

机器学习算法概述及其应用场景

张聪聪

华润智慧能源有限公司

April 17, 2020

目录

- ① 大数据与人工智能
- ② 机器学习算法概述
- ③ 机器学习算法应用场景

Section 1

大数据与人工智能

大物移云智



大数据



物联网



移动互联



云计算



人工智能

大数据与人工智能

① 大数据与人工智能

- 大数据
- 人工智能

② 机器学习算法概述

- 算法
- 机器学习

③ 机器学习算法应用场景

什么是大数据

定义 (广义)

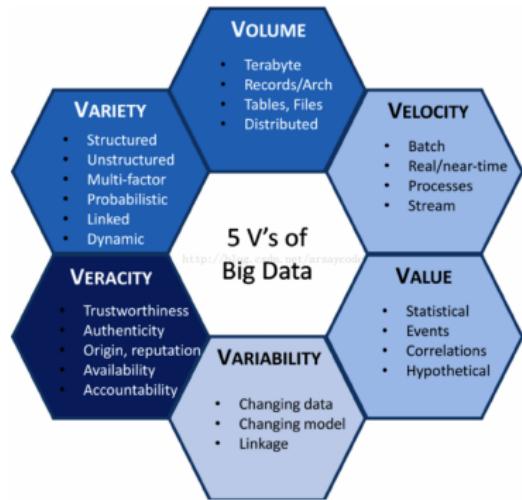
- 大数据是收集，组织，处理和收集大型数据集洞察所需的非传统策略和技术的总称。
- 大数据就是多，就是多。原来的设备存不下、算不动.
- 大数据，不是随机样本，而是所有数据；不是精确性，而是混杂性；不是因果关系，而是相关关系.
- 大数据，是指物理世界到数字世界的映射和提炼。通过发现其中的数据特征，从而做出提升效率的决策行为。

定义 (狭义)

大数据，是通过获取、存储、分析，从大容量数据中挖掘价值的一种全新的技术架构。

大数据特点

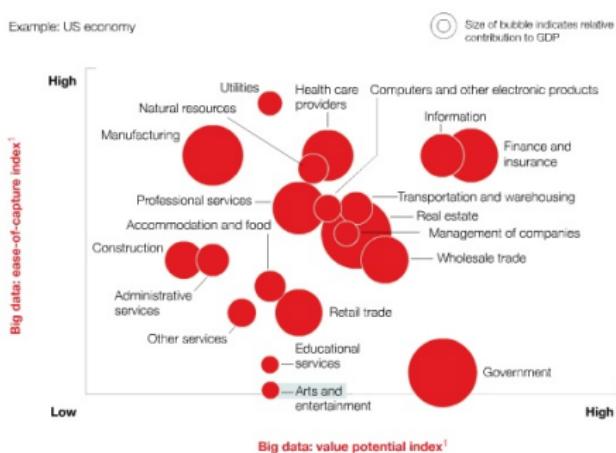
- **Volume**: 数据量大，包括采集、存储和计算的量都非常大。
- **Variety**: 种类和来源多样化。结构化、半结构化和非结构化数据；网络日志、音频、视频、图片、地理位置信息等等。
- **Value**: 数据价值密度相对较低，或者说是浪里淘沙却又弥足珍贵。
- **Velocity**: 数据增长速度快，处理速度也快，时效性要求高。
- **Velocity**: 数据的准确性和可信赖度，即数据的质量。



大数据的价值与应用场景

The ease of capturing big data's value, and the magnitude of its potential, vary across sectors.

Example: US economy



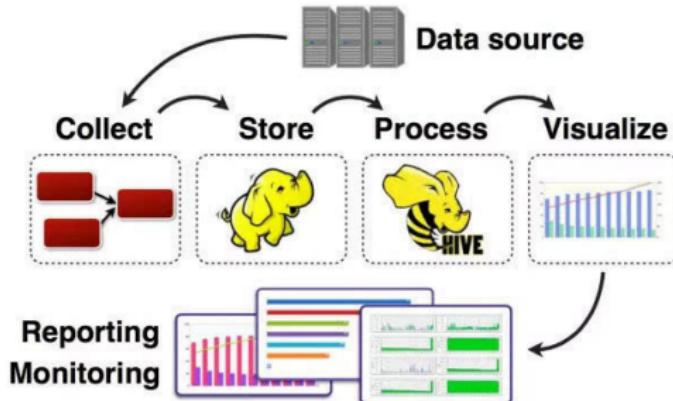
¹ For detailed explication of metrics, see appendix in McKinsey Global Institute full report *Big data: The next frontier for innovation, competition, and productivity*, available free of charge online at mckinsey.com/mgi.

Source: US Bureau of Labor Statistics; McKinsey Global Institute analysis

四类价值方法

- 客户群体细分，然后为每个群体量定制特别的服务。
- 模拟现实环境，发掘新的需求同时提高投资的回报率。
- 加强部门联系，提高整条管理链条和产业链条的效率。
- 降低服务成本，发现隐藏线索进行产品和服务的创新。

数据怎么来



大数据的六个环节

- 数据提取
- 数据存储
- 数据清理
- 数据挖掘
- 数据分析
- 数据可视化

大数据与人工智能

① 大数据与人工智能

- 大数据
- 人工智能

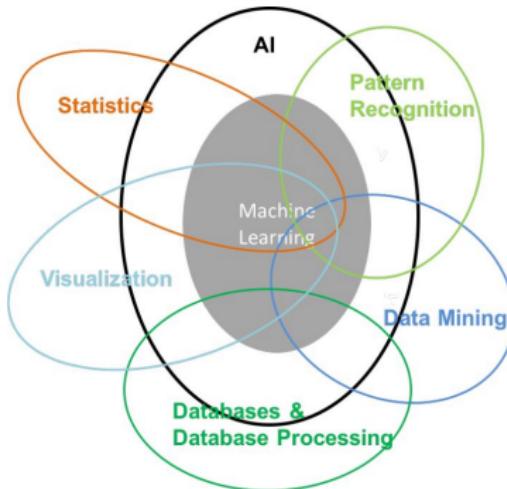
② 机器学习算法概述

- 算法
- 机器学习

③ 机器学习算法应用场景



人工智能是什么



定义

人工智能 (*Artificial Intelligence*) 它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学

