北京郵電大學
Beijing University of Posts and Telecommunications

大数据算法及其安全

石瑞生
网络空间安全学院

提纲

- 大数据算法基础
  - 数学模型
  - 案例：搜索引擎算法；电子商务协同过滤推荐算法
  - 新需求：机器学习算法，让计算机基于数据来自动构建（数学）模型
  - 机器智能不足怎么办：众包算法
- 大数据算法的安全问题

大数据算法基础

提纲

- 什么是数学模型?
- 什么是好的数学模型?
  - 天文学的例子：地心说 vs. 日心说
  - 搜索引擎中的网页相关性模型
- 数学模型的小结
- 大数据时代的新需求
  - 机器学习算法
  - 众包算法

科学的发展与人类知识体系的构建

- 科学的发展是一个知识积累的过程
- 什么是知识?
  - 解读"知识"：不仅"知道"其现象，还要深刻"认识"其本质
- 怎么创造知识?
  - 实验，假设与模型，实践验证，理论解释
  - 知识的发现与构建过程中，人们需要完成四个方面的工作：1）基于实验与观察来提出一些假设，2）然后基于这些假设来构建模型；3）有了模型，人们会（基于实践）对模型进行验证，4）还会尝试构建一个理论体系试图对其本质做出合理解释。
  - 观察与实验能够帮助我们知道其现象，假设和**模型**是认识其本质的起点，实践帮助我们验证与修正从而不断完善我们的假设与构建的模型，理论解释试图对其本质给出最终的解释,并帮助我们克服这种方法的局限性并扩大其应用范围,让其在更大范围内获得成功。

模型

- 几乎所有的科学领域都在用模型拟合数据。
  - 科学家们设计实验、进行观测并收集数据。然后，通过找寻能解释所观测数据的（简单）模型，尝试抽取知识。
  - 该过程称为归纳（induction），它是从一组特别的示例中提取通用规则的过程。
- 数学模型在人类认知过程中扮演着重要的角色。
  - 最广为人知的例子就是人类对宇宙的认知。人类对宇宙模型的认知，经历了从两千年前古罗马时代的托勒密提出的地心说到哥白尼的日心说、从牛顿的经典力学理论到爱因斯坦的相对论的演进过程。
  - 我们注意到，数据和模型在演进过程中起到了重要的作用。

数学模型

天文学：地心说 vs. 日心说

8

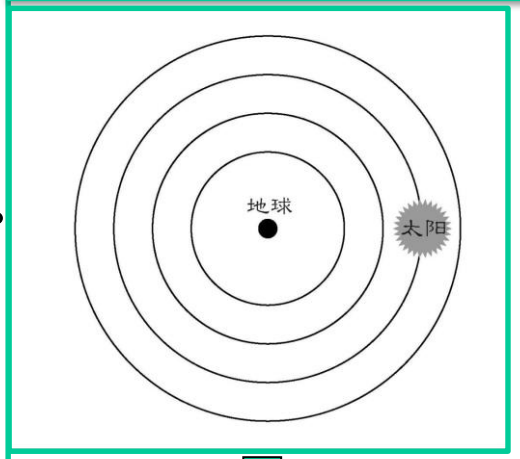
天文学 -- 地心说

- 古希腊、古罗马是西方文化的起源，很多结论现在看来可能是错的，但是，其研究方法与思想却永恒地保留了下来。
- 真正创立了天文学，并且计算出诸多天体运行轨迹的是两千年前古罗马时代的托勒密。
  - 托勒密发明了球坐标，定义了包括赤道和零度经线在内的经纬线，他提出了黄道，还发明了弧度制。
  - 最大的发明是地心说：从人们的观测出发，很容易得到地球是宇宙中心的结论。

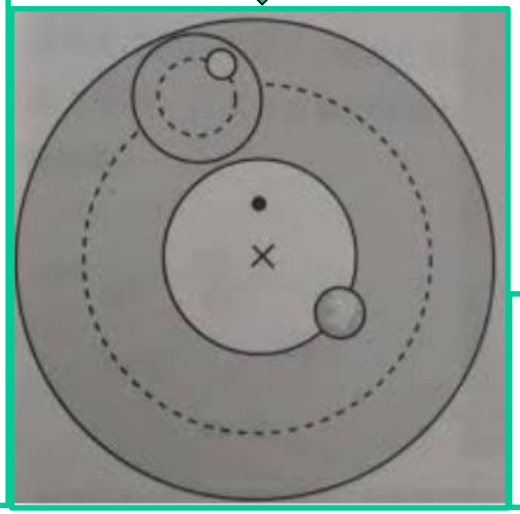
地心说的行星模型与应用

- 托勒密继承了毕达格拉斯的一些思想，他也认为圆是最完美的几何图形。
  - 假设：地球是宇宙的中心
    - 宇宙是以地球为中心的，所有的天体以均匀的速度按照圆形的轨道运动。
  - 模型：能够观测到的天文数据却和这一模型的预测不一样。
    - 为了让自己的模型更加严密，托勒密设想了非常复杂的“本轮”、“均轮”。每个行星都绕着“本轮”运转，每个“本轮”又沿着“均轮”绕着地球转动。
  - 计算方法：解出四十个套在一起的圆的方程
    - 从地球上看，行星的运动轨迹是不规则的，托勒密的伟大之处是用四十个小圆套大圆的方法，精确地计算出了所有行星运动的轨迹。
    - 一千五百年来，人们根据他的计算决定农时。但是，经过了一千五百年，托勒密对太阳运动的累积误差，还是差出了一星期。
- 托勒密模型的精度之高，让以后所有的科学家惊叹不已。
  - 即使今天，我们在计算机的帮助下，也很难解出四十个套在一起的圆的方程。每每想到这里，我们都不得不佩服托勒密。

