目录

1 引言

1.1 当今的时代背景——大数据

早在 20 世纪末, 互联网便已经展现出了势不可挡的趋势, 随着互联网的迅速发展, 大数据成为了这个信息时代的标志词. 它的发展迅速到甚至可能我们连什么是大数据都不清楚, 就已经在生活的各个方面受到它的影响了. IBM 提出大数据有 5 个特点, 分别是 Volume、Velocity、Variety、Value、Veracity, 分别代表着大量、高速、多样、低价值密度和真实性. **图 1** 描述了近过去十年以及未来五年的数据量增长趋势, 可以很明显的看出数据量的增长目前还是呈指数形式. 这也意味着数据量的增长会犹如指数爆炸一般十分迅猛, 因此, 如何有效地利用如此海量的数据便是当务之急.

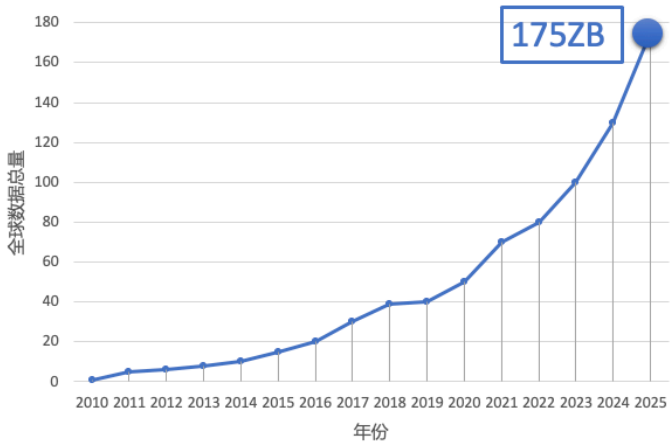

图 1: Rapid growth of data.

由于对于数据处理的需求, 一些技术如“数据挖掘”,、“机器学习”便被提了出来, 它们旨在运用现代强大的计算机和优秀的程序与算法. 对当今海量的数据进行处理, 从而找到一些有价值的信息. 随着数据量的大幅增大, 单台计算机已经无法满足数据处理的需要, 而以 Hadoop 为代表的开源分布式计算架构则提供了分布式计算的技术支持. 同时, 随着 Caffe 和 TensorFlow 等高效率的深度学习框架被开源, 许多小型公司甚至个人也具备了自主研发改进算法和模型的能力. 互联网在不断发展, 数据的生成也不会停下脚步.

如**图 1** 所示, 2020 年全球数字宇宙将会已经超过 50ZB, 虽然我们目前还不清楚这些数据如何被更加充分的利用, 不过可以肯定的是, 数据会成为一项重要的资源, 在大数据时代, 特别是未来的智能化时代, 如机器学习这种对于数据的处理技术一定会展现出更大的潜能. 正如马云所言, 人类社会的未来一定会进入数据处理技术 (Data Technology, DT) 时代.

图 2: Google Trend about Machine Learning.

图 2 是“机器学习”这一词语在研究领域的谷歌搜索热度趋势 [1], 可以看出, 自 2015 年以

来, 机器学习的搜索热度便急转直上, 这一结果可能要归功于 Google 旗下的 DeepMind 以及他们的 AlphaGo. AlphaGo 是一款通过机器学习构建出的围棋对弈系统, 其第一代系统于 2015 年 3 月与顶尖棋手李世乭进行了对弈, 并最终以 3:1 的大比分取得胜利. 经此一役, AlphaGo 以及其背后的机器学习进入了大众的视野.

值得留意的是, 这项数据是从 2004 年开始的, 也就是说 Google 公司至少从 2004 年就开始不惜耗费十分巨大的财力与物力来保存这些来自全球的庞大的搜索数据了. 同样的, Facebook 也在运用机器学习对每一个用户的每一条浏览记录, 每一次点赞进行分析, 从而为用户推送最为合适的内容以及广告. 国内公司对机器学习也有十分深入的应用, 如 Alibaba 的人脸支付, 能够在几亿用户中大规模正式上线人脸支付系统, 说明了 Alibaba 公司本身对于机器学习中人脸识别这一技术已经研发的十分完善. 而从 2000 年开始就有即时通讯业务的 Tencent 公司也将机器学习应用至很多产品, 如 QQ 中“可能认识的人”, 或者音乐软件, 视频软件里可能喜欢的视频或者音乐, 里面都有机器学习的影子.

类比于人脑的学习过程, 机器的学习似乎具有相似的性质, 人脑的学习过程 (**图 3**) 可以抽象为从经验之中总结出规律, 遇到新问题时基于过去总结而来的规律做出判断; 而机器学习的学习过程 (**图 4**) 则可以抽象为计算机从人为输入的历史数据中训练出模型, 对于新出现的的数据, 由已经训练好的模型给出预测. **图 5**

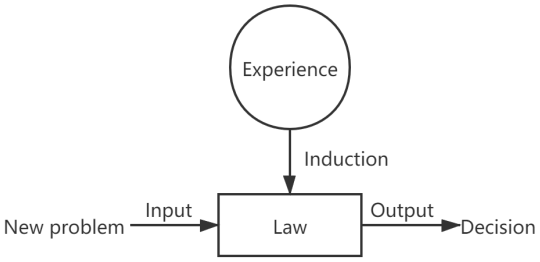

图 3: How our brians learn.

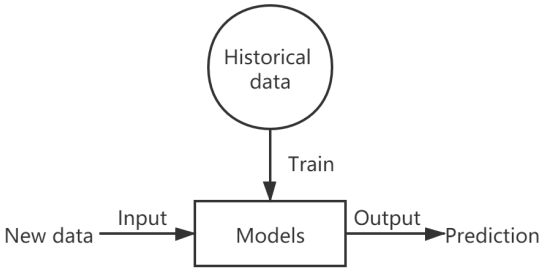

图 4: How machines learn.

