机器学习那些事*

作者：佩德罗·多明戈斯（Pedro Domingos）
译者：刘知远

关键词：机器学习

深入了解所需的“民间知识”可推进机器学习的应用。

机器学习系统自动地从数据中学习程序。与手工编程相比，这非常吸引人。在过去的20年中，机器学习已经迅速地在计算机科学等领域普及。机器学习被用于网络搜索、垃圾邮件过滤、推荐系统、广告投放、信用评价、欺诈检测、股票交易和药物设计等应用。麦肯锡全球研究院（the McKinsey Global Institute）最近一份报告指出，机器学习（又称数据挖掘或者预测分析）将驱动下一轮创新[15]。现在已经有几本优秀的机器学习教材书可以供感兴趣的研究者和实践者使用（例如米切尔（Mitchell）和维滕（Witten）等人的教材[16,24]）。但是，成功使用机器学习所应掌握的大量“民间知识”并没有出现在这些教材中。因此，很多机器学习项目浪费了大量时间，甚至最终也没有得到理想的结果。其实这些“民间知识”非常容易理解。本文的目的就是介绍这些知识。

机器学习有许多不同的类型，但为了展示方便，本文将主要介绍其中最常用的类型：分类。但是，本文所探讨的问题适用于所有的机器学习类型。一个分类器（classifier）是一个系统，系统输入是一个包括若干离散或连续的特征值（feature values）的向量，系统输出是一个离散值，代表分类的类别（class）。例如，一个垃圾邮件过滤器会将邮件信息分类到“是垃圾邮件”和“不是垃圾邮件”两个类别中。它的输入可以是一个布尔向量$\mathbf{x} = (x_1,\cdots,x_j,\cdots,x_d)$，其中如果词典中的第$j$个词出现在该邮件中，则$x_j$=1，否则$x_j$=0。一个学习器将一个训练集（training set）样例（x_i,y_i）作为输入，其中$x_i = (x_{i,1},\cdots,x_{i,d})$是观察到的输入，$y_i$是相应的输出，学习器的输出是一个分类器。对学习器的检验就是判断它输出的分类器是否能够对将来的输入样例x_t输出正确的y_t（例如，垃圾邮件过滤器是否能够将训练时没有见过的邮件信息正确分类）。

学习=表示+评价+优化

假设有一个应用，你认为机器学习有可能在其中发挥作用。那么，你面临的第一个问题是各种机器学习算法令人眼花缭乱。应挑选使用哪一个？现在有成千上万的机器学习算法，每年还有成百上千的新算法发表出来。避

* 本文译自*Communications of the ACM* 2012年第10期的“A Few Useful Things to Know About Machine Learning”一文。

免迷失在这么多算法中的关键是，要认识到这些算法都是由三个部分组成的，分别是：

表示（Representation） 一个分类器必须用计算机可以处理的某种形式语言来表示。反过来讲，为学习器选择一种表示，就意味选择一个特定的分类器集合。学习器可能学出的分类器只能在这个集合中。这个集合被称为学习器的假设空间（hypothesis space）。如果某个分类器不在该空间中，它就不可能被该学习器学到。与此相关的一个问题是如何表示输入，即使用哪些特征，本文稍后介绍。

评价（Evaluation） 我们需要一个评价函数（亦称为目标函数或打分函数）来判断分类器的优劣。机器学习算法内部使用的评价函数和我们希望分类器进行优化的外部评价函数有所不同。这是为了便于优化，接下来会讨论。

优化（Optimization） 最后，我们需要一个搜索方法，能够在假设空间中找到评价函数得分最高的那个分类器。优化技术的选择对学习器效率至关重要；而当评价函数有多个最优结果时，优化技术也有助于从中选择。初学者通常会采用现成的优化方法，之后再用定制专门的优化方法来替代。

表1展示了三个组成部分常见的例子。例如，对一个测试样例，k-近邻方法会寻找它的k个最相似的训练样例，并将这些样例中出现最多的类别作为该测试样例的类别。超平面方法会为每一个类别构造一个特征的线性组合，并将得分最高的组合所对应的类别作为预测结果。决策树方法会在树上的每个内部节点测试一个特征，每个特征值会对应一个分支，而不同的叶子节点会对应不同的类别。算法1展示了一个极简单的二分类决策树学习器，其中使用了信息增益（information gain）和贪心搜索（greedy search）[20]。InfoGain(x_j, y)表示特征x_j与类别y之间的互信息（mutual information）。MakeNode(x, c_0, c_1)会返回一个测试特征x的节

表1 机器学习算法的三个组成部分

表 示	评 价	优 化
基于实例的方法	准确/错误比率	组合优化
近邻方法	精确率和召回率	贪心搜索
支持向量机	平方误差	柱搜索
超平面方法	似然（likelihood）	分支界限法
朴素贝叶斯	后验概率	连续优化
逻辑斯蒂回归	信息增益	无约束
决策树方法	K-L距离	梯度下降
规则集的方法	成本/效用	共轭梯度
命题规则	利润	拟牛顿法
逻辑程序		有约束
神经网络		线性规划
图模型		二次规划
贝叶斯网络		
条件随机场		

表2 决策树算法

算法1：LearnDT(*TrainSet*)
if *TrainSet*中所有的样例有相同的类别y_* then
  return MakeLeaf(y_*)
if 不存在特征x_j能够满足InfoGain(x_j y)>0 then
  y_*←*TrainSet*中最常见的类别
  Return MakeLeaf(y_*)
x_*←argmax(x_j) InfoGain(x_j,y)
TS_0←*TrainSet*中x_*=0的样例
TS_1←*TrainSet*中x_*=1的样例
Return MakeNode(x_*,LearnDT(TS_0),LearnDT(TS_1))