假设是起点，模型需要不断改进



模型的优化



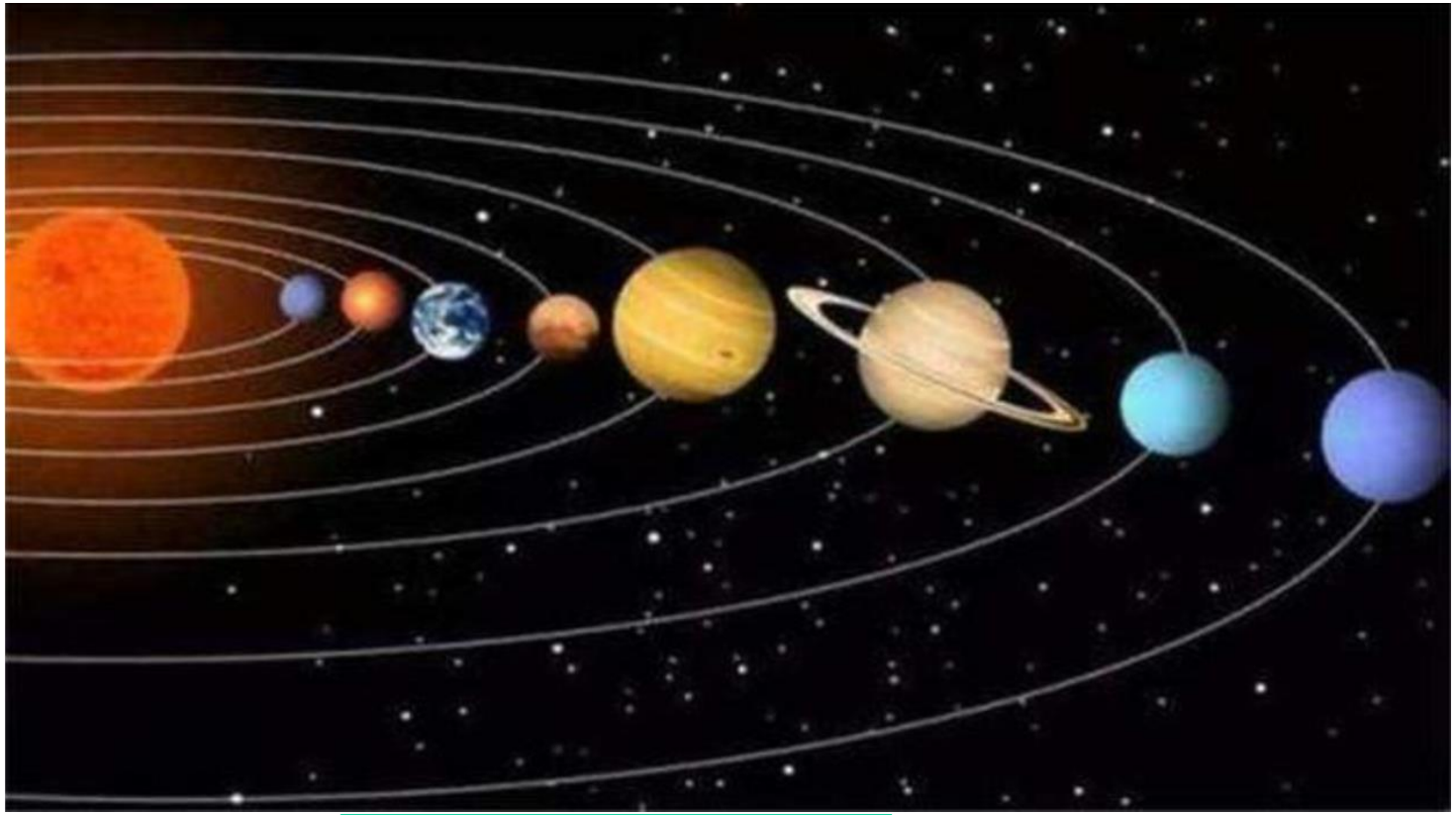
托勒密的小圆套大圆的地心说模型

哥白尼的贡献

- 哥白尼发现，如果以太阳为中心来描述星体的运行，只需要8-10个圆，就能计算出一个行星的运动轨迹，他提出了日心说。
- 很遗憾的事，哥白尼正确的假设并没有得到比托勒密更好的结果，哥白尼的模型的误差比托勒密地要大不少。
  - 日心说这个假设，可能更接近现实
- 新思想、新方法、新技术，一开始不见得效果更好？
  - “地心说”模型的基本假设是完全错误的，但随着不断的修改，却变得相当复杂、甚至可以说是精密。
  - 事实上，哥白尼的“日心说”远远不如托勒密的“地心说”深奥、巧妙。

新模型 – 数据的重要性

- 哥白尼提供一个崭新的思考问题的方向
  - 但是，哥白尼构建的模型误差很大，能否改进?



开普勒的行星模型

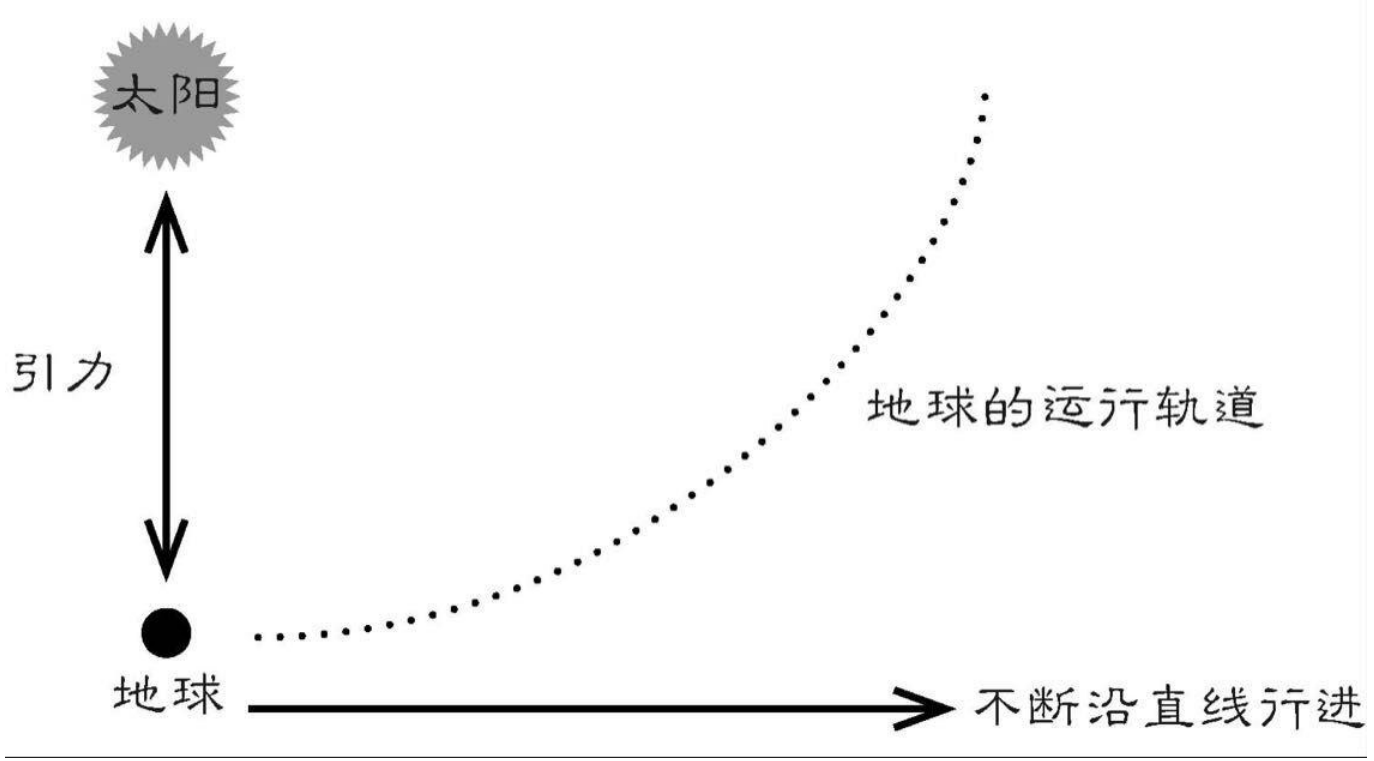
模型的构建

1609年，德国天文学家开普勒发表了《新天文学》，提出了一种彻底颠覆托勒密的宇宙模型。开普勒是怎么做的？很简单，他把天体运行的轨道改成了椭圆形。

开普勒从他的老师第谷手中继承的大量的、在当时最精确的观测数据。开普勒很幸运地发现了行星围绕太阳运转的轨道实际是椭圆形的，这样不需要用多个小圆套大圆，而只要用一个椭圆就能将星体运动规律描述清楚了。

新模型的理论解释

- 开普勒的知识和水平不足以解释为什么行星的轨道是椭圆形的。
- 问题的终结者：牛顿
  - 伟大的科学家牛顿用万有引力，对该模型给出了一个漂亮的理论解释。



永无止境 – 存在绝对真理吗？

- 爱因斯坦的相对论
  - 水星近日点的进动（时空弯曲）：天文观测记录了水星近日点每百年移动5600秒，人们考虑了各种因素，根据牛顿理论只能解释其中的5557秒，只剩43秒无法解释。
  - 广义相对论的计算结果与万有引力定律（平方反比定律）有所偏差，这一偏差刚好使水星的近日点每百年移动43秒。
- 在数学模型领域，有一句精辟的总结
  - All models are wrong ,but some are useful!

“所有模型都是错误的。实际问题是，它们要错到什么程度才会无用。”

数学模型

在搜索引擎中案例

20

搜索引擎中的例子

- 在网络搜索的研发中，单文本词频/逆文本频率指数（TF/IDF）和网页排名（page rank)都相当于是网络搜索中的“椭圆模型”，它们都很简单易懂。

影响搜索引擎服务质量的因素

- 相关性：TF/IDF
  - TF/IDF（term frequency/inverse document frequency，单文本词频/逆文本频率指数) 的概念被公认为信息检索中最重要的发明。
- 网页的质量
  - 例如，PageRank算法
- 个性化：用户偏好
- 基础架构：数据采集（网络爬虫系统），完备的索引

相关性排序的几个要素

- 词频:包含这关键词多的网页应该比包含它们少的网页相关
- 归一化
  - 一个明显的漏洞：长的网页比短的网页占便宜，因为长的网页总的来讲包含的关键词要多些。
  - 解决方案：需要根据网页的长度，对关键词的次数进行归一化，也就是用关键词的次数除以网页的总字数。
  - 概括地讲，如果一个查询包含关键词 $w_1,w_2,...,w_N$，它们在一篇特定网页中的词频分别是: TF_1, TF_2, ..., TF_N。（TF: term frequency)。 那么，这个查询和该网页的相关性就是: $TF_1 + TF_2 + ... + TF_N$。
- 权重:IDF
  - 假定一个关键词 w 在 D w 个网页中出现过，那么 D w 越大， w 的权重越小，反之亦然。
  - 在信息检索中，使用最多的权重是“逆文本频率指数” （Inverse document frequency 缩写为IDF），它的公式为ln(D/Dw)，其中D是全部网页数。
  - 利用 IDF，上述相关性计算个公式就由词频的简单求和变成了加权求和，即 $TF_1*IDF_1 + TF_2*IDF_2 + ... + TF_N*IDF_N$