但机器学习本身却并不是一个十分年轻的事物, 它最早的应用可以追溯至 1952 年 IBM 工程师 Arthur Samuel 写出的一个跳棋程序, 这个程序通过学习走得好和不好的棋局, 为下棋者做出指导 [2][3]. 在过去的几年里, 机器学习、人工智能乃至深度学习都一度成为了资本投资的风口, 但其实此三者之间其实是包含与被包含的关系, 如同俄罗斯套娃一般 (**图 5**).

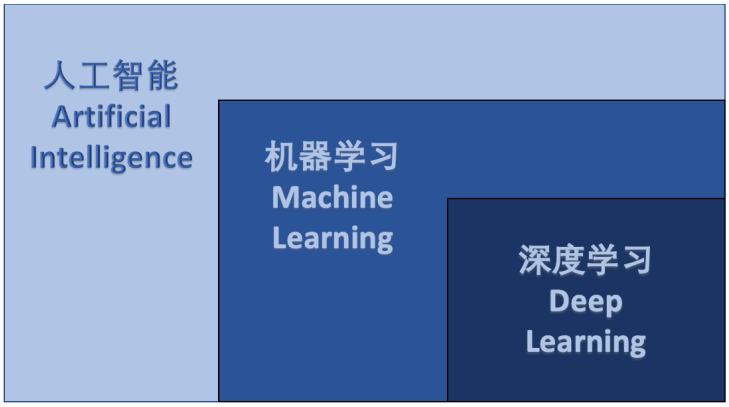

图 5: AI, Machine Learning & Deep Learning.

1.2 人工智能

“人工智能”的概念自 1956 年在达特茅斯会议 [4] 上首次正式提出以来, 随着技术手段的发展也在不断地变化, 但它的核心是可以总结为“将由人类完成的智力任务自动化”. 因此, 任何可以实现这一目标的手段与方法都可以冠以“人工智能”的名字.

但实现人工智能的方法并不仅限于机器学习, 例如早期的国际象棋程序 Deeper Blue. 国际象棋的棋局复杂度 (35^{80}[5]) 并非特别庞大, 因此可以通过穷举法将每一种棋子走法都由程序编写后交给计算机进行计算, 而 Deeper Blue 通过这一方法成功地战胜了国际象棋世界冠军. 在 Deeper Blue 的下棋过程中, 不需要学习什么, 仅仅通过内置的程序进行计算即可. 这种通过编写复杂规则来实现人工智能的方法被称为符号主义人工智能.

但后来, 随着人们需要处理的信息越来越复杂, 如图像、音频或者是棋局复杂度十分巨大的围棋 (250^{150}[5]), 符号主义人工智能已经不能够处理这些问题了, 因此, 一种新的人工智能实现方法便出现了, 它便是“机器学习”.

1.3 机器学习

1950 年, 人工智能先驱 Alan Turing 发表了“Computing machinery and intelligence”一文 [6], . 文中, 他对“计算机是否能够学习与创新”这一问题进行了思考, 并且给出了肯定的回答: 能.

经典的符号主义人工智能中, 我们输入的是数据以及精心编写的规则, 输出的是人们期望得到的答案 (**图 6**). 而在机器学习中, 输入的是数据和答案, 人们则希望计算机从这些数据与答案中学习到一些规则 (**图 7**).

图 6: What Symbolic AI do.

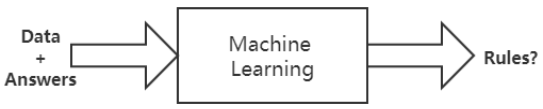

图 7: What ML do.

这一过程中, 计算机的学习系统是通过对输入数据以及对应的答案的观察来抽象出隐藏规则的. 到了现代, 甚至有许多问题我们只知道数据而不知道答案, 这时我们会只向计算机输入数据, 期待他能给我们答案, 这就是所谓的无监督学习. 事实表明, 一些并不借鉴人类输入数据的强化学习模型如 AlphaGo Zero, 给出的答案甚至要比人类自己的答案好很多.

1.4 深度学习

在普通的机器学习问题中, 我们需要计算机去学习的规则, 往往可以表达成输入数据的函数或者复合函数, 而在这些规则中有一部分的复合函数的复合次数甚至高达数十次乃至上百次, 对于这些复合函数十分复杂的问题, 我们称它们为深度学习问题, 深度学习的“深度”, 便指的是这种多次复合函数的次数.

而在实际的深度学习问题中, 这种多次的复合函数总是通过一种名叫“神经网络”的结构来实现的, **图 8** 便是一个三层的神经网络. “神经网络”这一术语的灵感来源于神经生物学. 虽然它的每一层十分简单, 但经过许多层的堆叠后, 往往可以对一些十分复杂的问题如图像识别等产生令人十分震惊的效果. 正像人的大脑虽然只由一个个细胞构成, 但却有十分复杂的智慧一样. 有