人工智能的三个级别 |



弱人工智能

也称限制领域人工智能（Narrow AI）或应用型人工智能（Applied AI），指的是专注于且只能解决特定领域问题的人工智能。

人工智能的三个级别 Ⅱ



强人工智能

又称通用人工智能或完全人工智能， 指的是可以胜任人类所有工作的人工智能。

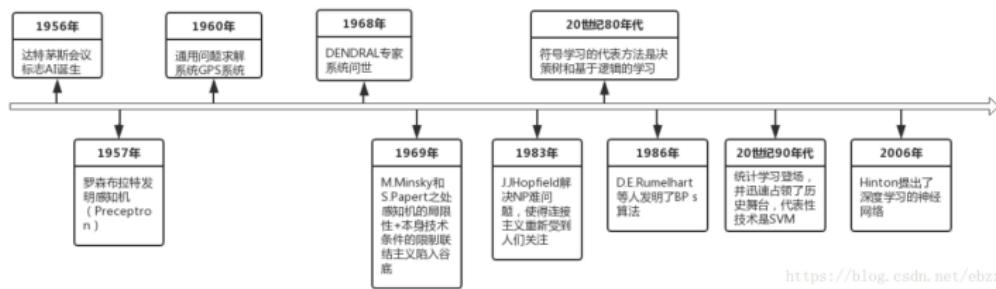
人工智能的三个级别 III



超人工智能

假设计算机程序通过不断发展，可以比世界上最聪明、最有天赋的人类还聪明，那么由此产生的人工智能系统就可以被称为超人工智能

人工智能发展史 |



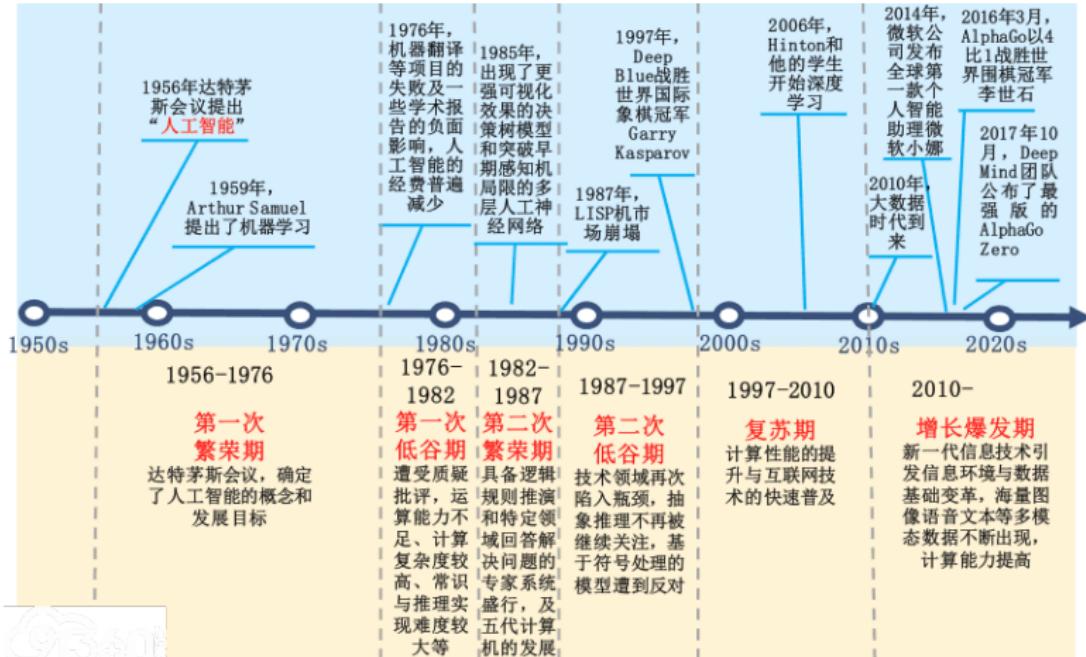
<https://blog.csdn.net/ebzxx>

人工智能发展史 II

大数据的六个环节

- 起步发展期：1956年—20世纪60年代初。
- 反思发展期：20世纪60年代—70年代初
- 应用发展期：20世纪70年代初—80年代中
- 低迷发展期：20世纪80年代中—90年代中
- 稳步发展期：20世纪90年代中—2010年
- 蓬勃发展期：2011年至今。

人工智能发展史 III



深度学习三巨头 |



(a)



(b)



(c)

图 1: a)Geoffrey Hinton; b)Yoshua Bengio; c)Yann LeCun

深度学习三巨头 II

Geoffrey Hinton

- ① **反向传播**: 证明了反向传播算法允许神经网络发现自己的数据内部表示，这使得使用神经网络成为可能网络解决以前被认为超出其范围的问题。
- ② **玻尔兹曼机**: 这是第一个能够学习不属于输入或输出的神经元内部表示的神经网络之一。
- ③ **卷积神经网络的改进**: 2012 年，Hinton 和他的学生 Alex Krizhevsky 以及 Ilya Sutskever 通过 Rectified Linear Neurons 和 Dropout Regularization 改进了卷积神经网络，并在著名的 ImageNet 评测中将对象识别的错误率减半，在计算机视觉领域掀起一场革命。

深度学习三巨头 III

Yoshua Bengio

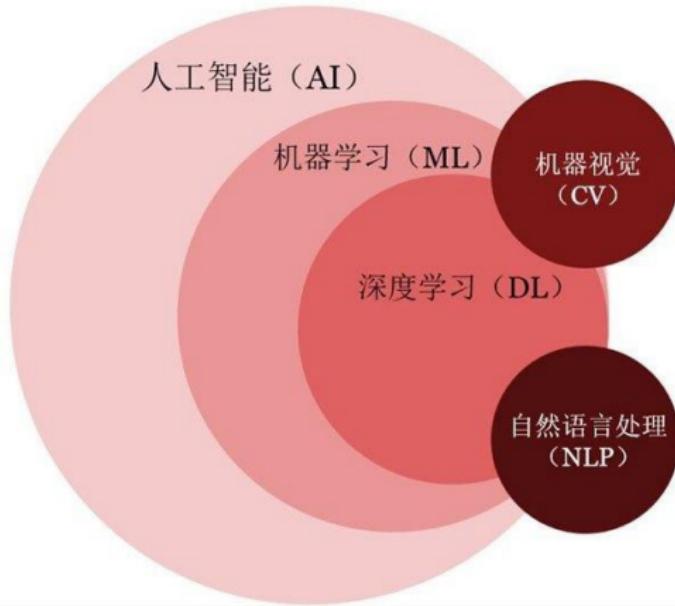
- ① 序列概率模型:现代深度学习语音识别系统正在扩展这些概念。
- ② 高维词嵌入和注意力模型:导致了机器翻译领域的突破，并构成了深度学习序列建模的关键组成部分；
- ③ 生成对抗网络:生成对抗网络(GANs), 在计算机视觉和计算机图形学领域引发了一场革命。

