

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。

大家早上好。欢迎回来。我想在讲座的一开始简单谈谈评估的问题，因为我不知道你们是否知道，截止日期还有两周多一点。我想，截止日期是 12 月 7 日，也就是两周零两天，不过我强烈建议你们尽量早一点提交。你们应该还记得，上次在我的截止日期那天，我们正好遇到了交稿中断的情况，那真是天时地利人和，让我们所有人都有点惊慌失措。所以你也会知道，如果发生这种全校范围的中断，只要你继续努力提交，就不会被扣分。话虽如此，但我相信，如果你受到了影响，你也会有点呀，如果不发生点什么，你就会陷入恐慌状态。所以，我建议大家尽量早点提交，这样如果再发生什么事情，你就可以坐下来放松一下，享受节目了。当你想到考试或类似的事情时，你现在有什么关于评估的问题想提出来吗？你在做评估吗？请告诉我你在做。好的，你已经开始了。你并不是完全没有开始。那就好很好 非常好 That's good.That's very good.因此，鉴于评估确实允许你种。使用任何你认为适合任务的方法显然，随着我们课程的深入。你可能会想，好吧，我会等到我们在课堂上讲到支持向量机或决策树什么的，然后再真正尝试应用它们。所以，我并不指望你已经完成了，但我强烈建议你尽量开始写起来，并进行总结，因为这样会在最后给你留出一点时间。考试中有什么问题吗？我记得上次你有几个相关的问题。你有机会再考虑一下吗？有啊是这样吗？幻灯片、推荐读物和幻灯片的内容略有不同，因为在幻灯片中，我们有更多关于分布和一切的内容，但推荐读物在技术上并没有谈及太多。因此，这两者之间是有区别的。还有什么是我们应该了解的？我们还有其他地方需要了解吗？有一般来说，我们在课堂上讨论的内容我告诉你的，或者你告诉对方的，或者我们在幻灯片上看到的，然后阅读大多是补充。因此，它是一种添加更多的深度和更多的理解。因此，有时阅读会给你带来一些不同的视角，或者是复述，如果没有，如果阅读中没有涉及，但你对我们讨论的内容有了一定的理解，你就不必再去寻找额外的阅读材料，如果我没有建议的话。尽管如此，如果对你有帮助，你显然可以进行额外的阅读。举例来说，当我们谈到分布的时候，你可能会很自然地选择教科书。相反，如果你觉得你想复习一下的话。好的还需要什么吗？还有输出结果还有问题吗？还有还是更多的只是理论？是的所以是基于理论的。基本上，课程的编码部分是通过小组作业来评估的，然后理论部分是通过考试来评估的。满意吗？差不多吧我听说你在谈论更多的课程作业。还有更多截止日期吗？什么时候？前一周有两个还有两个一样的哦，天哪。所以这是非常紧张的。是吗？是的，我能想象我认为这类课程的问题在于 我们试图教授大量的材料 然后很明显我们有一个结束阶段 在那里我们实际评估所有我们教过的东西是的，我能想象。这是非常紧张的。我在读硕士时也遇到过同样的问题。我们实际上是分块教学。这是完全不同的故事。我们先上一堂课，然后下一堂课，再下一堂课。所以两周只上一堂课，但最后还是要考试。所以，你必须在最后复习所有的材料，以应付考试。所以这是一个不同的挑战。我觉得我们现在的这种结构对大多数人来说更好一些，因为它能让一切都保持新鲜感。好了，我们可以开始了吗？上周第九周，我们讨论了决策树。你应该还记得，我们讨论了决策树在回归和分类问题上的应用。你还会记得，决策树的原理基本上就是建立一个决策结构。因此，你要从最顶端的某个地方开始，然后制定这些逐步的二进制分割标准，决定你的样本应该位于决策树的哪一边，然后使用你的训练数据构建一棵树，然后基本上将测试数据贯穿整棵树，并决定特定的测试数据样本会发生什么。这可能是分类，也可能是回归。你还记得我们说过，决策树本身往往不是超级准确。因此，它们可能会趋于过拟合。它们本身的准确率往往不会太高。但我们可以采取一些措施来提高它们的准确性。最值得注意的是，我们讨论过袋装法。这就是整个引导聚合，或者我喜欢把它叫做，把很多袋子扔到树上摇一摇。我们还有助推法。这是使用残差的想法。因此，基本上我们的树没有研究到的所有空间，都可以用来训练小型的迷你树，然后将它们连接到整个模型中。我们还谈到了随机森林，其理念是我们引入额外的随机化，而不仅仅是我们所有的树。因此，我们种植了很多不同的树。它们是随机的，但我们也限制了这些树的生长方式。具体来说，我们限制了这些树在每次分裂时可以使用的变量数量。这种增加的随机性和对每一棵树增加的压力，使得随机森林在一般情况下成为相当流行、相当强大的分类器和预测模型。因此，随机森林非常流行。在今天的讲座中，我们将研究另一种技术，这种技术也非常流行，因为在很多不同的问题上，它也被证明是一种令人惊讶的强大预测工具。因此，我们今天将讨论支持向量机。支持向量机是机器学习中非常有趣的一部分。当我们谈到支持向量机的实际工作原理时，你会发现支持向量机基本上来自优化领域，比机器学习的很多其他领域都要强。因此，我们会讨论很多机器学习的内容，比如决策树等等。它们给人的感觉非常直观。所以，在机器学习中，我们并不关心它为什么能成功，它就是能成功。所以我们对此很满意，对吧？支持向量机真正采取的是优化方法。因此，它给人的感觉更像是回归的概念。也许就是我们之前在课程中提到的。根据我的经验，SVM 在涉足机器学习的数学家中很受欢迎。