

订阅DeepL Pro以编辑此演示文稿。  
访问[www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document)，了解更多信息。

上来吧 是的 要么是真的 要么不是 好吧，我们可以开始了吗？ 好，大家都好吗？ 很好 忙着评估吗 很忙 很忙 你有很多课程 好吧，希望这一个不会增加那么多 在工作量方面，但是，是的，我们拭目以待。 还有几周或三周呢 好吧 评估截止日期 Right. 评估截止日期 所以，在其他截止日期之外，你们还有一些时间来做这件事。 上周有什么问题吗？ 嗯，任何投诉，有什么 你无法理解？ 有啊 到目前为止，你能掌握这些概念吗？ 你能接受吗？ 能 好的 慢一点 慢一点 有时候我想做笔记 好吧 - Yeah. 好的 好的 好的 好的 我会的 随便，随便 无论如何，我今天要做的就是稍微回顾一下我讲过的内容。 如果你还有什么需要问我的，尽管开口。 好的 今天将介绍多阶段随机编程。 嗯，概念可能有点复杂，难以理解，但我会尝试用构建两阶段模型的方法来构建它，也就是从一些天真的方法开始。 我们先尝试那些我们知道行不通的方法，然后了解它们为什么行不通。 然后，我们就开始构建相关概念。 好的，那么，嗯，首先，我将回顾一下，嗯，我们到目前为止所处理的问题。 我们从确定性建模开始。 我想大家对确定性优化都很熟悉了。 没错。 我们举了一个例子，涉及到发电扩容规划问题。 你能说说吗？ 能 能 我们举了个例子，涉及到发电扩建规划问题。 好的，我们试图将安装成本和发电成本降到最低。 然后，我们对发电机的数量有一些限制，以满足发电能力和需求。 因此，所涉及的成本是安装时间 x I t 的成本总和 I in I summation t in t。 嗯。 再加上发电成本 再乘以发电量。 这就是目标函数 成本最小化目标 我们有一些约束条件。 第一个约束条件是可用性。 所以我们说 呃 设置数量必须小于或等于可用数量 对于人工智能中的所有 I 在整个时间跨度内，我们在T上设置了屎的总和，而这应该小于我们投资组合中的数量。 好的 这就是发电机的可用性。 然后我们谈到了容量限制。 因此，我们的发电量应小于或等于 t 时间的发电量乘以 t 时间之前设置的发电机累计数量。 没错。 所以 x 我拆掉了小于或等于 t 的发电机，这就是输出能力。 然后我们说，我们需要满足需求。 所以这是个非常简单的问题，因为一切都是给定的。 因此，你有一个发电机组合，你部署它们，用它们发电，然后满足需求。 为了满足需求，所有发电机的发电量。 应该等于t时刻的需求量，所以这就是需求的满足。 对 这就是我们的确定性模型 我们讨论了确定性模型可能出现的各种问题。 如果成本不同呢？ 如果产能不同呢？ 如果需求不同呢？ 所以，当它们不同的时候，我们展示了一些例子，说明当它们不同的时候，实际上会使解决方案大相径庭。 没错。 我们还谈到了什么需要主动，什么需要被动。 所以涉及到不同类型的决策。 因此，你可以主动设置。 但我们不能被动地进行设置，因为设置需要很长时间才能完成。 因此，你不能等风量大了。 然后说我现在开始建造风力发电机。 这将在五年后发生。 因此，有些决策需要主动，有些决策需要被动。 正因为如此，我们建立了两阶段随机编程模型的概念。 于是，我们开始增加不确定性。 我们创建了两阶段随机程序。 在第一阶段，我们说： Um summation of I summation of d c I'd s exit, so that will be generated installation fixed plus. 然后我们对固定设置做出反应。 所以我们要设置。 首先，我们将固定设置作为输入，然后做出反应。 这就是我们所说的追索。 因此，我们要采取一些追索行动。 这样，我们就能最大限度地降低追索行动的平均成本，从而使追索行动考虑到第一阶段的决策和规划中的不确定性。 没错。 然后我们有了第一阶段的约束条件，很明显，对于所有的 I，第一阶段的约束条件保持不变 x I t over t 小于或等于 f I。 然后我们有一系列属于这个模型的第二阶段，呃，第二阶段约束条件。 Q 对。 所以要记住，这不仅仅是一个函数，而是一个完整的模型。 这个模型就是求助模型。 它规定了对你在第一阶段所作所为的反应。 好的 我们说的Q是这样的 我们说哪里？ Q本身就是一个模型 这涉及到最小化 第二阶段成本 并受到一些第二阶段的约束，这是输出能力的约束。 她撕毁旧城区小于等于茶和总结为什么它超过t等于TT。 对 这就是简单的两阶段追索模型 Y 追索。 这是求助函数。 这是对你在第一阶段所做事情的反应。 这种反应受限于你有多少可用的产能，以及你需要满足的需求是什么。 没错。 所以，这就是它的全部内容。 但也有一些东西，我没有和你一起去，在这个层面上的严谨性，在这个意义上，如果我们看看这个纯粹从概率的角度来看。 这只是 一个概率论的概念，它是一个函数在一个分布上的期望值。 所以从技术上讲，应该是这样的。 对 这个 这应该是对任何分布的期望值，无论它是连续分布还是离散分布。 但我们正在做的，至少在这门课程中，是从基于情景的角度来工作的，也就是我们有离散结果的情景，每个情景都被分配了一个概率。 那么 那么 是预期值还是预期概率？ 所以是预期值。 所以它是在概率 p 的情况下 q 的预期值。 所以，如果你想搜索某个不确定性函数的期望值，比方说，你知道它就是不确定性函数乘以不确定性概率的结果，对吧。 那么在一个无限的跨度上，你能不能一直走下去？ 可以啊 所以你试图得到分布情况 你要对分布进行积分 找出曲线下的面积 对吧 试着在这里找到它 沿着这个方向一直走 无论不确定性的范围在哪，都是正确的。 因此，从技术上讲，这应该是真实模型和真实模型。 它的问题在于包含了无限多的结果 对 而且很难求解 你如何求解一个模型的积分 一个本身就是优化模型的函数的积分？ 对 所以这一切都很复杂 所以我们做了一件有效的事 纯粹从严谨的角度来说 我们说 我们说，让我们定义 而不是，而不是，呃。 EPP。 让我们定义一个离散的近似值，这个近似值依赖于 n 个场景和离散的结果。 