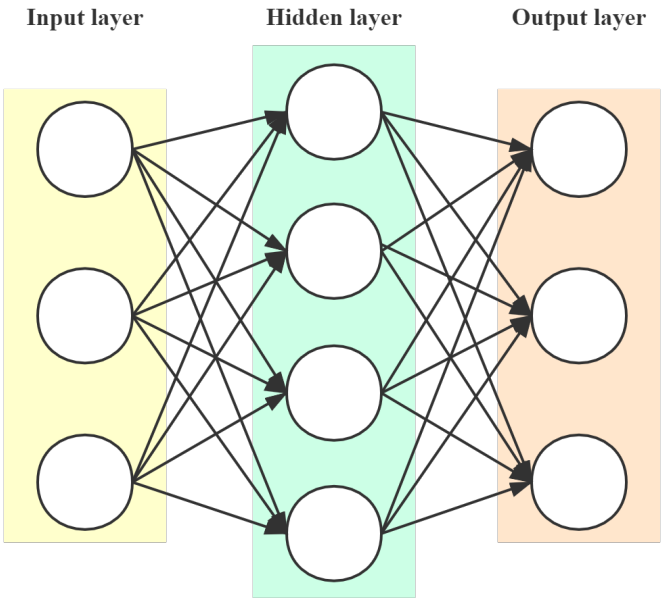

图 8: A three-layer full connected neural network

监督的神经网络机器学习的基本原理如**图 9** , 它通过预测值与真实值的误差来调整自己每一层的参数, 使预测值变得更加精准.

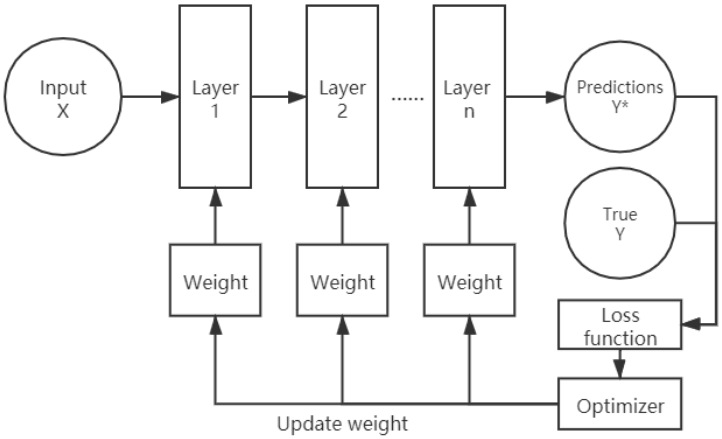

图 9: The workflow of neural network ML.

2 机器学习的发展与应用

2.1 发展历程

正如我们学习历史知识一样, 从机器学习的发展历史中了解它也是一种十分便捷的方式.

20 世纪 50 - 70 年代

此时, 人工智能正处于“推理期”, 即人们用编程赋予机器逻辑推理能力, 代表工作有: A. Newell 和 H. Simon 的“逻辑理论家”(Logic Theorist) 程序以及此后的“通用问题求解”(General Problem Solving) 程序等, 而他们二人也在之后获得了获图灵奖.

20 世纪 70 年代中期开始, 在 E.A.Feigenbaum 等的倡导下, 人工智能研究进入了“知识期”, 大量的专家系统问世. E.A.Feigenbaum 等人也获得了图灵奖. 但是后来, 知识工程也遇到瓶颈, 人们发现总结知识再教给计算机是相当困难的, 因此应该设法让机器自己学习知识. 而图灵在 1950 年的文章中, 肯定了机器学习的可能性；之后, 又陆续有基于神经网络的“连接主义”机器学习、基于逻辑表示的“符号主义”学习技术、以决策理论为基础的学习技术以及强化学习技术得到发展.

20 世纪 80 年代

1986 年, 机器学习 Machine Learning 创刊；1989 年, 人工智能权威期刊 Artificial Intelligence 出版机器学习专辑；20 世纪 80 年代机器学习成为一个独立学科领域, 同时, 各种机器学习技术也呈现着百花初绽的态势.

此时的研究热门都聚集于“从样例中学习”, 包括监督学习, 以及无监督学习. 其中, 以符号主义机器学习和基于神经网络的连接主义机器学习为主流, 前者包括决策树、基于逻辑的学习等, 后者虽然产生的是“黑箱”模型, 但由于能够有效地解决许多现实问题, 得到了广泛的应用.

20 世纪 90 年代

“统计学习”成为了这个年代的主流, 代表技术有支持向机 SVM、“核方法”等. 而统计学习事实上与连接主义学习关系密切, 支持向量机被普遍接受后, 核技巧 (kernel trick) 被广泛应用到机器学习各处, 核方法逐渐成为机器学习基本内容.

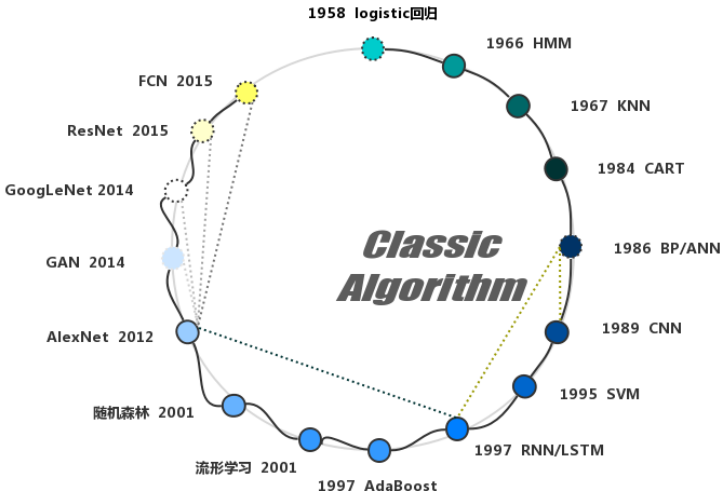

图 **10:** The evolution of the algorithm.

21 世纪

“深度学习”成为了当代的主流, 以往的机器学习技术对使用者要求比较高, 而相比于以往的机器学习技术, 深度学习技术将对于使用者的高要求转移到模型的复杂度上了, 以至于只要下功夫让计算机来调参, 性能往往也不会很坏. 因此, 虽缺乏严格的理论基础, 但它显著降低了机器学习应用者的门槛, 为其走向工程实践带来便利.