点，该节点以c_0作为x=0时的孩子节点，以c_1作为x=1时的孩子节点。

当然，并不是表1中从各列选出元素的相互组合都同样有意义。例如，离散表示很自然地与组合优化相结合；而连续表示则与连续优化相结合。然而，很多学习器同时包含离散和连续的部分。实际上，所有可能的组合也都快被实现过了。

大部分教科书是以表示为视角组织内容的。这通常会让人忽略掉一个事实，即其他部分也同样重要。虽然对如何在每个部分做出选择并没有简单的秘诀，但本文将涉及其中几个重要的问题。正如我们以后会看到的那样，机器学习项目中的某些选择甚至比学习器的选择更加重要。

泛化（Generalization）很重要

机器学习的基本目标是对训练集合中样例的泛化。这是因为，不管我们有多少训练数据，在测试阶段这些数据都不太可能会重复出现。（注意，如果在词典中有100000个词，前述垃圾邮件过滤器将会有种2^{100000}种可能的不同输入）。在训练集上表现出色其实很简单（只要记住这些训练样例即可）。机器学习初学者最常犯的错误是在训练数据上做测试，从而产生胜利的错觉。如果这时将选中的分类器在新数据上测试，它往往还不如随机猜测准确。因此，如果你雇人来训练分类器，一定要自己保存一些数据，来测试他们给你的分类器的性能。相反，如果你被人雇来训练分类器，一开始就应该将一部分数据取出来，只用它们来测试你选择的分类器性能，接下来再在整个数据上学习你最终的分类器。

你的分类器可能会在不知不觉中受到测试数据的影响，例如你可能会使用测试数据来调节参数并做了很多调节（机器学习算法有很多参数，算法成功往往源自对这些参数的精细调节，因此这是非常值得关注的问题）。当然，保留一部分数据用于测试会减少训练数据的数量。这个问题可以通过交叉验证（cross-validation）来解决：将训练数据随机地等分为若干份（如10份），其中的每一份均可用作测试，而剩下的数据用作训练，然后将每个学习的分类器在它没见过的样例上进行测试，将测试结果取平均后，就可用来评价不同参数设置的性能。

在机器学习研究早期，划分训练和测试数据的必要性没有受到广泛重视。部分的原因是，如果学习器的表示很有限（比如超平面表示），则训练误差和测试误差差别不大。但是对于比较灵活的分类器（比如决策树），甚至拥有大量特征的线性分类器，则训练和测试数据严格分开是非常必要的。

需要注意的是，将泛化作为目标给机器学习带来一个有趣的结果。与其他大部分优化问题不同，机器学习无法获得希望优化的那个函数！我们不得不用训练误差来代替测试误差（作为目标函数），而这非常危险（如何处理这个问题稍后会介绍）。从积极的角度讲，由于这个目标函数不过是真实目标的替身，我们也许没有必要完全优化它；而实际上，通过简单的贪心搜索返回的局部最优也许比全局最优更好。

仅有数据还不够

将泛化作为目标带来的另外一个重要结果是，仅有数据还不够，无论你有多少。考虑要从100万样例中学习一个包含100个变量的布尔函数。此时将有2^{100}-10^6 个样例的类别是不知道的[1]。你如何确定那些样例的类别呢？在没有更进一步信息的情况下，除了抛硬币随机猜之外将束手无策。哲学家大卫·休谟（David Hume）在200多年前首次指出这一问题（以某种不同的形式），但直到今天机器学习中的很多错误仍是由于没有意识到

[1] 这里2^{100}表示100个布尔变量的所有可能情况的个数，而10^6表示已经看到的100万样例，因此有$2^{100}-10^6$个可能情况是没有看到过的，因此也不知道它们的类别。

这一问题造成的。每个学习器都必须包含一些数据之外的知识或假设（assumption），才能够将数据泛化。这一概念被沃尔伯特（Wolpert）形式化为“没有免费的午餐”定理。根据该定理，没有学习器能够比在所有可能的布尔函数中随机猜测的结果更优[25]。

这似乎是一个非常让人失望的消息。那我们还能指望能学到什么东西吗？幸运的是，在真实世界中，我们要学习的函数并非均匀地来自所有可能的函数！实际上，一些非常泛泛的假设——比如平滑（smoothness），相似的样例有相似的类别，有限依赖，或者有限复杂度——通常足够起很大作用，这也是机器学习能够如此成功的重要原因。如同演绎（deduction）一样，归纳（induction，正是学习器所做的）起到知识杠杆的作用——它将少量的输入知识转化成为大量的输出知识。归纳是比演绎强大得多的杠杆，只要求很少的输入知识就可以产生有用的结果，但是它终归不能在没有知识的情况下工作。而且就像任何杠杆一样，输入越多，我们得到的输出就越多。

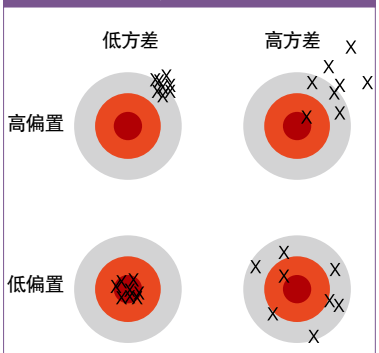
从中可以得到的一个推论是，选择表示的关键标准之一是，它比较易于表达什么类型的知识。例如，如果我们拥有大量关于在我们的领域是什么造成样例相似的知识，基于实例的方法也许就是合适的选择。如果我们拥有概率依赖的知识，图模型则比较适合。如果我们拥有每个类别要求的先决条件的知识，“If…Then…（如果…那么…）”规则的表示也许是最好的选择。在这一点上，最有用的学习器是那些并非将假设固化在其中，而是允许我们用显式规定假设，在大范围改变假设，并自动将其体现在学习中（例如采用一阶逻辑[21]或者语法[6]）的学习器。