然后我们将计算 这个分布的期望值 成本函数的分布 那这是如何改变的？ 这使得。 一个非常简单的。 求和 这让一个非常复杂的积分变成了一个简单的求和，对吗？ 这就是我们要找的函数 S的倍概率 对吧 所以这就是它的起点 如果我们从第一性原理出发，我们可以研究任何分布。 它可以是均匀分布 也可以是正态分布 不管它是什么 我们说不是 但我们的数据不是连续的 它不是无限的 所以我们用这些数据来近似这个分布 这些数据是以情景的形式出现的 这些情景是不确定性的离散现实 没错 所以这实际上已经是 P 的近似值了。 我们说，好吧。 现在的问题是，这能帮我解决之前的一个重要问题，那就是实现的无限性。 现在我有了有限的实现。 但这又如何呢？ 我如何估算这个概率？ 我有数据，但没有概率。 我有数据，但没有概率。 所以，我有的只是历史数据中的情景数据。 无论过去发生了什么，我都不知道未来发生的可能性有多大。 我试图规划未来。 使用过去的数据，并从这个角度估计概率几乎是不可能的，对吗？ 所以我们说，我们需要一个进一步的近似值。 我们称之为大数定律。 在大数定律中，我们知道真正的平均值 实际上等同于 无限样本的平均值 没错 所以我们取一些数据x来求取平均值。 取平均值 如果样本是无限的 渐近地会趋向于平均值 对 这就是大数定律的概念。 我们说过，我们现在可以把这个概念翻译过来。 因为如果我们能把这个概念转化到这里，那么发生的事情就是我摒弃了这个估计概率的概念，因为所有东西都是1大于n，对吧。 因此，我们建立了所谓的样本平均近似，即样本平均近似。 因此，样本平均近似值，也就是我们所说的 SAR 模型，就取代了这个。 成本函数的离散分布。 用 近似它的求和 s 元素的 SN 一个超过 n. 记录的每一个实现。 好了，我们尝试近似，然后我们说的是。 实际上 当 n 趋于无穷大时 这应该趋向于 没错 对不起 这个 Sorry. 没错 Right. 所以当n趋向于无穷大时 我将复制真正的原始分布 这是基于大数定律的分布 这就是我们提出样本平均近似概念的原因 从概念的角度来看。 它显然依赖于所有其他统计工具。 但从实施的角度来看，这让事情变得简单多了。 除了概率相等之外，与之前的模型没有什么不同，对吧？ 但是，我们不能随心所欲地实现样本平均近似，因为它必须依赖于这个概念。 没错。 如果我说，好吧，我做抽样平均近似，问题就来了。 我生成三个方案，然后尝试求解模型。 这样做的有效性如何？ 这完全没有复制我想要复制的东西。 只有当你有非常大的样本时，复制才是有效的，对吗？ 所以 我们说，为了分析样本平均近似，我们需要两个不同的指标。 第一个是收敛性。 所以我们需要了解模型的收敛性。 第二个指标是一些性能，一些性能分析。 那么在收敛性方面，我们在寻找什么呢？ 我们需要了解模型的性能何时趋于稳定。 这就是目标函数。 因此，你经常会看到这样的情况。 不稳定，然后开始稳定。 没错。 所以我们想知道我们的样本量应该有多大，才能使 SA 成为原始模型的有效近似值。 没错 你可以有这样的趋势 你也可以有这样的趋势。 所以并不需要减少。 只是它需要在某一点上稳定下来。 一旦稳定下来，你就明白了。 什么时候？ 你的样本平均近似是有效的。 那为什么不把样本量越做越大呢？ 你会遇到可操作性问题，对吧？ 因为约束条件的数量在不断增加，所以不可能求解出模型。 所以，就样本平均近似而言，我只需要把你带回到模型求解的问题上来。 好的。 我们说过两阶段模型，而两阶段模型变成了样本平均近似法。 两阶段模型也有一个确定性等价模型。 这个等效模型就是你在 Gams 中实际编码和求解的模型。 对不对？ 这只是求和。 呃，I t c I t 设置退出。 还有 嗯，现在，我只是保持它的概率。 每种情况下的生成成本乘以生成的数量 没错 所以，我把第二阶段的追索权行动与情景做了索引，以显示它对情景的依赖性。 然后，我这里有的只是第一阶段的约束条件。 和第二阶段。 第二阶段 约束。 和 Sdn 一样，对所有情况都有效。 这就是可操作性的问题所在。 如果我不断增加 n 不断增加样本量 就会不断增加约束条件的数量 因此，第一，我不确定是否可行。 其次，我也不确定模型的可解性。 所以随着模型越来越大，找出可行的点也就越来越难。 对吧 目前都清楚了吗？ 都清楚了 第二点及其收敛性 还有性能分析 对 - 对 - Yeah. Yeah. 这是测试模型的重要部分 所以第一点是收敛性，也就是测试近似质量。 对 第二部分是你的决策是如何执行的。 所以，我需要知道我的处方是否足够好。 要做到这一点，我需要生成各种方案。 我需要用机器学习的方法进行比较。 先训练，再测试，对吗？ 因此，我们有两种类型的性能分析。 一种是样本内分析。 样本内性能分析。 另一种是样本外。 对 因此，样本内分析是根据你所掌握的数据对你的决策进行绩效分析。 而样本外分析则是分析你在没有数据的情况下做出决策的绩效。 没错。 所以，假设任何样本分析都是这样进行的。 那么生成 作为一个样本。 求解模型。 确定第一阶段的决策。 然后测试 第二阶段的最佳性能。 那么到底是什么呢？ 我们试图从模型中获得的是第一阶段的决策。 因为第二阶段会根据实际情况、不确定性的实现而改变。 没错。 所以，我们想要得到的是第一阶段的决策，以及它是如何执行的，我们是如何在不同的不确定性和现实情况下，积极主动地修复、执行决策的。 因此，事实上，假设你解决了你生成的模型，比方说 X 星。 你想测试 Q. X 娃娃的性能。 这些 属于你的原始样本。 对 那又怎样？ 你能做什么样的测试？ 你可以计算这个资源的平均值。 你可以计算记录的方差。 举例来说，如果我想计算记录的平均值，可以是这样的：n s s n q。 这就是平均值。 方差是这样的 实际值减去平均值的平方，对吧。 所以你可以计算方差。 你可以计算变异系数，无论你想计算什么指标。 但这给你的是，第一阶段决策在样本上的表现如何？ 好的。 现在，样本外样本外的情况有点不同。 所以在样本外，我们要做的就是说。 嗯，所以我从这里开始。 修正第一阶段的决定 对 生成 而样本外集合 所以，样本外集合意味着，如果 S 是。 