因此，我们将讨论最大边际分类器和优化这些分类器。然后，我们将进一步讨论这些分类器是如何通过几个不同的步骤发展成为支持向量机的。这个想法是在计算机科学界形成的。它是几年前提出的。因此，实际上它是一种较早的技术，我们可以追溯到 20 世纪 90 年代，这也是为什么我们要把蜂群优化作为重点的原因。但这些技术越来越受欢迎，所以它们在过去和过去的日子里相当流行，但现在它们的发展势头更猛，变得越来越受欢迎。因此，在很多论文中，如果你查阅文献，实际上都有一个比较，例如，随机森林支持向量机和神经网络。这些就是你经常在文献中看到的三大算法。正如我在这里所说，它们通常被认为是更强大的分类器之一。我总是小心翼翼地说，因为这在很大程度上取决于你的问题。因此，不要陷入过度依赖一种模型的陷阱。我认为，很多数据科学家和分析师都会犯这样的错误：他们有自己的最爱。他们有自己喜欢的建模方法。我也在做同样的事情。我的意思是，我试着把聚类方法强加到每一个问题上，因为我觉得它太神奇了。但另一方面我觉得这会让你倾向于寻找一些不存在的东西。所以经常看到这样的问题，人们试图强制使用一个特定的模型，因为他们认为这是最好的分类器，可以是随机森林，可以是神经网络，等等。然后他们就会想方设法证明自己是对的。因此，这就涉及到了 "P 值黑客 "这一领域，你需要解释的是，嘿，这个模型的准确率可能与其他模型相同，甚至更差一些，但它在这一特定指标上更胜一筹。因此它是最好的模型，这只是再次证明了我的观点。所以，不要掉进这个陷阱。要有开放的心态。要明白，最佳模型总是取决于你试图回答的具体问题。我还提到，SVM 基本上是对最大边际分类器概念的概括。因此，我们将再次看到这是我们的线性模型或线性分类模型之一。如果你还记得我们在讨论线性回归时，我们基本上是这么说的：好吧，线性回归的优点之一是简单。缺点之一是线性。在这里，你会看到一个非常简单、与我们的最大边际分类器非常相似的概念，因为我们正在寻找可通过某种线性边界分离的类别。好在，我们将其从这一概念中概括出来。SVM 立场实际上是一种分类器，它可以通过一个小技巧（我们称之为核技巧）很好地适应非线性类边界。稍后我们将对此稍作介绍。不过，这基本上是一个将数据投射到空间中的巧妙想法，通过这种方式，你可以使用线性分类器来分离数据。这确实很聪明，因为分类器本身仍然处于我们绘制的线性边界，但因为我们首先将数据投射到不同的空间，所以实际上我们允许线性边界对非线性数据起作用。稍后你会看到这一点。我还会提供一张图片，希望能解释这一点。真聪明对不起。对不起。好了，在我们开始讨论分类器的这些想法之前，我们得简单谈谈超平面。超平面是一个非常重要的概念，因为我们刚刚说过，我们在寻找这些线性边界，这些我们试图绘制的线性决策，我们以超平面的形式绘制它们。现在，超平面其实就是任何平面，我们可以在空间中绘制出更广泛意义上的平面。所以在二维空间里，超平面就是一条线。举个例子你可以想想，如果你有两个维度，你有某种点的分布，你画一个超平面穿过它，把它分成两部分。超平面总是能够将分离的空间完全分成两部分。因此，在二维空间中会有一条线。我们可以画一条线穿过这个空间。在三维空间中，它是一个平面。因此，我们可以在整个空间中画出一个平面。我无法告诉你那是什么样子。我们看起来像四维、五维或六维。我没那么聪明。我相信有些人可以用四维空间来想象。我试着做过一次。网上的图片真的非常非常聪明。如果你想知道四维空间是什么样子，它们就像是抽象的可视化。但是，一旦我们的四维空间超过了五维，那就无法想象了，但在数学上却是可能的。而这正是我们所关心的。所以我在这里做了一个小小的修正。如果你有旧版本的幻灯片，你会发现多了一个变量，但这样更有意义。在二维空间中，超平面就是一条线。所以你可以把它想象成一条线性回归线。基本上，你有截距和斜率。而在三维空间中，它的维度会越来越多，因为你要描述的是一个平面。因此，当我们谈论 SVM 时，要始终牢记这一概念。因此，我们可以将其推广到 P 维设置中。这样就有很多不同的维度了。请记住，在我们的案例中，维度基本上都是指你所拥有的数据。因此，你可以从几个维度来衡量一个特定的观察结果。举例来说，如果你正在进行某种调查或收集某个人的数据，你要从几个维度来衡量这个人？所以会有他们的年龄、收入和所有这些不同的维度，所有这些问题。因此，P 基本上取决于数据集的大小或列的宽度。因此，有时会有一些点不符合超平面的定义。因此，它们可能位于超平面的一侧或另一侧。所以，再用一个简单的术语来想象一下我们的空间。你可以把它想象成二维空间中的直线。然后你就会在超平面的两边看到点。记住，这是一个分类器。很明显，我们试图以这样一种方式来画线，使点位于超平面的正确一侧。例如，我们认为有两个类别，我们就会在超平面上画一条线，这样我们就能以正确的方式将点分成这两个类别。那么这看起来像什么呢？就像这样想象一下我们的二维空间。我们有一些点，我们知道它们属于哪一类。我们有蓝色的点和粉色的点。我们试图找到一条线，一个超平面，把这个空间分成两半。它们描述了我们认为的蓝色和粉色这两个类别。你可以看到，我们可以用很多不同的方法来画这条直线，但总有一条直线是划分这些点的空间的完美方法。然后你可以看到，对于每一个点，我们基本上都可以查看它们是否位于该平面的右侧。也就是说，它们是否被正确分类了。