图 **10** 是机器学习中一些典型算法的发展过程.

2.2 应用案例

机器学习在当今其实已经“无处不在”, 它普遍应用于人工智能的各个领域, 包括数据挖掘、计算机视觉、自然语言处理、语音和手写识别、生物特征识别、搜索引擎、医学诊断、信用卡欺诈检测、证券市场分析、汽车自动驾驶、军事决策等.

异常检测 异常指的是某个数据对象由于测量误差、收集中的损失或自然变异等原因变得不同于正常的数据的场景, 而将这些异常辨别出的过程, 通常称为异常检测. 异常检测的训练样本都是非异常样本, 假设这些样本的特征都是服从高斯分布, 则在此基础可以建立一个概率模型, 用来估计就有某个特征的样本属于非异常样本的可能性. 比如当用户的信用卡发生了一笔特征与平常的交易的特征不同的交易请求时, 就有可能是信用卡被盗刷, 这是银行就可以根据风控系统来防止这样的盗刷发生.

用户画像 用户画像的核心工作就是给用户打标签, 如年龄、性别、地域、兴趣等. 公司从这些标签集合能抽象出一个用户的信息全貌. 从而在产品的运营和优化中, 根据用户画像深入理解用户需求, 从而设计出更适合用户的产品, 设定复合用户消费能力的价格区间, 提升用户体验等. 现在的各个 APP 对于用户的个性化推荐、支付宝的芝麻信用分等都属于用户画像的具体应用.

广告点击率预估 互联网广告是互联网公司主要的盈利手段, 当今的科技巨头 Google 便是以广告为主业. 互联网广告交易的双方是广告主和媒体. 为自己的产品投放广告并为广告付费；媒体是有流量的公司, 如各大门户网站、各种论坛, 它们提供广告的展示平台, 并收取广告费. 广告点击率是指广告的点击到达率, 即广告的实际点击次数除以广告的展现量. 在实际应用中, 我们将海量的历史点击数据作为样本来训练模型, 从而评估各方面因素对点击率的影响. 作为这一模型的应用, 当有新的广告位请求到达时, 就可以用训练好的模型, 根据广告交易平台传过来的相关特征预估这次展示中各个广告的点击概率, 结合广告出价计算得到的广告点击收益, 从而选出收益最高的广告向广告交易平台出价.

2.3 Uber 的具体应用

我们以 Uber 为例做一个具体的分析, 2017 年 9 月, Uber 介绍了他们的机器学习平台–Michelangelo. 应用 Michelangelo, Uber 做出了一些改变和发展.

Uber Eats

基于 Michelangelo, Uber Eats 会根据的预测到达时间、历史数据以及餐馆的实时信息, 来估算餐食的送达时间.

市场预测

Uber 的高峰期动态调价机制虽然是市场决定的, 但对用户来说绝对是 Uber 最大的槽点之一. Uber 的市场团队通过机器学习, 希望预测一大批客户群将在何时何地有打车需求. 这样, 在动态定价机制 (surge pricing) 启动之前, 让更多的 Uber 车辆能够提供服务. 在涨价之前, 道路上会有更多待命的司机等待这即将到来的大量打车需求, 而同时, 乘客们也不必苦等出租车. 但这也不意味着采用机器学习方式, 动态定价机制会消失. 因为对市场来说, 肯定还是价高者得, 通过价格的浮动, 使供求动态达到平衡.

预计到达时间 (ETAs)

对公司来说, 最重要的指标之一就是各种预估时间. 精确的预估时间对好的用户体验至关重要, 这些指标被输入无数其他的内部系统中, 来协助判定价格和路线.

Uber 的地图服务团队开发了一个复杂的分段路线系统, 用来预估基本的时间值. 这些基本的预估时间具有相同类型的错误. 地图服务团队发现他们可以使用机器学习模型来预测这些错误, 并用预测的错误来进行修正. 由于这个模型正逐个应用在各个城市, Uber 团队发现预估到达时间的准确性大幅提升, 在某些情况下, 平均预估到达时间的误差减小了 50% 以上.

3 统计学跟机器学习的对比

传统统计学统计与机器学习都是通过现有的数据预测或是还原一些关系的学科.

机器学习是目标驱动的, 为了解决在其他领域里遇到的挑战与困难, 不断地改进现有的方法或者提出新的想法. 最终目标是不断地更好地解决实际问题. 并且已经成功地在目标识别, NLP, 人机对弈等等领域里取得了传统统计学无法想象的成就. 人们想到机器学习的时候想到的不是一个一个数学定理, 而是机器学习所解决的这些关键的问题.

传统统计学, 在解决问题的同时也更多的注重于理论的完备性, 解决问题并且解释问题. 传统统计学的发展往往依赖并推动着随机矩阵, 随机过程, 泛函分析与几何学等应用数学甚至纯数学等领域里的的新方法的提出. 我们想到统计学的时候往往想到的是各种各样的统计方法以及这样的我们对这样的方法所能做出的评价.

同样一个问题在传统统计学与机器学习的视角下可以有这极大的研究方向的不同. 比如推荐系统. 根据用户的观看数据去预测用户的喜好从而达到给用户推荐新的电影小说或者是商品的这样一个任务. 机器学习跟传统统计学相比可能就有着完全不同的解决问题的途径.

3.1 机器学习视角-回归模型

在机器学习的视角下, 一种最简单的办法是训练出一套回归系统 $f : \mathcal{X} \to \mathcal{Y}, \mathcal{X} \subset \mathbb{R}^m, \mathcal{Y} \subset \mathbb{R}^n$, 对一个用户的年龄性别地址打分记录等一系列可能对电影喜好有影响的数据, 用特征提取的方法提取出来一个用户的特征 $x \in \mathcal{X}$, 将其输入网络, 就可以输出一个代表着他所喜爱的电影的相应特征 $y \in \mathcal{Y}$, 比如经过特征提取后的电影发布的时间, 电影的种类与电影的描述、影评电影的相关信息等等. 然后通过对所有电影库里的电影进行特征提取, 取最相近的一个, 我们就可以找到最合观众口味的电影.