这个 就是这个意思 对不起 意思是 S 不是 s n 的元素 对 这就是样本外的意思 这意味着样本外集合中的任何东西都不属于我们训练模型的原始集合。 这就是我们要解决模型问题的原因，我们要对没有数据的事情进行规划，对吗？ 我们有历史数据。 过去发生的事情在未来不会一模一样地发生。 所以我们接受这一点。 我们尝试生成新的情景，并观察我们在这方面的表现如何。 没错。 在制定性能指标方面，它们与此非常相似。 举例来说，样本外将是一个过度。 比方说，样本外的结算和质数，比方说，属于样本外的 s 和质数 q 的 n 次质数求和，再加上第一阶段的 v s。 所以这就是样本外的平均值，也是一样的。 你可以计算样本外性能的方差等。 现在一切都清楚了。 以上就是我们上周介绍的内容，但我们还介绍了几个非常特别的随机编程指标。 其中之一就是完美信息的期望值。 我今天要补充的第二个指标是，我上传了上周的讲义，在测试部分再次添加了这一概念，你可以看看这个。 Um in terms of this. 所以，我们有我们的随机编程。 我们先给它们起个名字，因为我们需要有一些界限。 因此，随机编程模型SP只是这个呃最小化C T S退出加上嗯一些S的S N P S Q X V S和受的东西。 对 这就是随机编程模型 然后我们制定了一个模型，说如果我们已经知道我们有一个甲骨文，它知道将要发生的情况，对吧。 以及即将发生的不同情况。 我们创造了一个所谓的 "静观其变 "问题 基本上就是这样 嗯 总结 嗯，超过S。 我坐在X S加。 嗯。 对不起 加q x v s。 右。 所以我们能改变一切 因为我们有完美的神谕 所以我们可以提前计划 我们知道会发生什么等等 因此，我们说的第一件事就是，让我们试着找出 SB 和 AWS 之间的关系。 例如，SP 的值是大于还是等于 w，还是小于 w？ 你觉得呢？ 所以，如果你只看这两个模型。 S p 会更大，对吗？ 为什么？ 为什么？ 好吧 说得好 那让我来吧 好吧 再大点 更大的权利。 就是这部分 还有另一面，如果你仔细想想，W S实际上只是SP Y的一个更受约束的版本，所以我可以通过这样做把SP转换成W S。 然后我移动 移动这个 移到外面 这样我就添加了一个约束条件 x I t s 的质数等于 x I s. 对于 s n 的所有 x 质数 s 元素 对吧 你看到了吗，这与 SP 模型完全相同，只是使用了 x 在不同情况下可变的概念。 所以我说的是，我只是复制了 WS，然后我说在所有情况下，设置必须是相同的。 这就产生了生产力。 所以，SP 就是 ORS，但更有约束性。 当限制更多的时候，最优值总是会更高。 好吧，因为这是一个最小化模块。 所以，我们还是把SP放回原位吧。 这个 你说的 "更多约束 "是指超过法国还是？ 是更多 它只是有更多的限制。 所以，比如说，嗯，如果我把这个已经包含了一些约束的 W s 模型，再加上这个约束，我再加上一个约束，说 x I t s prime 应该等于 x I t s for all s s prime。 你是说还没有数学约束的实际约束？ 没有实际约束 对 可行性空间缩小了 对 所以如果我加上这个 就能得到这个模型 对吧 因为这个模型在所有情况下都有固定的出口。 因此，无论在哪种情况下，你都能做出完全相同的决定。 在这里，我有各种不同的情况，但我们说的是在所有情况下都必须是一样的。 这就等于说这是一个固定的决定，对吗？ 因此，在这一部分，我们可以确定 SB 将大于或等于 ORS。 好的 就这样 现在我们来看另一个相当简单的模型，我们称之为 EV 模型，即期望值问题，就是这样。 嗯。 AWS有更多的约束条件，所以我就跟你说不清楚了。 是的，没错。 嗯，不，不。 S3有更多的约束条件 所以有一种方法可以通过添加更多的约束条件将W转换成SP，好吧。 对，没错 我们来定义一个简单的模型 数据的期望值。 我们称之为期望值模型 由此我们可以得到一个额外的关系。 所以呢？ 我们认为呢？ EV 和 AWS 之间有什么关系？ EV 是一个确定性模型，对吗？ 但我输入的数据是我所掌握的所有数据在各种情况下的平均值，对吗？ 这不是一个两阶段模型。 这是一个输入单一数据点的确定性模型。 这就是数据点。 那么我们怎么想？ 电动汽车会小于还是大于 AWS？ 是的 是的，应该会小一些，因为我们只是针对一种情况进行规划。 所以，如果我们要规划多种方案，就必须在所有方案中保持可行性。 没错。 所以显然我们的成本更高。 我不可能只针对其中一个方案进行规划，这样会产生更高的成本。 但因为我现在要保护系统免受更多可能发生的情况。 我在保护系统免受单一可能性的影响。 是的。 另一种观点认为，只有一种情况，这种情况的概率是 1。 而你把这个模型切成了电动车，对吧。 从这个角度来看 我们知道 AWS 大于或等于 EV。 最后，如果我们看一下这个模型，我们会说，让我们从中生成一个最优解。 因此，我们称之为 x e star。 然后计算这个模型。 顺便说一句，你会在文献中看到很多这样的模型。 这就是我们现在要计算 x I t e star 的模型。 再加上 q x I t，但 X，让我开始求和 p s s v s，受一些约束。 好的 我觉得这就像是期望值问题的期望值 你正在求期望值问题的解。 然后计算给定解的平均值。 对 两次是一个过程 对 找到一个 你找到一个，然后解决它，然后再求解。 在某种程度上，你是在用两步法 没错 但最主要的是，我们也可以找到一个边界。 比如说，我们比较一下V和SP。 我们会怎么想？ 夏娃和 SP 非常相似。 唯一变化的是 x，对吗？ 我把 x 固定为某个值。 无论如何，这都是一个 22 步的过程。 我从某处获取了 X，但我把 X 固定为某物，对吗？ 那么我们怎么想？ SP 是否会大于或等于 EV？ 所以，只要想一想就知道了。 在第二个过程中，我们将 X 固定为某个东西，而这个东西可能不会给你带来最小值。 它不可能给你一个更好的解决方案。 是的。 因为它是 C 的一个可行解，我们把它固定在一个可行解上。 它不可能给出比SP更好的解。 