深度学习三巨头 IV

Yann LeCun

- ① 卷积神经网络: 提出卷积神经网络(CNN), 卷积神经网络已经成为计算机视觉、语音识别、语音合成、图像合成和自然语言处理领域的行业标准.
- ② 改进反向传播算法: 加速了反向传播算法;
- ③ 拓宽神经网络: 他将神经网络发展为一种计算模型, 用到一系列任务中, 他早期工作中的一些概念已成为 AI 发展的基石。

AI、机器学习、深度学习之间的关系



Section 2

机器学习算法概述

机器学习算法概述

① 大数据与人工智能

- 大数据
- 人工智能

② 机器学习算法概述

- 算法
- 机器学习

③ 机器学习算法应用场景

什么是算法



算法

算法是任何良定义的计算过程，该过程取某个值或值的集合作为输入并产生某个值或值的集合作为输出。

算法特性

性质 (有限性)

算法在执行有限步之后必须终止。

性质 (确定性)

算法的每一个步骤，都有精确的定义。要执行的每一个动作都是清晰的、无歧义的。

性质 (输入)

一个算法有0个或多个输入，它是由外部提供的，作为算法开始执行前的初始值。

性质 (输出)

一个算法有1个或多个输出，输出是输入的某种函数。

性质 (可行性)

算法中有待实现的运算，都是基本运算。

算法设计原则

- ① 正确性：算法能够正确地执行预定的功能和性能要求；
- ② 可读性：利于让人理解，逻辑清晰、简单、结构化；
- ③ 健壮性：算法要有较高的容错性；
- ④ 高效率与低存储量需求：要求算法应有高效率和低存储量的需求。

算法时间复杂度与空间复杂度 |

时间复杂度

在计算机科学中，算法的时间复杂度（Time complexity）是一个函数，它定性描述该算法的运行时间。时间复杂度常用大O 符号表述，时间复杂度可公式表示为 $T(n) = O(f(n))$

空间复杂度

空间复杂度(Space Complexity)是对一个算法在运行过程中临时占用存储空间大小的量度，算法的空间复杂度可公式化表示为 $S(n) = O(f(n))$

算法时间复杂度与空间复杂度 II

表 1：常用排序算法时间复杂度和空间复杂度

排序算法	平均时间 复杂度	最坏时间 复杂度	最好时间 复杂度	空间复杂度	稳定性
冒泡排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
直接选择排序	$O(n)$	$O(n^2)$	$O(n)$	$O(1)$	不稳定
直接插入排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
快速排序	$O(n \log n)$	$O(n^2)$	$O(n \log n)$	$O(n \log n)$	不稳定
堆排序	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$	不稳定

算法表述——伪代码 I

Algorithm 1 Framework of ensemble learning for our system.

Input: The set of positive samples for current batch, P_n ; The set of unlabelled samples for current batch, U_n ; Ensemble of classifiers on former batches, E_{n-1} ;

Output: Ensemble of classifiers on the current batch, E_n ;

- 1: Extracting the set of reliable negative and/or positive samples T_n from U_n with help of P_n ;
- 2: Training ensemble of classifiers E on $T_n \cup P_n$, with help of data in former batches;
- 3: $E_n = E_{n-1} \cup E$;
- 4: Classifying samples in $U_n - T_n$ by E_n ;
- 5: Deleting some weak classifiers in E_n so as to keep the capacity of E_n ;
- 6: **return** E_n ;

算法表述——伪代码 II

Algorithm 2 An example for format For & While Loop in Algorithm

```
1: for each  $i \in [1, 9]$  do
2:   initialize a tree  $T_i$  with only a leaf (the root);
3:    $T = T \cup T_i$ ;
4: end for
5: for all  $c$  such that  $c \in RecentMBatch(E_{n-1})$  do
6:    $T = T \cup PosSample(c)$ ;
7: end for;
8: for  $i = 1; i < n; i++$  do
9:   // Your source here;
10: end for
11: for  $i = 1$  to  $n$  do
12:   // Your source here;
13: end for
14: // Reusing recent base classifiers.
15: while ( $|E_n| \leq L_1$ ) and ( $D \neq \phi$ ) do
16:   Selecting the most recent classifier  $c_i$  from  $D$ ;
17:    $D = D - c_i$ ;
18:    $E_n = E_n + c_i$ ;
19: end while
```