在这个过程里, 我们假定这样的用户的特征跟电影的特征之间存在着某种高度非线性关系, 并且机器学习让我们可以近似出这样的高度非线性的关系, 比如用最简单的 Deep ReLU Network(当然真的可不可以我没试过). 在海量用户观看数据的支持下, 设置好用户跟电影特征提取过程中的参数与学习率、网络深度、每一层神经元数等一系列超参数, 在训练集 $E \in \mathcal{X} \times \mathcal{Y}$ 上对性能度量 $||f(x) - y||_2^2, (x, y) \in E$ 做随机梯度下降, 用已经很成熟的反向传播算法就可以把网络优化得很好. 这个时候我们就已经的到了一个从用户特征空间 $\mathcal{X}$ 到电影特征空间 $\mathcal{Y}$ 的一个映射, 通过上面的步骤我们也就获得了一个最简单的电影推荐系统. 因为没有做过实验, 我们不对这一小节里的内容的严谨性负责, 但是对于更先进的推荐系统可以参考协同过滤算法 [7].

如果我们采用与机器学习同样的思路去尝试用传统统计学解决这个问题, 这样的实际问题中的高度非线性性给传统统计学带来了巨大的困难. 并且对于这种实际问题尽管我们知道他们相关, 可我们根本不知道这样的相关函数具体属于什么样的函数空间或者函数类, 这使得非参方法的合理性无法保证. 甚至就算我们知道了他属于哪些函数空间, 函数空间无穷维的特性也使得我们失去了有限维情况下的很多概率工具. 在这个具体的例子里面, 问题主要还是出在我们无法找到任何一个具体的函数类去合理地为这种特征与特征间非线性的关系建模. 基本上所有的假设都能自圆其说, 也就使得这些假设都失去了实际意义.

传统统计学在十年前已经对这样一个问题给出了一个十分漂亮优雅的答案 [8] , 并且由此使用的与发展出的方法都具有启发性.

3.2 传统统计学视角-矩阵补全

矩阵补全, 简单来说就是对于一个巨大的矩阵 $A \in \mathbb{R}^{n_1 \times n_2}$, 凭借我们所知道的其中一部分元素能否复原整个矩阵? 当然在不加任何假设的情况下, 这样一个问题不管用什么方法都是不可能的. 因为我们不知道任何关于这个矩阵的其他信息. 不过在我们知道了关于一个矩阵的其他信息比如这个矩阵低秩 [8] 或者列空间可分为低维空间的并 [9], 这个问题便有了一定的可解性. 并且如果这个矩阵满足了一定限制, 我们可以给出一个精确的补全 [10].

在这个应用场景下低秩是一个相对宽松但是又有一定合理性的假设, 因为在实际情况中往往少数因素就可以决定用户的喜好 [8]. 这样一个矩阵 A 可以代表着用户对所有电影的潜在的喜好程度, 其元素 $X_{i,j}$ 代表第 i 个用户对 j 个商品的喜好程度, 假设 $rank(X) = r, r << n$. 但是我们并没有对 A 的直接观测, 因为并不是每个用户都能接触到所有的商品并写出反馈. 所以我们实际上的观测只局限于在这个确定矩阵的元素里随机抽取其中的 m 个. 记我们观测到的矩阵为矩阵

$Y \in \mathbb{R}^{m\times n}$,

$$Y_{ij} := \delta_{ij} X_{ij}, \quad \delta_{ij} \sim \text{Ber}(p) \text{ are independent}, \quad p = \frac{m}{n_1 n_2} \tag{1}$$

我们希望能够最好的复原这样的矩阵, 最好的意思是复原出的矩阵 $\hat{X}$ 能够使得 $||\hat{X} - X||_F$ 最小. (注意这里虽然我们假设了 X 低秩但是观测到的矩阵 Y 不一定低秩).

统计学理论可以证明这样的矩阵补全可以被矩阵 $p^{-1}Y$ 的最佳 r 阶逼近所达成 [11]. 其中最佳 r 阶逼近可以写作

$$p^{-1}Y_r = \frac{1}{p}\sum_{i=1}^{r} s_i u_i \otimes v_i, s_i \in \mathbb{R}, u_i \in \mathbb{R}^{n_1}, v_i \in \mathbb{R}^{n_2} \tag{2}$$

即对矩阵 Y 的谱分解在第 r 个奇异值处截断, 最佳指的是

$$||Y_r - Y|| = \min_{rank(Y')\le r} ||Y - Y'||, \tag{3}$$

其中对 $||\cdot||$ 任意的酉不变范数, 包含 F 范数与 2-范数, 这样最佳逼近的求解实际上等价于一个凸优化问题. 那么这样的 $\hat{X}$ 不但可以作为矩阵补全的一个方式, 并且统计学还给出了它的性能能够达成到一个什么样的程度?

统计学的理论可以证明, 只要我们知道的元素够多, 多到 $m \ge ctn\log n$, $n = \max\{n_1, n_2\}$ 的时候, 我们就很有可能 (概率大于 $1-(2n)^{1-t}$) 能让估计误差变得很小

$$\frac{1}{n}||\hat{X} - X||_F \le C\sqrt{\frac{tn\log n}{m}}||X||_\infty, \tag{4}$$

其中 $t > 1, c, C$ 是两个不依赖于 X 的参数 [11].

