# 关于机器学习的12件有用的事情

机器学习算法可以通过从例子中归纳出如何执行重要任务。在手动编程不可行的情况下，这通常是可行和经济的。随着越来越多的数据可用，可以解决更多雄心勃勃的问题。因此，机器学习在计算机真诚等领域得到了广泛的应用。然而，开发成功的机器学习应用程序需要大量的“黑色艺术”，这在教科书中是很难找到的。

我最近读到了华盛顿大学的一篇令人惊叹的技术论文，题为“它总结了机器学习研究人员和实践者所学到的12个关键教训，包括要避免的陷阱、要关注的重要问题以及常见问题的答案。我想在本文中分享这些经验教训，因为它们在考虑解决下一个机器学习问题时非常有用。

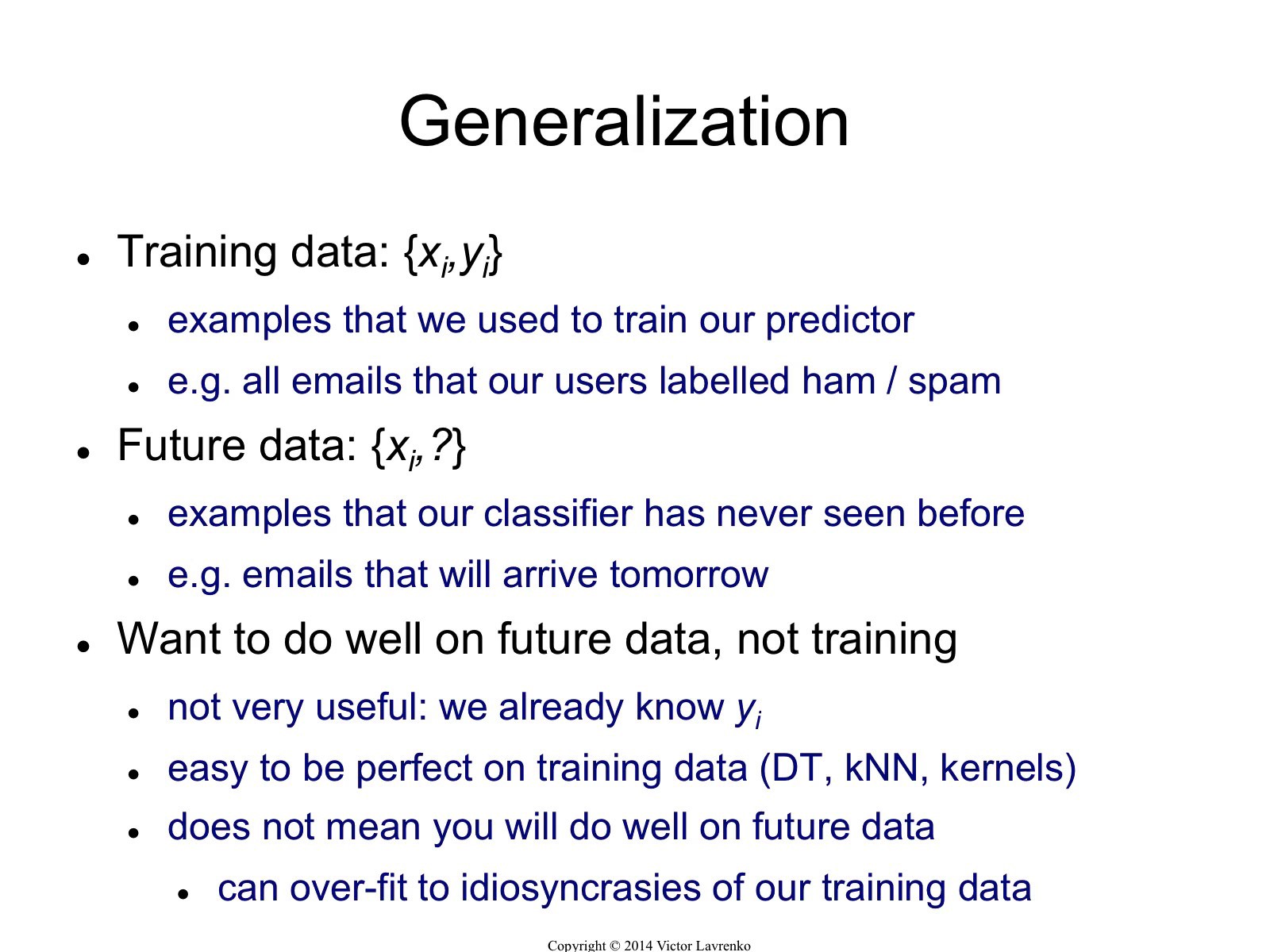
#### 1 - 学习=表现+评估+优化

所有机器学习算法通常由3个部分组成：

* 表示：分类器必须用计算机可以处理的某种形式语言表示。相反，为学习者选择一个表示就等于选择一组它可能学习的分类器。这个集合称为学习者的假设空间。如果分类器不在假设空间中，则无法学习。一个相关的问题是如何表示输入，即使用什么特性。
* 评价：需要一个评价函数来区分好分类器和坏分类器。算法内部使用的评估函数可能不同于我们希望分类器优化的外部评估函数，以便于优化，并且由于下一节讨论的问题。
* 优化：最后，我们需要一种方法在语言中的分类器之间搜索得分最高的分类器。优化技术的选择是学习者学习效率的关键，如果评价函数有多个最优值，则有助于确定所生成的分类器。对于新手来说，开始使用现成的优化器是很常见的，后来这些优化器被定制的优化器所取代。

