设所有聚合状态聚合概率都是正数
114 00:06:58,800 –> 00:07:01,530 and you consider
115 00:07:01,530 –> 00:07:04,740 all of those and scale them so that they
116 00:07:04,740 –> 00:07:06,690 add up to one you're going to obtain
117 00:07:06,690 –> 00:07:09,510 weights according to which Phi D is
118 00:07:09,510 –> 00:07:13,800 going to be a projection
你考虑这些所有的东西并且把他们标量话，他们增加到 1，你就可以得到权重 ΦD 是一个投影
119 00:07:13,800 –> 00:07:16,280 that's simple to verify
这可以很简单地被验证
120 00:07:18,300 –> 00:07:20,890 it turns out the hard aggregation is
121 00:07:20,890 –> 00:07:23,680 just about the only case for which the
122 00:07:23,680 –> 00:07:25,900 statement can be made if you have

123 00:07:25,900 –> 00:07:27,520 different kinds of aggregation however
124 00:07:27,520 –> 00:07:31,300 there is another statement which falls
125 00:07:31,300 –> 00:07:33,420 and is just as good
事实证明硬聚合是这种情况下唯一的例子，如果你有另一个聚合的方法，它会有另一个情况并且同样好
126 00:07:33,420 –> 00:07:35,950 suppose that we consider aggregation
127 00:07:35,950 –> 00:07:39,280 with representative features
假设我们考虑有代表性特征的聚合方法
128 00:07:39,280 –> 00:07:41,650 this is the case where the aggregate states are
129 00:07:41,650 –> 00:07:44,620 subsets of states of the original
130 00:07:44,620 –> 00:07:47,110 problem but they may not exhaust the entire space
这是一种聚合状态是原问题的状态的集合但是没有覆盖整个状态空间的例子
131 00:07:47,110 –> 00:07:50,980 remember that this was this
132 00:07:50,980 –> 00:07:54,820 contains a special cases both the hard
133 00:07:54,820 –> 00:07:58,240 aggregation case and the aggregation
134 00:07:58,240 –> 00:08:01,330 with representative states will you use
135 00:08:01,330 –> 00:08:04,510 simple state aggregate States single
136 00:08:04,510 –> 00:08:07,500 States okay single States realizations
你要记住这是一个特殊的情况，代表性状态聚合会使用单独的聚合状态实现
137 00:08:07,500 –> 00:08:12,820 then for this case it turns out that phi d
138 00:08:12,820 –> 00:08:18,190 is a projection but not with respect to a norm
这个例子证明 ΦD 是一种没有范数的投影
139 00:08:18,190 –> 00:08:21,700 a projection with respect to a
140 00:08:21,700 –> 00:08:25,200 semi norm how weighted the second one
141 00:08:25,200 –> 00:08:28,740 where the weights are proportional again
142 00:08:28,740 –> 00:08:32,890 to d however some weights may be zero
一个带有半范数权重的投影，同时权重与 d 成比例，一部分权重可能是 0
143 00:08:32,890 –> 00:08:36,010 that’s the meaning of semi-norm let me explain this
这代表 semi-norm，我来解释一下
144 00:08:36,010 –> 00:08:40,360 you remember the norm
145 00:08:40,360 –> 00:08:42,490 for the case of Euclidean weighted projection
还记得欧几里得范数加权投影么
146 00:08:42,490 –> 00:08:47,770 it involves the sum of the
147 00:08:47,770 –> 00:08:53,080 squares of the components of J weighted
148 00:08:53,080 –> 00:08:55,540 with positive components who has
149 00:08:55,540 –> 00:08:58,420 positive weights this is the case of a norm
它包括 J 的平方由正向量加权后累加的平方，这是一种范数
150 00:08:58,420 –> 00:09:01,480 and ξ here is the vector of this weight
这个 ξ 就是权重向量
151 00:09:01,480 –> 00:09:05,350 however suppose that you
152 00:09:05,350 –> 00:09:08,530 have some size to be positive and some
153 00:09:08,530 –> 00:09:11,350 others be zero then you don’t have a norm
但是假设这个向量一部分是正数，另一部分是 0，它就不是一个范数了
154 00:09:11,350 –> 00:09:16,390 you have instead a semi norm
就是一个 semi-norm
155 00:09:16,390 –> 00:09:21,100 a semi norm is is you corresponds to having
156 00:09:21,100 –> 00:09:24,400 norm zero for some J’s that not zero
semi-norm 表示你在计算 J 的范数的时候范数是 0 但是 J 不是 0
157 00:09:24,400 –> 00:09:27,280 standard property of the norm
158 00:09:27,280 –> 00:09:29,920 is that the norm of the zero vector is
159 00:09:29,920 –> 00:09:33,130 zero and any vector whose norm is 0 must
160 00:09:33,130 –> 00:09:36,460 be the zero vector okay
标准范数的性质是 0 向量的范数是 0，范数是 0 的向量一定是 0 向量
161 00:09:36,460 –> 00:09:39,730 norm of J equals zero if and only if J equals zero
只有当 J 是 0 向量的时候 J 的范数才是 0
162 00:09:39,730 –> 00:09:41,740 or semi norms this is not true。
但对于 semi-norm 这就不成立了
163 00:09:41,740 –> 00:09:46,000 you may have J norm of J equals zero but