举例说明

- 例如，查找关于“北邮的计算机学院”的网页。
  - 现在任何一个搜索引擎都包含几十万甚至是上百万个多少有点关系的网页。那么哪个应该排在前面呢？显然我们应该根据网页和查询“北邮的计算机学院”的相关性对这些网页进行排序。
  - 比如，在某个一共有一千词的网页中“北邮”、“的”和“计算机学院”分别出现了 2 次、35 次 和 5 次，那么它们的词频就分别是 0.002、0.035 和 0.005。 我们将这三个数相加，其和 0.042 就是相应网页和查询“北邮的计算机学院”相关性的一个简单的度量。
  - 引入权重后
    - 假定中文网页数是D = 1 0亿，应删除词“的”在所有的网页中都出现，即D w = 1 0亿，那么它的 IDF = ln(10亿/10 亿) = ln (1) = 0 。
    - 假如专用词“北邮”在两百万个网页中出现，即D w = 200 万，则它的权重 IDF = ln(500) =6.2。
    - 又假定通用词“计算机学院”，出现在五亿个网页中，它的权重 IDF = ln(2) 则只有 0.7。
  - 该网页和“北邮的计算机学院”的相关性为 0.0159，其中“北邮”贡献了 0.0126，而“计算机学院”只贡献了 0.0035。这个比例和我们的直觉比较一致了。

小结

- TD/IDF模型，把任意长度的文档简化为固定长度的数字列表
- 把搜索引擎的相关性排序问题简化为根据搜索关键词从文档的数字列表中选取相关的数字（相加），计算文档的相关性度量。
- 这种简化损失了什么？
  - 文档的上下文信息
  - 对文档这种表示，只提取了词的频率与权重信息，忽略了词的顺序。
- 分析一下该模型的几个要素
  - 假设，模型，计算方法，理论基础

理论解释

- 为什么这么计算效果很好?
  - 行星的轨道为什么是椭圆，而不是圆？牛顿的万有引力定律给出了理论解释
- 香农的信息论
  - 从信息论角度来解释，IDF 的概念就是一个特定条件下、关键词的概率分布的交叉熵 (Kullback-Leibler Divergence)

进一步的思考

- 分析一下该模型的几个要素
  - 假设，模型，计算方法，理论解释
  - 假设是什么？
    - 文档d和查询q的相关性可以由它们包含的共有词汇情况来刻画。

- 假设总是有效吗？这个假设的适用范围？
  - 对于有些文档，
    - 例如，图片较多的图文并茂的网页，这种模型还有效吗？如何改进？
    - 对于短文本？
  - 对于有些应用，关心的是主题，而不仅仅是单词的匹配程度
    - 例如，麦克和话筒，番茄和西红柿，是一回事；iOS和苹果手机、乔布斯的关系

这个模型总是有效的了吗？

- 能否找出一些反例?
- 一个例子
  - 有两个句子分别如下：
    - “乔布斯离我们而去了。”
    - “苹果价格会不会降？”
  - 这两个句子没有共同出现的单词，但这两个句子是相关的。
- 问题在哪儿?
  - TF-IDF模型没有考虑到文字背后的语义关联，可能在两个文档共同出现的单词很少甚至没有，但两个文档是相似的。
- 语义挖掘的利器是主题模型：
  - pLSI（probabilistic Latent Semantic Index）概率潜在语义索引
  - LDA（Latent Dirichlet Allocation）模型

四个要素

- 假设：方向
  - 猜想与直觉以及对现实问题的洞察力（Insights），是一切工作的起点。
  - 洞察力与想象力是最宝贵的，它决定了你的努力方向。
- 模型：方法
  - 发现规律：去粗存精（提取关键特征），去伪存真（过滤噪声），
  - 模型的选择：圆，还是椭圆？

实践：检验效果

- 用户只看结果，不问过程：理论，别人不关心；猜想，模型，别人不懂。
- 解决实际问题，创造价值：关键看效果
- 效果与成本的Tradeoff
- 统计语言模型：从高阶模型到二元模型，马尔可夫假设
- 计算方法：大数定律

理论体系：推广

- 解释规律，更好地预测未知领域的规律
- 解释模型与方法的原理，探索其本质规律
- 万有引力定律，初步解释了为什么椭圆型模型更有效
- 广义相对论，构建了一个更严密精准的理论系统

几点思考

- 一个正确的数学模型应当在形式上是简单的。（托勒密的模型显然太复杂。）
- 一个正确的模型在它开始的时候可能还不如一个精雕细琢过的错误的模型来的准确，但是，如果我们认定大方向是对的，就应该坚持下去。（日心说开始并没有地心说准确。）
- 大量准确的数据对研发很重要。
- 正确的模型也可能受噪音干扰，而显得不准确；这时我们不应该用一种凑合的修正方法来弥补它，而是要找到噪音的根源，这也许能通往重大发现。
- 没有绝对真理：All models are wrong ,but some are useful!

大数据时代的新需求

- 随着大数据时代的到来，各种各样的互联网服务都依赖于类似的数据分析算法。
- 很多场景，这样的数据分析已经不能再依赖人工完成了，原因有两个：一是数据量巨大，场景多变，并且很多时候人很难用手工编程来完成。二是能够做这种分析的人非常少而且人工分析又很昂贵。

- 因而，对于能够分析数据并自动从中提取信息的计算机模型，也就是说对于学习，人们的兴趣正在不断地增长。

能否让计算机基于数据来自动构建（数学）模型，完成任务？

从冷兵器到现代武器装备：从手工构建模型到自动构建模型

机器学习

- 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。
- 本节围绕着以下几个问题来展开：
- 1）为什么需要机器学习？
- 2）机器学习的基本原理
- 3）机器学习的常用方法

为什么需要机器学习

- 对于许多问题，我们已经知道如何求解。
  - 例如，欧几里得告诉我们可以用辗转相除法求两个整数的最大公约数；Dijkstra告诉我们如何有效地求两点之间的最短路径；Hoare向我们展示了怎样将杂乱无章的对象快速排序；等等。对于这些问题，我们清楚地知道求解步骤。
  - 因此，让计算机求解这些问题只需要设计算法和数据结构、进行编程，而不需要让计算机学习。
- 然而，还有一些问题，人们可以轻而易举地做好，但是却无法解释清楚我们是如何做的。
  - 尽管桌子千差万别、用途各异，但是我们一眼就能看出某个物体是否是桌子；尽管不同的人的手写阿拉伯数字大小不一、笔画粗细不同，但是我们还是可以轻易识别一个数字是不是8；尽管声音时大时小，有时可能还有点沙哑，但是我们还是可以不费力气地听出熟人的声音；等等。对于这些问题，我们不知道求解步骤。
  - 因此，对于这些任务我们希望计算机能够像人类一样自己来学习怎么做。

机器学习的基本原理

- 机器学习使用实例数据或过去的经验训练计算机，以优化性能标准。
- 当人们不能直接编写计算机程序解决给定的问题，而是需要借助于实例数据或经验时，就需要学习。
- 一种需要学习的情况是人们没有专门技术，或者不能解释他们的专门技术。
  - 以语音识别为例，这个任务需要将声学语音信号转换成ASCII文本。看上去我们可以毫无困难地做这件事，但是我们却不能解释我们是如何做的。
  - 由于年龄、性别或口音的差异，不同的人读相同的词发音却不同。在机器学习中，这个问题的解决方法是从不同的人那里收集大量发音样本，并学习将它们映射到词。

机器学习的基本原理

- 那么，机器如何来学习呢？简单地说，就是从经验中发现规律。我们知道桌子不是木材和各种材料的随机堆砌，手写数字不是像素的随机分布，熟人的声音也不是各种声波的随机混合。
- 现实世界总是有规律的。机器学习正是从已知实例中自动发现规律，建立对未知实例的预测模型；根据经验不断提高，不断改进预测性能。
- 所谓的“学习”， 其实就是模型训练；更简单点说，是根据一些东西，推导出了一个结论，这个结论是一个函数，函数的某些部分是一个常量，但是常量本身并不是已知的；我们需要基于大量数据，去进一步推断出缺失的这些常量。

机器学习的常用方法

- 预测建模是建立一个能够进行预测模型的通用概念。通常情况下，这样的模型包括一个机器学习算法，以便从训练数据集中学习某些属性做出这些预测。
- 预测建模的任务可以分成两类。
- 1）回归模型基于变量和趋势之间的关系的分析，以便做出关于连续变量的预测。如天气预报的最高温度的预测。
- 2）分类任务是分配离散的类标签到特定的实例作为预测的结果。在天气预报中的模式分类任务可能是一个晴天、雨天或雪天的预测。

机器学习的常用方法

分类任务可被分成两个主要的子类别：监督学习和无监督学习。

1）在监督学习中，用于构建分类模型的数据的类标签是已知的，如图7-1所示。

2）无监督学习任务处理未标记的实例，并且这些类必须从非结构化数据集中推断出来。通常情况下，无监督学习采用聚类技术，使用基于一定的相似性（或距离）的度量方式来将无标记的样本进行分组。

还有一类学习算法使用“强化学习”这个概念来描述。在这种算法中，模型是通过一系列的操作而最大化“奖励函数”来进行学习。奖励函数的最大化，可以通过惩罚“坏行为”，和/或通过奖励“好行为”来实现。强化学习的一个常见的例子是根据环境反馈而进行学习自动驾驶的训练过程。