算法表述——流程图

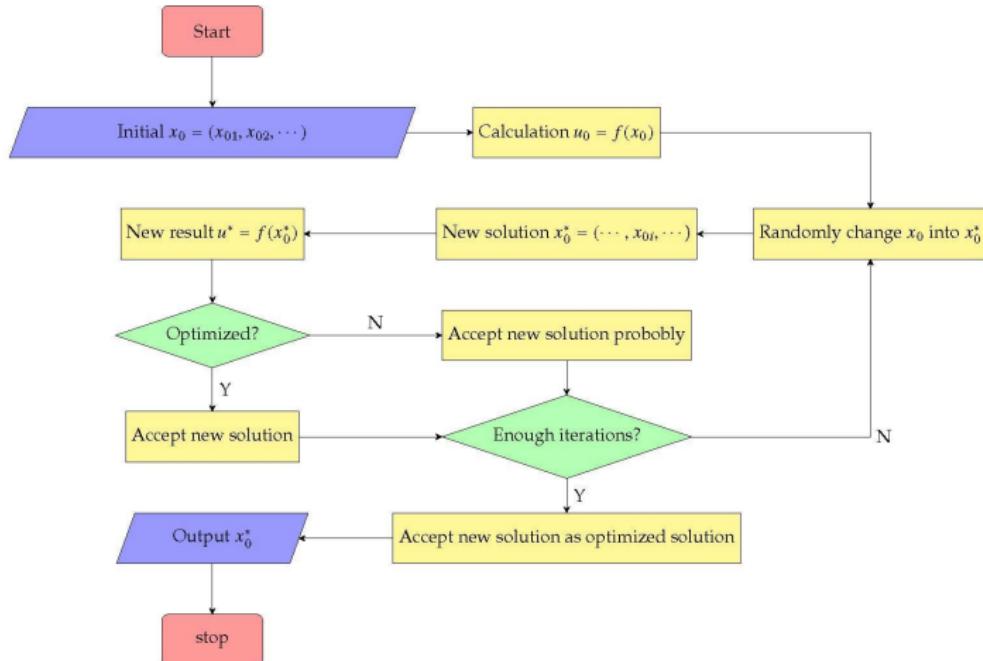


图 2: 算法流程图

算法表述：N-S图

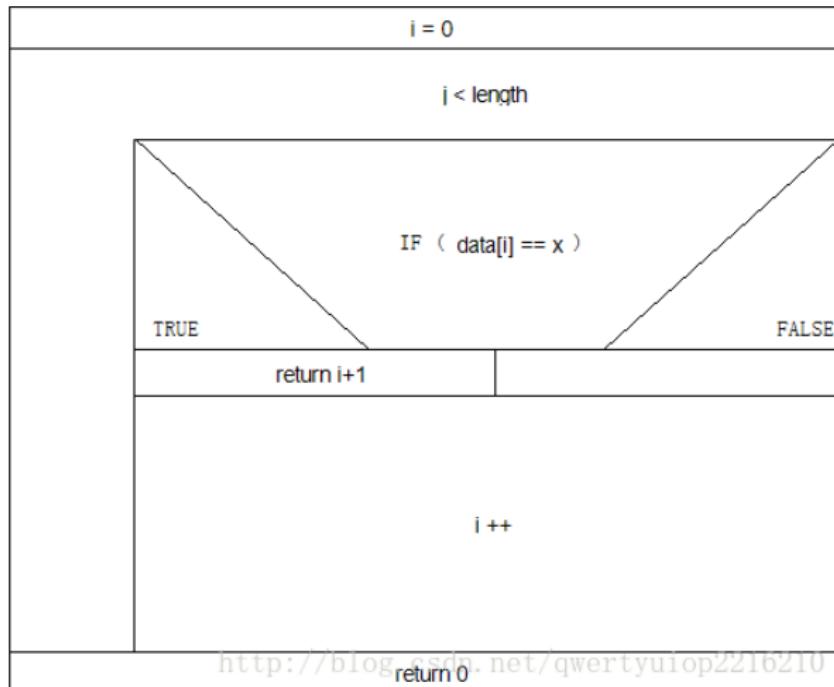


图 3: 算法N-S图

数据结构

定义 (数据结构)

数据结构是计算机 **存储、组织数据的方式**。数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。

程序的性能瓶颈往往都跟算法和数据结构有关系，简单的说，可以让你程序跑的更快

基本数据逻辑结构——集合结构

定义 (集合结构)

集合结构的里面的元素关系是孤立的

若 x 是集合 A 的元素，则记作 $x \in A$ 。集合中的元素有三个特征：

- ① 确定性：集合中的元素必须是确定的；
- ② 互异性：集合中的元素互不相同 $\forall a \in A, b \notin A, \exists a \neq b$ ；
- ③ 无序性：集合中的元素没有先后之分。

基本数据逻辑结构——线性结构

定义 (线性结构)

线性结构里面的元素关系：一对一

常用的线性结构有：

- ① 线性表：一个线性表是n个具有相同特性的数据元素的有限序列，线性表主要由顺序表示或链式表示；
- ② 栈：它是一种运算受限的线性表，限定仅在表尾进行插入和删除操作的线性表；
- ③ 队列：先进先出；
- ④ 双队列：可以在队列任意一端入队和出队；
- ⑤ 数组：用于储存多个相同类型数据的集合；
- ⑥ 串

基本数据逻辑结构——树形结构

定义 (树形结构)

树是包含 $n(n \geq 0)$ 个节点的有穷集合，内部元素是一对多的关系，其中：

- ① 每个元素成为节点 (*Node*)；
- ② 有一个被称为根节点的特定节点 (*Root*)；
- ③ 除根节点外其余元素被分为 $m(m \geq 0)$ 个互不相交的集合 $\{T_i, i \in \{0, 1, \dots, m\}\}$ ，每个子集 T_i 被成为原树的子树。