并且更进一步, 统计学的理论可以证明, 对于一些稍微更特殊的矩阵 (A 的左右特征向量组是两组正交向量组), 只要我们观测到的元素数量够多, 多到 $m \ge Cn^{1.2}\log(n)$, 我们就很有可能 (大于 $1 - cn^{-3}\log n$), 我们能够精准的补全 A, 其中 c, C 是两个不依赖于 X 的参数 [10]. 再进一步, 统计学的理论还可以证明, 这样的 Completion 是近乎最完美的, 因为他接近了信息论所允许的不可能再提升的下界 (差一个 $\log n$)[8].

这是为数不多的统计学理论几乎能与机器学习在实战中相媲美, 同时又给出很漂亮理论解释的场景. 但是就在这样的场景下我们也可以看出他们研究风格的巨大区别. 在更多的场合里, 机器学习的前沿发展在一些项目上已经使人类高手相形见绌, 而统计学理论解释机器学习的分支却还处在婴儿阶段.

机器学习突破了传统统计学在数学严谨性要求上的枷锁, 本着黑猫白猫抓住了老鼠就是好猫的精神不断地在工程上取得了一个又一个惊人的突破. 同时机器学习也对传统统计学理论的发展提供了新的方向与灵感, 比如研究网络结构的 Expressiveness[12] 以及学习过程的 Empirical Process[13] 的探索等等, 而这些催生出的统计理论在量子物理、数据科学等方向的再次应用也远远超出了机器学习或是人工智能的范畴.

4 机器学习基本步骤

机器学习虽然是一门具体理论比较复杂的学科, 但其基本步骤却可以高度简单的抽象出来, 下文就简单介绍一下典型的机器学习中的几个步骤. **图 11** 是机器学习几个步骤的图示.

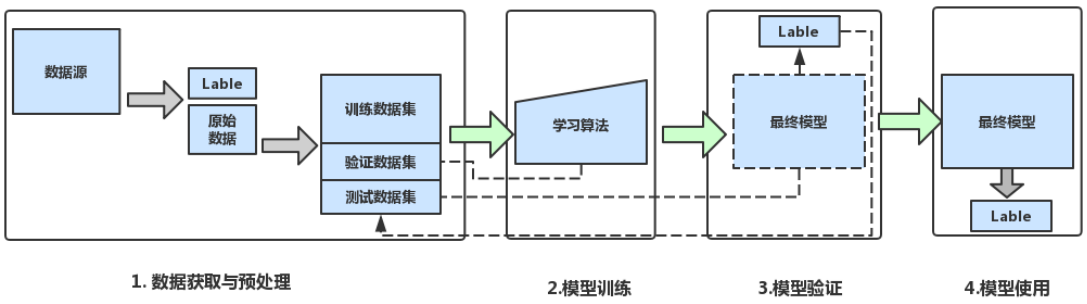

图 11: Procedure of Machine learning.

4.1 数据获取与预处理

所有机器学习算法尽管有各方面的不同, 但大都有一点是相同的——都是数据贪婪的, 即任何一个算法, 都可以通过增加训练数据来达到更好的结果. 因此我们的首要工作便是获得尽可能多的有效的数据. 在获得数据集之后, 进行数据清洗、数据预处理、以及特征选择、降维等工作, 而这些预处理, 特征选择, 降维等工作都是为了模型可以更好地来学习数据.

同时, 对于原始的数据, 我们还应该将其分割为不同的部分, 一般会分为训练集、测试集与验证集. 训练集主要是用来对于模型进行训练, 测试集用来判断模型的效果, 在模型进入真实环境前改进, 防止模型没有机会通过实际调试就直接应用到实际当中. 当使用了所有的原始数据去训练模型, 得到的结果很可能是该模型最大程度地拟合了原始数据, 对原始数据进行三个数据集的划分, 也是为了防止模型过拟合.

4.2 模型的训练

在开始训练模型前, 一般有三或四个需要人为选定的参数, 它们是决策函数集、目标函数 (表现度量) 和优化算法, 有时还有超参数.

决策函数集 即要选定使用什么样的决策函数集合来构建模型.

目标函数 (表现度量) 即需设定衡量决策函数好坏的度量方法.

优化算法 即使用什么样的算法根据表现度量来对决策函数中的参数进行优化.

超参数 超参数是指需要预先给定的一类模型参数, 机器往往不具备学习出这些参数的能力, 如决策树中树的数量或树的深度、学习率、神经网络的隐藏层数、k 均值聚类中的簇数等. 因此我们先要给定一个超参数的初始值. 在后期的在训练过程中使用验证集进行测试来决定它的最优数值.

在确定好上述参数后, 便需要使用收集好的数据对于模型进行训练.

4.3 模型的验证

在得到最终的模型后, 一般需要使用测试集对它的性能作出评价. 评价模型好坏的方法有很多, 如在分类模型中常用的就有错误率、精准率、召回率、F1 指标和 ROC 等.

4.4 模型的实际使用

使用训练好的模型对于新的数据进行预测.

5 机器学习的方法分类

机器学习这一学科发展到现在, 已经衍生出了许多种方法, 大多数可归为如下的类别.

5.1 按照模型的学习形式分

按照学习形式可将它们分为监督学习、无监督学习、半监督学习、强化学习等. 区别在于, 监督学习需要提供标注的样本集, 无监督学习不需要提供标注的样本集, 半监督学习需要提供少量标注的样本, 而强化学习则需要反馈机制.