164 00:09:46,000 –> 00:09:48,130 J may be nonzero
165 00:09:48,130 –> 00:09:50,170 it’s just that nonzero components
166 00:09:50,170 –> 00:09:54,640 correspond to zero weights
J 的范数等于 0 但是 J 可能不是 0 向量，有可能非零元素对应 0 权重
167 00:09:54,640 –> 00:09:56,650 okay so this is another concept that I’m throwing at you
这就是我要告诉你的另一个概念
168 00:09:56,650 –> 00:10:00,340 it’s possible to consider semi norms
169 00:10:00,340 –> 00:10:02,920 and projection with respect rejected
170 00:10:02,920 –> 00:10:06,150 equation with respect to semi enormous
可以考虑 semi-norm 的投影方程
171 00:10:06,150 –> 00:10:08,650 in order for this approach to make sense
172 00:10:08,650 –> 00:10:11,530 it has to be the case that when you
173 00:10:11,530 –> 00:10:13,480 project on the approximation subspace
174 00:10:13,480 –> 00:10:15,550 with a semi norm there’s a unique
175 00:10:15,550 –> 00:10:19,420 projection associated with it
为了让这个方法有意义，必须满足的条件就是你在近似子空间用 semi-norm 进行投影的时候，存在个唯一的投影
176 00:10:19,420 –> 00:10:21,850 okay if you have that property then everything I
177 00:10:21,850 –> 00:10:23,380 have told you about earlier about
178 00:10:23,380 –> 00:10:26,350 projections norm projections hold also
179 00:10:26,350 –> 00:10:27,310 for semi norms
如果有这个性质，那么我之前讲的关于范数投影对 semi-norm 同样成立
180 00:10:27,310 –> 00:10:30,730 the key idea is that this pi gives you
181 00:10:30,730 –> 00:10:33,430 a unique projection and then in that
182 00:10:33,430 –> 00:10:35,290 case with a unique solution to this equation
主要想法是 Π 给了你一个唯一的投影，这样这个方程组 $(\Phi r = \Pi T(\Phi r))$ 就有一个唯一解
183 00:10:35,290 –> 00:10:38,350 now in order to get a unique
184 00:10:38,350 –> 00:10:40,420 projection for every vector to be
185 00:10:40,420 –> 00:10:43,390 approximation subspace
为了对近似子空间的所有向量都有一个唯一的投影
186 00:10:43,390 –> 00:10:46,900 what you need is this matrix here to be invertible
你需要做的就是保证这个矩阵 $(\Phi\Xi\Phi)$ 可逆
187 00:10:46,900 –> 00:10:49,330 when you’re calculating projections you
188 00:10:49,330 –> 00:10:51,760 invert this matrix here involves
189 00:10:51,760 –> 00:10:56,380 evolving phi prime xi the diagonal
190 00:10:56,380 –> 00:11:01,480 matrix of weights and Phi which is this matrix here
当你计算投影的时候，你需要求这个矩阵 $(\Phi\Xi\Phi)$ 的逆
191 00:11:01,480 –> 00:11:06,310 in the norm projection case
192 00:11:06,310 –> 00:11:13,480 it’s xi is is has all diagonal elements positive
范数投影情况下，这个 Ξ 的所有对角元素都是正数
193 00:11:13,480 –> 00:11:17,820 this is invertible provided
194 00:11:17,820 –> 00:11:20,740 there’s a full rank assumption for this
195 00:11:20,740 –> 00:11:24,180 phi matrix
假设矩阵 Φ 满秩，则可逆性能够保证
196 00:11:24,180 –> 00:11:27,450 however this matrix here is small dimensional
但是这个矩阵维度很低
197 00:11:27,450 –> 00:11:32,200 and and it is possible that it is
198 00:11:32,200 –> 00:11:34,360 invertible in fact it’s very common that
199 00:11:34,360 –> 00:11:36,700 this matrix is invertible even if some
200 00:11:36,700 –> 00:11:40,230 of the components of xi are zero
但是这个矩阵可能是可逆的，实际上经常是可逆的，即使 Ξ 的一些元素是 0
201 00:11:40,230 –> 00:11:43,630 basically this matrix is a weighted sum
202 00:11:43,630 –> 00:11:49,540 of rank one matrices
203 00:11:49,540 –> 00:11:51,370 and and if sufficient number with rank one matrices
204 00:11:51,370 –> 00:11:53,320 are included with positive weights in
205 00:11:53,320 –> 00:11:57,360 the sum then this matrix is invertible
基本上，这个矩阵是秩为 1 的矩阵的加权和，如果秩是 1 的矩阵数量足够多并且加权都是正数，那么这个矩阵就是可逆的

206 00:11:59,070 –> 00:12:02,650 so if this matrix is invertible then the
207 00:12:02,650 –> 00:12:07,360 entire theory of algorithms and and
208 00:12:07,360 –> 00:12:09,660 contraction properties and so on
209 00:12:09,660 –> 00:12:12,280 generalizes to semi norm projected
210 00:12:12,280 –> 00:12:15,460 equations and therefore this entire
211 00:12:15,460 –> 00:12:17,620 theory applies to aggregation problems
212 00:12:17,620 –> 00:12:19,830 as well with this kind of aggregation

如果这个矩阵可逆，那么算法和收缩性等整套理论就可以推广到 semi-norm 投影方程，同样可以应用于所有聚合问题中

213 00:12:19,830 –> 00:12:22,930 and the interesting part about it is
214 00:12:22,930 –> 00:12:26,050 that it provides you a way to bring into
215 00:12:26,050 –> 00:12:28,660 play algorithms that are well known and
216 00:12:28,660 –> 00:12:30,820 trusted in the projected equation area
217 00:12:30,820 –> 00:12:35,860 such as calles LSTD,LSPE,TD lambda all of
218 00:12:35,860 –> 00:12:38,290 those who work in the context of
219 00:12:38,290 –> 00:12:40,900 aggregation based on the fact that they
220 00:12:40,900 –> 00:12:42,580 can be generalized to semi norm projections

一个很有趣的事情就是这为投影方程提供了很多很有名的算法，比如 LSTD，LSPE 和 $TD(\lambda)$，所有这些算法都可以在之前的条件满足的时候对一个 semi-norm 投影方程使用

221 00:12:42,580 –> 00:12:45,580 so sweeping generalization
222 00:12:45,580 –> 00:12:47,890 of all these algorithms anything that
223 00:12:47,890 –> 00:12:51,070 you can do essentially with projected
224 00:12:51,070 –> 00:12:53,970 equations you can do with aggregation
225 00:12:53,970 –> 00:12:59,050 equations the aggregation context the
226 00:12:59,050 –> 00:13:02,250 only limitation really is the fact that
227 00:13:02,250 –> 00:13:07,540 Phi and D have to be have to have roles
228 00:13:07,540 –> 00:13:09,100 that are probability distributions
229 00:13:09,100 –> 00:13:11,380 that's the only restriction

概括一下就是，所有这些算法都可以用在投影方程上，你可以对他们进行状态聚合，唯一的一个限制就是 Φ 和 D 必须表示概率分布，只有这一个限制条件

230 00:13:11,380 –> 00:13:13,780 before the logically there is no there is no
231 00:13:13,780 –> 00:13:18,190 difficulty in using projected equation
232 00:13:18,190 –> 00:13:22,110 methodology to aggregation

也就是说用投影方程来求解聚合问题一点难度都没有

233 00:13:29,579 –> 00:13:32,110 this is a recent work and here's a
234 00:13:32,110 –> 00:13:37,930 reference for it a paper from 2012

这是一个最近做的工作，2012 年的文章

235 00:13:37,930 –> 00:13:42,550 it's also briefly discussed in bi9 my 2012
236 00:13:42,550 –> 00:13:47,050 book but not in great detail

这篇文章简洁地讨论了我在 2012 写的那本书，但是没有深入讨论

237 00:13:47,050 –> 00:13:49,509 this paper also describes the method of freeform
238 00:13:49,509 –> 00:13:51,579 sampling that I mentioned earlier in the lecture

这篇文章同样介绍了我在之前的课程提到的 freeform sampling 方法

239 00:13:51,579 –> 00:13:56,139 and so it's a broader reference
240 00:13:56,139 –> 00:14:00,629 on the material of of today's lecture