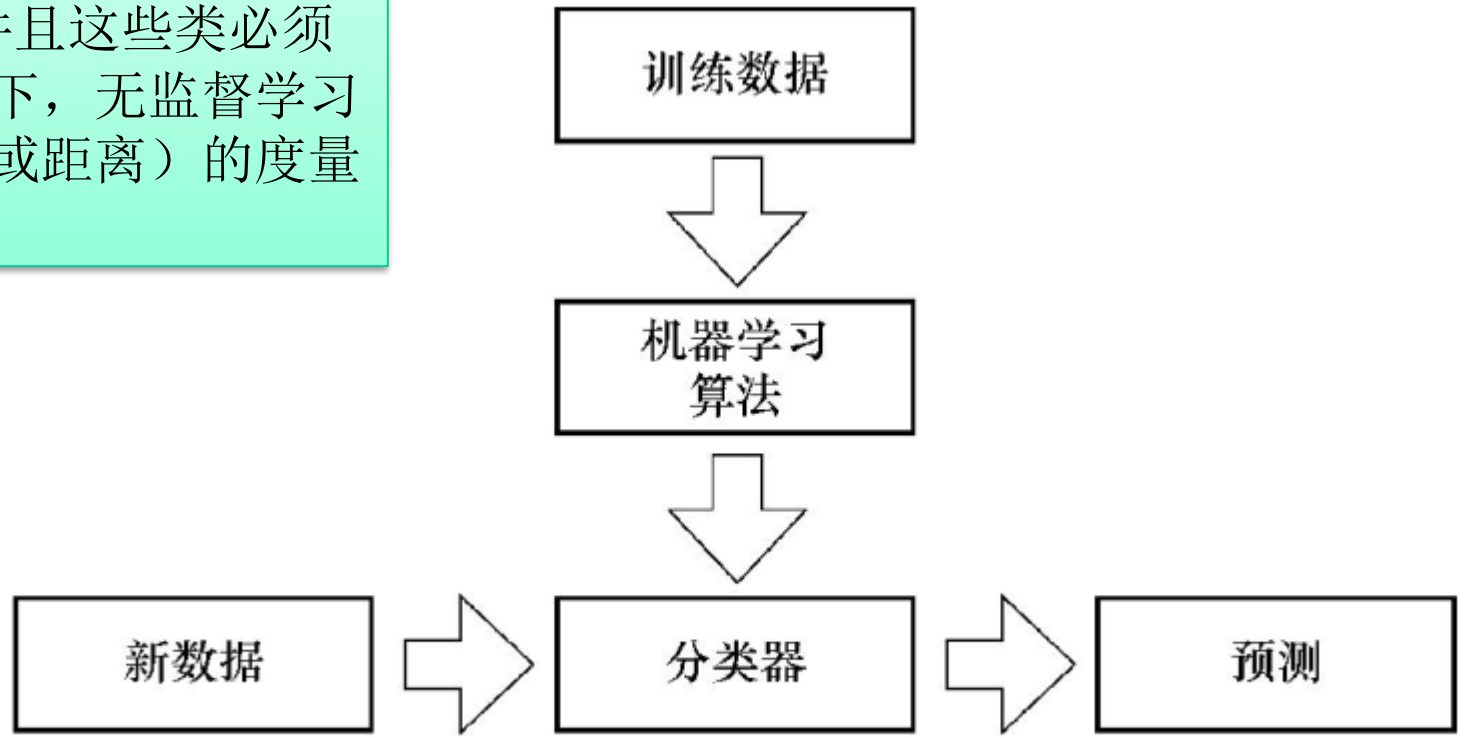


图 7-1　监督学习

基本概念

- 分类：分配预先定义的类标签到特定实例，将它们分成不同的类别的一般方法。
- 实例： 实例是“observation”或“样本”的同义词，描述由一个或多个特征（或称为“属性”）组成的“对象”。
- 我们用一个简单的例子来解释一下这些基本概念。
- 著名的“鸢尾花（Iris）”数据集可能是最常用的一个例子。1936年，R.A.Fisher创建了Iris数据集。Iris现在可以从UCI机器学习库中免费得到。

Iris 中的花被分为三类：Setosa，Virginica 和 Versicolor。而 Iris 数据集的 150 个实例中的每一个样本（单花）都有四个属性：①萼片的宽度；②萼片的长度；③花瓣的宽度；④花瓣的高度。

关于特征提取的方法可能包括花瓣和萼片的聚合运算，如花瓣或萼片宽度和高度之间的比率。

特征选择：相对于三种不同的花，花瓣包含的辨别信息相对于花萼来说要更多一些，因为花萼的宽度和长度差别更小一些。那么，该信息就可以用于特征选择，以去除噪声和减少数据集的大小。

可以使用 Iris 得到一个非常简单的决策树来完成对样本数据的分类，具体如图 7-2 所示。当花瓣长度小于 1 cm 时，判定为 Setosa。当花瓣长度大于等于 1 cm 时，继续用花瓣的宽度进行判定，当花瓣宽度小于 1.75 cm 时，判定为 Versicolor；当花瓣宽度大于等于 1.75 cm 时，判定为 Virginica。

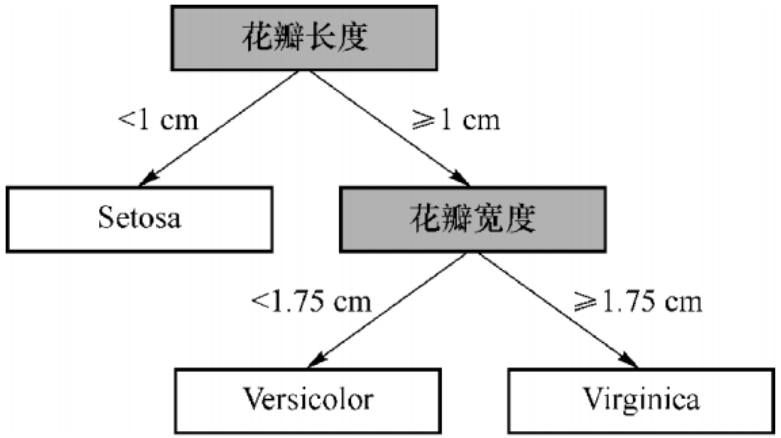


图 7-2　决策树

常用的监督学习算法

① 决策树分类器是树形图，其中，图中的节点用于测试某个特征子集的特定条件，然后分支把决策分割到叶子节点上。图 7-2 中的决策树的叶子节点表示最低级别，用于确定类的标签。

决策树的核心算法是确定决策树分枝准则，该准则涉及两个方面问题：①如何在众多的输入变量中选择出一个最佳的分组变量；②如何在分组变量的众多取值中寻找到最佳的分割值。

② 支持向量机(SVM)是利用采样超平面分隔两个或多个类的分类方法。最终，具有最大间隔的超平面被保留，其中“间隔”指的是从采样点到超平面的最小距离。组成间隔的采样点称为支持向量，从而建立起最终的 SVM 模型。

③ 贝叶斯分类器基于一个统计的模型(即贝叶斯定理：后验概率的计算基于先验概率和所谓的似然)。一个朴素贝叶斯分类器假定所有属性都是条件独立的，因此，计算似然可以简化为计算带有特定类标签的独立属性的条件概率的乘积就可以了。

④ 人工神经网络(ANN)是模仿人或动物“大脑”的图类分类器，其中相互连接的节点模拟的是神经元。

评价指标

- 准确率是我们最常见的评价指标，而且很容易理解，就是被分对的样本数除以所有的样本数，通常来说，正确率越高，分类器越好。
- 准确率确实是一个很好很直观的评价指标，但是有时候准确率高并不能代表一个算法就好。

比如某个地区某天地震的预测，假设我们有一堆的特征作为地震分类的属性，类别只有两个：0：不发生地震、1：发生地震。一个不加思考的分类器，对每一个测试用例都将类别划分为0，那那么它就可能达到99%的准确率，但真的地震来临时，这个分类器毫无察觉，这个分类带来的损失是巨大的。

为什么99%的准确率的分类器却不是我们想要的，因为这里数据分布不均衡，类别1的数据太少，完全错分类别1依然可以达到很高的准确率却忽视了我们关注的东西。

再举个例子说明下。在正负样本不平衡的情况下，准确率这个评价指标有很大的缺陷。比如在互联网广告里面，点击的数量是很少的，一般只有千分之几，如果用acc，即使全部预测成负类（不点击）acc也有 99% 以上，没有意义。因此，单纯靠准确率来评价一个算法模型是远远不够科学全面的。

评价指标

	Positive
True	True Positive (TP)
False	False Positive (FP)

实际类别	预测类别			
		Yes	No	总计
	Yes	TP	FN	P（实际为Yes）
	No	FP	TN	N（实际为No）
	总计	P'（被分为Yes）	N'（被分为No）	P+N