没错 如果我们更进一步 你会发现 什么 什么？ 我们不 我们不需要在这里走得更远。 我想我们已经确定了 好吧 Right. 这不是基于游泳的系统吗？ 反正也不是小数目 好吧，我们假设它是最有效的。 会相等吧？ 是的，不可能少，因为一个是最小化问题。 所以，如果存在的话，就永远不会少。 如果存在 "少"，它就应该是速度的最佳值。 没错 所以我们现在有了这些边界，甚至大于或等于SP，大于或等于Wait，大于或等于EV。 我们在上周的课堂上提出了一个指标，那就是VPI。 即完美信息的期望值。 让我在这里写一下。 夏娃大于或等于 ESP，而 ESP 大于或等于 W S。 我们在上一堂课中提出了一个指标，那就是完美信息的期望值。 我们称之为 EVP 对吧 嗯 预期值 完美信息的期望值 对 我们说EVP就是ESP减去WS 对吧 WS是一个概念 有一个甲骨文 在那里你预测。 所以你有完美的信息，对吧？ 拥有完美信息的价值是什么？ 当你没有完美信息时，你会得到什么价值？ 这两者之间有什么区别？ 还有 现在我要定义一个新概念，叫做 VSS，也就是随机解的价值。 这一点很重要，因为当我们对随机编程建模时，我们想知道的是，不确定性是否有建模的价值？ 对不对？ 如果不确定性不是必需的，那就忘了随机编程吧。 只要用平均值作为输入求解确定性模型，就不会有太大损失，对吧？ 这就是 夏娃 减去 ESP 夏娃是什么？ 夏娃是我们求解的模型，第一阶段的输入决策是确定性模型，带有平均数据的确定性模型。 有平均数据的确定性模型 因此，与模型实际值的一个差值就能得出随机性为模型增加了多少价值。 对 如果你看看这些界限 首先，很明显，我觉得这一点应该很清楚。 所有的边界都大于或等于零。 因为我们要把更大和更小的东西 东西减去更小的东西。 所以它们都会大于或等于零。 还有一个有趣的地方。 它们都应该小于或等于 EV 减 EV。 从同一事物到可能的最远边界。 第一个可能的反弹之间的差值应该是其他差值的上限。 好吧，也就是说，如果夏娃等同于EV 也就是说，你在平均数据上求解的模型，与你从平均数据中提取第一阶段的解，然后求解成本函数的模型是完全相同的，它们是同一个模型。 这样一来，你就没有任何添加完美信息的价值，也没有任何随机性的价值。 对吗？ 如果你回想一下第一讲，这意味着如果我在方案二中出售了我的模型，这是我的平均方案，我看到了我的模型。 场景二的最优结果与我的随机编程模型完全相同。 或者等于零 或者等于零 这是一个最优解还是一个？ 是的。 那么最优解就变得非常简单了。 这只是模型在平均数据上的解。 所以你只需解决第一讲中的第二种情况的确定性问题。 没错 如果这两种情况相等，你就有了 呃，实际上你已经有了一个完美的模型。 你不需要建立任何随机性模型。 一般来说 值和这些更大的值 这取决于你的解释。 如果你在比较两个模型，其中一个模型的完美信息值更高。 这意味着该模型离完全信息模型更远。 因此，这意味着你。 你会为获得完美信息付出很大代价。 同样，如果你的随机解值很高，就意味着随机性起了很大的作用。 有问题吗？ 有问题吗？ 我只补充了这部分内容，但我也把这部分内容补充到了上周的讲座中，因为这很重要。 在某些时候，你会读到很多关于随机编程的论文。 你会看到这个，你会看到 VPI，你会看到所有这些类似的技术。 这将帮助你直观地理解它们的含义。 有问题吗？ 有问题吗？ 没有 你可以休息一下 休息五到八分钟 休息五到八分钟 因为三点的时候就会很完美了 好的 所以休息八分钟 我们再回来 开始讨论多阶段的问题 谢谢 - 哦 - Thank. 好的 好的，那么，嗯，你 你对我们之前讲的有什么问题吗？ 有问题吗？ 没有，我理解这些模型 就像你看到的那些 我只是不明白 我们是在不同的情况下使用这些模型，还是在处理同一个问题时使用它们？ 或者这些模型有什么用？ 这个有什么用？ 这个的主要用途是评估两阶段随机编程。 所以你有不同的评估方法。 一种是两阶段随机编程的近似值。 所以你想对近似值进行评估。 第二种是对决策进行评估。 所以平均值等等等等。 第三是评估随机编程。 那么，这个模型是否需要随机编程呢？ 这就是这个模型发挥作用的地方。 因此，你可以建立不同的模型，这些模型与随机编程并不相似。 或者它们在某种模糊的方式上类似于随机编程，因为你修正了一些东西或改变了一些东西，然后你就会发现随机编程与这些模型产生的结果之间有什么区别。 所以从某种意义上说，如果，比如说，我的演化为零，那么随机编程就不是一个有用的工具，它几乎就像你在机器学习中用来验证交叉验证的东西。 对 差不多就是交叉验证。 但在这里，我们是对一个模型的结构进行交叉验证。 所以我们必须想出其他结构来进行交叉验证。 所以我们本质上就是从不同角度观察同一个模型 如果是的话 我们看着模型，然后说 "好吧"。 这个模型的一个特点是我们固定了第一阶段。 如果没修好呢？ 所以我们就有了等待和观望的问题。 其次，它包含了概率性。 如果没有呢？ 所以我们把它固定为一个确定性数据的平均值。 所以我们试着问，如果不同的数据产生不同的模型怎么办？ 好的 还有其他问题吗？ 目前一切都还好吗？ 我是说，你能跟上讲座的流程吗？ 因为现在会稍微有点紧张 是的 有点 是的 所以我的意思是多阶段就像我跟你说的那样 对你来说是可选的 嗯 我第一次教这门课的时候 我说每个人都必须做多阶段模型 没错。 所以，从那时起，我就说，如果你做的是两阶段模型，那么，你就能拿到B。 如果你没有雄心壮志，你就说，我对 B 级很满意，我不想在这个项目上花太多时间。 好极了。 你会的，但你必须做得很好。 两阶段模型，对吗？ 如果你在两个阶段有错误，那么一切，呃，A，B是高达69，我认为60，60至69。 60到69 但为了区分，你需要给我一些好的多阶段模型。 很显然，如果你只想做一个多阶段模型，并且包含很多错误，那是没有用的。 好吧 突然间问题就变得随机了 最后你做出来的数值是60。 这是随机函数吗？ 这算不算是失败，还是已经证明了？ 嗯。 一般来说，这种情况是有可能发生的，但极少发生。 这说明你的可行性空间非常有限，所以你没有太多选择。 