#### 2 - 一般化才是最重要的

机器学习的基本目标是超越训练集中的例子。这是因为，无论我们有多少数据，在测试时我们都不太可能再看到这些精确的例子。在训练场上做得好很容易。机器学习初学者最常见的错误是对训练数据进行测试，产生成功的错觉。如果选择的分类器在新的数据上进行测试，通常不会比随机猜测更好。所以，如果你雇了人来构建一个分类器，一定要把一些数据留给你自己，并在上面测试他们给你的分类器。相反，如果你被雇来构建一个分类器，那么从一开始就把一些数据放在一边，只在最后测试你选择的分类器，然后在整个数据上学习你的最终分类器。



#### 仅凭3 - 数据是不够的

作为目标的泛化有另一个主要后果：数据本身是不够的，不管你有多少数据。

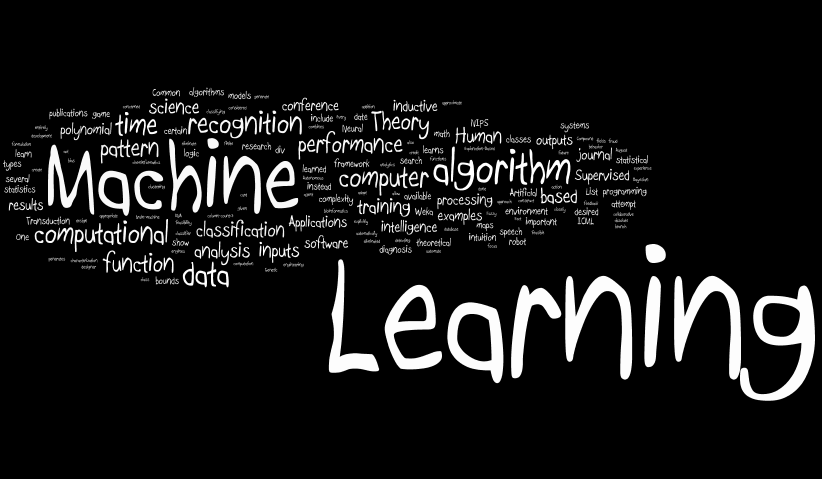
这似乎是一个相当令人沮丧的消息。那我们怎么能指望学到什么呢？幸运的是，我们想在现实世界中学习的函数并不是从所有数学上可能的函数集合中统一得出的！事实上，非常一般的假设-类似平滑度，相似的类具有相似的类，有限的依赖性，或者有限的复杂性，通常是很好的，这也是机器学习如此成功的主要原因。与演绎法一样，归纳法（学习者的行为）也是一种知识杠杆：它将少量的输入知识转化为大量的输出知识。归纳法是一个比演绎法强大得多的杠杆，它需要更少的输入知识才能产生有用的结果，但它仍然需要更多的零输入知识才能发挥作用。而且，和任何杠杆一样，我们投入的越多，我们能得到的就越多。