说到这里，学习需要知识，这并不让人惊讶。机器学习不是魔术，它无法凭空变出东西。它所做的是由少变多。编程就像所有的工程技术那样，意味着大量的工作，必须从头开始建造一切。而机器学习更像是种田，它让大自然做大部分工作。农夫将种子与肥料混合种出庄稼。学习器将知识和数据结合“种出”程序。

过拟合（Overfitting）有多张面孔

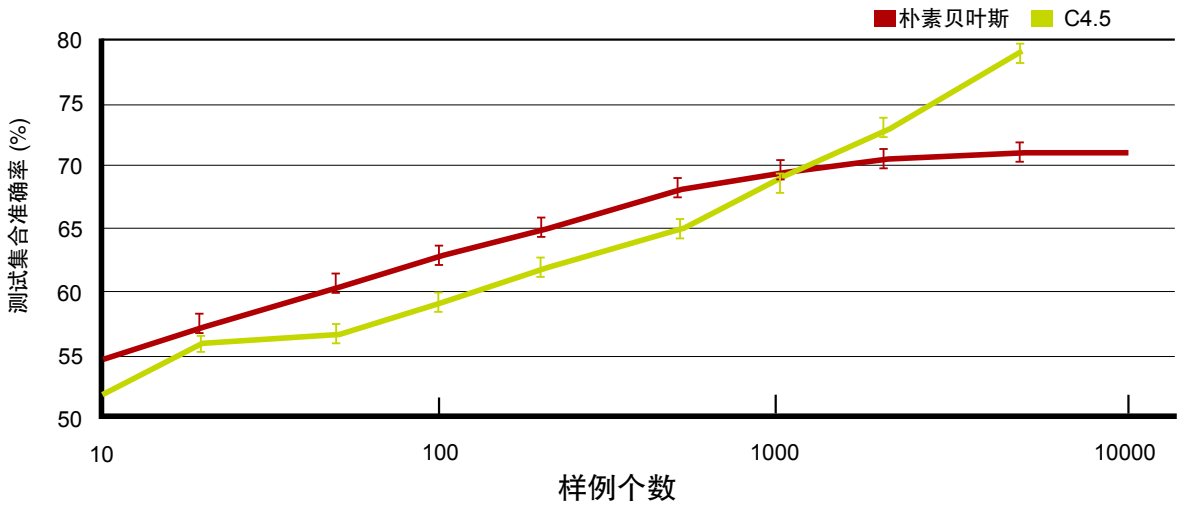
如果我们拥有的知识和数据并不足以学习出正确的分类器，将会怎样呢？我们就得冒风险构建一个分类器（或者其中一部分），这个分类器并非建立在现实基础上，而是将数据随机表现加以解读。这个问题称为过拟合，它是机器学习中的棘手问题。当你的学习器输出的分类器在训练数据上准确率为100%，而在测试数据上仅有50%的时候（而本来可以学到一个分类器能够在两个数据上均达到75%的准确率），说明这个分类器发生过拟合了。

图1 扔飞镖中的偏置与方差



机器学习领域的每个人都了解过拟合，但过拟合会以多种并不明显的形式出现。一种理解过拟合的方式是将泛化误差（generalization error）分解为偏置（bias）和方差（variance）[9]。偏置度量了学习器倾向于一直学习相同错误的程度。方差则度量了学习器倾向于忽略真实信号、学习随机事物的程度。图1用朝板子扔飞镖作为类比进行了直观说明。一个线性学习器有较高的偏置，因为当两个类别的交界不是超平面的时候，这个学习器就无法进行归纳。决策树就不会有这个问题，因为它可以表示任意的布尔函数，但在另一方面，决策树会面临高方差的问题：在同一现象所产生的不同训练数据上学习的决策树往往差异巨大，而实际上它们应当是相同的。类似道理也适用于优化方法的选择上：与贪心搜索相比，柱搜索的偏置较低，但方差较高，原因是

图2 即使真正的分类器是一个规则集合，朴素贝叶斯方法仍然可以优于最好的学习器（C4.5规则）



柱搜索会尝试搜索更多的假设。因此，与直觉相反，一个学习能力更强的学习器并不见得比学习能力弱的效果更好。

图2示例说明了这一点[2]。即使真正的分类器是一个规则集合，但根据1000个样例学习的朴素贝叶斯学习器仍比一个规则学习器的准确率更高。甚至当朴素贝叶斯错误地假设分类面是线性的，也依然如此。这种情形在机器学习领域很常见：一个强错误假设比那些弱正确假设更好，这是因为后者需要更多的数据才能避免过拟合。

交叉验证可以帮助避免过拟合，例如通过交叉验证来选择决策树的最佳大小。但这不能彻底解决问题，因为假如我们利用交叉验证做太多的参数选择，它本身就会开始过拟合[17]。

除了交叉验证以外，还有很多方法可以避免过拟合。最常用的方法是对评价函数增加一个正则项（regularization term）。这样做可以惩罚那些包含更多结构的分类器，偏好更小的分类器，从而降低过拟合的可能性。另一个方案是在决定是否增加新的结构时进行诸如卡方测试（chi-squre）等统计显著性检验（statistical significance test），用来决定类别分布是否会因为增加这个结构而不同。当数据非常缺乏时，这些技术非常有用。然而，你应该对那些宣称某项技术“解决”了过拟合问题的说法持怀疑态度。我们会很容易在避免过拟合（或者说“方差”）时，造成另外一个相反的错误——欠拟合（underfitting，或者说“偏置”）。要学习一个完美的分类器来同时避免过拟合和欠拟合，事先又没有足够知识，这种情形下没有任何单一技术能够总是表现最好（没有免费的午餐）。

对过拟合的一个常见误解是认为它是由噪音造成的，例如有些训练样例的标注类别是错误的。这的确会加剧过拟合，因为分类器会调整分类面让那些样例保持在分类器认为正确的一侧。但是即使没有噪音依然会发生严重的过拟合。例如，假如我们学习一个布尔类型分类器，它是训练数据中所有标为“true”的样例的析取（disjunction）。（换句话说，这个分类器是一个析取范式（disjunctive normal form）的布尔类型公式，其中每一项是