所以这篇文章在今天的课程中被引用的很广泛

(someone asking questions

问题：概率 d_{xi} 中那些元素是正数，那些元素是 0

回答：硬聚合中聚合状态 x 中的状态 i 对应的元素是 1，其他的是 0

241 00:14:04,110 –> 00:14:09,149 okay I need questions on this

有什么问题么

242 00:14:20,030 –> 00:14:23,870 that is very recent probability is not
243 00:14:23,870 –> 00:14:28,970 Arabic for the data like we have all
244 00:14:28,970 –> 00:14:31,360 stayed
245 00:14:33,369 –> 00:14:41,589 act politics or predation probability
246 00:14:44,319 –> 00:14:48,799 probability the act is not ever only for
247 00:14:48,799 –> 00:14:57,549 that yeah your question has to do with
248 00:14:57,549 –> 00:15:00,049 which ones of these probabilities are

249 00:15:00,049 –> 00:15:03,129 positive and which are zero yeah
250 00:15:03,129 –> 00:15:05,779 these probabilities in heart aggregation
251 00:15:05,779 –> 00:15:10,160 are always zero for States I outside the
252 00:15:10,160 –> 00:15:12,439 aggregate state so you have this group
253 00:15:12,439 –> 00:15:18,639 of state and D is 0 for X going outside
254 00:15:18,639 –> 00:15:22,489 and it could be positive for states
255 00:15:22,489 –> 00:15:25,489 inside the assumption here is that for
256 00:15:25,489 –> 00:15:27,230 every aggregate state all these
257 00:15:27,230 –> 00:15:31,839 probabilities are positive ok
258 00:15:38,010 –> 00:15:49,060 the fine matrix here no it's not
259 00:15:49,060 –> 00:15:53,020 identity it is it is the matrix that
260 00:15:53,020 –> 00:15:56,140 that involves the weights involves this
261 00:15:56,140 –> 00:16:00,670 probabilities along some diagonal if all
262 00:16:00,670 –> 00:16:05,560 the the probabilities are equal
263 00:16:05,560 –> 00:16:08,020 okay one over m let's say if m is the
264 00:16:08,020 –> 00:16:11,709 number of states in a good state then it
265 00:16:11,709 –> 00:16:15,430 becomes the identity matrix yes well
266 00:16:15,430 –> 00:16:17,470 even even that I'm not quite sure it may
267 00:16:17,470 –> 00:16:21,910 be okay if all the aggregate states have
268 00:16:21,910 –> 00:16:26,110 the same number of elements okay then if
269 00:16:26,110 –> 00:16:27,430 all the aggregate seeds have the same
270 00:16:27,430 –> 00:16:29,529 number of elements in all states within
271 00:16:29,529 –> 00:16:34,630 each one is equally likely then this Phi
272 00:16:34,630 –> 00:16:38,279 D is going to be the identity matrix
273 00:16:38,279 –> 00:16:42,520 otherwise it will be a matrix that name
274 00:16:42,520 –> 00:16:47,680 of nonzero weights okay I mean not
275 00:16:47,680 –> 00:16:50,430 non-uniform way it's non-uniform weights
276 00:16:50,430 –> 00:16:53,050 and the same thing happens here for this
277 00:16:53,050 –> 00:16:55,180 case except that some of the weights may
278 00:16:55,180 –> 00:17:02,339 be 0 it's a similar process okay
问题：硬聚合是投影的唯一情况么
回答：是的，剩下的都是 semi-projection(semi-norm)
279 00:17:14,439 –> 00:17:16,069 okay you're asked an interesting
280 00:17:16,069 –> 00:17:19,130 question is hard aggregation the only
281 00:17:19,130 –> 00:17:22,849 case for which you have a projection if
282 00:17:22,849 –> 00:17:24,890 you look at the proofs if you look at
283 00:17:24,890 –> 00:17:28,220 the analysis you'll see that it must be
284 00:17:28,220 –> 00:17:32,090 the only one proving this theorem right
285 00:17:32,090 –> 00:17:36,429 maybe it requires some formal argument
286 00:17:36,429 –> 00:17:41,900 but if you just calculate the Phi D and
287 00:17:41,900 –> 00:17:43,549 you look what has to happen you know you
288 00:17:43,549 –> 00:17:45,799 have a projection you see that that this
289 00:17:45,799 –> 00:17:48,650 must be the only only case I believe
290 00:17:48,650 –> 00:17:50,330 that under certain assumptions is the
291 00:17:50,330 –> 00:17:52,820 only case which gives you a projection
292 00:17:52,820 –> 00:17:54,950 but there are many many other cases that
293 00:17:54,950 –> 00:17:58,360 give you a seminar projection
294 00:18:08,039 –> 00:18:10,080 yes exactly
295 00:18:10,080 –> 00:18:18,070 yeah yeah yeah if if X consists of a
296 00:18:18,070 –> 00:18:21,640 number of those then the requirement is
297 00:18:21,640 –> 00:18:26,380 that peas are all positive and the
298 00:18:26,380 –> 00:18:29,049 others are 0 for such they have to be in
299 00:18:29,049 –> 00:18:32,010 card aggregation
asking completed)
300 00:18:38,470 –> 00:18:41,440 okay so so there's a close connection
301 00:18:41,440 –> 00:18:43,059 between this methodology of the
302 00:18:43,059 –> 00:18:45,879 aggregation in projection and it can be
303 00:18:45,879 –> 00:18:48,460 exploited very nicely in many different

304 00:18:48,460 –> 00:18:49,149 ways
这两种方法之间有很紧密的联系，并且可以推广到很多不同的方法
305 00:18:49,149 –> 00:18:52,509 I mentioned error bounds for projected
306 00:18:52,509 –> 00:18:55,059 equations this error bounds Terry over to aggregation
比如误差上界投影方程，计算聚合的误差上界
307 00:18:55,059 –> 00:18:58,179 I mentioned TD lambda L
308 00:18:58,179 –> 00:19:02,500 STD LSP for projected equations these
309 00:19:02,500 –> 00:19:15,789 can also be this method can also be be
310 00:19:15,789 –> 00:19:20,490 used for aggregation in the involve
311 00:19:20,490 –> 00:19:23,980 multi-step simulation okay from
312 00:19:23,980 –> 00:19:28,960 aggregate state two to regular state to
313 00:19:28,960 –> 00:19:31,149 aggregate state and so on over multiple steps
我之前提到的 LSTD，LSPE 和 $TD(\lambda)$ 投影方程都可以用来求解多阶段仿真的聚合问题，从聚合状态到原装胎，再到聚合状态，再到原状态
314 00:19:31,149 –> 00:19:34,509 it's it's easy to generalize all
315 00:19:34,509 –> 00:19:36,159 these algorithms to the aggregation
316 00:19:36,159 –> 00:19:39,159 context and this special case is very
317 00:19:39,159 –> 00:19:41,980 general as I mentioned earlier it sort
318 00:19:41,980 –> 00:19:46,029 of includes some of the basic examples
319 00:19:46,029 –> 00:19:55,570 useful examples of aggregation
他们很容易被应用到聚合问题中，这些例子也非常一般，我之前提到的例子都是比较基本而且有用的例子