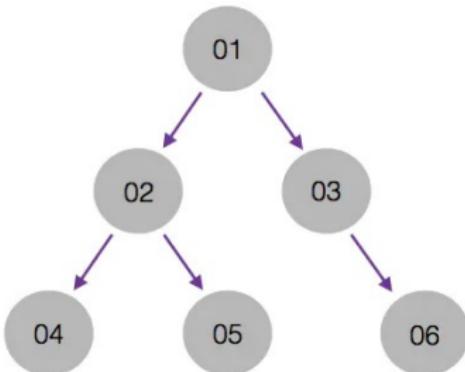


图 4：二叉树

基本数据逻辑结构：图结构 |

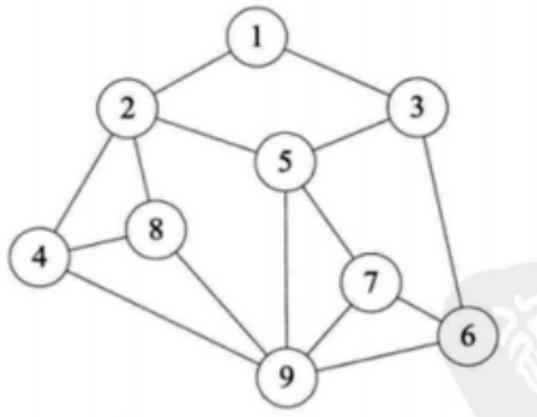
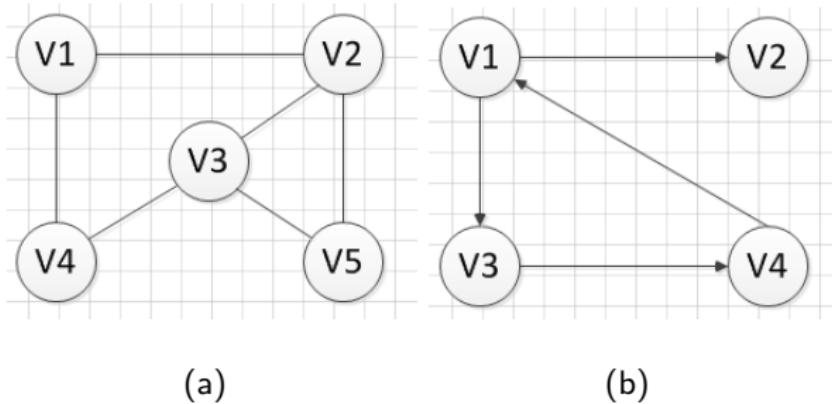


图 5: 图

定义 (图结构)

图是由顶点的有穷非空集合和顶点之间边的集合组成，表示为 $G(V, E)$ ，其中 G 表示一个图， V 是图 G 中顶点的集合， E 是图 G 中边的集合，内部顶点间是多对多的关系。

基本数据逻辑结构：图结构 II



(a)

(b)

图 6: a)无向图; b)有向图

定义 (有向与无向)

若顶点 v_i 到 v_j 之间的边没有方向，则这条边为无向边，表示为 (v_i, v_j) 或 (v_j, v_i) ，反之为有向边（或弧），表示为 $< v_i, v_j >$ ，其中 v_i 称为弧尾， v_j 称为弧头。若图中所有边都为无向边，则该图为无向图，反之若都为有向边，则为有向图。

基本数据逻辑结构：图结构 III

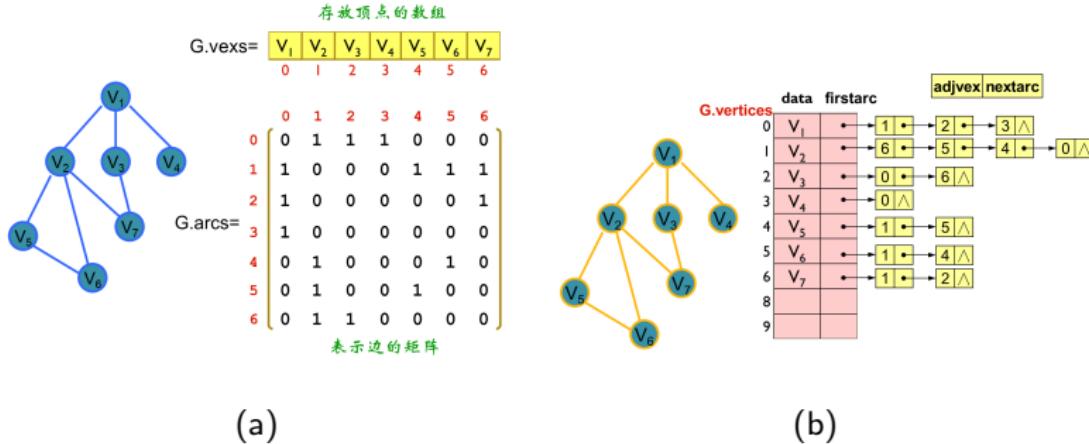


图 7: a) 邻接矩阵； b) 邻接链表

图的表示

对于图 $G = (V, E)$ 来说，可以有两种标准的表示方法，一个是邻接矩阵，另一个是邻接链表，这两种方法都可以表示有向图和无向图

机器学习算法概述

① 大数据与人工智能

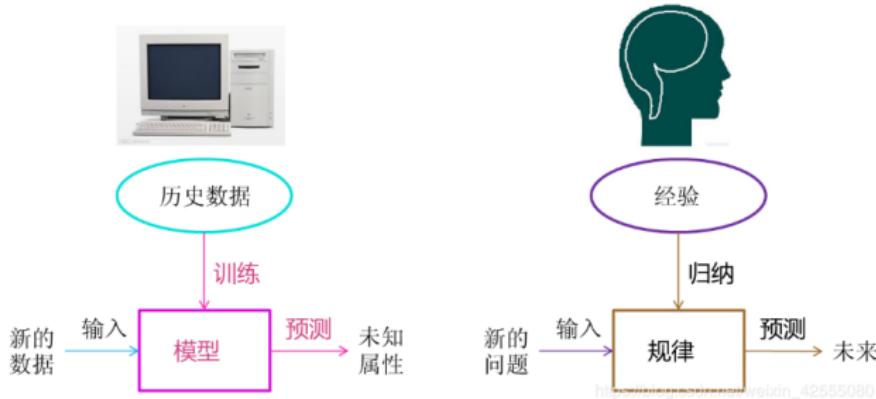
- 大数据
- 人工智能

② 机器学习算法概述

- 算法
- 机器学习

③ 机器学习算法应用场景

机器学习是什么



定义 (机器学习)

- 机器学习是一门人工智能的科学，该领域的研究对象是人工智能，特别是如何在**经验**学习中改善具体算法的性能。
- 机器学习是对能通过**经验**自动改进的计算机算法的研究。
- 机器学习是用**数据**或以往的经验，以此优化计算机程序的性能标准。

机器学习的目的

目的

- ① 对数据进行预测与分析，特别是对未来的数据进行预测与分析；
- ② 使计算机更加智能化，也即让计算机的某些性能得到提高；
- ③ 让人们获取新的知识，给人们带来新发现。

对数据的预测与分析是通过构建机器学习模型实现的。机器学习的目标就是考虑学习什么样的模型和如何学习模型，以使模型能对数据进行准确的预测和分析，同时考虑尽可能的提高学习效率。

机器学习三要素



机器学习是由模型（Model）、策略（Strategy）、算法（Algorithm）三部分构成，可简单的标识为

$$\text{方法} = \text{模型} + \text{策略} + \text{算法}$$

机器学习三要素：模型 |

定义 (模型假设空间)