[2] 训练样例含有64个布尔类型特征和1个根据一个集合的“如果…那么…”的规则集合计算得到的布尔类型的类别。图中的曲线是对100次运行结果的平均，每次对应不同的随机产生的规则集合。误差条（error bar）代表两个标准方差。具体细节请参考论文[10]。

某个特定训练样例的所有特征值的合取（conjunction）。）这个分类器对所有的训练样例都分类正确，但对测试样例中的每个正例都分类错误，不管训练数据是否有噪音。

多重检验（multiple testing）[13]问题与过拟合密切相关。标准的统计检验中只有一个假设被检验，而现代学习器在结束学习前会轻易地检验上百万个假设。因此，那些看上去很显著的结论实际并不如此。例如，一个连续十年跑赢市场的共同基金（mutual fund）看上去很引人注目。但当你发现，如果有1000家基金，每家都有50%的概率在某年跑赢市场，在这种情况下，极有可能会有一家基金能够凭侥幸而连续10次都跑赢市场。这个问题可以通过在显著性检验中将假设的个数考虑进去来解决，但这样也会导致欠拟合。更好的途径是控制错误接受的非零假设（non-null hypotheses）的比率，该方法通常被称为错误发现率（false discovery rate）方法[3]。

直觉不适用于高维空间

机器学习中紧接过拟合之后的最大问题就是维度灾难（curse of dimensionality）。这一概念是由贝尔曼（Bellman）在1961年首先提出的，用来描述以下事实：许多在低维空间表现很好的算法，当输入是高维度的时候，就变得计算不可行（intractable）了。但在机器学习领域，这有更多的意义。随着样例维度（即特征数目）的增长，正确泛化的难度会以指数级增加，原因是同等规模的训练集只能覆盖越来越少的输入空间比例。即使对于中等大小的100维布尔空间，一个包含1万亿样例的大型数据集合也只能覆盖输入空间的10^{-18}左右[3]。这体现了机器学习存在的必要性，也是它的难点所在。

更严格地讲，机器学习算法所（显式或隐式）依赖的基于相似度的推理在高维空间不再有效。现在考虑一个采用汉明距离（hamming distance）作为相似度度量的最近邻分类器，并设定样例的分类类别是$x_1 \wedge x_2$。如果没有其他特征，这是一个很容易的问题。但是当增加98个不相关的特征$x_3,\cdots,x_{100}$的时候，来自这些特征的噪音会淹没来自x_1和x_2的信号，导致所找到的最近邻相当于做出随机预测。

更多的困扰是，即使所有的100个特征都是相关的，最近邻方法依然会有问题。这是因为在高维空间所有的样例都变得很相似。例如，假设所有样例分布在规则的网格上，现在考虑一个测试样例x_t。如果网格是d-维的，会有个$2d$个x_t最近邻样例与x_t的距离相等。因此，随着维数的增加，越来越多的样例会变成x_t的最近邻，以致最后最近邻的选择实际上变成随机的（类别选择也因此变成随机的）。

这只是高维空间上更广泛问题的一个实例。我们的来自三维世界的直觉在高维空间通常并不奏效。在高维空间，多元高斯分布（multivariate Gaussian distribution）的大部分质量（mass）并不分布在均值附近，而是在逐渐远离均值的一层“壳”上；打个比方，一个高维的橘子的大部分质量不在瓤上，而是在皮上。如果数量一定的样例均匀分布在一个（维数不断增加的）高维的超立方体中，那么超出某个维数后，大部分样例与超立方体的某一面的距离要小于与它们最近邻的距离。如果我们在超立方体中内接一个超球面，那么超立方体的几乎所有质量都会分布在超球面之外。这对机器学习是一个坏消息，因为机器学习常常用一种类型的形状来近似另一种类型的形状。

在二维或三维空间构建分类器很简单，我们可以仅通过肉眼观察发现不同类别样例的分界线（甚至可以说，假如人们有在高维空间中观察的能力，机器学习就没有存在的必要了）。但是在高维空间中很难理解正在发生什么。因此也就很难设计一个好的分类器。人们也许会天真地认为收集更多的特征永远不会有什

[3] 这里作者指的是输入为布尔量时的情形。

么坏处，因为最坏的情况也不过是没有提供关于类别的新信息而已。但实际上这样做的好处可能要远小于维度灾难带来的问题。

幸运的是，有一个效应可以在一定程度上抵消维度灾难，那就是所谓的“非均匀性的祝福”（blessing of nonuniformity）。在大多数应用中，样例在空间中并非均匀分布，而是集中在一个低维流形（manifold）上面或附近。例如在手写体数字识别任务中，即使数字图片的每个像素都单独作为一个特征，近邻方法在该任务上表现依然良好，这是因为数字图片的空间要远小于整个可能的空间。学习器可以隐式地充分利用这个有效的更低维空间，也可以显式地进行降维（例如特南鲍姆（Tenenbaum）的工作[22]）。

理论保证（Theoretical Guarantees）与看上去的不一样

机器学习论文充满了理论保证。最常见的类型是能保证泛化所需样例数目的边界（bound）。你应当如何理解这些保证呢？首先，需要注意的是它们是否可行。归纳与演绎相反：在演绎中你可以保证结论是对的；在归纳中就难说了。这是很多世纪以来的普遍共识。最近几十年的一个重要进展是我们认识到可以有归纳结果正确性的保证，特别是如果我们愿意接受概率保证。