回顾过去，学习中对知识的需求并不奇怪。机器学习不是魔术，它不能从无到有。它所做的是从更少中得到更多。编程，就像所有的工程一样，是一项艰巨的工作：我们必须从头开始。学习更像是耕作，让大自然做大部分的工作。农民把种子和养分结合起来种植庄稼。学习者将知识与数据结合起来，以开发程序。

#### 4 - 过盈有许多面

如果我们掌握的知识和数据不足以完全确定正确的分类器呢？然后我们冒着仅仅产生一个不基于现实的分类器（或部分分类器）的风险，这个分类器只是在数据中编码随机的怪癖。这个问题被称为过度拟合，是机器学习的错误。当你的学习者输出的分类器在训练数据上100%准确，但在测试数据上只有50%准确时，当事实上它可能输出的分类器在两个数据上都是75%准确时，它就过拟合了。

机器学习中的每个人都知道过拟合，但它有很多形式，并不是很明显。理解过度拟合的一种方法是将泛化误差分解为偏差和方差。偏见是一种学习者不断地学习相同错误的东西的倾向。方差是一种学习随机事物的趋势，与真实信号无关。线性学习者有很高的偏差，因为当两个类之间的边界不是一个超平面时，学习者无法诱导它。决策树不存在这个问题，因为它们可以表示任何布尔函数，但另一方面，它们会遭受高方差的困扰：在同一现象产生的不同训练集上学习的决策树往往非常不同，而事实上它们应该是相同的。



交叉验证有助于克服过度拟合，例如使用它来选择要学习的决策树的最佳大小。但这并不是万能的，因为如果我们用它来做太多的参数选择，它本身就可能变得过大。

除了交叉验证外，还有许多方法可以克服过度拟合。最流行的方法是在评价函数中加入一个正则化项。例如，这可以惩罚结构更复杂的分类器，从而有利于空间更小的分类器。另一个选择是在添加新结构之前执行一个像卡方检验这样的统计显著性测试，以确定在有或没有这种结构的情况下类的分布是否真的不同。当数据非常稀少时，这些技术特别有用。尽管如此，你还是应该对某一特定技术“解决”了过度拟合问题的说法持怀疑态度。很容易避免过度拟合（方差），陷入相对误差的不足（偏差）。同时避免这两种情况需要学习一个完美的分类器，而且没有一种技术可以永远做到最好（没有免费的午餐）。

#### 5 - 直觉在高维度上失败

在过度拟合之后，机器学习中最大的问题是维数的诅咒。这个表达式是Bellman在1961年发明的，它指的是当输入是高维时，许多在低维下运行良好的算法变得难以处理但在机器学习中，它指的更多。随着示例的维数（特征数）的增加，正确的泛化变得指数级困难，因为固定大小的训练集覆盖了输入空间的一小部分。