所以最终你要么选择 A，要么选择 B，不管随机性如何，你要么选择 A，要么选择 B，对吧。 因此，你可以对模型的结构做出一些结论，并说明为什么随机编程不适合这个模型。 但一般来说，如果你的问题只有非常有限的可行性空间，那为什么还要有问题呢？ 没错。 所以你并没有真正的问题。 你可以尝试不同的解决方案。 是啊 我想知道有没有延期的可能 因为两天后我们就有两个任务了 是的，我想你们都可以申请延期，对吧？ 我的意思是，不能由我来申请 因为我已经确定了截止日期 所以教务处不允许我更改截止日期。 所以 但从你们的角度来说，如果你们需要延期，我想延期是很容易批准的。 所以，你只需要求延期或申请延期。 我不清楚是什么制度。 然后你就得到了。 然后你就得到了。 不，不是这样的。 No, no. Yeah. Um, yeah. 即使是特殊情况 也不会影响分数 对 所以我总是根据作品的优劣打分 然后，如果有像惩罚 延迟提交或什么的， 这发生在后面。 好的 还有其他问题吗？ 好吧，我们现在来谈谈多阶段随机编程吧 我想你们很多人之前都问过我这个问题。 因此，我们在这里所做的是，我们有不同的时间段，但只是为了建立一个两阶段模型。 所以，信息揭示的时间点是单一的。 没错。 所以两阶段随机程序设计就是这样一个模型，在这个模型中，你有这样的情景三，如果你还记得的话，你有这样的情景三。 因此，你要在实现之前做出决策 X，然后再根据一些实现情况做出决策。 数据附带一些概率。 等等。S3. 你做了一些决定。 为什么？ 对 但为什么是有时间戳的决定呢？ 你有一代的时间，一代的时间二。 时间三代。 所以，两阶段模型背后有一个大假设。 这个大假设就是 所有的信息都在一个点上显露出来。 因此，在第二阶段，我有第一时间的信息，直到第五时间或其他时间。 规划范围是 对不对？ 这是一个很大的假设。 所以，假设你是决策者。 你要解决这个两阶段随机编程模型。 它的原理是，你有第一阶段的决策。 因此，你要执行第一阶段的决策。 你设置了所有的生成器和一切。 现在你开始等待。 你说，好吧，在第一年年底，我要开始发电。 我将在第一年年底开始发电以供应电力。 我只有第一年或第一年的数据。 我只有第一年的数据。 在这种情况下，我不能使用两个阶段。 对 因为我需要的是每年的数据，然后才能开始生成。 没错。 所以这就是阶段和信息启示或其他东西的地方。 因此，两阶段模型只有一个信息革命阶段。 但阶段和时间是两码事。 而在多阶段结构中，我们要做的，就是努力调和阶段和时间，写下所揭示的一切。 有了时间。 我们将其添加为有条件的启示。 所以，我会再讲一讲这是如何发挥作用的。 那么。 首先，我打算做的是在这里做一个决定X。 等待信息。 第一时间启示。 比方说，假设情景 S1、S2、S3，然后建立一个 y1 s1，直到 y1 s3。 一旦我做出这些决定 我就等待第二阶段的信息启示。 然后针对这些启示做出决定 没错 所以我想这样 这样我就有时间 我可以继续这样 对 我可以一直这样继续下去。 所以，无论我做了什么，如果你还记得两阶段结构，第二阶段的决定是以第一阶段的决定为条件的。 在这里，我说的第二阶段决策就是第一阶段的决策。 以这个为条件，我会产生第二个时间段的决策，以这个为条件，我会产生第三个时间段的决策。 所以最后，这里的决定将以你过去所做的一切为条件。 对，就是这样。 在某些情况下，这就是所谓的动态程序设计，但最常见的是多阶段随机程序设计。 这就是它的主要思想。 我们现在要做的就是复制这个模型。 想象一下我们正在做的事情。 我们有一个嵌套在另一个模型中的模型。 我们有一个嵌套在大模型中的模型队列，对吗？ 在这个大模型中，嵌套着一个在第一代时产生决策的第一个模型，嵌套在这个模型中。 另一个模型在第二代产生决策，不断嵌套，嵌套，再嵌套。 好了，你有了一个模型循环。 因此，我们将尝试解决这个问题。 我们将尝试复制这一点。 但首先，我们要做的和以前一样。 所以，让我们尝试一些不太好用的方法。 虽然很幼稚，但却很直观。 我们会告诉你为什么嵌套很重要。 好的，我们首先要做的就是说，好吧，嗯，我们为什么不解决。 不同的问题 不同的两个阶段的问题，对吧？ 没有嵌套 没有连接 我们解决两个不同的两阶段问题 而且我们对每一个 t 都有所有的决定，对吧。 所以像这样 我建立 我为时间一建立两阶段树 做出X决定 做Y决定 我再建另一棵树 第三阶段 X2 Y2 时间二 对 我一直这样做，直到 我有一个时间 T 的 Cardinality of T, right. 所以，如果我这样做，我解决的模型。 这是一个简单的解法，因为我在求解 t 个独立的两阶段模型，而这些独立的两阶段模型我可以表述为。 就像这样 所以假设我把这个叫做 Z 方法之一 呃 这就包含了概率 记住，概率还是存在的，因为有所有的两阶段模型 每个时间段都会有一个模型。 没错 这就像我说的 "最小化 x I t 的求和 I 加上概率 第二阶段的预期值，包括x xt 输出和 T 的不确定性实现，并受到约束。 好的 嗯，我想我最好还是把约束条件写出来，因为只是为了让你明白它们的意思。 为什么小于或等于 m v I t 乘以 x I t 和 y I t 对 I 的求和等于 rt？ 我没有过去的累积决策。 我在独立求解 t 两阶段模型。 没错 所以，如果我解决这个问题，你首先会注意到，这和这个模型是一样的。 我去掉了这里的 t。 我去掉了你这里 我去掉了你这里 没错 所以，在这种情况下，是的，这是完全一样的模型，因为 t 没有任何区别。 T只是t个不同独立模型的不同决定。 没错 改变的是数据 我只要改变数据，重新求解两阶段模型，就会得到不同的解。 因此，这个模型就是我们上面所做的。 现在，让我们来了解一下这个模型可能会出什么问题。 你认为这个模型会出什么问题？ 可行性 可行性？ 因为这里有一点。 我不是在和过去约会。 对 所以我最初的约束条件是 So my original, my original constraint is. 是这样的 是这样的 我对产出能力的原始约束是这样的。 在这种情况下 当我获得X和Y时 这两个都不可行吗？ 不会 两个都不可行 不可行。 