1.4 DISTRIBUTED AGGREGATION I

320 00:19:55,570 –> 00:19:57,340 okay now let me go into the last topic for
321 00:19:57,340 –> 00:19:59,789 today's lecture
让我们来看看今天的最后一个话题
322 00:19:59,789 –> 00:20:02,440 which has to do with parallel computation distributed
323 00:20:02,440 –> 00:20:08,110 computation using aggregation
聚合问题的并行/分布式计算
324 00:20:08,110 –> 00:20:10,690 so we want to consider the composition of the state
325 00:20:10,690 –> 00:20:13,419 space and distribute solution of large
326 00:20:13,419 –> 00:20:16,389 scale discounted problems using hard aggregation
我们想要考虑的是大型折扣问题的硬聚合状态空间分解和分布式方案
327 00:20:16,389 –> 00:20:23,879 we have discussed already
328 00:20:23,879 –> 00:20:27,159 distributed value iteration whereby we
329 00:20:27,159 –> 00:20:29,100 can do value directions simultaneously
330 00:20:29,100 –> 00:20:33,309 for different states in fact the
331 00:20:33,309 –> 00:20:35,080 directions may be a synchronous and
332 00:20:35,080 –> 00:20:37,720 still the method will work
我们之前讨论过分布式值迭代，我们可以使用仿真对不同的状态进行值迭代，事实上迭代可以是同步进行的，而且这个方法能够工作
333 00:20:37,720 –> 00:20:40,120 we're considering something similar here we
334 00:20:40,120 –> 00:20:44,019 partition the the big original state
335 00:20:44,019 –> 00:20:47,740 space into subsets so how much identical
336 00:20:47,740 –> 00:20:49,659 ate space we cut it down into many
337 00:20:49,659 –> 00:20:50,720 pieces
我们考虑一个相似的问题，把一个大的原始状态空间划分为几个子集合
338 00:20:50,720 –> 00:20:53,210 and we have a parallel processor that
339 00:20:53,210 –> 00:20:56,030 will do value iteration on part of the state space
我们有几个并行的处理器，可以对一部分状态空间进行值迭代
340 00:20:56,030 –> 00:20:58,850 okay so that would be the
341 00:20:58,850 –> 00:21:02,180 way to do it and in fact with earlier
342 00:21:02,180 –> 00:21:04,190 theory that we're discussing the second lecture applies
我们在第二次课程已经讲过这个理论了
343 00:21:04,190 –> 00:21:08,540 however still each
344 00:21:08,540 –> 00:21:11,690 processor has to consider not only the

345 00:21:11,690 –> 00:21:14,810 states within its own subset but also
346 00:21:14,810 –> 00:21:17,270 the states in the other subsets
在这里每一个处理器不仅需要考虑一个状态在它所属的子集合，同时还要考虑其他子集合的状态
347 00:21:17,270 –> 00:21:20,420 because there may be transitions from his States
348 00:21:20,420 –> 00:21:23,300 into the states of other processors and
349 00:21:23,300 –> 00:21:26,060 they have to be taken into account when
350 00:21:26,060 –> 00:21:29,030 you write Bellman's equation or you do the value iteration
因为状态转移有可能从当前状态到其他处理器处理的状态，因此它必须考虑到这些事情，然后你可以写出 bellman 方程或者进行值迭代
351 00:21:29,030 –> 00:21:33,470 so the idea of
352 00:21:33,470 –> 00:21:36,170 aggregation within this context is to
353 00:21:36,170 –> 00:21:40,730 consider exact values for the local
354 00:21:40,730 –> 00:21:44,180 states and aggregate values for the other states
聚合的思路就是考虑当前状态子集合的值迭代和其他状态的聚合状态
355 00:21:44,180 –> 00:21:47,030 and these aggregate values
356 00:21:47,030 –> 00:21:49,220 will be computed by the other processors
357 00:21:49,220 –> 00:21:52,580 and sent over to each processor who will
358 00:21:52,580 –> 00:21:55,330 then do an approximate value iteration
359 00:21:55,330 –> 00:21:59,780 using exact values for its own aggregate
360 00:21:59,780 –> 00:22:05,750 values for the remainder of the world
这些聚合值被其他处理器计算，使用自己的状态进行精确值迭代，使用其他处理器的聚合状态值进行近似值迭代
361 00:22:05,750 –> 00:22:07,130 so this is the distributed value iteration scheme
这就是分布式值迭代方法
362 00:22:07,130 –> 00:22:10,790 we have a processor l associated
363 00:22:10,790 –> 00:22:14,360 with a subset of states as L
我们有处理状态子集合 l 的处理器 l
364 00:22:14,360 –> 00:22:17,990 and this processor maintains detail and the exact
365 00:22:17,990 –> 00:22:22,130 local cost J sub I for every original
366 00:22:22,130 –> 00:22:26,210 system state that is local within its own subset
这个处理器计算它所属的状态子集合中每一个原始状态的精确局部成本 $J(i)$
367 00:22:26,210 –> 00:22:30,590 and considered aggregate
368 00:22:30,590 –> 00:22:34,850 costs for the other subsets one number for each subset
考虑其他子集合的聚合状态的成本，每一个子集合都对应一个数
369 00:22:34,850 –> 00:22:38,420 so I need to maintain in
370 00:22:38,420 –> 00:22:44,420 my memory just the costs of my own state
371 00:22:44,420 –> 00:22:49,810 and one number for every other processor
所以我需要在我自己的内存中维护的只有我自己的子集合的状态的成本和其他处理器的成本
372 00:22:50,710 –> 00:22:54,830 so it maintains an aggregate cost within
373 00:22:54,830 –> 00:22:58,700 its own local states and it sets
374 00:22:58,700 –> 00:23:00,260 this aggregate cost to the other
375 00:23:00,260 –> 00:23:07,190 aggregate states or processors
所以一个处理器需要维护的聚合成本包括他自己的状态成本和其他处理器计算的聚合状态成本
376 00:23:07,190 –> 00:23:10,910 so how is this done processor L now the l-th
377 00:23:10,910 –> 00:23:15,620 processor updates J sub I and also the
378 00:23:15,620 –> 00:23:19,160 aggregate cost as follows it does a
379 00:23:19,160 –> 00:23:21,940 value iteration for its local states
380 00:23:21,940 –> 00:23:26,060 which involves the current local cost
381 00:23:26,060 –> 00:23:29,080 vector and an aggregate cost vector
382 00:23:29,080 –> 00:23:34,580 received by other processors
所以现在的第 l 个处理器需要更新 $J(i)$ 和聚合状态成本，对处理器的状态进行值迭代，包括当前局部成本向量和其他处理器算出的聚合成本向量
383 00:23:34,580 –> 00:23:40,520 so RK is the vector of aggregate cost vector of
384 00:23:40,520 –> 00:23:43,370 aggregate costs associated with the
385 00:23:43,370 –> 00:23:49,130 other professors at time k
所以 R_k 是其他处理器在时间 k 时计算的聚合成本向量
386 00:23:49,130 –> 00:23:52,430 and this mapping is one of those generalized