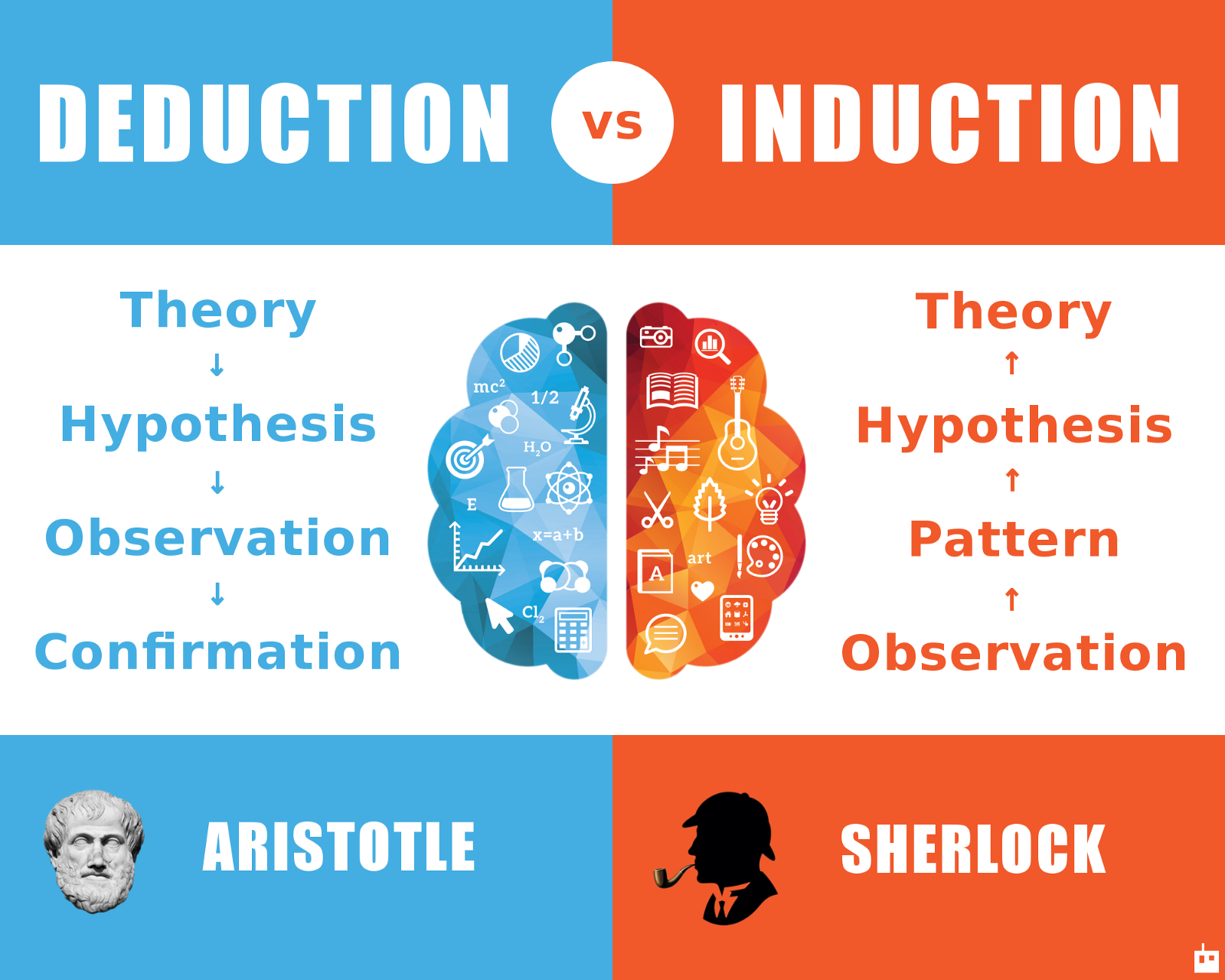
高维的普遍问题是，我们来自三维世界的直觉，往往不适用于高维世界。在高维中，多变量高斯分布的大部分质量并不接近平均值，而是在其周围越来越远的“壳”中；而高维橙色的大部分体积在皮肤中，而不是果肉中。如果一个高维超立方体中均匀分布着大量的例子，那么除了某些维数外，大多数例子都更接近于超立方体的一个面，而不是它们的最近邻。如果我们用超立方体来描述超球面，在高维中，几乎所有超立方体的体积都在超球面之外。对于机器学习来说，这是一个坏消息，其中一种类型的形状常常被另一种类型的形状所近似。

建立一个二维或三维的分类器是很容易的，我们可以通过视觉检查在不同类别的例子之间找到一个合理的边界。但在高维空间里，很难理解发生了什么。这反过来又使设计一个好的分类器变得困难。天真地说，人们可能会认为收集更多的特性不会有什么坏处，因为最坏的情况下它们不会提供关于类的新信息。但事实上，它们的好处可能会被维度诅咒所抵消。