假设空间 \mathcal{F} 可以定义为决策函数的集合

$$\mathcal{F} = \{f | Y = f(X)\}$$

其中， X 和 Y 是定义在输入空间 \mathcal{X} 和输出空间 \mathcal{Y} 上的变量。

机器学习三要素：模型 II

定义（概率模型假设空间）

假设空间 \mathcal{F} 可以定义为决策函数的集合

$$\mathcal{F} = \{P|P_{\theta}(Y|X), \theta \in \mathbb{R}^n\}$$

其中， X 和 Y 是定义在输入空间 \mathcal{X} 和输出空间 \mathcal{Y} 上的随机变量，参数向量 θ 为 n 维欧式空间 \mathbb{R}^n 中元素， \mathbb{R}^n 也称为参数空间。

机器学习三要素：模型 III

定义（非概率模型假设空间）

假设空间 \mathcal{F} 是由一个参数向量决定的函数族

$$\mathcal{F} = \{f | Y = f_{\theta}(X), \theta \in \mathbb{R}^n\}$$

其中，参数向量 θ 为 n 维欧式空间 \mathbb{R}^n 中元素， \mathbb{R}^n 也称为参数空间。

机器学习三要素：策略 |

监督学习问题是从假设空间 \mathcal{F} 中选择模型 f 作为决策函数，使得对输入 X ， $f(X)$ 尽可能的逼近 Y ，一般使用损失函数来衡量逼近程度。常用损失函数有：

① 0 – 1 loss function:

$$L(Y, f(X)) = \begin{cases} 1, & Y \neq f(X) \\ 0, & Y = f(X) \end{cases}$$

② quadratic loss function:

$$L(Y, f(X)) = (Y - f(X))^2$$

机器学习三要素：策略 II

③ absolute loss function:

$$L(Y, f(X)) = |Y - f(X)|$$

④ loglikelihood function:

$$L(Y, P(Y|X)) = -\log P(Y|X)$$

机器学习三要素：策略 III

定义（经验风险）

给定样本集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, 模型 $f(X)$ 关于样本集 T 的经验风险 (*empirical risk*) , 记作 R_{emp} :

$$R_{emp}(f) = \frac{1}{N} \sum L(y_i, f(x_i))$$

在假设空间 \mathcal{F} 、损失函数 L 、样本集 T 确定的情况下，经验风险 R_{emp} 也就可以确定，监督学习问题可表述为下述优化问题：

$$\arg \min_{f \in \mathcal{F}} \frac{1}{N} \sum L(y_i, f(x_i)).$$

机器学习三要素：策略 IV

定义（结构风险）

给定样本集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, 模型 $f(X)$ 关于样本集 T 的结构风险 (*structural risk*) , 记作 R_{srm} :

$$R_{srm}(f) = \frac{1}{N} \sum L(y_i, f(x_i)) + \lambda J(f)$$

在假设空间 \mathcal{F} 、损失函数 L 、样本集 T 确定的情况下，经验风险 R_{emp} 也就可以确定，那么增加结构风险的监督学习问题可表述为下述优化问题：

$$\arg \min_{f \in \mathcal{F}} \frac{1}{N} \sum L(y_i, f(x_i)) + \lambda J(f)$$

机器学习三要素：算法 |

算法是指学习模型的具体计算方法。在三要素的框架下，机器学习的算法成为求解优化问题的算法。一般求解优化问题的算法有以下几种：

- ① Gradient Descent: Batch Gradient Descent, Stochastic Gradient Descent, Conjugate Gradient, gradient projection method;
- ② Newton's method, Quasi-Newton Method;
- ③ 遗传算法
- ④ 模拟退火

机器学习三要素：算法 II

令 f 为样本集 T 的模型假设空间 \mathcal{F} 中的一个元素，对于 $\forall (x_i, y_i) \in T$ 有：

$$y_i \approx \hat{y}_i = f(x_i; \theta)$$

根据经验风险与结构风险最小原则，构建优化问题

$$\arg \min_{\theta} L(X, Y; \theta) \quad (1)$$

$$\text{s.t. } \theta \in \mathbb{R}^n \quad (2)$$

其中 X 为样本集 T 中样本的特征部分， Y 为样本的目标值部分，下文以 $L(\theta)$ 作为 $L(X, Y; \theta)$ 的缩写。

机器学习三要素：算法 III

Batch Gradient Descent

$$\theta^{(i+1)} = \theta^{(i)} + \alpha \nabla_{\theta} L(\theta^{(i)})$$

其中 θ 为模型中 f 中要学习的参数， α 为学习步长。

- **优点：**保证得到一个最优解；
- **缺点：**初始下降速度快，最优解领域内下降速度慢；计算量大。

机器学习三要素：算法 IV

Stochastic gradient descent

$$\theta^{(i+1)} = \theta^{(i)} + \alpha \nabla_{\theta} L(\theta^{(i)}; x_i, y_i)$$

其中 θ 为模型中 f 中要学习的参数， α 为学习步长。

- **优点：**更新频次快，优化速度更快；一定的随机性导致有几率跳出局部最优；
- **缺点：**随机性可能导致收敛复杂化，即使到达最优点仍然会进行过度优化，因此SGD得优化过程相比BGD充满动荡。

机器学习三要素：算法 V

Mini-batch gradient descent

$$\theta^{(i+1)} = \theta^{(i)} + \alpha \nabla_{\theta} L(\theta^{(i)}; x_{i:i+n}, y_{i:i+n})$$

其中 θ 为模型中 f 中要学习的参数， α 为学习步长。

- **优点：**参数更新时的动荡变小，收敛过程更稳定，降低收敛难度；可以利用现有的线性代数库高效的计算多个样本的梯度；
- **缺点：**学习率选择与更新策略，非凸问题如何全局寻优。