这种概念的有趣之处在于，如果你注意到这里，有些点会比其他点更靠近直线。因此，在某些情况下，这种分类会有点不确定。也许这是有道理的。因此，你可以看到，即使这条线向一个或另一个方向稍稍摆动，也会影响到这些更靠近这条线的点。我认为这是一个非常重要的概念，每当我们试图对空间进行分类和分隔时，对某些点来说，超平面方程的微小改变所产生的影响要比对其他点产生的影响大得多，而对其他点来说，分类要清晰得多。没错。所以，如果我们有这个方程，在这种情况下，这是我们的多维度，我们的 p 维度例子。这里有两个维度。那么我们基本上可以说，这些点中的每一个点的 x 值，对于一边的点来说，都会大于零，而对于另一边的点来说，都会小于零。因此，我们基本上是在计算这些点位于直线的哪一边。现在，刚才提到的关于点离直线有多远的有趣问题，实际上就是我们如何开始思考这个分类器以及如何找到最佳分离线的问题。因此，我们称之为最大边际分类器。我们的想法是，我们要以这样一种方式来画线，即让所有这些点的边际最大化，也就是让离边际最近的点的边际最大化。因此，你可以看到，我们试图以这样一种方式来绘制这条线，即中间的实线，这条线左边和右边的边际是虚线，我们希望将空间最大化。因此，我们要最大化最近点之间的距离。我们为什么要这么做呢？回想一下我们的例子。如果我们的线条不是最优的，那么它们就会离其中一些点太近。你可以看到这里。例如，我们有一条非常靠近点的线。这条线也很糟糕，因为它离这个点更近。因此，我们知道这两类之间有一定的空间。我们要以这样一种方式来画线，即最大限度地扩大我们这条线的决策边界左侧和右侧的空间。当然，我之前提到过，我们要把这变成一个优化问题，为什么不呢？我们的想法是，根据更好的参数，最大化边际 m。记住，更好的参数就是截距和斜率。因此，它们基本上就是描述直线的样子。因此，我们希望通过调整直线在空间中的位置来最大化余量。我们这样做有几个限制条件。因为这是一个优化问题。在这种情况下，我们必须想出一些限制条件。这就是两个问题。其中一个比另一个更容易理解。其中一个很容易理解，因为这个限制条件基本上就是每个观测点都应该在超平面的右侧。因此，只要 M 是正值，因为它是边际值，显然就应该是正值。我们先看一下这些点中的每一个点，以及它们与我们的直线之间的关系值，然后我们希望以这样一种方式绘制直线，即这些点位于直线的右侧。原因就在这里。你可以看到，在这个区域，我们可以画另一条线，这条线可能离所有点都更远。所以边距可能会更大。但所有这些点都会位于这条线的错误一侧。因此，如果我们在这里画一条线，就边距的大小而言可能会更好，但就点的分类而言会更差。因此，这是一个最佳位置，我们可以在这里画一条线，这样既能使边际最大化，又能确保点位于线的右侧。此外，我们还有另一个限制。不用想太多。这在数学上是说得通的。这个想法基本上与上文非常相似，即我们所有的贝塔值总和应该为一，或者说贝塔值的平方和应该为一。这是一个很长的数学理由。我就不多说了，但它对我们模型的影响基本上是，每个限制条件、每个观测值都在超平面的正确一侧。因此，它是一种加倍的约束，离超平面至少有 m 的距离。这样就说得通了。我们不希望有任何点在边际范围内。边距只能与点的距离相等。因此，边距基本上就是我们所拥有的最小距离，因为这是最接近的点。确定边距的宽度。显然，这就是我们的两个限制条件。并不总是这样，因为事实并非如此。我的意思是，你的数据不会是这样的，这你是知道的。所以在大多数情况下，我们的数据都是这样的。所以我们会有也许有一条隐约可能的线穿过这个地方，你可以想象这条线可能会在这里的某个地方，对吗？但这并不是真正的线性边界。这就是典型的情况。如果在这种情况下，我们说样本的数量小于预测因子的数量，那么情况就尤其常见。因此，n 小于 n，p 小于 p，这是唯一一种不太可能出现这种情况的情况，因为此时你的样本量太小，而预测因子的数量太大。这样一来，数据就有很大的传播空间。这样就更容易找到线性边界。通常情况下，我们将不可分离数据或噪声数据之类的数据称为可分离数据。我们仍在试图在其中画出一条分界线。因此，我们先来看看比较简单的情况，因为这可以解释我们如何处理比较复杂的情况。在某些情况下，我们的数据可以通过我们的线性边距、线性线来分离。因此我们可以看到，在左边我们可以画出完美的分离。我们就像之前那样做。但我们可以把这个例子称为有噪声的例子。或者说，我把这个例子叫做有影响点的例子。例如，你可以看到右边这里，如果我们在这里多加一个点，会发生什么。你可以看到左边没有点。然后我们多加一个蓝色的点，我们原来的线就会突然不对了。我们应该画这条新线。因此，只需增加一个点，直线的斜率就会发生很大变化。观察。现在，我们可能会有这样的疑问：奇异的观察结果真的如此重要，以至于我们要确保我们的线条完美地分隔我们的空间吗？那么，我们对奇异点上的这种奇异线条、奇异个体的重视程度有多高？也许我们可以说，好吧，我们正在改变我们辛辛苦苦建立的整个分类器。这就是我们所说的有影响力的点。基本上，这是一个对模型和模型构建方式有很大影响的点。你有时会听到我们谈论有影响点的作用，但它们并不一定是异常值。我们不会把这个点称为离群点，因为它并没有极端值。与同类的其他点相比，维度只是表现得有点奇怪。所以它对模型的影响很大。所以它是一个有影响力的点。