387 00:23:52,430 –> 00:23:55,430 mappings that I talked to heard in the
388 00:23:55,430 –> 00:23:58,670 class and it involves a form of
389 00:23:58,670 –> 00:24:01,580 Bellman's equation modified to
390 00:24:01,580 –> 00:24:05,090 accommodate aggregate cost

所以这个映射 ($H_l\left(i,u,J,R\right)$) 是一个一般性的映射，它包括被修正的包括聚合成本的 bellman 方程

391 00:24:05,090 –> 00:24:09,650 so this is the one stage cost this is the one
392 00:24:09,650 –> 00:24:14,860 stage cost associated with state I

这是状态 i 一个阶段的成本

393 00:24:14,860 –> 00:24:20,180 this is the expected cost to go to local
394 00:24:20,180 –> 00:24:26,360 states to J's within SL and get cost to
395 00:24:26,360 –> 00:24:30,080 Go for going into states into other subsets

这一项 (第二项) 是本集合的长期期望成本，这一项 (第三项) 是其他处理器计算得到的长期成本

396 00:24:30,080 –> 00:24:35,390 so it's an iteration that
397 00:24:35,390 –> 00:24:39,080 involves a much smaller factor the
398 00:24:39,080 –> 00:24:43,630 vectors of J and one number per
399 00:24:43,630 –> 00:24:47,350 other processor that sort of aggregates
400 00:24:47,350 –> 00:24:54,740 the the cost of aggregates the current
401 00:24:54,740 –> 00:24:57,290 cost value of the other processors

所以这个迭代只包含很少的元素，向量 J，其他处理器的当前成本

402 00:24:57,290 –> 00:24:59,660 and the interesting thing about this mapping
403 00:24:59,660 –> 00:25:01,790 is that it is a contraction mapping a
404 00:25:01,790 –> 00:25:04,670 souped norm contraction and therefore
405 00:25:04,670 –> 00:25:07,260 this iteration converges in
406 00:25:07,260 –> 00:25:09,540 converges to a unique solution but also
407 00:25:09,540 –> 00:25:11,280 converges as synchronously when
408 00:25:11,280 –> 00:25:13,740 implemented totally synchronously

一个很有意思的事情是这个映射是一个 sup-norm 压缩映射，所以这个迭代会收敛于一个唯一解，使用同步算法时也会收敛

409 00:25:13,740 –> 00:25:15,750 so each one of these processors computes
410 00:25:15,750 –> 00:25:18,210 whenever he wishes sends out aggregate
411 00:25:18,210 –> 00:25:22,250 values whenever he wishes that it seems
412 00:25:22,250 –> 00:25:25,710 unpredictably values from other
413 00:25:25,710 –> 00:25:28,260 processors combines them in within this
414 00:25:28,260 –> 00:25:30,330 iteration and even with the
415 00:25:30,330 –> 00:25:38,870 communication some health
416 00:25:45,880 –> 00:25:47,680 I'm concerned that somebody may be
417 00:25:47,680 –> 00:25:50,500 shooting at me maybe I'll just I'll just
418 00:25:50,500 –> 00:25:54,580 take it off okay how about that a new
419 00:25:54,580 –> 00:25:58,480 battery we don't really believe that I
420 00:25:58,480 –> 00:26:00,670 would say that since I'm wearing the
421 00:26:00,670 –> 00:26:03,640 microphone for the video I think we're
422 00:26:03,640 –> 00:26:07,420 fine I just talk a little louder okay
423 00:26:07,420 –> 00:26:11,410 okay can you take it off it's this one
424 00:26:11,410 –> 00:26:14,710 right okay okay maybe a matter of battery
425 00:26:14,710 –> 00:26:18,850 okay so oops that's not the
426 00:26:18,850 –> 00:26:19,270 problem
427 00:26:19,270 –> 00:26:21,010 it may be your problem your your
428 00:26:21,010 –> 00:26:28,480 microphone is is the problem okay how

话筒突然坏了，笑死了

429 00:26:28,480 –> 00:26:36,640 about now okay okay let me review here
430 00:26:36,640 –> 00:26:37,690 and maybe you can think of your questions

我再复述一遍，你可以想象有没有什么问题

431 00:26:37,690 –> 00:26:42,310 a parallel computing setting
432 00:26:42,310 –> 00:26:44,380 with a partition of the state space two
433 00:26:44,380 –> 00:26:49,000 subsets each processor controls the does
434 00:26:49,000 –> 00:26:52,030 value duration for for the states in the subset

把状态空间费城很多子空间，每个处理器对一个子空间中的状态进行值迭代，多个处理器构成并行计算

435 00:26:52,030 –> 00:26:56,740 and it does the value duration by

436 00:26:56,740 –> 00:26:59,440 maintaining local costs corresponding to

437 00:26:59,440 –> 00:27:02,050 the local state and aggregate costs that

438 00:27:02,050 –> 00:27:04,930 receives from other processors and also

439 00:27:04,930 –> 00:27:06,940 computes aggregate cost which it sends out to other processors

处理器值迭代时需要维护自己的子集的状态和聚合状态的成本，每一个处理器计算自己的子集的成本后通知其他处理器

440 00:27:06,940 –> 00:27:09,910 and this is the

441 00:27:09,910 –> 00:27:12,340 value duration and if you look at this

442 00:27:12,340 –> 00:27:14,320 expression here it's not hard at all to

443 00:27:14,320 –> 00:27:16,450 show that's a contraction mapping and

444 00:27:16,450 –> 00:27:19,060 has a unique fixed point

这就是值迭代，如果你看一下这个表达式，这一点都不难发现这是一个压缩映射并且由一个不动点

445 00:27:19,060 –> 00:27:20,860 now that unique fixed point of this mapping here that we