这个1可能不可行，对吗？ 不会。 那我们就来生成它 然后假设从这里开始 我们生成了X 我停滞不前 对吧 Y 我停滞不前 我知道她偷了东西 所以每一个T，我都会产生一些狗屎。 所以，让我们让我们接近一个。 写一个 对于每一个T，我生成一个X，我停滞不前。 所以我知道这是真的 我能确认这是真的吗？ 这就是新的漏洞实例。 这就是求和的原因。 可能不可行，对吧？ 因为这可能是Fi，它的完成就像下一个时间段。 不管产生的是什么，都在上面。 如果我当我的总和一切权利。 所以，如果这是可行的，它并不能保证这一个将是可行的。 但这是不同的。 这个约束条件是不同的，因为如果我能够用更少的发电机产生满足需求的东西。 发电机。 那么我也应该可以用更多的发电机来满足需求。 没错。 所以这个的可行性意味着这个的可行性。 对 所以并不是所有的约束条件都会造成同样的后果。 有些约束条件会变得不可行。 有一些约束条件会变得可行，但不一定是最优的。 但我们现在并不关心最优性。 对 这就是问题所在 第一个问题 就这样 我开始依次求解两个阶段模型 一个接一个 最终的结果是，我的解决方案可能无法满足原始问题的约束条件。 在有些情况下，如果我没有任何积累，我的解决方案就会满足要求，对吗？ 所以，如果我的模型不包含任何从时间 1 到时间 t 发生器数量累积的情况，那就没问题，完全没问题。 这个模型在 T 时是可分离的，你可以求解不同的两阶段模型。 因此，在某些情况下，我实际上可以求解许多不同的两阶段模型。 我复制了一个嵌套结构，因为在这种情况下，模型已经与 T 无关了。 所以，这就是看待问题的一种方式。 让我们试试 换一种方式 我们在这里观察到的主要情况是 我们没有阶段之间的联系 当一切都独立时，我们就会遇到原始问题的可行性问题。 所以我说，假设我们想要建立一些联系。 我之所以这么做，是因为我说过，我观察到可能会有隐蔽性，因为我没有考虑到过去的设定。 对不对？ 我把这个输入 进入时间二。 我正在做决定x2，我有决定x1。 从解决之前的模型。 我现在求解 y2 s，得到一个解。 我一直这样做下去。 直到 直到时间 t，我将求解 XT。 我就会得到 X1 商店的信息，直到 X t 减去一个商店。 没错 然后我就会有一个两阶段三 没错 这就是实时数据 这只是你获得的数据。 Y是你通过模型得到的。 这就是为什么我们说，就像前面的例子，X1、X2，你是在拆分它。 X不是数据，而是决策。 对，就是这样。 所以你是在不同的树里做决定，对吧。 你在第一棵树上做决定，然后得到 X1。 然后把 X1 发送到这里。 然后你再处理这个 X1，防止出现我们之前看到的可行性问题。 然后，再次求解模型，得到新的 x2 和 y2。 因此，数据本身将是不同的数据。 这里的数据 将是 V1S1。 V1, s2, v1 s3. 对 这里的数据将是 V2 S1 V2 S2 V2 S3。 这里的数据将是 v. B, t, S1. 等等就是 x 的决策结果。 所以，Y 是以数据为条件的决策结果。 所以，当数据发生时，我们就会做出这个决定。 为什么？ 为什么？ 好吧 我们所做的就是把所有东西都独立出售。 所以，两个阶段独立。 为了让事情更清楚，我还会在这里添加数据。 这是 V1S1V1S3。 这是 V2S1V2S3 对吧。 所以是的，我们有不同的认识，所以有不同的树。 我们看到的每棵树都是独立的 所以我们看到的树都是独立的 我们得到了不同的 x 值和 y 值，所以在不同的时间段做出了不同的决定。 主要的观察结果是，如果我们这样做， 发电机的可用性可能会被违反，因为我有免费的发电机可用，而且在最后，当我把我在所有不同时间段规定的所有发电机加起来时，如果我没有这个保证，我就不会超出。 对吗？ 因此，无论我在这里得到什么解决方案，都有可能对原始问题不可行，对吗？ 于是我们说，好吧。 那么，主要问题就在于我没有把不同的阶段连接起来。 如果我连接起来，如果我说，好吧，我已经在这里设置了两个，现在只剩下五个了，我应该从五个中选择，而不是从原来的七个中选择，对吗？ 所以，如果我这么说，我就会确保满足约束条件。 没错。 这就是我们在第二部分要做的。 在第二部分，我们说它不应该是独立的。 我们在做出决定之前，会先输入阶段性的决定。 在第二阶段，我们要关注我们所做的事情。 在第一阶段，我们看看还剩下多少发电机，然后从剩下的发电机中提取，对吧。 因此，我们正在尝试制定这个特殊的模型。 因此，我们从时间开始。 让我们从第一时间段说起。 第一时间段 这只是对 IXI1 加上期望值的求和的最小化。 q 取 x1 和 v1 没错 受制于约束条件 约束条件是 XI1，XI1 小于等于 f I y i1 小于等于 VI1，VI 1，XI1 和 y，我赢了。 I 的总和等于 d 1，对吗？ 在第一时间段。 和之前的方法完全一样。 对 所以这和 ZP1 方法一基本相同，因为现在我在第一时间段没有任何条件。 我在求解前四个时间段。 前三个。 现在我开始求解第二棵树。 那么第二棵树 或者说三棵 好的 让我们先看看第二棵树 第二棵树是我们要最小化的。 总和 IXI2 加上一个期望值。 所以我从这棵树里生成了XI1颗星 对不对？ 我正在生成解决方案 在给定X1的情况下试图获得最佳X2的概率。 V一店。 发生了什么事后，V2店？ 对。 好开始。 第一个要来。 好的 好的 所以不是V2商店V2不确定。 对 所以我有第一阶段实现的数据，在第一阶段实现。 第一阶段的决策。 对 这就是我所掌握的数据。 因此，我可以说 XI1 受限于 XI1 星加 XI2 小于或等于 f I，对吗？ 因为现在我知道了第一时间段内有多少台发电机。 我正在处理这个问题。 我不是在独立求解。 而我要做的就是小于等于 v I。 加上Z2 对 然后对I求和YI2等于D2 大家都同意 大家都同意 这个模型包含了第一个时间段的决策 以及第一个时间段的数据 然后我们再看这个模型 我们来看看这个模型。 我们注意到的第一件事是，第一个时间段的数据根本不重要。 没错。 所以在这个模型中，数据之间没有联系。 在其他模型中可能会出现这种情况。 