- 我们需要对训练出来的分类器的性能进行评估。混淆矩阵是一种用于性能评估的方便工具。通常，使用预测"准确率"或"差错率"来报告分类性能。
  - 准确率定义为正确分类的样本占总样本的比值。准确率的计算公式是：$\frac{TP+TN}{P+N}$

分类性能的其他指标：精度（precision=TP/P'）和灵敏度（**sensitive**=TP/P=**recall**，召回率）。
灵敏度（召回率）和精度用来评估二元分类问题中的"真阳性率"。
1）精度高，意味着误报率低（FP小，误报数量少）；误报率 + 精度 = 1
2）灵敏度高，意味着漏报率低（FN小，漏报数量少），灵敏度+漏报率 = 1

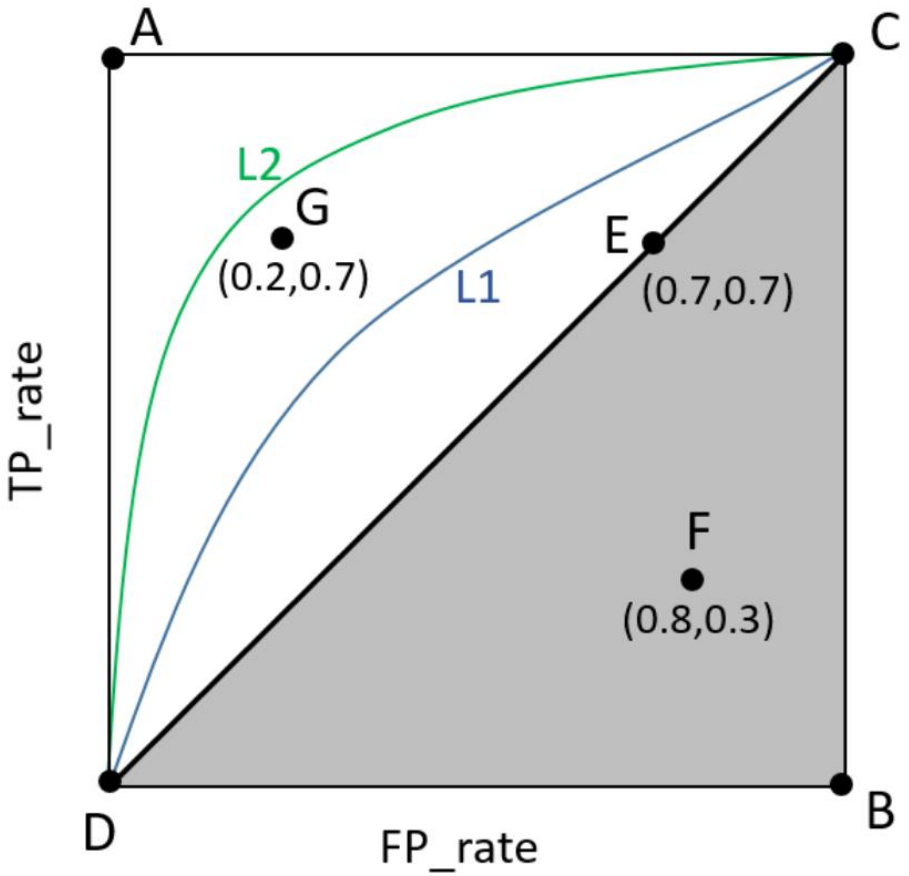
当参数α=1时，就是最常见的F1，也即

$$F1=\frac{2*P*R}{P+R}$$

P和R指标有时候会出现的矛盾的情况，这样就需要综合考虑他们，最常见的方法就是F-Measure（又称为F-Score）。**F-Measure是Precision和Recall加权调和平均**。
F1综合了P和R的结果，当F1较高时则能说明试验方法比较有效。

ROC和PR曲线

1、ROC曲线：
ROC（Receiver Operating Characteristic）曲线是以假正率（FP_rate）和真正率（TP_rate）为轴的曲线，ROC曲线下面的面积我们叫做AUC



2、PR曲线：即，PR（Precision-Recall）曲线。

举个例子（例子来自Paper：Learning from eImbalanced Data）：
假设N_c>>P_c（即Negative的数量远远大于Positive的数量），若FP很大，即有很多N的sample被预测为P，因为$FP_{rate} = \frac{FP}{N_c}$，因此FP_rate的值仍然很小（如果利用ROC曲线则会判断其性能很好，但是实际上其性能并不好），但是如果利用PR，因为Precision综合考虑了TP和FP的值，因此在极度不平衡的数据下（Positive的样本较少），PR曲线可能比ROC曲线更实用。

模型选择

- 在一个典型的监督学习的工作流程中，为了能够选出一个具有满意性能的模型，我们将会评估特征子空间、学习算法和超参数的各种不同的组合。
- 交叉验证法是一种好的方法，可以避免过拟合我们的训练数据。
- 把我们的数据随机分成训练和测试数据集。训练数据集将被用于训练模型，而测试数据集的作用是评价每次训练完成后最终模型的性能。重要的是，我们对测试数据集只使用一次，这样在我们计算预测误差指标的时候可以避免过度拟合。
- 过度拟合导致分类器在训练的时候表现良好，但是泛化能力一般。例如，对人类特征的识别，如果训练集中都是黑人，过度拟合的分类器就会将白人识别为非人类。

泛化

- 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。
  - 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。
- 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。

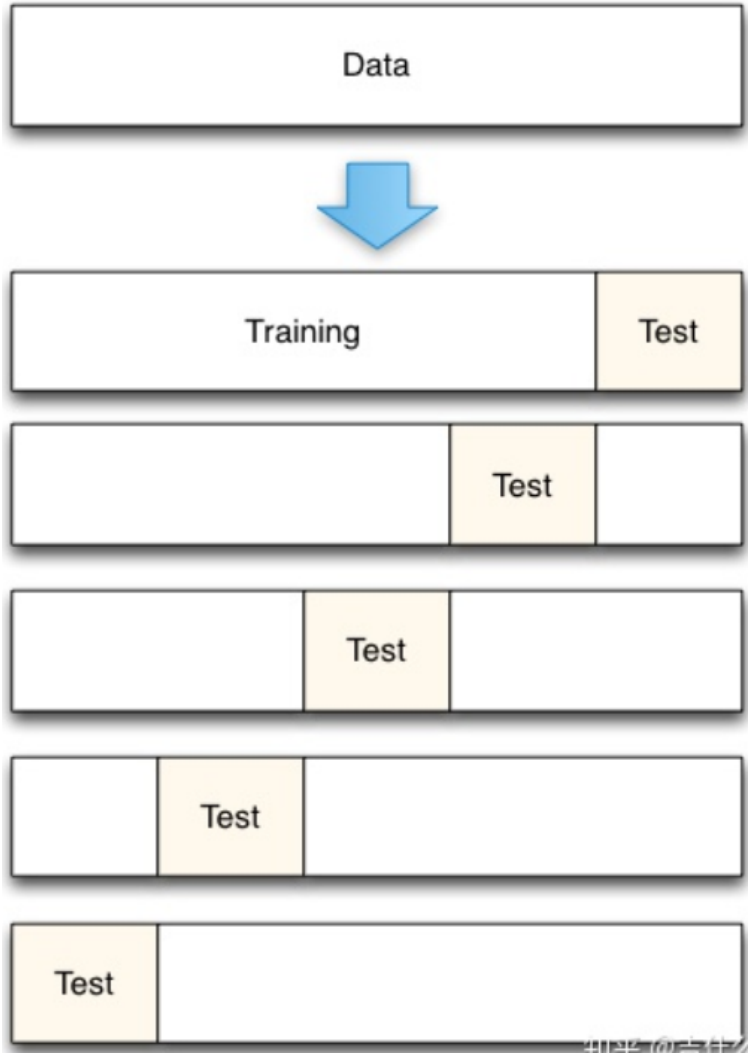
交叉验证

- 交叉验证是评估特征选择，降维，以及学习算法的不同组合的最有用的技术之一。
- 交叉验证有许多种，最常见的一种很可能是k折交叉验证了。
  - 在k-折交叉验证中，原始训练数据集被分成k个不同的子集，其中，1个被保留作为测试集，而另外的K-1个被用于训练模型。
  - 例如，如果我们设定k等于4（即，分为4份），原始训练集的3个不同的子集将被用于训练模型，而第四个子集将用于评价。经过4次迭代后，可以计算出最终模型的平均错误率（和标准差），这个平均错误率可以让我们看到模型的泛化能力如何。

交叉验证（cross-validation）

划分出训练集/测试集的不同会使得模型的准确率产生明显的变化。
为了消除这一变化因素，我们可以创建一系列训练集/测试集，计算模型在每个测试集上的准确率，然后计算平均值。
这就是 **K-fold cross-validation** 的本质。