这就给了我们一个基本思路：我们该如何扩展我们的最大边际分类器，使其允许这些有影响的点？我们是真的想把整个分类器都改成这个方向，还是对某些点影响模型或不影响模型感到满意？思考如何更好地把握时机。好的，在休息之前，我会再讲一会儿。我们谈到了最大边际分类器的概念。请记住这一点，因为支持向量机最初就是这样开发出来的。不过，现在先把它放到一边。一般来说，支持向量机就像我们讨论过的其他方法一样。就像我们之前提到的决策树一样，支持向量机也是一组方法。因此，在支持向量机的框架下，我们有不同的方法。一般来说，我们可以使用它们，就像使用决策树进行分类和回归一样。我们将专注于分类。在分类任务中，支持向量机更为流行。从技术上讲，它们也可以用于回归。但这种方式比较少见。因此，你会发现支持向量分类器是最常见的。但因为它们太受欢迎了，所以我们仍然称它们为支持向量机。所以这就有点混淆了。如果你读到 "支持向量机"，可能就是指支持向量分类器，但我们并不经常使用这个词。基本逻辑还是一样的。我们仍然在寻找最大边际分类器。因此，这仍然是一个优化问题，我们试图将其最大化。你还记得我们正在绘制的分类线周围的边际吗？所以这条线不一定是一条线。它可以是任意维度。但我们现在所做的，是允许有一些松弛。因此，我们增加了模型的这种能力，使其对这些有影响的单个点不那么敏感。所以我们说，这一个点会改变整个模型，而不是说，这一个点会改变整个模型。小音乐所以，我们不再说我们允许单个编辑点改变整个模型的设置，而是说，好吧，我们将允许一点点松弛，一点点误差。不一定非要用完美的线来划分。再多说几句术语，正确的术语应该是把它们称为线性支持向量分类器，因为它们是线性边界，使用支持向量。稍后你会听到更多关于分类的内容。所以，线性支持向量分类器。没有人真的这么称呼它们。大家都只说 SVM。他们可能会说线性 SVM，或者像线性 SVM 这样的分类器，但这才是技术上正确的说法。尽管如此，我之前确实说过，我们将使用一个技巧来对非数据进行分类，这样你就可以在线性分离的情况下使用它们。它们在这方面的作用非常完美。它们有时可能有点矫枉过正，但还是能用的。但你可以这么做，而且通常会将它们用于非线性情况。所以，让我们先看看线性版本。线性支持向量分类基本上是我们之前模型的一个变体。它是一个最大边际分类器。但我们只是在整个模型中增加了一点松弛。这和你之前说的是一样的。我们想要最大化最大边际值，可以通过改变参数来实现，这些参数决定了线在空间中的位置。我们仍然有一个略显奇怪的约束条件，你只需相信我，它能确保边际永远不会是两个。点不在边际内。我们仍在尝试最大化，我们仍在尝试在之前的相同约束条件下解决整个问题，确保各点都在右边。但这次我们在这里添加了一个小项，即 1 减ε。我们要确保所有的ε都为正或零。我们要确保所有这些项的总和小于 C。因此，这里的这个项是在增加松弛度。基本上，它允许是的，所有的点都应该在分类器的右侧。在大多数情况下，我猜是这样。所以要稍微加一点是的。它不会对我们造成太多限制。所以我们要增加这一点，这一点松弛，但我们还是要确保不会有太多的松弛。因此，我们仍在努力寻找一条既能很好地分离，又不完美的线，如果这能说得通的话。那么这看起来像什么呢？这些小的ε项就是我们所说的松弛变量。因此，它们允许个别观测值处于超平面边缘的错误一侧。你可能会遇到三种情况。要么一切完美。观测值处于边际的右侧，一切都很好。或者是项比零稍高一点。然后是观测值在边际的错误一侧。所以我们会说这是一种违反边际的情况。这可不好。所以，你可以看到这里有一些点是在边际范围内的。我之前说过，我们不希望有点在边距内，但并不是说我们不希望有点在边距内，而是对于某些点，我们会允许它在边距内。有时，你会发现一些点完全偏离了超平面。在这种情况下，我们的误差项实际上会大于 1。因此，我们的误差范围基本上是渐进的。如果一切顺利，误差为零，如果没有误差，误差大于零，就会向错误的方向移动。在这里，你可以看到八号观测点正从它的蓝色点群向边界边缘移动。它在边际范围内。所以这并不好。一号观测点也是如此，正在向边缘移动。所以这将是一个大于零的ε。在某些情况下，一切都会出错。观测值的ε大于1。然后就是完全错误的一边。例如，11 号观测点就在平面的错误一侧。应该在这边。12 号观测值也是如此。超平面的一边应该和所有其他蓝色点在另一边。所以我们要计算每个观测点的误差有多大。从理论上讲，这与我们处理回归问题的原理是一样的。例如，当我们计算点的误差时。你还记得我们在讨论线性回归时，每个观测点离回归线有多远？我们总结了所有的误差，完美地预测了点的位置。然后我们想让误差保持最小。没错。这是一个非常简单的类似过程。我们计算误差。有不同类型的误差，误差的严重程度也不同。然后我们限制总和。因此，所有误差的总和都应受到约束或限制。会有一个最大值。在我们继续之前，我们先休息一下，这样你们也有机会提几个问题。我们将在十分钟后继续。好的，请说 All right.继续吧好的，我们刚刚聊完，或者还没聊完。我们聊到一半的时候，通过我们的松弛变量。有人问我是的，没错。我们的每个观测值都有一个松弛变量。因此，你可以把它们看作是回归中的误差项。但你也可以为每个观测值计算一个误差项，计算它离直线有多远。在这种情况下，你计算的基本上不仅是观测值离直线的距离，还包括观测值是在直线的右侧还是在直线的左侧。