但在这个模型中，V1 在任何地方都不会出现。 所以我可以去掉V1，对吧 对 D2. D2 很高兴你们都在听 对，D2 是的，D2 是的 That. 所以，你听到了，VI？ 听到Y I到F I？ 我们在前三次会议上做出的决定的基础上不断积累 前三个 这就是成本函数 如果你看一下模型 V，就会发现没有任何地方出现 "1"。 即使我们掌握了这些信息 但我们并没有真正使用这些信息 Even though we have this information, we don't actually use the information. 是啊 所以，比如说，如果，嗯，我们什么时候会用到。 假设我们讨论的是温室气体排放，我们试图限制温室气体排放。 那么我们也需要时间一的发电量。 再加上时间二的发电量。 我们试图限制温室气体排放。 那么它将包含V1V1 V1V1 现在我们可以把它概括起来 概括到 t 所以在时间t，这意味着x I。 这里是时间T的X 没错 所以所有额外的 x 到一 直到 x 点 t 减去 1VT. VTi，然后退出。 总结。 冷毛巾 汝小于或等于t减1。 加退出。 嗯 琌摆 所有以前的。 所以，T在这里T在这里。 这里也有类似的东西 所以我们在这里 v I t x I 求和 tau 小于等于 t 减一 tau。 而 x 确实和 t 相等。 现在我们可以将其推广。 那么，T 时刻的模型是什么呢？ 第一阶段的决策是在时间 t 上做出的，同时考虑到之前时间段内做出的最优决策。 这样一来，我们就不可能不满足原始问题的这个约束条件了。 没错 我们也不可能不满足这个约束条件。 反正已经满足了 需求的满足也是一样的 那么 这个模型有什么问题？ 这个模型有什么问题？ 如果一切都可行，这个模型还有什么问题？ 所以，这里可行的一切对于原来的问题都是可行的。 因为现在我们的积累非常多。 我们想要的是积累。 我们在这里也在积累。 这将是可行的。 这将是可行的。 但这仍然不是一个好的模式。 为什么不是好模型？ 因为它在求解。 所以它要做的是优化第一个模型，然后用第一个模型作为下一个模型的信息。 没错 所以，我们基本上是说，在第一个时间段，找到我们能在第一个时间段做的最好的事情，即使这对接下来的时间段来说很糟糕。 没错。 所以，我们在这里陷入了局部最优的想法。 所以我们现在要做的就是解决这个问题。 而这会影响和损害我们的未来。 对 所以这是主要问题所在 这不是可行性什么的问题。 所以我们尝试了两种不同的方法 第一种方法是我们分别解决不同阶段的问题。 我们分别解决了时间段的问题。 这就存在可行性问题。 然后我们通过向前反馈信息来解决可行性问题。 然后我们就有了最优性问题。 对 这就是建立多阶段随机程序的主要动机 明白了吗 清楚了吗 明白了 不难理解 至少这部分 好吧 Right. 嗯，所以随机程序，它将做的是它将开始， 一棵包含条件的树 对吧 所以我们的主要问题在于构建这棵树的方式 如果你看这里，我们正在构建独立的树。 而在这里，我们是通过一个决策来连接这棵树的。 现在，我可以批评的另一点是，数据没有连接起来。 这是一个大问题。 因为阶段并不一定代表时间段，并不总是如此。 所以，我认为你已经做了一点，比方说，在第一阶段，在第一课程中的人道主义物流。 没错。 我认为道格拉斯非常关注人道主义问题。 所以，假设我给你一个这样的数据。 有一场灾难，有人受伤，诸如此类。 然后会有一场捐款活动。 灾难发生后，人们开始募集捐款，然后再把钱拿出来。 没错。 你能看出数据是相互关联的吗？ 数据是有条件的。 所以，如果我遇到了大灾难，我就会开展大的捐赠活动，收集大量捐款。 如果我遇到了小灾难，我就会开展小规模的捐赠活动。 小灾难和大捐款活动是没有意义的。 这样就会剩下很多东西，而且什么也用不上。 反过来说也是没有意义的，那就是你发生了大灾难，却没有人对灾难敏感。 就好像我们不捐款一样，对吧。 因此，数据是有条件的，这两者是分阶段的。 因此，假设捐赠额是，我不知道，是受害者需求的 20%到 30%。 那么根据受害者的需求，你会有不同的捐赠。 好吧。 所以，数据条件是非常重要的。 在某种程度上，信息揭示纯粹是关于数据中的条件。 我正在添加阶段和时间段之类的东西。 这些都不重要。 如果时间段之间没有联系，那也没关系。 我们应该独立解决问题，对吧？ 因此，阶段纯粹是基于数据中的联系。 而建立多阶段树的方法，就是要表现数据中的这种联系。 避免局部最优化，保持一切的可见性。 就是建立这样一棵决策树。 因此，我们从决策开始。 我们从做决定开始。 第一阶段 第一阶段 退出 好的 所有的茶 因此，我们不把它与时间段。 我们仍然希望修复。 我们仍然希望有一个能够部署的计划，而不必每次都要改变生成安装计划。 然后，我们的数据就出来了。 V1 s1 v1 s2 直到 v1 s n。 我还有其他数据。 启示 所以我在这里做的决定是第二阶段的决定。 这就是我做 s 的原因。 所以我在做决定的时候，是为了在场景中生成人工智能。 而所有这些场景都会在这里发生，然后以这个为条件，我有了进一步的信息启示，就会做出下一个决定。 因此，在此条件下，我有了比方说 v one uh s one 条件 S1V1S1 条件 s two。 一般来说，我们需要为它选择不同的符号。 但为了让你明白，第一种情况的条件是 S1，第二种情况的条件是 S1，等等。 还有，我不知道有多少种情况，比方说，s n 的首要条件是 s 2。 这里也是一样。 我们会有V1 S1 S N的条件 V1S和S N的首要条件 在这里，我们正在做第三阶段的决定 这就是为什么我认为应该是 y1 y2 s，而不是那些条件情景。 是啊。 所以这就是我们要用来建立多阶段随机编程模型的框架。 好的 还有 我现在还没开始讲模型 我会在下一堂课再讲的 但这只是为了让你了解我们下一讲的方向，呃，还有，呃。 好的 所以我会坚持。 因为我还有一个建模挑战需要解决。 我们需要做的。 我忘了 但无论如何，我会从讲义中复述某些东西，因为我不认为我，我去了讲义中的所有。 嗯，所以是的，只是复述。 好的 确定性模型 大家应该都能接受。 主要问题是缺乏对不确定性的考虑。 