#### 6 - 理论上的保证并不是看起来的那样

机器学习的论文充满了理论保证。最常见的类型是对确保良好泛化所需的示例数量的限制。这些保证你该怎么办？首先，值得注意的是，它们甚至是可能的。归纳法传统上与演绎法相反：在演绎法中，你可以保证结论是正确的；在归纳法中，所有赌注都被取消。或者这是许多世纪以来的传统智慧。最近几十年的一个主要发展是认识到，事实上，我们可以对归纳的结果有保证，特别是如果我们愿意接受概率保证的话。

我们得小心这样的束缚意味着什么。例如，它并没有说，如果你的学习者返回了一个与特定训练集一致的假设，那么这个假设可能很好地概括了。所说的是，给定足够大的训练集，学习者很有可能会返回一个概括良好的假设，或者找不到一个一致的假设。边界也没有说明如何选择一个好的假设空间。它只告诉我们，如果假设空间包含真正的分类器，那么学习者输出错误分类器的概率会随着训练集的大小而降低。如果我们缩小假设空间，界限会改善，但它包含真正分类器的机会也会缩小。



另一种常见的理论保证是渐近的：在给定无限数据的情况下，保证学习者输出正确的分类器。这是令人放心的，但是选择一个学习者而不是另一个学习者是鲁莽的，因为这是渐进的保证。实际上，我们很少处于渐近状态（也称为“渐近”）。而且，由于上面讨论的偏差-方差权衡，如果在给定的无限数据下，学习者A比学习者B好，那么B通常比给定的有限数据好。

理论保证在机器学习中的主要作用不是作为实际决策的标准，而是作为理解的源泉和算法设计的驱动力。在这种能力下，它们是非常有用的；事实上，理论和实践的密切互动是机器学习多年来取得如此大进步的主要原因之一。但需要提醒的是：学习是一个复杂的现象，仅仅因为学习者有理论依据并在实践中发挥作用并不意味着前者就是后者的原因。

#### 7 - 功能工程是关键

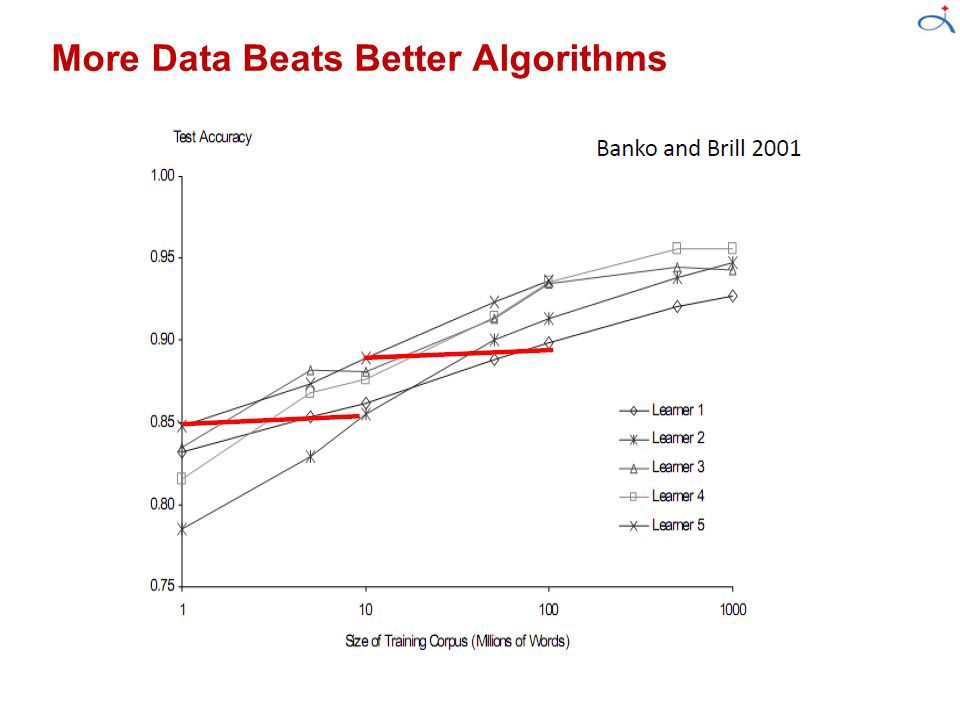
最后，有些机器学习项目成功了，有些失败了。有什么不同？最重要的因素很容易就是使用的特性。如果你有许多独立的特点，每一个都与课程有很好的关联，那么学习是很容易的。另一方面，如果类是一个功能非常复杂的函数，您可能无法学习它。通常，原始数据的形式不适合学习，但您可以从中构造特性。这通常是机器学习项目中大部分工作的地方。它通常也是最有趣的部分之一，直觉、创造力和“黑色艺术”与技术性的东西一样重要。