K-fold cross-validation的步骤：
1.将原始数据集划分为相等的K部分（“折”）
2.将第1部分作为测试集，其余作为训练集
3.训练模型，计算模型在测试集上的准确率
4.每次用不同的部分作为测试集，重复步骤2和3 K次
5.将平均准确率作为最终的模型准确率



展望

- 自动构建模型的三要素：数据，算力，算法（学习的技巧）
  - 当我们有了足够的内存和计算能力，我们可以使用相对简单的算法来完成很多任务；这里的技巧是学习，或者从实例数据中学习，或者使用增强学习通过试错来学习。
  - 只要我们为机器提供足够的数据（不必是监督的）和计算能力，如果机器可以自己学习，我们则不需要提出新的算法。在人工智能的许多领域这种态势都将继续，而关键是学习。
  - 从算法角度来看，机器学习是一种元算法：生成算法（函数、模型）的算法

- 可以预见，机器学习技术将被应用到越来越多的领域，为研究者提供新的思路，还将给应用者带来更多的回报。

机器学习：大数据+大算力 =》大模型（多模态、预训练）

众包算法

- 很多任务对人类来说很容易，但是对于计算机来说，却很困难。例如，验证码的识别。
  - 验证码（CAPTCHA）是全自动区分计算机和人类的图灵测试（Completely Automated Public Turing test to tell Computers and Humans Apart）的缩写。
- 说起验证码的起源，还要从一个故事开始。
  - 2001年，雅虎为垃圾邮件问题所困扰，找到了CMU的教授，教授把这个任务分给他刚刚入学的博士生Luis von Ahn，该生想出了一个简单有效的解决方案：验证码。Luis von Ahn就是验证码的明人。

reCAPTCHA项目

- 2007年的时候，已经是教授的Luis von Ahn又来了一个点子。他认为人们输入CAPTCHA所花费的大量时间可以用得更有意义：帮助推进书籍数字化。
  - 于是他发起了reCAPTCHA项目。事情的起因是《纽约时报》古老的报纸存档打算数字化，但是由于时间久远且字迹不清楚，其中有很大的比例不是计算机AI能认识的，而人却能非常轻松地凭着模糊直觉和望文生义，识别其中的绝大多数。于是就有了reCAPTCHA这个新一代验证码系统：用户对于污染、扭曲文字的识别能力被用来处理数字化古籍中不能被计算机自动识别的文字。
  - reCAPTCHA的应用效果非常之好：reCAPTCHA被超过10万家网站使用，每天数字化超过4千万个单词。《纽约时报》所存的130年的资料，本来需要巨大的时间和人力资源的工程。通过reCAPTCHA系统，在几个月之内就由网友们完成了，而且是在网友们事前无知、事后惊讶中完成的。
  - reCAPTCHA系统的创新之处在于：让电脑去向人类求助。

具体做法是：将OCR软件无法识别的文字扫描图传给世界各大网站，用以替换原来的验证码图片；那些网站的用户在正确识别出这些文字之后，其答案便会被传回CMU。

解决方案的核心：“用户行为”（正确识别验证码）数据的二次开发。而用户行为数据几乎是没有成本的。

技术挑战

- 但是，挑战还是有的。技术挑战是什么？
  - 返回的结果对吗？
  - 有人会问，既然机器都看不明白那他怎么判断你输对了还是错了呢？

难点是什么？

我们问别人的问题，我们并没有答案，那么我们如何判断别人的回答是否正确？

思路

1）判断回答者的水平：如果水平高，就可以假设其回答是正确的。
2）多问几个人：如果大多数人都这么说，就假设其回答是正确的。

具体方法

针对这个问题，一个典型的方案是使用两个验证码，两个验证码里面有一个是正确的，被人审核过的，而另一个是未被审核的。当你把那个正确的输对以后我们就会默认另外一个也是对的，这样，你每输入一次验证码，就为人类的知识宝库里增加了一个单词。进一步，我们还可以通过把未被审核的验证码发给多个用户来提高结果的可靠性：如果2个用户的识别结果相同，无疑该结果会更为可靠。

基于人的计算（Human-based computation）

- Luis von Ahn提出了基于人的计算（Human-based computation）的概念，具体而言就是机器把要实现功能分解为很多微任务，把其中某些步骤外包给人来完成。（a machine performs its function by outsourcing certain steps to humans, usually as microwork.）
- 众包就是一个典型的情况。
  - 众包（Crowdsourcing）是指把一个问题分解为很多子问题，然后把这些问题外包给可能是一组分布在不同地域的大众，然后聚合其结果，计算出最终的答案。例如，《纽约时报》面临的识别大量的文字扫描图任务。

图 7-3　众包模型

- 例如，对图片的分类，可以把待分类的图片分发多个人，整合其结果，按照算法计算出最终的结果。如图7-4所示。著名的数据集ImageNet就是通过众包的方式对图片进行标注的。

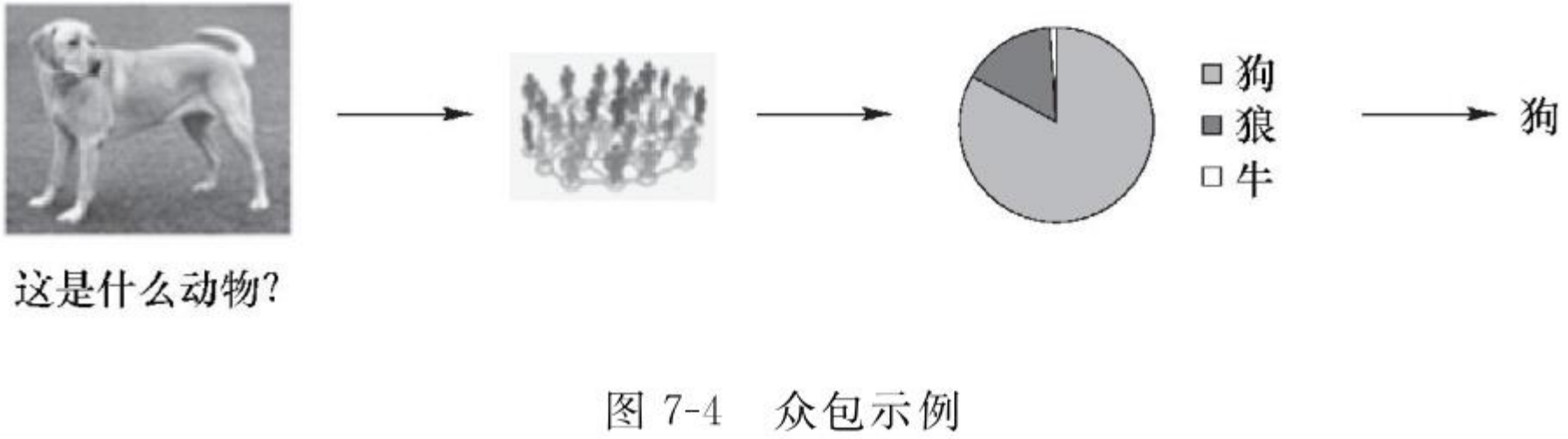


图 7-4　众包示例

在众包计算中，人首次作为一个计算设备出现在计算机系统中，并且能够引入大量的人参与到任务中来，来协助计算机完成其不能胜任的计算步骤。
这种新的模式，有效地将计算机智能与人的智能结合，并通过巧妙的设计来低成本地完成任务。

大数据算法的安全问题

典型案例

- 对传统大数据算法的攻击
  - 通过伪造共同访问对推荐系统进行攻击
  - 搜索引擎优化（SEO）
- 对机器学习算法的攻击（*）
  - 诱导分类器产生错误分类
  - 诱骗视觉分类算法

推荐系统

推荐系统（Recommender System）在现代电子商务和广告平台中起着关键作用。面对数以万计的商品或服务，用户往往需要依赖推荐系统来找到自己真正感兴趣的东西。因此，商家也愈加依赖推荐系统作为用户入口，例如亚马逊公司提出的基于物品的协同过滤算法（item-based collaborative filtering，ICF)推荐系统。

ICF 算法的思想是给用户推荐那些与其之前喜欢的项目相似的项目，首先计算项目相似度矩阵，然后根据用户行为数据为用户提供推荐列表。

2003_Item-to-Item Collaborative Filtering

推荐系统

ICF算法并不是利用项目的自身属性来计算项目之间的相似度，它主要通过分析用户的历史行为数据，计算得到项目相似度。比如，喜欢项目A的用户大部分也喜欢项目B，那么项目A和项目B就具有较高的相似度。用户在访问亚马逊网站的商品页面时，网页中会出现“购买此商品的顾客也同时购买”的部分。

如图当用户搜索《Java 编程思想》时，网站会为用户推荐《Java 核心技术》和《深入理解Java 虚拟机》等书籍。

推荐攻击

这类推荐系统中，一些恶意用户为了谋求利益，会向推荐系统中注入虚假用户，希望提升或降低某些项目的评分，使推荐系统的预测产生偏差，产生利于自己的推荐结果，这类攻击行为称为“托攻击”。