446 00:27:20,860 –> 00:27:23,230 compute this way is not the exact optimal cost

这个映射的不动点不是精确最优成本

447 00:27:23,230 –> 00:27:25,810 it is something that's approximate

这是一个近似解

448 00:27:25,810 –> 00:27:28,450 it is locally exact you

449 00:27:28,450 –> 00:27:31,480 might say but it uses approximate

450 00:27:31,480 –> 00:27:34,810 quantities from other processors

它的局部解是精确的，但是从其他处理器获得的数值是近似的

451 00:27:34,810 –> 00:27:37,210 so it's something that is related within some

452 00:27:37,210 –> 00:27:39,940 error bound of the exact optimal cost

453 00:27:39,940 –> 00:27:43,530 vector but it is not the optimal thing

所以这个解不是最优的，是存在误差上界的

454 00:27:43,530 –> 00:27:48,280 however this value Direction involves a

455 00:27:48,280 –> 00:27:50,860 lot less computation than if you were to

456 00:27:50,860 –> 00:27:53,500 do value iteration using exact quantities

这种值迭代减少了很大的计算量，如果与精确值迭代相比的话

1.5 DISTRIBUTED AGGREGATION II

457 00:27:53,500 –> 00:28:01,170 okay so this slide shows

458 00:28:01,170 –> 00:28:04,000 basically many other things I've said already

这一页展示了我之前说过的基础内容

459 00:28:04,000 –> 00:28:06,460 we can show that this iteration

460 00:28:06,460 –> 00:28:08,350 involves a sup norm contraction of modules alpha

我们可以看到这个迭代包括关于 α 的 sup-norm 收缩

461 00:28:08,350 –> 00:28:10,840 so it converges to the

462 00:28:10,840 –> 00:28:13,240 unique solution of this system of equations

它会收敛到这个系统的唯一解

463 00:28:13,240 –> 00:28:15,370 the system of equations that

464 00:28:15,370 –> 00:28:18,490 you are operating on this is not bellman

465 00:28:18,490 –> 00:28:21,190 equation for the original problem

这个方程不是 bellman 方程

466 00:28:21,190 –> 00:28:22,630 it's something else it's something approximate

这是一个近似方程

467 00:28:22,630 –> 00:28:31,090 and you can show that it's

468 00:28:31,090 –> 00:28:33,550 actually a contraction mapping using the

469 00:28:33,550 –> 00:28:36,630 fact that this is a the disaggregation

470 00:28:36,630 –> 00:28:39,670 probabilities a former probability

471 00:28:39,670 –> 00:28:41,760 distribution

使用分解概率分布 d_{li} 能够让 H 是一个压缩映射

472 00:28:46,930 –> 00:28:48,790 it's also possible to make a connection

473 00:28:48,790 –> 00:28:50,800 with an aggregate dynamic programming
474 00:28:50,800 –> 00:28:52,930 problem the difference being that the
475 00:28:52,930 –> 00:28:56,710 mapping H involves the exact values of
476 00:28:56,710 –> 00:28:59,170 J within the subset rather than the aggregate value
这个方程可以叫聚合动态规划，与 bellman 方程的区别是映射 H 包括精确值 J 而不是聚合值
477 00:28:59,170 –> 00:29:04,180 I mentioned earlier a
478 00:29:04,180 –> 00:29:06,250 synchronous implementation in an a
479 00:29:06,250 –> 00:29:08,440 synchronous version of the algorithm The
480 00:29:08,440 –> 00:29:11,170 aggregate costs may be outdated to
481 00:29:11,170 –> 00:29:13,090 account for communication delays between
482 00:29:13,090 –> 00:29:15,730 aggregate states so the process was
483 00:29:15,730 –> 00:29:18,310 making me communicating irregularly but
484 00:29:18,310 –> 00:29:20,290 no matter what the communication delays
485 00:29:20,290 –> 00:29:22,330 are as long as the processes keep
486 00:29:22,330 –> 00:29:26,200 computing this method will work
我之前提过同步版本的算法，聚合成本由于聚合状态通信延迟而过时，处理器一直在计算，但通信时间是不规则的，这个方法用在聚合问题上会工作
487 00:29:26,200 –> 00:29:27,640 and the proof of convergence is based on the
488 00:29:27,640 –> 00:29:28,900 general theory of a synchronous
489 00:29:28,900 –> 00:29:31,450 computation a single distributed
490 00:29:31,450 –> 00:29:33,700 computation that we described in the
491 00:29:33,700 –> 00:29:35,800 second lecture and there is a theory
492 00:29:35,800 –> 00:29:37,570 associated with that that you can find in the references
收敛性的证明基于我们第二次课程讲的同步计算的一般性理论，你可以在引用列表中找到它
493 00:29:37,570 –> 00:29:44,800 okay so we reached the
494 00:29:44,800 –> 00:29:48,520 end of a lecture there is we covered in
495 00:29:48,520 –> 00:29:52,350 these last two lectures a very large
496 00:29:52,350 –> 00:29:57,970 domain of problems and methods
这次课程结束了，我们在这两次课程中讲了非常多的问题和方法
497 00:29:57,970 –> 00:30:00,520 the problems are challenging approximation
498 00:30:00,520 –> 00:30:11,440 is is is is a difficult process with the
499 00:30:11,440 –> 00:30:14,650 outcome not very certain there are very
500 00:30:14,650 –> 00:30:17,110 few guarantees in this process you try
501 00:30:17,110 –> 00:30:19,120 to solve very difficult problems and
502 00:30:19,120 –> 00:30:21,520 sometimes we succeed sometimes you do not
这个问题很有挑战性，近似是很难的，而且没有理论保证，你会尝试解决很难的问题，但是有时候能成功有时候不能
503 00:30:21,520 –> 00:30:24,700 there are some interesting methods
504 00:30:24,700 –> 00:30:26,800 that you can choose from and you can try
505 00:30:26,800 –> 00:30:30,280 none of them is guaranteed to have success
你会尝试很多有趣的方法，但是这些方法中没有一个能够保证成功
506 00:30:30,280 –> 00:30:33,820 however you can try it and if
507 00:30:33,820 –> 00:30:36,040 you have some insight and the standard
508 00:30:36,040 –> 00:30:37,900 theory behind them you have a better
509 00:30:37,900 –> 00:30:42,480 probability of success
但是你在使用那些有标准理论支撑的方法时就很可能会成功
510 00:30:42,480 –> 00:30:44,920 the two signposts projected equation aggregation are
511 00:30:44,920 –> 00:30:46,990 related I think with an important point
我认为投影方程和聚合方程有关联是一个非常重要的点
512 00:30:46,990 –> 00:30:50,020 and they are suitable for different types of problems
他们很适合不同类型的问题
513 00:30:50,020 –> 00:30:53,560 there are all sorts of
514 00:30:53,560 –> 00:30:55,330 other issues that we have not covered
还有很多话题我们没有涉及
515 00:30:55,330 –> 00:30:56,890 and
516 00:30:56,890 –> 00:30:59,529 I've just in the last few slides and
517 00:30:59,529 –> 00:31:01,419 just went through some of them but both
518 00:31:01,419 –> 00:31:03,850 projections projected equations and