初学机器的人常常会惊讶于，在机器学习项目中，真正花在机器学习上的时间是多么的少。但是，如果考虑到收集数据、集成数据、清理数据和预处理数据是多么耗时，以及在功能设计中可以进行多少尝试和错误，这是有意义的。此外，机器学习不是建立数据集和运行学习者的一次性过程，而是运行学习者、分析结果、修改数据和/或学习者以及重复的迭代过程。学习往往是其中最快的部分，但那是因为我们已经很好地掌握了它！特征工程更为困难，因为它是特定领域的，而学习者在很大程度上是通用的。然而，两者之间没有明显的边界，这也是最有用的学习者是那些有助于整合知识的人的另一个原因。

#### 8 - 更多的数据胜过更聪明的算法

在大多数计算机科学中，时间和内存是两种主要的有限资源。在机器学习中，有第三种方法：训练数据。哪一个是瓶颈已经从十年变成了十年。在20世纪80年代，它往往是数据。今天常常是时候。大量的数据是可用的，但是没有足够的时间来处理它，所以它就不用了。这就导致了一个悖论：尽管原则上，更多的数据意味着可以学习更复杂的分类器，但实际上，更简单的分类器最终会被使用，因为复杂的分类器需要花费太长时间才能学习。部分答案是提出快速学习复杂量词的方法，事实上在这方面已经取得了显著的进展。

使用聪明算法的一部分原因比你预期的要小，因为，在第一次近似下，它们都是一样的。当你将表示视为不同的规则集和神经网络时，这是令人惊讶的。但事实上，命题规则很容易被编码成神经网络，而且其他表示之间也存在类似的关系。所有的学习者基本上都是通过将附近的例子分组到同一个类中来工作的；关键的区别在于“附近”的含义。使用非均匀分布的数据，学习者可以产生广泛不同的边界，同时在重要的区域（有大量训练例子的区域）做出相同的预测，因此也就是大多数文本示例可能出现的地方）。这也有助于解释为什么强大的学习是不稳定的，但仍然准确。