我们建立函数来创建成本函数，我们称之为函数模型。 因此，当函数取决于 v 时，它是 q 括号，当函数取决于第一阶段决策和 v 时，它是 q x v。 有了这个符号，我们就建立了两阶段随机编程模型。 现在应该很直观了。 我们有第一阶段和第二阶段。 在第二阶段，所有的事情都会在所有的时间段内发生，我们就可以建立第二阶段的所有时间段。 好的。 我们还看到，确定性模型实际上是一个随机编程模型，在这个模型中，你有一个概率为一的单一场景。 对，就是这样。 所以我们也可以看到这一点。 有了它，我们就能建立情景树，对分布进行离散近似。 嗯，是的，我在这里说概率应该是一。 但实际上，它可以是任何东西，但它总是等同于一个概率为1的模型。 没错 你只是需要用不同的方式来权衡 Q。 嗯，有了这个离散分布，我们就能建立一个 n 种情况概率乘以 q 的模型，从而得出所有情况下的期望值。 根据大数定律，当 n 趋于无穷大时，其极限渐近于平均值，即平均值。 我们在这里看到了这个条件。 样本平均数近似是一种简单的转换，它将 n 的概率变为 1。 这就是流程图，你需要记住的性能评估流程图。 所以 首先，你要假设一个分布。 这一点非常重要。 我想直到现在，每个人都已经意识到了这一点。 随机编程依赖于概率分布。 这是主要假设。 如果没有最初的分布假设，我就无法进行任何随机编程。 对不对？ 所以，如果我从正态分布中生成程序，我的收敛性就会不同。 如果我从随机均匀分布生成考奇分布，等等。 对吧。 所以，分布很重要，但你需要选择一个。 这就是随机程序的主要缺点。 但你需要选择一个分布，生成一个大样本量的样本平均近似值，然后收集第一阶段的决策。 现在，在第一阶段决策中，你要么计算你所拥有的方案的求助，然后计算统计数据，要么生成样本外的其他方案，然后根据统计数据求解求助。 没错。 所以，你有这两样东西。 这是我今天想再详细说说的 完美信息的期望值 是的 这就是前两堂课的精华部分了 然后我们谈到了决策时机 没错 所有的Y决策都是在包含所有时间段的所有数据的情况下做出的。 我们说，这不是我们想要的。 我们想要的是一些不同的东西。 因此，我们的第二阶段三实际上是说，这里没有信息启示。 嗯，生成 X，并在此固定 X，在未来五年内进行信息披露。 呃，给一代人的一代计划 无信息启示 三在你的四在你的五 无信息启示后。 没错 这就是第二阶段第三阶段的局限性。 然后我们说，好吧，我们需要扩大这个范围。 我们需要做出决定，让信息随着时间的推移而显现出来。 这就创造了一种决策随时间变化的适应性。 这就是多阶段模型的由来。 我们尝试了一些天真的方法。 嗯，我想现在如果你看到一开始的幻灯片，就不会觉得太直观了。 我想现在你可以从我写的东西中理解这些幻灯片的意思了。 第一张，第一种方法是我们独立求解 t 树。 所以是两个阶段的模型。 没错。 每个两阶段模型都是这样的，在一个时间点上做出决定。 我说过这个模型的问题，也就是可行性问题。 因此，就可用发电机的数量而言，你可能会遇到可行性问题。 而且我们没有考虑到过去的信息。 没错。 所以我们每次都从零开始。 然后我们说，让我们通过这种方法来解决隐形问题，我们在这里做一个生成设置。 然后发送 然后，我们不断将前一时间段的生成器设置发送到下一时间段。 这样我们就 我向大家展示了我们是如何建立这个模型的。 我还解释了为什么会出现这种情况。 在第一时间段，我们仍然只是在求解一个两阶段模型。 问题是，我们陷入了局部最优的困境，我们仍然没有把不确定性联系起来。 呃，现实。 这是我给你的 Gams 代码。 我想所有的 Gams 代码都是可用的。 顺便说一句，我不会去看游戏的代码。 如果你们对 Gams 有问题，请告诉我，这样我就能在课上讲解了。 我假设每个人都知道 gams。 每个人都懂。 所以如果有问题，就告诉我，好吗。 如果你求解了我在 Gams 代码中提供的模型，你就会看到这些可见性和局部最优性。 而主要的。 局部最优性的主要问题其实更多的是从我们的计算角度出发的。 如果你还记得，我是把所有的除数都除以 T 的幂，对吗？ 所以，最好晚点再设置，因为这样就可以按净现值折现，这样更好。 但是，当我先优化第一个时间段，然后用它来优化下一个时间段时，我根本没有考虑到贴现的问题。 好吧，这就会产生一些问题。 因此，我们要做的主要事情是这样的。 决定 X，观察第一个时间段的结果，然后决定 Y。 然后观察第二个时间段的结果，再决定 Y 二，等等，直到观察到两个时间段的结果并做出决定。 所以，我们在这里要建立的是一棵树的概念，这棵树上有信息条件。 好的。 你会看到，这里显然涉及到很多符号，因为我需要为第一时间段的情景设置一个符号。 在第二个时间段，我需要为条件等添加符号。 所以需要很多符号。 但我们的想法是，现在可以用条件概率来代替概率。 因此，在时间三发生的概率不仅仅是绝对概率。 它是在时间二发生了什么的情况下，在时间三发生什么的概率。 因此，你有一系列条件概率来连接这棵树。 这样做的好处是非常非常通用。 比如说，如果你的树没有连接，就没有条件。 所有这些都只是一种情况，对吗？ 它只是变成了一种情况。 一个场景。 如果你保持一个场景，一个场景无处不在，你就回到了两阶段树。 没错 所以，这是信息启示多阶段结构的一般表述。 我相信这就是我停下来的地方。 接下来，我们还要建立多阶段随机编程模型。 嗯，是的，我想我们没有时间去讨论建模难题了，不过我已经顺便给你们提供了一些东西。 就在第二讲的最后。 你们会看到，我要求你们添加需求的不确定性，然后观察需求的不确定性会发生什么变化。 所以，增加需求的不确定性。 建立一个两阶段框架，建立确定性等价物，即样本平均近似，编码并求解。 这对你来说应该相当不错。 有问题吗？ 有问题吗 没有 有 下周见 So? 下周见 顺便说一下，总的来说，我提早了十分钟结束。 我的一贯做法是允许你们最后提问 因为我知道不是每个人都喜欢在全班面前提问的 所以，如果你们有问题，尽管问。 我会在这里。 来问我吧，好吗。 没关系，只要我知道旅行是什么样子就可以了。 你来自哪里？ 你从哪里来？ 我来自毛里求斯 但是