机器学习三要素：算法 VI

Newton's method

$$\theta^{(i+1)} = \theta^{(i)} - H^{-1} \nabla_{\theta} L(\theta^{(i)})$$

其中 θ 为模型中 f 中要学习的参数， H^{-1} 为 L 的 **Hessian matrix**。

$$\nabla_{\theta}^2 L = \begin{bmatrix} \frac{\partial^2 L}{\partial \theta_1^2} & \frac{\partial^2 L}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 L}{\partial \theta_1 \partial \theta_N} \\ \frac{\partial^2 L}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 L}{\partial \theta_2^2} & \cdots & \frac{\partial^2 L}{\partial \theta_2 \partial \theta_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L}{\partial \theta_N \partial \theta_1} & \frac{\partial^2 L}{\partial \theta_N \partial \theta_2} & \cdots & \frac{\partial^2 L}{\partial \theta_N^2} \end{bmatrix}$$

机器学习三要素：算法 VII

Quasi-Newton Method

牛顿法的搜索方向是 $d^{(i)} = -H_i^{-1}\nabla_{\theta}L(\theta^{(i)})$, L 的一阶导和二阶导存在以下关系：

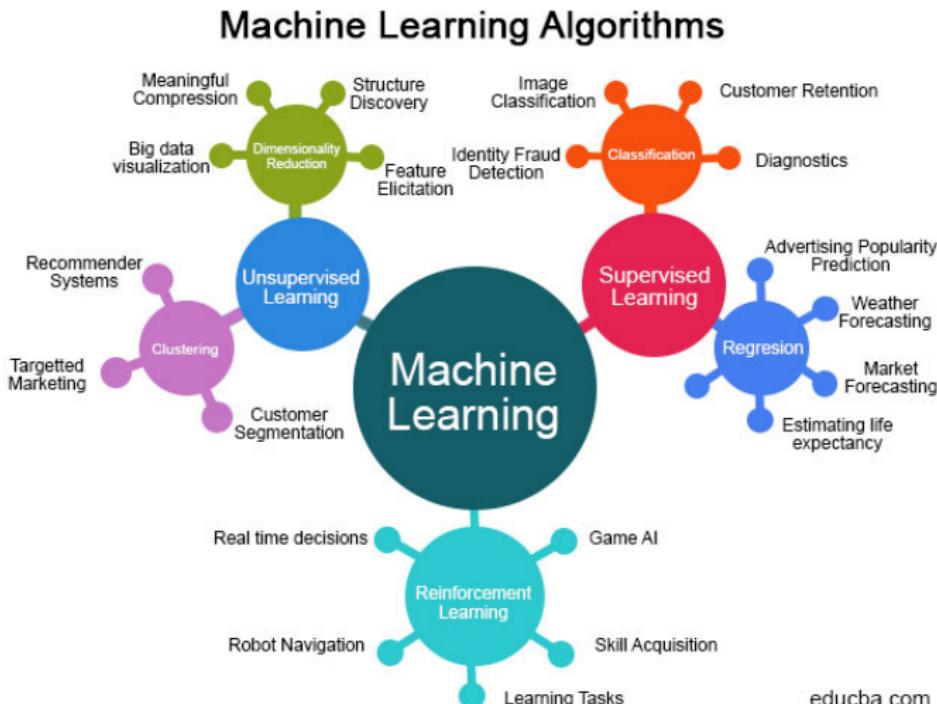
$$\theta^{(i+1)} - \theta^{(i)} = H^{-1}[\nabla_{\theta}L(\theta^{(i+1)}) - \nabla_{\theta}L(\theta^{(i)})]$$

H^{-1} 的计算量较大，改为构造近似矩阵 U 且满足一下条件

$$\theta^{(i+1)} - \theta^{(i)} = U_{i+1}[\nabla_{\theta}L(\theta^{(i+1)}) - \nabla_{\theta}L(\theta^{(i)})]$$

拟牛顿法和牛顿法的不同在于如何确定近似矩阵 U 及每次更新 θ 时进行步长寻优，由此引出 DFP、BFGS、L-BFGS 等算法。

机器学习算法归类 |



educba.com

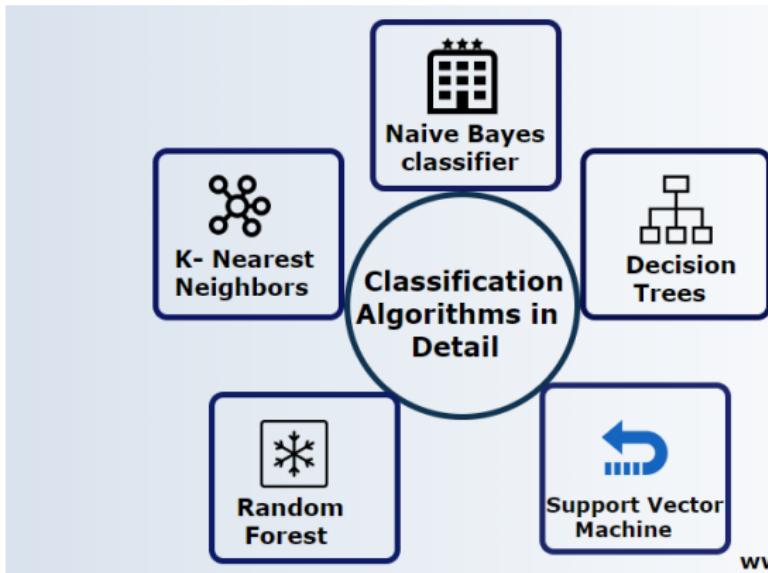


机器学习算法归类 II

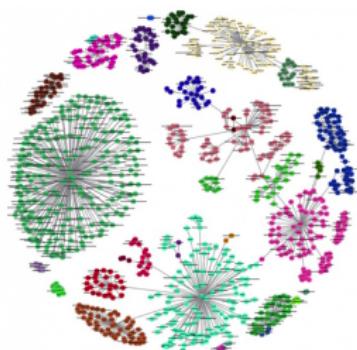
机器学习是基于数据构建的统计模型从而对数据进行分析与预测，可分为监督学习、无监督学习、半监督学习、强化学习四部分。

- ① 有监督学习：训练数据既有特征(feature)又有标签(label)；
- ② 无监督学习：训练样本的标记信息未知；
- ③ 半监督学习：训练集同时包含有标记样本数据和未标记样本数据，不需要人工干预；
- ④ 强化学习：强调如何基于环境而行动，以取得最大化的预期利益。