根据攻击目的的不同，“托攻击”可以被分为“推攻击（Push Attack）”和“核攻击（Nuke Attack）”，

- 推攻击的目的是提升目标项目的推荐频率，使目标项目获得更高的关注。
- 核攻击是降低目标项目的推荐频率，达到干扰竞争对手或是降低推荐系统的推荐质量的目的。

推荐攻击

假设推荐系统中有用户 User 和用户 1~用户 7 共八个用户，项目 1~项目 6 共六个项目，给出用户项目评分矩阵，用户对项目的评分为 1~5，推荐系统需要判断要不要将项目 6 推荐给 User。

如果推荐系统采用基于用户的协同过滤算法，首先通过用户相似度计算 User 的近邻用户，假设该推荐算法采用皮尔逊相似度，那么计算的结果是用户 1 为 User 的最近邻用户，那么根据用户 1 的用户行为分析，用户 1 对于项目 6 的评分为 2 分，那么用户 User 很可能对项目 6 不感兴趣，推荐系统不会做出推荐。

	项目 1	项目 2	项目 3	项目 4	项目 5	项目 6	与 User 的相似度
User	5	3	-	3	2	?	-
用户 1	4	3	3	-	3	2	0.94
用户 2	-	1	2	-	3	-	-1.00
用户 3	3	1	3	2	3	1	0.21
用户 4	-	1	5	-	5	1	-1.00
用户 5	4	1	-	3	2	1	0.72
用户 6	2	-	4	4	-	1	-1.00
用户 7	3	-	1	3	1	2	0.76
攻击者 1	5	2	-	2	2	5	0.93
攻击者 2	5	-	2	3	-	5	1.00
攻击者 3	5	-	2	4	1	5	0.89
与项目 6 的相似度	0.85	0.48	-0.59	0.00	-0.55	-	-

托攻击示例

推荐攻击

假设推荐系统中有用户 User 和用户 1~用户 7 共八个用户，项目 1~项目 6 共六个项目，给出用户项目评分矩阵，用户对项目的评分为 1~5，推荐系统需要判断要不要将项目 6 推荐给 User。

攻击者通过概貌注入的方式向推荐系统中加入三个攻击用户：攻击者1~攻击者3，这三个攻击用户的攻击目标是项目 6，他们想要提高项目 6 的推荐频率，于是他们努力成为其他用户的邻居用户，并给项目 6 评价最高分，那么此时推荐系统重新计算出 User的最近邻用户是攻击者 2，由于攻击者 2 给予项目 6 的评分是 5 分，于是推荐系统会认为 User 对项目 6 感兴趣。

	项目 1	项目 2	项目 3	项目 4	项目 5	项目 6	与 User 的相似度
User	5	3	-	3	2	?	-
用户 1	4	3	3	-	3	2	0.94
用户 2	-	1	2	-	3	-	-1.00
用户 3	3	1	3	2	3	1	0.21
用户 4	-	1	5	-	5	1	-1.00
用户 5	4	1	-	3	2	1	0.72
用户 6	2	-	4	4	-	1	-1.00
用户 7	3	-	1	3	1	2	0.76
攻击者 1	5	2	-	2	2	5	0.93
攻击者 2	5	-	2	3	-	5	1.00
攻击者 3	5	-	2	4	1	5	0.89
与项目 6 的相似度	0.85	0.48	-0.59	0.00	-0.55	-	-

托攻击示例

推荐攻击

假设推荐系统中有用户 User 和用户 1~用户 7 共八个用户，项目 1~项目 6 共六个项目，给出用户项目评分矩阵，用户对项目的评分为 1~5，推荐系统需要判断要不要将项目 6 推荐给 User。

如果推荐系统采用基于物品的协同过滤算法，攻击者会将目标项目与流行项目联系起来，我们演示的示例中项目 1 为流行项目，于是攻击者为项目 1 和项目 6 都评价 5 分，使项目 6 成为项目 1 的近邻项目，从而增加项目 6 被推荐的可能性。

托攻击会改变推荐系统的预测结果，推荐结果向着托攻击者希望的方向产生偏移，如果这种情况继续蔓延下去，用户很可能产生厌烦心理，从而丧失对该推荐系统的信任。

	项目 1	项目 2	项目 3	项目 4	项目 5	项目 6	与 User 的相似度
User	5	3	-	3	2	?	-
用户 1	4	3	3	-	3	2	0.94
用户 2	-	1	2	-	3	-	-1.00
用户 3	3	1	3	2	3	1	0.21
用户 4	-	1	5	-	5	1	-1.00
用户 5	4	1	-	3	2	1	0.72
用户 6	2	-	4	4	-	1	-1.00
用户 7	3	-	1	3	1	2	0.76
攻击者 1	5	2	-	2	2	5	0.93
攻击者 2	5	-	2	3	-	5	1.00
攻击者 3	5	-	2	4	1	5	0.89
与项目 6 的相似度	0.85	0.48	-0.59	0.00	-0.55	-	-

托攻击示例

阅读材料 – 推荐系统的安全与隐私

1) 2003_Item-to-Item Collaborative Filtering
2) 2017_Two Decades of Recommender Systems at Amazon.com
3) Guolei Yang, Nei Zhenqiang Gong, Ing Cai. Fake Co-visitation Injection Attacks to Recommender System[C]//In Proceedings of the Network and Distributed System Security Symposium (NDSS), 2017.
4) 周俊,董晓蕾,曹珍富. 推荐系统的隐私保护研究进展[J]. 计算机研究与发展, 2019, 56(10): 2033-2048.

搜索引擎优化（SEO）：概念和应用

搜索引擎优化是指通过了解各类搜索引擎如何抓取互联网页面、如何进行索引以及如何确定其对某一特定关键词的搜索结果排名等技术，来对网页内容进行相关的优化，使其符合用户浏览习惯，在不损害用户体验的情况下提高搜索引擎排名，从而提高网站访问量，最终提升网站的销售能力或宣传能力的技术。

在国外，SEO开展较早，那些专门从事SEO的技术人员被Google称之为“搜索引擎优化工程师SEOers”。由于Google是世界最大搜索引擎提供商，所以Google也成为了全世界SEOers的主要研究对象，为此Google官方网站专门有一页介绍SEO，并表明Google对SEO的态度。

对于任何一家网站来说，要想在网站推广中取得成功，搜索引擎优化是极为关键的一项任务。科学规范的SEO对搜索引擎是一种人性化的完善。

搜索引擎优化（SEO）：作弊

由于不少研究发现，搜索引擎的用户往往只会留意搜索结果最开始的几项条目，所以不少商业网站都希望透过各种形式来干扰搜索引擎的排序，在网站里尤以各种依靠广告为生的网站最甚。SEO技术被很多目光短浅的人，用一些SEO作弊的不正当手段，牺牲用户体验，利用搜索引擎的漏洞，来提高排名。

自从有了搜索引擎，就有了针对搜索引擎网页排名的作弊(SPAM)。因此，有时候用户会发现在搜索结果中排名靠前的网页不一定是高质量的信息。

搜索引擎优化（SEO）：作弊方法

- 搜索引擎的作弊，虽然方法很多，目的只有一个，就是采用不正当手段提高自己网页的排名。
- 早期最常见的作弊方法是重复关键词。比如一个卖数码相机的网站，重复地罗列各种数码相机的品牌，如尼康、佳能和柯达等等。为了不让读者看到众多讨厌的关键词，聪明一点的作弊者常用很小的字体和与背景相同的颜色来掩盖这些关键词。其实，这种做法很容易被搜索引擎发现并纠正。
- 在有了网页排名(page rank)以后，作弊者发现一个网页被引用的连接越多，排名就可能越靠前，于是就有了专门卖链接和买链接的生意。
  - 比如，有人自己创建成百上千个网站，这些网站上没有实质的内容，只有到他们的客户网站的连接。这种做法比重复关键词要高明得多，但是还是不太难被发现。因为那些所谓帮别人提高排名的网站，为了维持生意需要大量地卖链接，所以很容易露马脚。

搜索引擎优化（SEO）：人工的手段

- 随着网络水军的出现，还有一些人开始雇佣水军，在博客和论坛上注入垃圾评论，把链接注入排名高的博客网站和论坛，从而欺骗搜索引擎，获得更好的排名。
- 2005年初，Google为网页链接推出一项新属性nofollow，使得网站管理员和网络博客作者可以做出一些Google不计票的链接，也就是说这些链接不算作“投票”。nofollow的设置可以抵制垃圾评论。