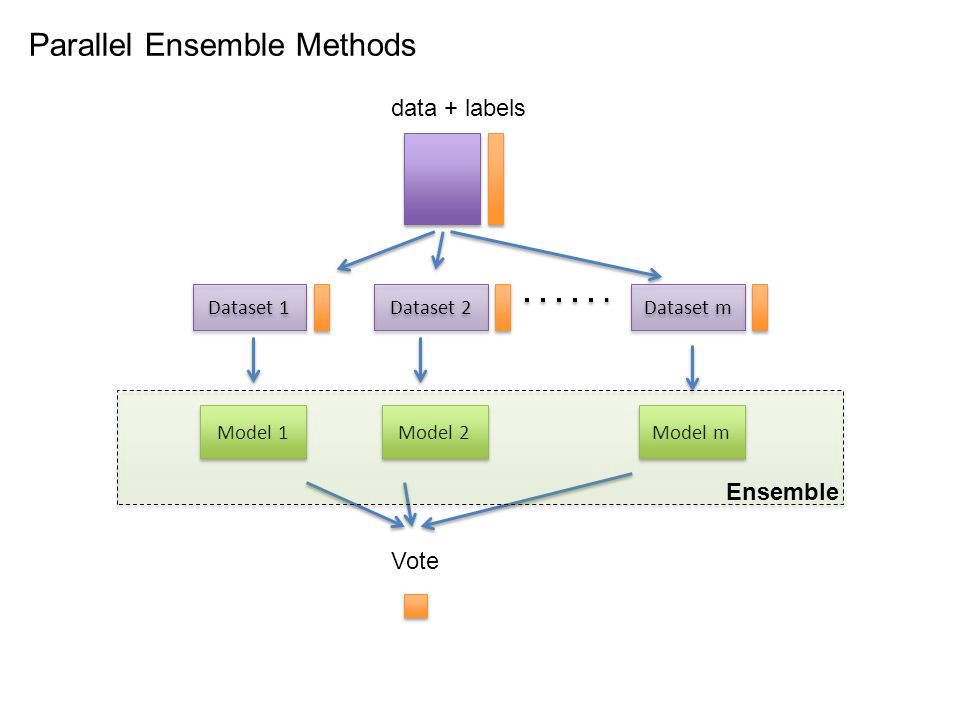


通常，首先尝试最简单的学习者是值得的（例如，logistic回归之前的朴素贝叶斯，支持向量机之前的k近邻）。更成熟的学习者很有诱惑力，但他们通常更难使用，因为他们有更多的旋钮，你需要转动它们才能获得好的结果，而且因为他们的内部更不透明。

学习者可以分为两大类：一类是表示具有固定大小的，如线性分类器；另一类是表示可以随着数据增长的，如决策树。固定规模的学习者只能利用这么多数据。变大小学习者原则上可以在给定足够数据的情况下学习任何函数，但在实际应用中可能不会，因为算法或计算成本的限制。此外，由于维数灾难，没有足够的数据量就足够了。基于这些原因，聪明的算法——那些能充分利用数据和计算资源的算法——往往最终会有回报，只要你愿意付出努力。在设计学习者和学习分类器之间没有明显的边界；相反，任何给定的知识片段都可以编码在学习者中或从数据中学习。因此，机器学习项目往往最终成为学习者设计的重要组成部分，而实践者需要在这方面具备一些专业知识。

#### 9 - 学习许多模型，而不仅仅是一个

在机器学习的早期，每个人都有自己喜欢的学习者，还有一些先验的理由相信它的优越性。大部分的努力都花在尝试它的多种变体和选择最好的。然后系统的实证比较表明，最佳学习者在不同的应用程序中有所不同，包含许多不同学习者的系统开始出现。现在，人们开始尝试许多学习者的多种变体，但仍然只选择了最好的一种。但后来研究人员注意到，如果我们不选择找到的最佳变异，而是将许多变异组合起来，结果会更好，通常会好得多，而且对用户来说，只需付出很少的额外努力。



创建这样的模型集合现在是标准的。在最简单的技术中，我们称之为bagging，我们只需通过重采样生成训练集的随机变化，在每个训练集上学习一个分类器，然后通过投票来组合结果。这是因为它大大减少了方差，而只稍微增加了偏差。在boosting中，训练的例子有权重，这些权重是不同的，因此每个新的分类器都将重点放在前面那些容易出错的例子上。在叠加过程中，单个分类器的输出成为“更高层次”学习者的输入，学习者知道如何最好地组合它们。

许多其他技术存在，趋势是越来越大的合奏。在Netflix奖中，来自世界各地的团队竞相构建最佳视频推荐系统。随着比赛的进行，团队发现他们通过将自己的学习者和其他团队结合起来获得了最好的结果，并合并成越来越大的团队。优胜者和亚军都是由100多名学习者组成的叠合奏，将这两个合奏结合起来进一步提高了成绩。毫无疑问，我们将来还会看到更大的。