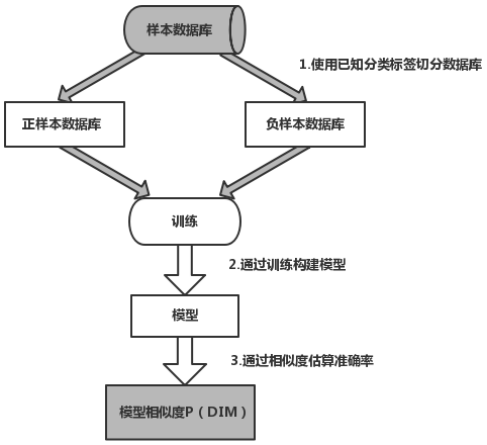

图 12: Workflow of supervised learning.

监督学习

监督学习是利用已标记的有限训练数据集, 通过某种学习方法建立一个模型, 实现对新数据的标记. 监督学习要求训练样本的分类标签已知, 分类标签的精确度越高, 样本越具有代表性, 学习模型的准确度就会越高. 监督学习在自然语言处理、信息检索、文本挖掘、手写体辨识、垃圾邮件侦测等领域获得了广泛应用. 监督学习的输入是标注分类标签的样本集, 通俗地说, 就是给定了一组标准答案. 监督学习从这样给定了分类标签的样本集中学习出一个函数, 当新的数据到来时, 就可以根据这个函数预测新数据的分类标签. 监督学习的过程如图 12 所示.

无监督学习

无监督学习的特点是寻找隐藏在未标记数据中的规律. 无监督学习不需要训练样本和人工标注数据, 便于压缩数据存储、减少计算量、提升算法速度, 还可以避免正负样本偏移引起的分类错误问题. 它主要用于经济预测、异常检测、数据挖掘、图像处理、模式识别等领域, 如社交网络分析、市场分割、天文数据分析等. 与监督学习相比, 无监督学习的样本集中没有预先标注好的分类标签, 即没有预先给定的标准答案. 它没有告诉计算机怎么做, 而是让计算机自己去学习

如何对数据进行分类, 然后对那些正确分类行为采取某种形式的激励. 常见的无监督学习算法有 Apriori 算法、KMeans 算法、随机森林 (random forest)、主成分分析 (principal component analysis) 等.

半监督学习

半监督学习介于监督学习与无监督学习之间, 其主要解决的问题是利用少量的标注样本和大量的未标注样本进行训练和分类, 从而达到减少标注代价、提高学习能力的目的. 它的应用场景有分类与回归, 算法主要有一些对常用监督学习算法的延伸. 这些算法首先试图对未标识数据进行建模, 在此基础上再对标识的数据进行预测. 如图论推理 (Graph inference) 算法或者拉普拉斯支持向量机 (Laplacian SVM) 等.

强化学习

强化学习是智能系统从环境到行为的学习, 以使强化信号函数值最大. 由于外部环境提供的信息很少, 强化学习系统必须靠自身的经历进行学习. 强化学习的目标是使得智能体选择的行为能够获得环境的最大奖赏. 在这种学习模式下, 输入数据作为对模型的反馈, 不像监督模型那样, 输入数据仅仅是作为一个检查模型对错的方式. 在强化学习下, 输入数据直接反馈到模型, 模型必须对此立刻做出调整. 常见的应用场景包括动态系统以及机器人控制等. 常见算法包括 Q-Learning 以及时间差学习 (Temporal difference learning).

5.2 按照处理问题的方法分

分类

分类是机器学习要处理的最广泛的问题之一. 分类是要依据历史数据刻画出不同类事物的典型特征, 进而预测未来数据的归类情况. 这个问题的主要目的是学会一个分类函数或分类模型, 该模型能把数据集中的事物映射到给定类别中的某一个类.

聚类

聚类是指将物理或抽象的集合分成由相似的对象组成的多个类的过程. 由聚类生成的簇是一组数据对象的集合, 这些对象与同一个簇中的对象彼此相似, 与其他簇中的对象相异. 在机器学习中, 聚类是一种无监督的学习, 在事先不知道数据分类的情况下, 根据数据之间的相似程度进行划分. 常用的算法有 KMeans、LDA 等.

回归

回归是使用已有的数值来预测未知数值的过程, 与分类模式不同, 回归更侧重于“量化”. 一般会使用分类方法预测离散值, 使用回归方法预测连续或有序值. 如对电影的评分等.

降维

近几年以来, 来自文本分析、图像检索、消费者关系管理等方面的特征数据急剧增加, 这种数据的海量性会使得大量机器学习算法在有效性和学习性能方面产生严重问题. 例如, 有成百上千维特征的高维数据, 会包含大量的无关冗余信息, 这些信息可能极大地降低学习算法的性能. 因此, 当面临高维数据时, 特征降维对于机器学习就显得十分重要. 特征降维是从初始高维特征集合中选出低维特征集合, 以便模型能够更好地学习. 通常会将降维作为机器学习的预处理步骤. 大量的实践证明, 特征降维能有效地消除无关和冗余特征, 提高挖掘任务的效率, 改善预测精确性等学习性能, 增强学习结果的易理解性.

6 机器学习的未来展望

贯穿整个深度学习的发展历史, 不管是逻辑启发的符号主意还是生物学启发的链接主意, 其已经将人类文明带到了新的高度. 但是它也遇到了很多无法破除的限制. 那么现在机器学习面临的困难有哪些? 未来的发展方向又在哪里? 这是一个远远超出了我们能回答范围的问题, 不过 Hinton 和 LeCun 在 2019 年图灵奖颁奖时的讲话也许能给我们些许的灵感.

首先 Hinton 认为 [14] 在计算机视觉的领域, 对于目标识别任务, 我们目前的神经网络认知的方式与我们人认知的方式并不一致. 人类可以利用并且依赖于的各个部件之间的几何关系, 在视角变换下的不变关系, 来认知事物, 而神经网络往往忽视了这一点. [15]

其次我们在神经网络里只有两种时间尺度, 一种缓慢的权重适应, 一种激活的神经元的快速变化. 这与生物学上神经元所具有的有适应各种时间尺度变化的能力有所差距. 所以一个方向时能够引入一些更多的时间尺度, 使得神经网络更迅速地对权重完成调整, 实现对短期记忆的快速重建.