基本论证非常简单[5]。如果一个分类器的真实错误率（true error rate）大于ε，我们称该分类器是坏的。那么一个坏分类器在n个随机独立训练样例上都保持正确的概率小于 。设b是学习器的假设空间H中坏分类器的个数，其中至少有一个分类器能保持正确的概率小于$b(1-\varepsilon)^n$，即所谓“一致限（union bound）”。假设学习器返回的都是保持正确的分类器，那么这个分类器是坏的概率小于$|H|(1-\varepsilon)^n$，这里我们利用了$b \leq |H|$这个事实。所以，如果我们希望这个概率小于δ的充分条件是使$n>1/\varepsilon(\ln|H|+\ln 1/\delta)\geq \ln(\delta/|H|)/\ln(1-\varepsilon)$[4]。

不幸的是，对这类保证得十分小心。这是因为通过这种方式获得的边界往往非常松散（loose）。这种边界的突出优点是所要求的样例数目只随$|H|$和$1/\delta$呈对数增长。但遗憾的是，大多数假设空间是随着特征数目呈双指数级增长的，这就要求我们提供的样例数目d也随着呈指数增长。例如，考虑包含d个布尔变量的布尔类型函数空间。如果有e个可能不同的样例，就会有2^e个可能不同的函数。因此，由于有2^d个可能的样例，函数总数达到个2^{2^d}。即使对“仅仅”为指数级的假设空间，这个边界仍然很松，因为一致限非常保守。例如，如果有100个布尔特征，假设空间是层数最多为10的决策树，为了保证$\delta=\varepsilon=1\%$，我们需要50万个样例。但实际上，只需要其中的一小部分数据就足以精确学习了。

而且，我们必须留意边界所包含的意义。例如，边界并不意味着，假如你的学习器返回了一个在某个特定训练集上保持正确的假设，这个假设就可能实现了泛化。边界的意思是，给定一个足够大的训练集，告诉你在很大的概率上你的学习器会返回一个成功泛化的假设，还是无法找到一个保持正确的假设。这个边界也无法告诉我们如何选择好的假设空间。它只能告诉我们，如果这个假设空间包含真实分类器，那么学习器输出一个坏分类器的概率随着训练数据规模的增长而降低。如果我们缩小假设空间，边界就会得到改善，但是空间包含真实分类器的几率也降低了（在真实分类器不在假设空间中的情况下也会有边界，以上讨论同样适用）。

另一类常用理论保证是渐进（asymptotic）：给定无穷数据，学习器将保证输出正确的分类器。这个保证让人欣慰，但如果只是因为有渐进保证而选择一个分类器则是非常草率

[4] 原文公式有误，根据参考文献[5]应为该公式。

的。在实践中，我们很少处于渐进状态（或称为渐进态（asymptopia））。而且，由于我们前面探讨过的偏置-方差的权衡（trade-off），如果对无穷数据，学习器A比学习器B好，那么在有限数据的情况下B通常比A好。

机器学习中理论保证的主要作用并不是在实践中作为决策的标准，而是在算法设计中作为理解和驱动的来源。在这方面，它们作用巨大；实际上，理论与实践的紧密结合是机器学习在过去几年中取得重大进展的重要原因。但是使用者需要谨慎：学习是一个复杂现象，因为一个学习器既有理论证明又有实际应用，而前者并未成为后者的依据。

特征工程（Feature Engineering）是关键

在考虑所有情况之后，有的机器学习项目成功了而有的则失败了。这是什么原因造成的呢？无疑最重要的因素是所利用的特征。如果你有很多与类别非常相关的独立特征，学习起来很容易。但另一方面，如果特征与类别的关系非常复杂，你就不一定能够学到它了。通常原始数据不能直接拿来学习，你需要从中构建特征。这是机器学习项目的主要工作。这通常也是最有趣的部分，在这里直觉、创造性和魔法与技术一样都很重要。

初学者往往惊讶于机器学习项目中真正用于机器学习的时间是如此之少。但假如你考虑到对数据的收集、整合、清理和预处理是多么费时，以及特征设计需要经历多少试验和错误，就会理解这个过程了。还有，机器学习无法做到一次性就能完成构建数据集合和运行学习器，它是一个反复迭代的过程，包括运行学习器，分析结果，修改数据和/或学习器等，不断重复。学习往往是这其中最快完成的部分，原因在于我们已经非常精通它了！特征工程更加困难，原因是它是领域相关（domain-specific）的，而学习器则很大程度是通用的。不过，两者并没有明确界限，这也是最有用的学习器往往是那些有助于融入领域知识的学习器的原因之一。

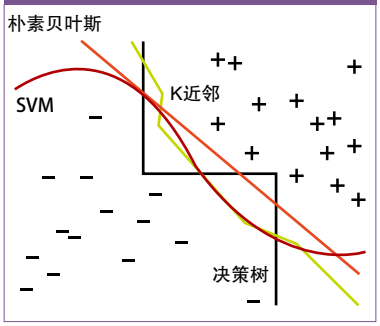
当然，机器学习的一个终极目标就是将特征工程过程越来越多地自动化。现在经常采用的一种方式是先自动产生大量的候选特征，然后根据它们与分类类别的信息增益等方法来选取最好的特征。但需要牢记在心的是，特征独立地看也许与分类无关，但组合起来也许就相关了。例如，如果分类类别是取个输入k个特征的“XOR（异或）”，那么每个特征单独看都与分类没有关系（如果你想给机器学习找点乱子，就祭出XOR来吧）。但是，运行包含大量特征的学习器来寻找有用的特征组合太耗时，也容易导致过拟合。因此，归根到底你仍需责无旁贷地介入特征工程的工作。

更多的数据胜过更聪明的算法

假设你已经尽你所能构建了最好的特征集合，但分类器的效果仍不够好，这时候应该怎么办呢？有两个主要选择：设计更好的学习算法，或者收集更多数据（包括更多的样例和不致造成维度灾难的更多可能的原始特征）。机器学习研究者更关注前者，但从实用角度来看，最快捷的方法是收集更多数据。作为一条经验，有大量数据的笨算法要胜过数据量较少的聪明算法。（毕竟，机器学习就是研究如何让数据发挥作用的。）