搜索引擎优化（SEO）：自动化防作弊方案

除了人工的手段，搜索引擎公司还提出了自动化防作弊方案，其基本的思路和信号处理中的去噪音的办法很类似。

在信号处理技术中，原始的信号混入了噪音，在数学上相当于两个信号做卷积。那么，去除噪声的过程就是一个解卷积的过程。只要噪声的频率是固定的，由于噪声总是重复出现的，只要采集几秒钟的信号就可以利用解卷积算法把噪声消除。从理论上讲，只要噪音不是完全随机的、并且前后有相关性，就可以检测到并且消除。

搜索引擎的作弊者所作的事，就如同在原始信号中加入了噪音，打乱了搜索结果的正确排名。但是，这种人为加入的噪音并不难消除，因为作弊者的方法不可能是随机的，否则就无法提高排名了。而且，作弊数据在时间上是具有前后相关性的，因为作弊方法不可能老变，不然作弊的成本太高，如果作弊成本高于收益，作弊者自然就放弃作弊了。因此，搞搜索引擎排名算法的人，可以在搜集一段时间的作弊信息后，将作弊者抓出来，还原原有的排名。

搜索引擎优化（SEO）：对抗

由于作弊信息的采集过程需要时间，在这段时间内作弊者可能会尝到些甜头。因此，有些人看到自己的网站通过作弊，排名在短期内靠前了，以为这种所谓的优化是有效的；但是，过不了多久就会发现排名又掉下去了。

当然，作弊者也会不断改进自己的方法，降低成本，延长有效时间，以达到作弊的目的。作弊与反作弊和所有安全问题一样，是一种相生相伴、长期对抗的关系，至今还没有一个一劳永逸地解决作弊问题的方法。

诱导分类器产生错误分类

- 机器学习在很多应用领域取得了成功。
- 很多安全领域的难题也用到机器学习技术来解决，比如垃圾邮件分类、僵尸号检测、恶意软件分类等。
  - 2015年微软在Kaggle上赞助了一个Windows恶意软件分类比赛，冠军队赛前并没有任何恶意软件知识，仅凭基本的机器学习技能就赢得第一名，模型准确率接近100%。
- 2016年，在安全领域的著名会议NDSS上有一篇论文指出，机器学习做安全只是看起来很美。该论文采用遗传编程（Genetic Programming）随机修改恶意软件的方法，成功攻击了两个号称准确率极高的恶意PDF文件分类器：PDFrate和Hidost。
  - 论文还披露了Gmail内嵌的恶意软件分类器更加脆弱，只须4行代码修改已知恶意PDF样本就可以达到近50%的逃逸率，10亿Gmail用户都受到影响。然而Google安全团队表示恶意软件检测是个大难题，他们暂时也无能为力。

诱骗视觉分类算法

以往的对抗攻击需要进行复杂的数据处理。Evtimov发现表明在物理世界中的图像进行轻微的改变，就能成功地诱骗视觉分类算法。

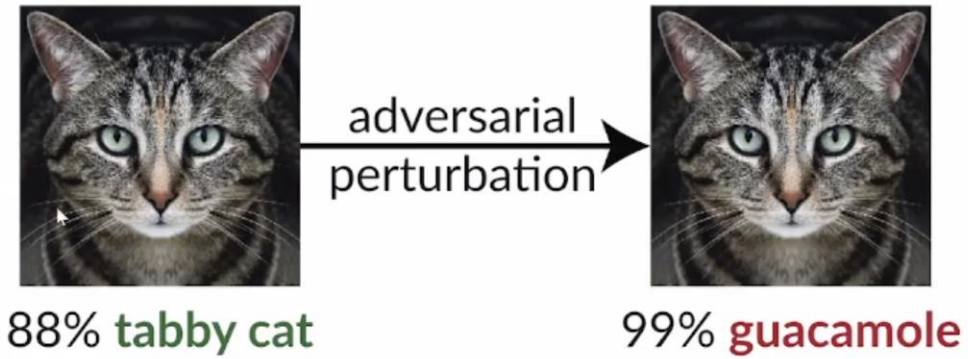
人类非常难以理解机器人是如何“看”世界的。机器的摄像头像我们的眼睛一样工作，但在摄像头拍摄的图像和对于这些图像能够处理的信息之间的空间里，充满了黑盒机器学习算法。训练这些算法通常包括向机器显示一组不同的图像（比如停止标志），然后看看机器能否从这些图片中提取足够的常见特征，从而可靠地识别出那些没有在训练集中出现过的停止标志。

你只需要在停车标志上加一点喷漆或一些贴纸，就能够愚弄一个深度神经网络分类器，让神经网络将停止标志看成是限速标志。

Evtimov I, Eykholt K, Fernandes E, et al. Robust physical-world attacks on machine learning models[J]. arXiv preprint arXiv:1707.08945, 2017.

- 这些事例都体现了安全领域的特殊性。
  - 攻击者会不断改变策略，至少不能以传统的机器学习视角来看待安全类问题了。
  - 在安全领域应用机器学习构建分类器，不仅仅需要关注准确率、误报率之类的传统度量，更需要深入分析其分类特征是否能够有效的应对攻击者的挑战，否则机器学习做安全只是看起来很美而已。

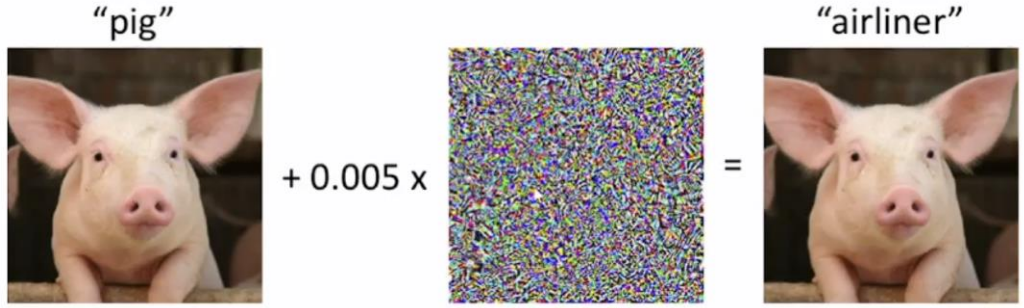
对抗样本



Szegedy等人首先注意到图像分类中存在对抗性的例子。他们通过向原始样本中加入精心设计的扰动，生成带有扰动的样本。

但是，这种带有扰动的样本会导致目标模型对其给出**高置信度的错误分类**结果。

Szegedy等人称这种带有精心设计的扰动的样本为对抗样本。



Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[C]//2nd International Conference on Learning Representations, ICLR 2014.

理论解释 – Shamir 2021

In our model, easy training and sensitivity to adversarial attacks are two sides of the same coin:

- When we train a network: the data points are fixed and the decision boundary is movable
- When we create adversarial examples, the decision boundary is fixed and the data points are movable

- The third property implies that small dimples suffice to completely change the classification for the training examples, but also creates adversarial examples which are very close to any training example

Our new theory makes three easily testable claims:

- Natural images are located on a low dimensional manifold
- The decision boundary clings very closely to the image manifold
- There is a large gradient whose direction is roughly perpendicular to the image manifold.

https://arxiv.org/abs/2106.10151 The Dimpled Manifold Model of Adversarial Examples in Machine Learning

小结

- 本章首先学习了数学模型的几个要素：假设，模型，实践，理论解释；并通过人类对宇宙模型的认知这个最广为人知的例子，解释了各个要素的作用和关系。然后，以我们每天都在使用的搜索引擎为例，介绍了搜索算法的基本原理，并解读了数学模型和算法在其中发挥的作用。
- 介绍了机器学习的基本概念和常用方法，围绕着以下三个问题来详细展开：1）为什么需要机器学习？2）机器学习的基本原理。3）机器学习的常用方法。
- 针对人工智能的不足，介绍了众包的思想与应用案例。
- 最后，介绍了几种针对典型的大数据算法的攻击。1）针对电子商务系统中的推荐算法的攻击。2）针对搜索引擎的作弊与防范手段。3）针对恶意软件分类器的攻击。4）针对视觉分类器的攻击。

阅读材料 – 机器学习算法安全

1) Weilin Xu, Yanjun Qi, David Evans. Automatically evading classifiers[C]// Proceedings of the 2016 Network and Distributed Systems Symposium. 2016.
2) Weilin Xu, David Evans, Yanjun Qi. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks[C]// 2018 Network and Distributed System Security Symposium. San Diego, California, 2018: 18-21.
3) Evtimov I, Eykholt K, Fernandes E, et al. Robust physical-world attacks on machine learning models[J]. arXiv preprint arXiv:1707.08945, 2017.
4) Evan Ackerman. Slight Street Sign Modifications Can Completely Fool Machine Learning Algorithms[J]. IEEE Spectrum, 2017.

5) Guolei Yang, Nei Zhenqiang Gong, Ing Cai. Fake Co-visitation Injection Attacks to Recommender System[C]//In Proceedings of the Network and Distributed System Security Symposium (NDSS), 2017.

北京郵電大學
Beijing University of Posts and Telecommunications

感谢聆听！