虽然我们身处大数据时代, 但是有一些数据依然是无法获得的. 比如虚拟世界中表现优异的强化学习在现实应用中往往因为无法接受的数据成本 (比如自动驾驶) 而发展受阻. 毕竟我们不能为了收集每一个样本就撞一次车, 或是摔坏一个机器人, 尽管这样的成本在虚拟场景下比如游戏里面是唾手可得的. 但是我们的纯粹的强化学习却需要大量这样的数据去完成学习. 或者我们需要更多的数据, 或者我们需要改变我们的策略.

LeCun 认为机器学习的未来在自监督学习与半监督学习. 因为对于纯强化学习而言网络被输入的关于每个样本的有效信息往往是很少的, 像是蛋糕上的樱桃. 对于监督学习而言这个信息量就比较巨大, 像是蛋糕上面的冰层. 而对于自监督学习而言这些单个样本提供的信息量就更为巨大, 像是真正的蛋糕. 这个世界不是完全可预测的, 也不是完全随机的, 所以在有限的样本下, 我们需要尽可能多的信息去克服这样的不确定性.

最后, 对智能的理论依旧是空缺, 机器学习理论或者人工智能理论在未来会如何挑战我们对自己与对机器学习的认知? 而这些理论上的突破又会如何知道我们的发明与创造? 这些事情都值得我们期待.

A 机器学习的发展历程

年份	事件	时期
1949 年	Hebb 基于神经心理学提 Hebb 学习规则	基础奠定的 热烈时期
1950 年	Alan Mathison Turing 创造了图灵测试	
1956 年	John McCarthy, Cloude Shannon 等人 提出“人工智能 (Artificial Intelligence)”	
1957 年	Frank Rosenblatt 设计出了第一个 计算机神经网络——感知机 (The Perceptron)	
1956 年	Arthur Samuel 发明“机器学习”(Machine Learning) 这个词, 并以他开发的西洋跳棋程序打败了当时的西洋棋大师	
1967 年	k 最近邻分类算法 (KNN) 出现	
停滞不前的冷静时期：20 世纪 60 年代中叶到 70 年代末		
1980 年	卡内基梅隆大学 (CMU) 召开了 第一届机器学习国际研讨会	重拾希望的 复兴时期
1984 年	Breiman 等人提出分类与回归树 (CART)	
1986 年	国际性杂志 *Machine Learning* 创刊	
1986 年	Rumelhart 和 McClelland 为首的科学家 提出 BP 神经网络	
1989 年	Yann LeCun 在贝尔实验室就开始使用 卷积神经网络 (CNN) 识别手写数字	现代机器学习的 成型时期
1995 年	Vladimir Vapnik 提出支持向量机 (SVM)	
1995 年	Freund 和 schapire 提出了 AdaBoost 算法	
1997 年	循环神经网络 (RNN) 提出	
2000 年	流形学习方法 (Manifold Learning) 在 *Science* 上被提出, 现代应用包含 Isomap、LE、PCA	
2001 年	Breiman 提出随机森林 (Random Forest)	
2006 年	Hinton 等人在 *Science* 上提出了一种 训练深层神经网络的方法	大放光芒的蓬勃 发展时期
2010 年	Leslie Valiant 因 PAC 理论获得图灵奖	
2011 年	Judea Pearl 因概率图模型获得图灵奖	
2012 年	Hinton 小组发明的深度卷积神经网络 AlexNet 首先在图像分类问题上取代成功	
2014 年	Lan.J.Goodfellow 等人提出 GAN	
2017 年	AlphaGo 战胜柯洁	
2019 年	Geoffrey E Hintion、Yann LeCun、Yoshua Bengio 因在深度学习方面的贡献获得图灵奖	

参考文献

[1] GoogleTrend. 机器学习 - 探索 - google 趋势. https://trends.google.com/trends/explore?date=all&q=%2Fm%2F01hyh_.

[2] EncyclopædiaBritannica. Arthur samuel | american computer scientist | britannica. https://www.britannica.com/biography/Arthur-Samuel.

[3] IEEEComputerSociety. Arthur samuel | ieee computer society. https://www.computer.org/profiles/arthur-samuel.

[4] John McCarthy. A proposal for the dartmouth summer research project on artificial intelligence. https://www.computer.org/profiles/arthur-samuel.

[5] Silver D. Huang A. Maddison C. et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

[6] Alan M. Turing. *Computing Machinery and Intelligence*, pages 23–65. Springer Netherlands, Dordrecht, 2009.

[7] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to Weave an Information tapestry. *Communications of the ACM*, 1992.

[8] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 2010.

[9] Brian Eriksson, Laura Balzano, and Robert Nowak. High-rank matrix completion. In *Journal of Machine Learning Research*, 2012.

[10] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2009.

[11] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 2011.

[12] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 2017.

[13] Vladimir Vapnik, Esther Levin, and Yann Le Cun. Measuring the VC-Dimension of a Learning Machine. *Neural Computation*, 1994.

[14] Geoffrey Hinton and Yann LeCun. Turing award lecture "the deep learning revolution". https://www.youtube.com/watch?v=VsnQf7exv5I.

[15] Adam Kosiorek, Sara Sabour, Yee Whye Teh, and Geoffrey E Hinton. Stacked capsule autoencoders. In *Advances in Neural Information Processing Systems*, pages 15486–15496, 2019.

[16] Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

[17] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning, 2017.