#### 10 - 简单并不意味着准确

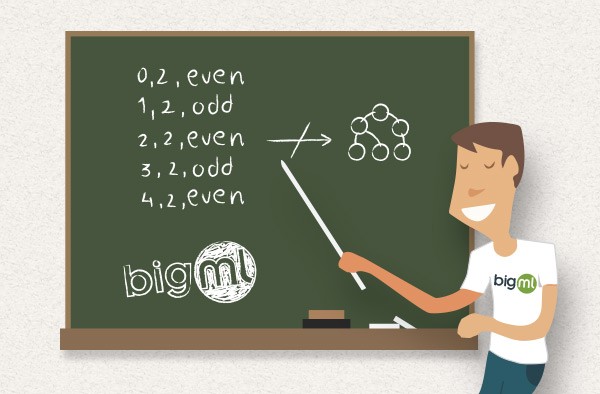
奥卡姆的剃刀（razor）有一个著名的观点，即实体不应在必要性之外成倍增长。在机器学习中，这通常被认为是指，给定两个具有相同训练错误的分类器，其中较简单的两个分类器可能具有最低的测试错误。这一论断的据称证据在文献中经常出现，但事实上有许多反例，而且“没有免费午餐”定理暗示它不可能是真的。

我们在上一节看到了一个反例：模型集合。即使在训练误差达到零之后，增强集成的泛化误差仍然通过添加分类器而得到改善。因此，与直觉相反，一个模型的参数数量与其过度拟合的倾向之间没有必要的联系。

相反，更复杂的视图将复杂性与假设空间的大小等同起来，基于较小的空间允许假设用较短的代码来表示。像上面关于理论保证的一节中那样的界限可能会被视为意味着较短的假设会得到更好的概括。这可以通过在我们有一些先验偏好的空间中为假设分配较短的代码来进一步细化。但将此视为精确性和简单性之间权衡的“证明”是循环推理：我们通过设计使我们更喜欢的假设更简单，如果它们是准确的，那是因为我们的偏好是准确的，而不是因为假设在我们选择的表示中是“简单的”。

#### 11 - 可代表并不意味着可学习

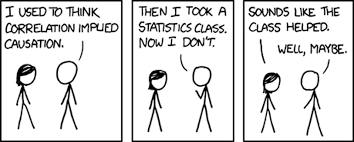
本质上，在可变大小的学习者中使用的所有表示都有相关的定理：“每个函数都可以用这种表示来表示或近似地紧密地逼近。”然而，仅仅因为一个函数可以被表示并不意味着它可以被学习。例如，标准的决策树学习者不能学习比有训练例子更多的叶子的树。在连续空间中，使用一组固定的基元来表示即使是简单的函数，通常也需要无穷多个组件。



此外，如果假设空间有许多评价函数的局部最优值（通常是这样），学习者可能找不到真正的函数，即使它是可表示的。给定有限的数据、时间和记忆，标准学习者只能学习所有可能函数的一小部分，而对于具有不同表示的学习者来说，这些子集是不同的。因此，关键问题不是“它能被代表吗？”？答案往往是微不足道的，但是“它能被学习吗？”？“尝试不同的学习者是值得的（也可能是将他们结合起来）。

#### 12 - 相关性并不意味着因果关系

关联并不意味着因果关系的观点经常被提出，这也许不值得一味地重复。但是，尽管我们讨论过的这类学习者只能学习相关性，但他们的结果通常被视为表示因果关系。这是不是错了？如果是，那人们为什么要这么做呢？



通常，学习预测模型的目的是将其作为行动指南。如果我们发现啤酒和尿布经常在超市一起买，那么也许把啤酒放在尿布区旁边会增加销量。但如果不真正做实验，就很难说了。机器学习通常应用于观察数据，其中预测变量不受学习者的控制，而实验数据则相反。一些学习算法可能从观测数据中提取因果信息，但其适用性受到很大限制。另一方面，相关性是潜在因果关系的标志，可以作为进一步研究的指导。

#### Conclusion

Like any discipline, machine learning has a lot of “folk wisdom” that can be hard to come by, but is crucial for success. Professor Domingos’ paper summarized some of the most salient items that you need to know.