因此，你要计算它是被正确分类、被错误分类，还是被正确分类，但违反了你的边距。这就是你可能遇到的三种情况。这是对我们每个观察结果的计算结果。然后我们再计算所有这些的总和。记住，它们总是正数。所以它们的总和也是正数。C 是我们的正则化参数。这是你选择的一个数字。这是你选择的调整参数，用来决定在整个模型中允许多少松弛。你允许多少错误分类？你允许多少违反边际值的行为？所有这些的总和就是你的参数。C 你会在计算机实验室里看到，你会在计算机实验室里看到。你会在计算机实验室里看到，在 scikit learn 中，这也是 SVM 模型功能的输入参数之一，它有一个默认值。我想这是一个我不完全确定的值，但你可以看到，如果你减少或增加它，会发生什么。它必须是正值。它可以小于 1。记住，如果我们的epsilons只是违反了边际值，但仍能正确分类，那么它就会小于1。因此，所有ε的总和 C 可以小于 1，但总是至少为零，这样才是完美的。从可视化的角度看，这基本上是允许的。它决定了边距的宽窄。那么，我们该如何绘制边距，从而使违反边距的情况最少呢？你可以看到，这里左上角的边距非常宽。因此，我们允许大量违反边距的行为。而右下角的边距非常窄，我们不允许大量违反边距。因此，我们不希望有很多点落在其中。所以我们把它们画得更小。从这个意义上说，模型的限制更多一些。它能同时确定违反边际和超平面的数量和严重程度。它基本上就像一个预算。所以你允许多少违规行为以及违规行为的总数。如果 C 为零，就没有预算。一切都必须是完美的。每个点都必须恰好在边界的右侧，并且不在你的边际范围内。然后你就可以增加预算。你增加的越多，就越允许出现这些错误。实际上，这意味着 C 基本上就是我们的调整参数。因此，你可以通过交叉验证来选择参数，因为这取决于数据的可分离性。因此，如果数据的可分离性非常好，那么很明显，你会找到一条违规较少的好线。如果数据的可分离性不是很好，那么就会出现违规现象。而你只是想限制违规的数量。你也可以将 C 视为偏差方差权衡的控制因子。所以，你还记得我们在讨论这个问题吧。我们希望模型的拟合范围有多窄？对训练数据的限制应该有多大 对训练数据的限制可能是好事，因为它可以提高准确性，但在某些时候也会达到一个极限。模型的范围太窄，只局限于从训练数据中学习，以至于在测试数据上表现不佳。因此，如果出现这种情况，我们就会说这个模型过拟合了，因为它在训练数据上的表现太好，而在测试数据上却完全不行。实际上，有一种方法可以很好地看到这一点，如果你看一下准确率分数，如果你有一个分数，如果你的训练准确率非常好，而测试准确率不是很好，这就表明你的模型是过拟合的，因为它在训练数据上的表现比测试数据好得多。通常情况下，你的模型在训练数据上的表现会更好一些，但你不希望它在训练数据上表现完美，而在测试数据上却一点也不好。这是模型过拟合的明显迹象。在这种情况下，如果你要拟合 SVM，就需要放宽一些限制。那么，C 的值意味着什么呢？C 值越小，边际越窄。因此，你需要寻找罕见的违规情况。记住，这是你的预算。因此，误差预算越小，意味着你的模型偏差越小，但方差越大。如果 C 越大，边际就越宽。你就允许出现更多的误差。这意味着你可能会有更多偏差，但模型的方差会更小。因此，从其他方面来说，换句话说，你对训练数据的拟合可能并不完美，但你能够更好地概括训练数据。因此，较大的 C 可以让你更好地概括测试数据。我认为这是一个非常重要的概念，因为当你听到 "这是你的误差预算 "时，你可能会想，我不想犯任何错误。当然，C 应该始终为零，或者至少应该非常小。但这样一来，你就掉进了过度拟合模型的陷阱。过小的 C 值会使 SVM 过拟合，因此必须留出一点余量。请记住，每个观测值都会增加误差，所以如果观测值很多，自然会增加误差。举例来说，因为你的观测点越来越多，可能会有一点违规。因此，在这种情况下，预留一点误差是非常重要的。现在，我们之前说的都是线性边界。因此，我们有了一些空间，二维的线和三维的板或超平面，但有时并不奏效。因此，有时你的数据看起来就像这样，这对使用线性模型的人来说简直就是噩梦。你可以看到，我试图通过线性边界来绘制数据，很明显，就是这样。就是不行。太糟糕了中间是粉红色的点。这里有蓝色的点，那里有蓝色的点。线性边界根本无法将其分为两类。所以，你认为价值是什么并不重要，你的边界或边际如何也不重要。这是不可能的。那我们现在该怎么办？有几种方法。人们首先尝试的是扩大特征空间。因此，我们使用预测因子的函数，而不是使用预测因子本身。你还记得吗，这有点类似于这样的想法：嘿，让我们对所有数据进行对数变换吧。或者在这种情况下，让我们使用所有预测因子的二次项，而不是直接使用预测因子本身。有时这样做也行得通。例如，在这种情况下，我们会对所有的 x 值使用二次函数，然后使用线性函数拟合 SVM。这样做效果会很好。问题就在于此。它变得非常复杂。计算变得非常困难。因此计算效率非常低。举例来说，我们仍然有同样的最大化问题。最大化我们的 m 值。我们仍在努力使它的值在右边，允许有松弛。我们仍在努力限制总的松弛量，并尽可能确保它们在右侧，而不是在余量范围内。因此，有很多不同的方法可以扩大特征空间。因此，你可以使用二次函数和所有这些方法。问题是，如果你尝试这样做，最终往往会得到大量的特征。