519 00:31:03,850 –> 00:31:09,010 aggregation involve a lot of interesting
520 00:31:09,010 –> 00:31:11,169 theory and also a lot of unanswered
521 00:31:11,169 –> 00:31:13,720 questions having to do with exploration
522 00:31:13,720 –> 00:31:17,350 having to do with with oscillations some
523 00:31:17,350 –> 00:31:19,529 of the questions that you asked are unresolved

我要用剩下的几页来讲一下，投影方程和聚合方程还有很多有趣的理论和没有回答的问题还没有涉及，比如探索，震荡等你们问过的事情

524 00:31:19,529 –> 00:31:26,769 so let me stop here and let
525 00:31:26,769 –> 00:31:30,570 you ask questions have any questions

下面你们有什么问题就可以问了

(question and answer

问题：你讲的聚合动态规划方法能求解 140 页的那个方程组么

回答：能，这个问题不是原问题，是与原问题相关的一个问题，求到的是近似解

526 00:31:41,880 –> 00:31:44,640 yes indeed you are asking whether this
527 00:31:44,640 –> 00:31:48,799 method whether whether this method
528 00:31:48,799 –> 00:31:51,809 solves this problem it will solve this
529 00:31:51,809 –> 00:31:54,720 problem it will solve this system of
530 00:31:54,720 –> 00:31:57,120 equations but that's not the exact
531 00:31:57,120 –> 00:31:58,679 system of equations that we want to
532 00:31:58,679 –> 00:32:01,850 solve it's only another system related
533 00:32:01,850 –> 00:32:04,559 conceptually to the origin no but the
534 00:32:04,559 –> 00:32:06,000 problem is approximately an
535 00:32:06,000 –> 00:32:07,799 approximation in the solution is also an
536 00:32:07,799 –> 00:32:10,530 approximation to what we want to solve
537 00:32:10,530 –> 00:32:13,640 the optimal cost factor

问题：为什么映射 H 是一个压缩映射

回答：没听懂，得去看看压缩映射的定义，说是很简单

538 00:32:21,090 –> 00:32:24,030 one day there's a website and the other
539 00:32:24,030 –> 00:32:27,200 way the rest of the other states
540 00:32:27,200 –> 00:32:32,300 difficult to think about why there's
541 00:32:35,460 –> 00:32:38,619 [Music]
542 00:32:51,760 –> 00:32:55,429 you okay so why is this mapping a
543 00:32:55,429 –> 00:33:01,610 contraction mapping you simply verify
544 00:33:01,610 –> 00:33:03,620 the definition of a soup norm
545 00:33:03,620 –> 00:33:06,980 contraction in other words you take take
546 00:33:06,980 –> 00:33:12,559 two vectors JR and J bar R Bar and form
547 00:33:12,559 –> 00:33:14,990 the difference and then the fact that
548 00:33:14,990 –> 00:33:16,789 these are probability distributions is
549 00:33:16,789 –> 00:33:19,250 very critical and there is also an alpha
550 00:33:19,250 –> 00:33:23,059 term here and this term is going to drop
551 00:33:23,059 –> 00:33:25,820 out from B from from the from the
552 00:33:25,820 –> 00:33:27,950 difference in what's left is weighted by
553 00:33:27,950 –> 00:33:29,570 alpha and multiplied by probability
554 00:33:29,570 –> 00:33:33,679 distributions and and and okay there are
555 00:33:33,679 –> 00:33:35,630 also these distributions in here the
556 00:33:35,630 –> 00:33:36,650 fact that these are probabilities
557 00:33:36,650 –> 00:33:38,270 tribution and these are probability
558 00:33:38,270 –> 00:33:40,460 distributions is what makes the proof go
559 00:33:40,460 –> 00:33:44,110 through I think I can give a
560 00:33:44,110 –> 00:33:46,159 mathematical proof more easily than I
561 00:33:46,159 –> 00:33:48,710 can give you a conceptual proof as to
562 00:33:48,710 –> 00:33:52,990 why this map is a contraction mapping

回答：直接说结论了，所有的方法都是硬聚合与代表性聚合的基础上发展来的，也有可能是这两者的综合

563 00:34:01,010 –> 00:34:03,710 yeah I understand that
564 00:34:03,710 –> 00:34:06,980 three is a special
565 00:34:06,980 –> 00:34:11,590 for what's the relationship between
566 00:34:11,590 –> 00:34:15,450 I think