然而这带来了另外一个问题：可扩展性（scalability）。在绝大多数计算机科学问题中，两个主要资源是有限的——时间和内存。而在机器学习中，还有第三个：训练数据。其中哪一个资源会成为瓶颈是随着时间变化而不断变化的。在20世纪80年代，瓶颈是数据。现在的瓶颈则是时间。我们有海量数据，但没有足够的时间处理它们，只能弃之不用。这就造成一个悖论：即使理论上说，更多数据意味着我们可以学习更复杂的分类器，但在实践中由于复杂分类器需要更多的学习时间，我们只能选用更简单的分类器。一个解决方案是对复杂分类器提出快速学习算法，在这个方向上已经有了一些

图3 非常不同的分类边界会产生类似的预测（+和-表示两类训练样例）



引人注目的进展（例如赫尔滕（Hulten）和多明戈斯（Domingos）的工作[11]）。

采用更聪明的算法得到的回报比预期要少，一部分原因是，机器学习的工作机制基本上是相同的。这个论断也许让你吃惊，特别是当你想到诸如规则集与神经网络之间差异巨大的表示方法的时候。但实际上，命题规则的确可以轻易地表示成神经网络，其他表示之间也有类似的关系。本质上所有的学习器都是将临近的样例归类到同一个类别中；关键的不同之处在于“临近”的意义。对于非均匀分布的数据，不同的学习器可以产生迥乎不同的分类边界，同时仍能在关心的领域（即那些有大量训练样例、测试样例也会有很大概率出现的领域）保证得到相同的预测结果。这也有助于解释为什么能力强的学习器虽然不稳定却仍然很精确。图3在二维空间展示了这一点，在高维空间这个效应会更强。

作为一条规则，首先尝试最简单的学习器总是有好处的（例如应该在逻辑斯蒂回归之前先尝试朴素贝叶斯，在支持向量机之前先尝试近邻）。更复杂的分类器固然诱人，但它们通常比较难驾驭，原因包括我们需要调节更多的参数才能得到好的结果，以及它们的内部机制更不透明。

学习器可以分为两大类：一类的表示是大小不变的，比如线性分类器；另一类的表示会随着数据而增长，比如决策树。（后者有时候会被称为非参数化学习器（nonparametric learners），但不幸的是，它们通常需要比参数化学习器学习更多的参数。）数据超过一定数量后，大小不变的学习器就不能再从中获益。（注意图2中朴素贝叶斯的准确率是如何逼近大约70%的。）而如果有足够的数据，大小可变的学习器理论上可以学习任何函数，但实际上却无法做到。这主要是受到算法（例如贪心搜索会陷入局部最优）和计算复杂度的限制。而且，由于维度灾难，再多的数据也不会够。正是由于这些原因，只要你努力，聪明的算法——那些充分利用已有数据和计算资源的算法——最后总能取得成功。在设计学习器和学习分类器之间并没有明显的界限；因为任何知识要么可以被编码进学习器，要么可以从数据中学到。所以，机器学习项目通常会有学习器设计这一重要部分，机器学习实践者应当在这方面积累一些专门知识[12]。

终极而言，最大的瓶颈既不是数据，也不是CPU速度，而是人力。在研究论文中，学习器一般都在准确率和计算复杂度方面进行比较。但更重要的是节省的人力和得到的知识，虽然这些更难度量。这使那些产生人类可理解的输出的学习器（比如规则集合）更为受到青睐。机器学习成果最丰硕的，是那些建立了机器学习的基本条件，能够便捷地在多个学习器、数据来源和学习问题上方便有效地开展实验，并实现机器学习专家与领域专家的密切合作的组织。

要学习很多模型，而不仅仅是一个

在机器学习早期，每个人都有一个最喜欢的学习器，并由于一些先入为主的原因坚信它的优越性。人们花费大部分精力来尝试它的各种变种，从中选择最好的那个。后来，系统的实验比较表明在不同应用上的最佳学习器并不相同，因此开始出现包含多种学习器的系统。这时，人们尝试不同学习器的各种变种，仍然只是找出其中表现最好的那个。后来研究者注意到，如果不是只选最好的那个，而是将多个学习器结合，结果会更好——通常是好得多——而这只需要花费人们很少的精力。

现在建立模型集成（model ensembles）已经实现标准化[1]。最简单的集成技术是bagging（装

袋）方法，该方法通过重采样（resampling）随机产生若干个不同的训练集，在每个集合上训练一个分类器，然后用投票（voting）的方式将结果合并。该方法比较有效，原因是它在轻度增加偏置的同时，极大地降低了方差。在boosting（强化提升）方法中，每个训练样例都有权重，权重会不断变化，每次训练新分类器的时候都集中在那些分类器之前倾向于分错的样例上。在stacking（堆叠）方法中，每个单独分类器的输出会作为更高层分类器的输入，更高层分类器可以判断如何更好地合并这些来自低层的输出。

此外，还有很多其他技术，现在的趋势是越来越大型的集成。在Netflix大奖赛中，来自世界各地的团队竞争建立最好的视频推荐系统（http://netflixprize.com）。随着竞赛的开展，团队们开始发现与其他团队合并学习器会取得最好的结果，因此团队开始合并，越来越大。竞赛的第一名和第二名团队都合并了超过100个学习器，将这两者集成后又进一步提升了效果。毫无疑问，未来我们会看到更大的集成学习器。