如果你把每个特征都看成是一个变量，再考虑到有成千上万个变量，突然间你就会发现数据变得难以管理。数据量太大了。所以这是有可能的，但效率真的很低。于是，人们想到了我们称之为内核技巧的东西。内核技巧确实相当聪明，因为它允许我们在非线性分离情况下使用线性边界。因此，如果我们谈论支持向量机，我们通常会谈论使用核的支持向量机。因此，对于非线性问题，从技术上讲，我在这里添加了一个小注，说明了所有先前的模型。我们之前谈到的模型也是支持向量机。我们有时称之为线性支持向量机。你会在 scikit learn 中看到，你也可以拟合普通的支持向量分类器，只需添加线性作为内核即可。然后你就能拟合出我们之前在模型中提到的东西了。但当你阅读文献时，在大多数情况下，他们指的是应用了核技巧的 SVM，特别是在 SVM 中，高斯核是非常流行的。这种核方法背后的理念是，我们希望以这样一种方式来扩大特征空间，即我们可以通过它画出一个线性边界，同时还能分离出一个空间，这是一件计算效率很高的事情。因此，这涉及到一些数学问题。具体来说，我们要研究两个向量的内积。现在我把所有的数学知识都写在这里了。就是这个意思。如果你不记得线性代数中的内积是什么，我不怪你。但内积的概念就是这样。我们正在寻找描述两个向量之间相似性的方法。例如，两个矢量之间的点积可以通过它们的长度来描述，我把矢量看成是箭头和它们之间的角度。那么，这两个向量的长度是多少，它们之间的夹角是多少？这就是它们的点积。所以这是一个奇数，我们可以用它来描述这两个向量之间的关系。然后，这些内积就可以被概括了。这就是内积背后的整个概念。在这里，我们研究的是两个向量之间的相似性或关系。这一点不用太担心。举例来说，如果你想到两个向量 a 和 d，我们可以这样描述内积。所有这些观测值的总和，直到 J。然后取这两个向量的乘积。把它们看成是一种关系数。对于向量来说这样就够了。重要的是，如果我们对线性支持向量分类器做了这样的处理，那么每个训练观测值都有一个这样的观测值。因此，每个训练观测值都有一个这样的项。所以从某种意义上说，这真的有点非常抽象。我们必须计算每个新点和每个训练点之间的内积，这非常昂贵。好在，现在终于到了我要告诉你支持向量机为什么叫支持向量机的部分了。我们不需要对每个观测点都进行训练。所以，如果你有一个非常大的空间，并思考我们在哪里可以找到。我们在哪里能找到这些线条？我们没有考虑如何将这些线与我们正在寻找的所有点进行拟合？让我找张好的图来我们基本上是在寻找所谓的支持向量。这些锚点决定了我们的直线应该在哪里。例如，在这种情况下，可能是这一点、这一点和这一点。我们称它们为支持向量。因为请记住，每一个点都是通过一个向量来描述的。你的每一个观察结果都可以通过一个向量来描述，这个向量的维度就是你对这个点的变量数量。这就是为什么我们称它们为支持向量。它们在空间中支持着我们的超平面或直线或其他任何东西。所以要决定这条线的确切位置。我们在哪里？没错好在这个值 alpha 在这里。天哪，只有支持向量的阿尔法值才不为零。所以任何不是支持向量的训练观测值都不为零。想想那些远离直线的训练观测值。我们并不关心它们。那么阿尔法值就是零。因此，换句话说，我们只需查看这些支持点、支持向量。然后我们就能比较容易地计算出我们的直线、我们的分割线在空间中的具体位置。现在是内核思想。这个想法就是我们把这个内积放在这里。我们刚刚决定，α 永远只是我们支持向量的一个值。现在我们来看看 x 和 x I 之间的内积。我们使用核方法来替代这个内积。我发现核函数和核方法非常混乱和复杂，我真的不知道。好吧，这到底是什么意思？所以我一直在思考它们。这有点这有点傻。当我第一次听到 "上校 "这个词时，如果英语不是你的母语，我想到的是爆米花核，就像你用的那种小爆米花核。然后我想，好吧，如果我，比如说，有。一个二维空间，对吧。所以我是那种二维空间。如果我有一个爆米花核，把它放在中间，把它向上推到三维空间，会怎么样？这就是我对爆米花核的看法它们就像一个个小东西，把我们的维度推高了一个维度。因此，两个维度被推高到三个维度。一个维度会被推高到两个维度。这就是我对内核的理解。我觉得这比所有这些都容易记住。那么问题来了，你的小果核是什么形状的呢？所以我得到了一个小爆米花核，而评论可以有不同的形状。我之前提到的最常见的形状是高斯核。所以，我想到了高斯形状。这就是一个小内核。你把它推向你的空间。你可以有多项式内核。你可以有径向内核，不同的应用案例有不同的内核形状。这都是一样的。选择哪种内核取决于你的数据。你的数据将决定你所需要的形状，从而使数据摆脱简单的维度。让我们将其可视化，因为这有助于思考这个案例。这看起来和我们之前做的很相似，对吧？中间是一个空间。我们有这些红色的点，然后周围有很多紫色的点。你会觉得我们不可能在这里画出一条线性边界，但实际上，如果我们把二维空间推到三维空间，我们就能画出一条线性边界。因此，如果我们在这里使用一个核，例如，这应该是一个高斯核。你可以看到，它突然变成了三维空间中的这个形状。现在，底部的所有红色点或紫色点都在空中。你可以画出这样一个绿色的空间，这个绿色的平面穿过它，完美地将两者分开。这就是内核技巧的原理。我还发现了这个非常漂亮的可视化效果。类似的例子还有，我们在中间有一些点，周围有一些点，然后我们把内核推过去。突然间，我们可以找到很多东西。