567 00:34:26,679 –> 00:34:29,090 okay the question is the following I
568 00:34:29,090 –> 00:34:32,510 have this aggregation scheme where every
569 00:34:32,510 –> 00:34:34,879 aggregate state is a subset of the
570 00:34:34,879 –> 00:34:37,460 original but it is possible also that
571 00:34:37,460 –> 00:34:39,800 this aggregate states do not exhaust the
572 00:34:39,800 –> 00:34:42,949 same or the entire space and I argued
573 00:34:42,949 –> 00:34:45,379 that with these destructions this
574 00:34:45,379 –> 00:34:49,369 contains special cases all the previous
575 00:34:49,369 –> 00:34:54,260 schemes so for case for example three I
576 00:34:54,260 –> 00:34:59,660 have single a single state to every
577 00:34:59,660 –> 00:35:03,109 representative feature state in example
578 00:35:03,109 –> 00:35:08,180 one which is hard aggregation then I
579 00:35:08,180 –> 00:35:11,869 have the sets again disjoint sets but we
580 00:35:11,869 –> 00:35:15,050 exhaust the entire space example two is
581 00:35:15,050 –> 00:35:17,390 hard aggregation is a special case of
582 00:35:17,390 –> 00:35:21,710 hard aggregation because I am exhausting
583 00:35:21,710 –> 00:35:24,500 the space of features and every state
584 00:35:24,500 –> 00:35:27,710 maps into a unique feature so the states
585 00:35:27,710 –> 00:35:29,750 that correspond to any subset of
586 00:35:29,750 –> 00:35:35,240 features is some subset over here but
587 00:35:35,240 –> 00:35:37,970 all of these subsets exhaust the entire
588 00:35:37,970 –> 00:35:39,710 space it's just that the partition he is
589 00:35:39,710 –> 00:35:43,130 quite irregular it's the partitions
590 00:35:43,130 –> 00:35:45,200 regular in this feature space but it is
591 00:35:45,200 –> 00:35:48,260 irregular in the state space so this is
592 00:35:48,260 –> 00:35:50,930 hard aggregation in the two extreme
593 00:35:50,930 –> 00:35:53,180 cases are hard aggregation and represent
594 00:35:53,180 –> 00:35:57,740 all states exhausted representative
595 00:35:57,740 –> 00:36:01,730 States single state eigenstate so it's
596 00:36:01,730 –> 00:36:03,770 it's covering these two base cases and
597 00:36:03,770 –> 00:36:06,430 everything in between
问题：所有的方法都基于仿真，那样会花费很长时间，一般需要多久
回答：取决于你，看你想让仿真运行多久
598 00:36:08,980 –> 00:36:26,029 yes how long is enough yeah okay okay so
599 00:36:26,029 –> 00:36:27,980 the question is all of these methods
600 00:36:27,980 –> 00:36:31,240 require a great deal of simulation and
601 00:36:31,240 –> 00:36:33,559 potentially you don't have enough time
602 00:36:33,559 –> 00:36:35,809 for simulation what's enough time it
603 00:36:35,809 –> 00:36:37,369 depends on you how much patience you
604 00:36:37,369 –> 00:36:40,160 have if you are willing to wait for a
605 00:36:40,160 –> 00:36:43,730 week Vence one thing if you're ready to
606 00:36:43,730 –> 00:36:46,400 only for a day to another but if you are
607 00:36:46,400 –> 00:36:47,839 really pressed for time and you want an
608 00:36:47,839 –> 00:36:49,130 answer in five seconds
609 00:36:49,130 –> 00:36:51,019 then probably it's not going to be
610 00:36:51,019 –> 00:36:54,589 enough it there are possible there are
611 00:36:54,589 –> 00:36:57,740 cases where enough to have enough
612 00:36:57,740 –> 00:36:59,630 simulation time you have to live a
613 00:36:59,630 –> 00:37:02,690 hundred years and so you should avoid
614 00:37:02,690 –> 00:37:04,549 methods like that
615 00:37:04,549 –> 00:37:09,170 they it's always part of the art nothing
616 00:37:09,170 –> 00:37:11,299 is guaranteed there are no failsafe but
617 00:37:11,299 –> 00:37:13,940 there's no failsafe advice here you have
618 00:37:13,940 –> 00:37:15,799 to look at your problem you have to look
619 00:37:15,799 –> 00:37:18,140 at your options you have to look at what
620 00:37:18,140 –> 00:37:20,630 environment you're operating in in some
621 00:37:20,630 –> 00:37:22,670 cases you may need an answer within five
622 00:37:22,670 –> 00:37:25,670 seconds if you have real-time operation

623 00:37:25,670 –> 00:37:28,069 and you have to have a decision ready in
624 00:37:28,069 –> 00:37:31,130 five seconds then you may have to do
625 00:37:31,130 –> 00:37:33,079 your simulation within five seconds and
626 00:37:33,079 –> 00:37:35,239 then perhaps simulation may not be the
627 00:37:35,239 –> 00:37:38,299 right idea maybe a simpler solution may
628 00:37:38,299 –> 00:37:41,329 a simpler approach we may have to be
629 00:37:41,329 –> 00:37:44,769 adopted so I can't answer your question
630 00:37:44,769 –> 00:37:49,670 how much time can we afford because that
631 00:37:49,670 –> 00:37:50,839 depends on your personal circumstances
632 00:37:50,839 –> 00:37:53,410 and also how you apply these methods
633 00:37:53,410 –> 00:37:56,180 these are very complicated methods I
634 00:37:56,180 –> 00:37:57,529 don't want to fool you that they are
635 00:37:57,529 –> 00:38:00,440 easy to apply complicated because of the
636 00:38:00,440 –> 00:38:03,019 many anomalies because of oscillations
637 00:38:03,019 –> 00:38:06,739 explorations what have you also because
638 00:38:06,739 –> 00:38:08,539 they require a lot of computation time
639 00:38:08,539 –> 00:38:11,569 in some problems they may work but in
640 00:38:11,569 –> 00:38:14,890 some other problems they may not
641 00:38:16,080 –> 00:38:20,760 you have to make a judgment for yourself

问题：一些方法会用到仿真，比如投影方程什么的，那么如何选择一个长轨迹仿真还是很多短轨迹仿真呢

回答：从探索和试错的角度出发，一个长轨迹效果不如多个短轨迹，特别是探索，实际上这个东西还是得根据你对问题的认识来选择，具体问题具体分析，没有一个统一的理论指导选择

642 00:38:56,250 –> 00:38:59,070 okay so the question is the following
643 00:38:59,070 –> 00:39:02,980 let's say we consider some method that
644 00:39:02,980 –> 00:39:07,360 requires simulation like the simulation
645 00:39:07,360 –> 00:39:09,460 based solution of projected equations I
646 00:39:09,460 –> 00:39:12,130 mentioned one possibility to use a
647 00:39:12,130 –> 00:39:14,710 single very long simulation trajectory
648 00:39:14,710 –> 00:39:17,410 and also other possibilities of using
649 00:39:17,410 –> 00:39:19,240 shorter trajectories many shorter
650 00:39:19,240 –> 00:39:23,950 trajectories which one to use I think
651 00:39:23,950 –> 00:39:27,220 that it has to do with exploration of a
652 00:39:27,220 –> 00:39:31,300 lot and also it's a matter of trial and
653 00:39:31,300 –> 00:39:35,560 error I would say that if one long
654 00:39:35,560 –> 00:39:37,990 simulation trajectory does not work well
655 00:39:37,990 –> 00:39:40,390 it's unlikely that five trajectories or
656 00:39:40,390 –> 00:39:42,220 four trajectories will work well you
657 00:39:42,220 –> 00:39:45,360 probably need many many shorter
658 00:39:45,360 –> 00:39:47,200 trajectories in order to deal with a
659 00:39:47,200 –> 00:39:53,470 problem of exploration we so so the
660 00:39:53,470 –> 00:39:55,060 again there is no unique answer to your
661 00:39:55,060 –> 00:39:58,120 questions to your to this particular
662 00:39:58,120 –> 00:40:01,450 question you need to you need to have
663 00:40:01,450 –> 00:40:04,710 some insight into your problem and and
664 00:40:04,710 –> 00:40:08,440 and develop a better insight on which
665 00:40:08,440 –> 00:40:16,500 method to use some other questions
666 00:40:24,090 –> 00:40:28,900 okay I think we're done for today and we
667 00:40:28,900 –> 00:40:31,600 have one more election Friday which has
668 00:40:31,600 –> 00:40:35,470 to do with things that we haven't
669 00:40:35,470 –> 00:40:37,330 covered so far and I'm not quite sure
670 00:40:37,330 –> 00:40:39,910 what they will be
671 00:40:39,910 –> 00:00:00,000 [Applause]