模型集成不应与贝叶斯模型平均（bayesian model averaging，BMA）混淆，后者是学习的一种理论最优化方法[4]。在贝叶斯模型平均方法中，对新样例的预测是对假设空间中的所有分类器的预测取平均得到的，每个分类器会根据它解释训练数据的能力和我们对它的先验信任度而有不同的权重。虽然模型集成与贝叶斯模型平均方法表面上很相似，它们其实非常不同。集成方法改变了假设空间（例如从单独的决策树变成了决策树的线性组合），而且可以采用多种多样的形式。贝叶斯模型平均方法只是根据某个准则对原始空间的假设赋予不同的权重。贝叶斯模型平均方法的权重与bagging或者boosting等集成方法产生的权重非常不同。后者很平均，而前者波动很大，甚至出现某个权重最大的分类器占据统治地位的情况，导致贝叶斯模型平均方法实际上等同于直接选择这个权重最大的分类器[8]。一个实际的后果是，模型集成已经成为机器学习工具的重要组成部分，而贝叶斯模型平均方法则少有人问津。

简单并不意味着准确

著名的奥坎姆剃刀（occam's razor）原理称：若无必要，勿增实体（entities should not be multiplied beyond necessity）。在机器学习中，这经常被用来表示成：对于有相同训练误差的两个分类器，比较简单的那个更可能有较低的测试误差。关于这个断言的证明经常出现在文献中，但实际上对此有很多反例，而且“没有免费的午餐”定理也暗示了这个断言并不正确。

我们前面已经看到了一个反例：模型集成。集成模型的泛化误差会一直随着增加新的分类器而改进，甚至可以优于训练误差。另一个反例是支持向量机，它实际上可以有无限个参数而不至于过拟合。而与之相反，函数可以将轴上任意数量、任意分类的数据点划分开，即使它只有1个参数[23]。因此，与直觉相反，在模型参数的数量和过拟合之间并无直接联系。

一个更成熟的认识是将复杂度等同于假设空间的大小。这是基于以下事实：更小的假设空间允许用更短的代码表示假设。那么“理论保证”一节中的边界就暗示了，更短的假设可以泛化得更好。这还可以进一步改善为，为有先验偏好的空间中的假设分配更短的代码。但如果将此看作是准确（accuracy）和简单（simplicity）之间权衡的“证明”，那就变成循环论证了——我们将所偏好的假设设计得更加简单，而如果结果是准确的是因为我们的偏好是准确的，而不是因为这些假设在我们选择的表示方法中是“简单的”。

问题的复杂性还来自这样一个因素：几乎没有学习器能穷尽搜索整个假设空间。一个在较大的假设空间搜索较少假设的学习器，比一个在较小空间中搜索较多假设的学习器更不容易过拟合。正如珀尔（Pearl）[18]指出的，假设空间的大小只是对对确定影响训练误差和测试误差的关键因素有初步的指导意义。

多明戈斯[7]调研了机器学习中奥坎姆剃刀原理问题的主要论证和论据。结论是，应当先选择简单假设，这是因为简单本身就是一个优点，而不是因为所假设的与准确率有什么联系。这也许正是奥坎姆最初想表达的意思。

可表示并不意味着可学习

从本质上讲，用于大小可变的学习器的所有表示都有其形式为“每个函数都可以表达为或以无限接近的方式近似表达为××表示”的定理与之伴随。正因为如此，某种表示方法的拥趸往往会忽略其他方法。但是，仅仅因为一个函数可以被表示，并不意味着它是可被学习的。例如，标准的决策树学习器无法学习出比训练样例更多的叶子节点。在连续空间中，用一个固定的基元（primitives）族来表示哪怕很简单的函数，也常常要由无限多项组成。更进一步，如果假设空间有许多评价函数的局部最优点，正如经常发生的那样，学习器可能根本无法找到这个真正的函数，即使它是可表示的。给定有限数据、时间和内存，标准学习器只能学到所有可能函数中很有限的子集。这个子集会随着表示方法的不同而不同。因此，关键问题不是“它是否可表示？”（这个问题的答案通常无关紧要），而是“它是否可以被学习？”这值得我们尝试不同的学习器（或者它们的组合）来寻找答案。

对某些函数来讲，一些表示方法会比其他方法更加精简，从而只需要更少的数据来学习那些函数。很多学习器的工作机制是将简单的基函数（basis function）进行线性组合。例如，支持向量机就形成了集中在某些训练样例（也就是那些支持向量）上的核（kernels）的组合。如果用这种组合方法来表示n个比特的奇偶性（parity），将需要2^n个基函数。但如果采用多层表示（也就是说在输入和输出之间存在多步），奇偶性就可以用一个线性规模的分类器表示。探索这种深层表示的学习方法是机器学习的主要研究前沿之一[2]。

相关并不意味着因果

相关不意味着因果，这一点经常被提起，好像在这儿已经不值得再加赘述了。但是，即使我们讨论的这些学习器只能学习到相关性，它们的结果也经常被作为因果关系来对待。这样做错了么？如果是错的，为什么人们还这样做呢？

更多时候，人们学习预测模型的目标是作为行动指南。如果我们发现超市里的啤酒和尿布经常被一起购买，那将啤酒放在尿布旁边将会提高销售量。（这是数据挖掘领域的著名例子。）但除非真的做实验，不然很难发现这一点。机器学习通常应用于观测（observational）数据，在观测数据中预测变量并不在学习器的控制之下，这与实验（experimental）数据相反，后者的预测变量在控制范围内。一些学习算法其实有潜力做到从观测数据发现因果信息，但它们的可用性比较差[19]。而另一方面，相关性是因果关系的标志，我们可以将其作为进一步考察的指南（例如试图理解因果链可能是什么样）。

很多研究者相信因果只是一种为了方便而杜撰的概念。例如，在物理定律中并没有因果的概念。因果是否真的存在是一个深奥的哲学问题，现在并没有一个确定的答案。但对于机器学习有两个实用的要点。首先，无论我们是否称它们为“因果关系”，我们都希望能预测我们行动的效果，而不仅仅是观测变量之间的相关性；其次，如果你能够获取到实验数据（例如能够随机分配访问者到一个网站的不同版本），那么务必尽量获取[14]。