我们可以找到一个超平面来分割这些空间。这就是我记忆中的核技巧。我认为这更直观地反映了所有其他的数学公式，这些都非常重要。所以，我们找到了这个超平面，它可以在这个额外的空间中进行分割。然后，如果你从二维的角度来看，我们真正找到的是这个完美的分割圆。这就是内核技巧背后的理念。这是一种基本上可以让我们分割空间的方法，而这些空间是不应该被线性边界分割的。把它们推向额外的维度。所以我们之前提到过，我有不同形状的内核。例如，在这种情况下，很容易看出这应该是一个高斯核。因此，这是一个完美的空间。我们可以把它推高或推低，然后找到一个完美的分离空间。但是还会有其他空间。你们明天在计算机实验室里也会看到一些这样的例子。所以我们觉得没问题。好的我还以为发生了什么事明天你们要测试多种不同的内核问题是，我们很难提前知道什么是完美的形状。所以在某些情况下，这相对容易。你可以绘制它，然后看看什么是现实的内核形状。大多数情况下，我们会再次进行交叉验证。因此，明天我们将在计算机实验室使用 Gridsearchcv 测试不同的内核形状。我们将测试不同的 C 值。最后，你基本上要寻找完美的组合、允许的完美松弛量以及你想要寻找的完美内核形状。因此，使用内核的优势在于，它可以让我们分离非线性，而无需真正在大型特征空间中工作。在大型特征空间中工作，计算成本非常高。因此，我们要做的基本上就是查看我们的支持向量、支持点，然后决定我们要在这里拟合，基本上就是核函数。这样我们就可以进行除法运算。这就是我们之前的结果。你现在可以看到，这将与一个阶数为 3 的多项式核完全吻合。或者右边这个例子。这是一个高斯核的漂亮例子。所以你可以绘制这些空间图，然后尝试不同数量的核。是啊这是个很好的例子，说明了核的不同数值。明天你在计算机实验室也会看到，很明显，我们刚才说这是一个三度的多项式核。我之所以知道，是因为我测试了多项式核的不同值。所以，不幸的是，你需要做大量的参数拟合工作。因此，有些模型的参数要少得多。在 SVM 中，你确实需要选择 C、正则化参数、核的形状、度数和参数，以及所有与核相关的参数。因此，这三个参数就是 SVM 必须做出的三个决定。这还是比较容易拟合的。好了，关于支持向量机我们已经聊了很多。我相信你还有一些问题，但我们还是先谈谈你的邻居吧。再来一次想想支持向量机。你可以将支持向量机与其他方法进行比较。你们可以想一想，我们到底讨论了什么，可能有哪些优点和缺点。你们有机会思考我一上午的精彩独白。然后我们会再次收集你们的分数。你们花十分钟左右，然后我们11点30分再谈。好的，你们喜欢我的鼓掌吗？我总觉得我会喜欢老师这样。啊，你可以这样是不是这只狐狸？这只安静点狐狸这是你们老师在学校教的吗不是 好吧 很好好的谁想开始？我们有优点、缺点 或者你们想发表的意见 或者你们想问的问题我们应该从中间开始改变吗？这不合适真的适用于。正式。解密，所以必须适用。是的。所以多类分类有点棘手。你说得很对。所以这是可能的。是有难度。它的基本原理是使用多个二元模型，然后将它们组合在一起，解决多类问题。所以我们所做的每一步基本上都是在做二元分类。你必须将多个不同的模型结合起来，才能解决多类问题。因此，SVM 可以用于多类问题，但并不像其他情况那么简单。完全正确。中间还有其他东西没错。我觉得你说的建立模型时对异常值非常敏感。比如如果你考虑到异常值，就会生成另一个新模型。如果你拒绝接受异常值，包括是的所以它对异常值和噪声特别敏感。多亏了我们的松弛变量，我们才能让它更好地工作。但是，如果你真的在数据集的完全另一侧出现了异常值，那么 SVM 的拟合就会有点棘手。完全正确。所以，我们把这称为对异常值和噪声敏感。是这样吗？是的，这取决于。这取决于你使用的成本函数。基本上取决于你使用的 c 参数。但在使用 SVM 之前，你必须仔细清理。有什么优势吗？有什么优势吗？有你觉得呢？对这里还有如果这个参数能让我们一概而论的话嗯哼 - 对 - Uh huh.是的所以基本上是为了控制过度拟合。对所以你真的有一个参数可以选择，你可以仔细选择，以避免模型过度拟合。所以我们在讨论如何拟合边际值，要拟合得多接近，要多严格。还有。举例来说，与双网络相比，我们有很多参数需要拟合。因此，从这个意义上讲，调整过程可能会更加复杂。或者说，你有像聚类这样的东西，这很难与之相比。但要控制模型的表现则更加困难。因此，在这里我们有了这种控制或调整参数C，例如，你可以用它来控制过拟合。是的，非常好。非常好但也有一个缺点，那就是。是的你必须选择 C。你必须调整参数你不仅要调整参数。这就是你的C语言和内核参数。但你还必须选择你的内核。这很棘手。因此，我们在二维中看到了这些相对简单的例子，我只是告诉你，嘿，你有这两个维度。绘制它们。嘿，这看起来像一个高斯核。那么如果你有 50 个维度，你该怎么办呢？你不可能真的绘制出 50 个维度的图，然后再决定你的高斯核如何与之匹配。在这种情况下，你必须仔细选择内核，通常是基于交叉验证，并对参数 c 以及内核根据其形状自带的任何参数进行同样的处理。这就是参数调整。这并不容易。我的论文有五个优点。对此，我还会更多。是的是因为为什么它允许我们做 指定非线性？是的所以非线性分类。谢谢。双核技巧。真方便。我们刚才谈到了线性分类和线性回归，其中一个主要缺点就是模型的线性。