结论

就像其他任何一个学科那样，机器学习拥有的很多“民间智慧”并不是那么容易就能了解到，但这些知识对于成功运用机器学习至关重要。这篇文章总结了其中最主要的几条知识。当然这只是对机器学习的传统学习内容的补充。读者可以参加一个有完整内容的机器学习在线课程，其中融合了正式

和非正式的知识，网站是http://www.cs.washington.edu/homes/pedrod/。此外，在http://www.videolectures.net上还有大量宝贵的与机器学习相关的学术报告。Weka[24]是一款优秀的机器学习开源工具包。

祝大家学习快乐！■

作　者：

佩德罗·多明戈斯：美国西雅图华盛顿大学计算机科学与工程系教授。

pedrod@cs.washington.edu

译者：刘知远

CCF会员。清华大学博士后。主要研究方向为自然语言处理、信息检索与社会计算。

lzy.thu@gmail.com

参考文献

[1] Bauer, E. and Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Machine Learning 36 (1999), 105~142

[2] Bengio, Y. Learning deep architectures for AI. Foundations and Trends in Machine Learning 2, 1 (2009), 1~127

[3] Benjamini, Y. and Hochberg, Y. Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society, Series B, 57 (1995), 289~300

美国会发难华为、中兴

CCF发表声明

美国众议院情报委员会于当地时间2012年10月8日发布针对两家中国通信企业华为和中兴“可能对美国带来安全威胁”的调查结果，随后，华为、中兴遭美国国会封杀一事愈演愈烈。10月12日，中国计算机学会青年计算机科技论坛（CCF YOCSEF）在京举办了“为什么美国国会发难华为中兴”特别论坛，就该事件中所涉及的政治、经济、技术及法律原因进行了相关探讨。会后，CCF就美国会发难华为和中兴发表严正声明，全文如下：

近日，本学会注意到，美国国会众议院情报委员会发表调查报告（以下简称“报告”）称，中国华为技术有限公司和中兴通讯股份有限公司对美国国家安全构成威胁，建议阻止这两家企业在美开展投资贸易活动。对此，中国计算机学会发表声明如下：

1 “报告”没有举出确切证据可以证明华为、中兴给美国国家安全带来威胁，而且“报告”对华为、中兴提出的种种要求并没有同等地施加于美国市场上的其他同类企业，因而，华为、中兴遭到这样的待遇是不公正的。

2 多年来，华为、中兴在美国市场上的业务活动完全遵循了美国的法律，符合WTO的相关规则，“报告”以国家安全为名借助政治手段剥夺了它们参与美国市场平等竞争的权利，既违背了美国长期标榜的自由竞争的市场经济原则，也不符合WTO的有关规则及全球一体化的世界潮流。

3 华为、中兴都是中国计算机学会的会员单位，本学会对它们在美国遭到的不公正待遇表示严重关切。我们呼吁中国有关方面帮助华为、中兴维护它们的正当权益，必要时对美国实施的这种贸易保护主义行为采取反制措施。

[4] Bernardo, J.M. and Smith, A.F.M. Bayesian Theory. Wiley, NY, 1994

[5] Blumer, A., Ehrenfeucht, A., Haussler, D. and Warmuth, M.K. Occam's razor. Information Processing Letters 24 (1987), 377~380

[6] Cohen, W.W. Grammatically biased learning: Learning logic programs using an explicit antecedent description language. Artificial Intelligence 68 (1994), 303~366

[7] Domingos, P. The role of Occam's razor in knowledge discovery. Data Mining and Knowledge Discovery 3 (1999), 409~425

[8] Domingos, P. Bayesian averaging of classifiers and the overfitting problem. In Proceedings of the 17th International Conference on Machine Learning (Stanford, CA, 2000), Morgan Kaufmann, San Mateo, CA, 223~230

[9] Domingos, P. A unified bias-variance decomposition and its applications. In Proceedings of the 17th International Conference on Machine Learning (Stanford, CA, 2000), Morgan Kaufmann, San Mateo, CA, 231~238

[10] Domingos, P. and Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29 (1997), 103~130

[11] Hulten, G. and Domingos, P. Mining complex models from arbitrarily large databases in constant time. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Edmonton, Canada, 2002). ACM Press, NY, 525~531

[12] Kibler, D. and Langley, P. Machine learning as an experimental science. In Proceedings of the 3rd European Working Session on Learning (London, UK, 1988). Pitman

[13] Klockars, A.J. and Sax, G. Multiple Comparisons. Sage, Beverly Hills, CA, 1986

[14] Kohavi, R., Longbotham, R., Sommerfield, D. and Henne, R. Controlled experiments on the Web: Survey and practical guide. Data Mining and Knowledge Discovery 18 (2009), 140~181

[15] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A. Big data: The next frontier for innovation, competition, and productivity. Technical report, McKinsey Global Institute, 2011

[16] Mitchell, T.M. Machine Learning. McGraw-Hill, NY, 1997

[17] Ng, A.Y. Preventing "overfitting" of cross-validation data. In Proceedings of the 14th International Conference on Machine Learning (Nashville, TN, 1997). Morgan Kaufmann, San Mateo, CA, 245~-253

[18] Pearl, J. On the connection between the complexity and credibility of inferred models. International Journal of General Systems 4 (1978), 255~264

[19] Pearl, J. Causality: Models, Reasoning, and Inference. Cambridge University Press, Cambridge, UK, 2000

[20] Quinlan, J.R. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993

[21] Richardson, M. and P. Domingos. Markov logic networks. Machine Learning 62 (2006), 107--136

[22] Tenenbaum, J., Silva, V. and Langford, J. A global geometric framework for nonlinear dimensionality reduction. Science 290 (2000), 2319~2323

[23] Vapnik, V.N. The Nature of Statistical Learning Theory. Springer, NY, 1995

[24] Witten, I., Frank, E. and Hall, M. Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition. Morgan Kaufmann, San Mateo, CA, 2011

[25] Wolpert, D. The lack of a priori distinctions between learning algorithms. Neural Computation 8 (1996), 1341~1390