所以我们在这里使用了核技巧来避免线性。所以，是的，这是一个很大的非线性问题。我的论文里还有什么？嗯还有这也是很多人使用 SVM 的原因之一。它能很好地处理高维数据。因此，与其他方法相比，SVM 在高维数据中的表现非常出色。我们谈到了 k 和 N，你还记得 Kanan 不喜欢 SVM 的高维。这对大数据很有帮助，对吗？是的，它的计算成本有点高。所以它在处理大型数据集时表现不佳。计算昂贵，但也不是超级昂贵。并不可怕。你可以在相当大的数据上使用它，但它真正的优势在于中等大小的数据，比如中等大小的数据，很多维度的数据。SVM 在这方面非常出色。我说的中等规模，是指几千到一万个数据点，这取决于你的电脑性能如何。是的，同理。有了大型数据集，就更难了。没错这也是他们挣扎的原因之一它们不仅计算成本高，而且如果你有大量数据点，找到分离空间也会变得更加困难。因此，数据集可以非常大，但也可以分离得很好，这要看具体情况。但在很多情况下，收集的数据越多，数据的随机性就越大，数据自然也就越嘈杂。因此，你必须更加小心。这又回到了参数调整的问题上，根据数据集的大小来选择 C。所以要允许更多的噪音，允许更多的违规行为。是的嗯，我一开始就提到了 SVM。非常笼统嗯？所以它们表现得很好。我的意思是，这是我们可以说的，因为我们通常是一个相当不错的分类器。我们说过决策树，单个的决策树，它们本身的 SVM 性能并不是很好。你不需要任何形式的套袋方法或类似方法。它们本身的表现相当不错，所以它们是相当不错的分类器，也是一种模型，如果这样说有意义的话。我们有吗？我们有这个。我们有这个我还有两个缺点是的 抱歉 Yes.Sorry.可解释性？是的可解释性非常困难。它的能力。所以与决策树相比，比如与 K 和 N 相比，它们都非常容易解释。你注意到了吗，当我向你解释它们时，真的很容易，对吗？作为虚拟机，你可以很快理解它们。我的意思是，我希望你能理解它们是如何工作的，但如果你想象一下要写一份商业报告，试图解释 SVM 是如何工作的，然后你开始解释：好吧，我们有这些支持向量，它们就像这些重要的点。然后我们使用一个核技巧。我们把它推到一个更高的特征空间。然后我们使用一个分割超平面。这真的有点棘手吧。所以它们更难解释。它们也很黑箱。不是一个很大的盒子，但我的意思是，是的。所以你可以在一定程度上解释它们。所以它们不像神经网络那样完全是黑盒子。但是，SVMs 要想知道它们究竟是如何做出决策的，尤其是在维度较多的情况下，难度可想而知。因此，如果是高维情况，有大量数据点，那么即使你掌握了技术知识，也很难对其进行解释和研究。因此，我区分可解释性和黑盒模型的方法是，对我来说，可解释性是指能够传达模型是如何工作的。而黑箱模型是指，即使你掌握了技术知识，即使你是模型的建立者，你又能多好地研究模型，并解释它为什么会做出这样的决定呢？因此，我还想补充一点，那就是规模敏感性。这一点非常重要。每当你有一个模型，我们正在寻找一种多维距离，这就是我们所做的一切。回想一下你的最大边际分类器。我们看的是线，我们看的是点到线的距离。因此，只有在刻度相同的情况下，这才能起作用。因此，你必须做大量细致的清理工作。你必须清除异常值。你必须移除或识别这些有影响的点。你还必须处理数据中可能存在的不同尺度。因此，这对 SVM 非常重要。而我们讨论过的决策树并不关心尺度问题。所以这两者之间会有很大的区别。就是这两点。最后，我还想补充一个优势，那就是可以用于回归和分类。这也是一个非常重要的例子。还记得我们刚才说的逻辑回归吗？逻辑回归是一种二元分类器。完全正确。这就是模型的作用。我们只谈线性回归。我们只有一个数字输出。SVM 可以调整为同时用于分类和回归问题。我们在这里只介绍了分类，但也可以用于回归。因此，SVM 与决策树的工作方式类似，基本上都是预测每个新测试点的数值，而不是类别成员。从这个意义上说，它们的工作方式与决策树类似。但它们可以同时用于这两种情况，这就使得它们具有相当大的通用性。用途广泛。总的来说，svm 非常适合高维数据，也适合中等规模的数据集。功能相当强大，但很难解释，也很难调整。这就是高度概括。是的。对是的。是的。是的原因是想想你认为你决定的空间。决定。所以，如果你想在这个空间中画一个超平面，如果你有混合数据，理论上可能行得通，但可能会完全搞砸你的优化问题。所以我还没见过在分类数据、二进制数据或类似数据上运行良好的 SVM。所以数字数据是最合适的。嗯。所以，如果我们使用的是分类数据，就像我们希望对其进行热编码，使其成为数值数据，这会增加更多维度。所以你所以你可以总是有点小心翼翼的。你到底想说什么？通过一个热编码 你基本上创造了很多二进制变量而二进制变量对于这类模型来说真的很难处理。因为如果你回想一下我们的例子，画出这些点，然后在它们之间画一条线，如果你有二进制数据。这些点其实并不分散。没有真正的空间让你画线。因此，理论上有很多模型适用于二进制数据，所以我相信你能做到。这个变量对整个模型的影响有多大？另一个问题好吧你满意吗？满意差不多吧满意吗？很好 Very good.如果你还有其他问题，我还会再呆几分钟。否则明天见另外，请记住下周是我们的最后一堂课。所以接下来我们将介绍神经网络。这将是最后一讲。因此，如果你有任何问题 任何最后的课程问题 请在下周带来。然后我们再讨论好的 谢谢 Okay.谢谢