监督学习:分类 |



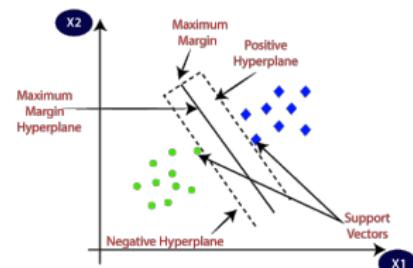
监督学习:分类 II



(a)



(b)



(c)

图 8: a)KNN; b)Decision Tree; c)SVM

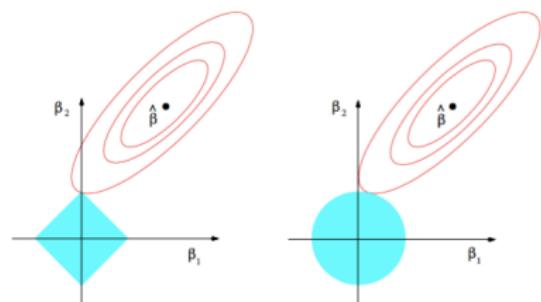
监督学习: 回归 |

① 多元线性回归:

$$\arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i); \theta)$$

② Lasso 回归:

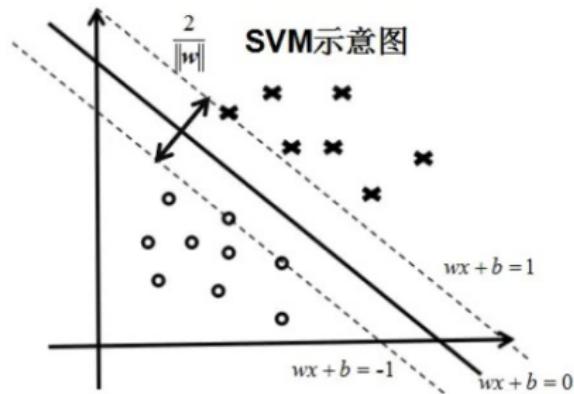
$$\arg \min_{\dot{\theta}} (y - X \dot{\theta})^\top (y - X \dot{\theta}) + \lambda \sum_j |\dot{\theta}_j|$$



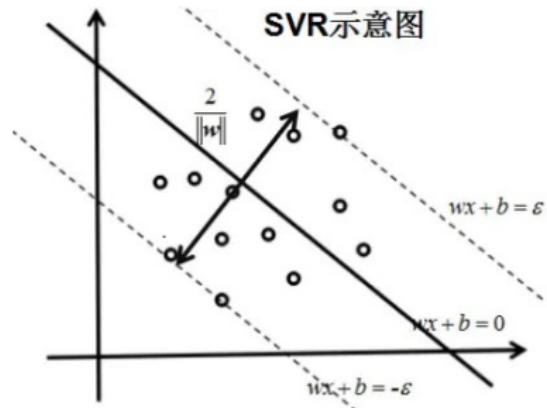
③ Ridge 回归:

$$\arg \min_{\dot{\theta}} (y - X \dot{\theta})^\top (y - X \dot{\theta}) + \frac{\lambda}{2} \sum_j \dot{\theta}_j^2$$

监督学习: 回归 II



$$\begin{cases} \min \frac{1}{2} \|w\|^2 \\ s.t. y_i(wx_i + b) \geq 1, \quad \forall i \end{cases}$$



$$\begin{cases} \min \frac{1}{2} \|w\|^2 \\ s.t. |y_i - (wx_i + b)| \leq \varepsilon, \quad \forall i \end{cases}$$

监督学习: 回归 III

CART回归树的损失函数为

$$\min \frac{1}{n} \sum_{m=1}^M \sum_{x_i \in R_m} (c_m - y_i)^2$$

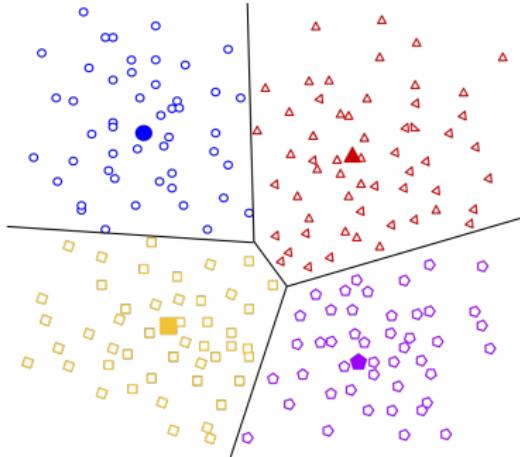
其中 R_m 为第 m 个叶节点对应的取值, c_m 表示第 m 个叶节点的预测值

$$c_m = \text{ave}(y_i | x_i \in \text{leaf}_m)$$

CART选择切分特征量 j 和切分点 s 的公式如下:

$$\min_{j,s} [\min_{c_1} \sum_{x_i \in R_1 \{j,s\}} (y_i - c_1) + \min_{c_2} \sum_{x_i \in R_2 \{j,s\}} (y_i - c_2)]$$

无监督学习：聚类 |



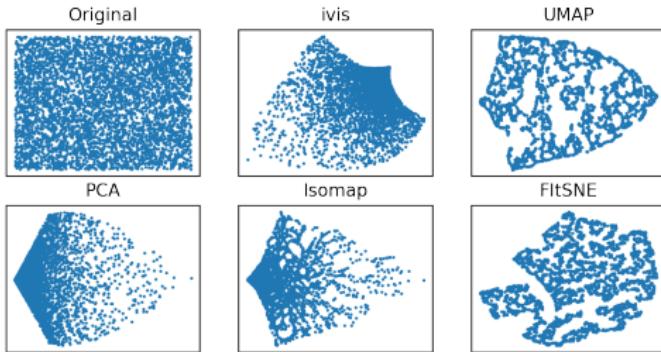
聚类就是按照某个特定标准(如距离准则)把一个数据集分割成不同的类或簇，类

内差异小，类间差异大。

无监督学习：聚类 II

- ① 基于划分：类内的点都足够近，类间的点都足够远，K-means；
- ② 基于层次：合并的层次聚类和分裂的层次聚类，BIRCH算法；
- ③ 基于密度：定义两个参数，一个是圈儿的最大半径，一个是一个圈儿里最少应容纳几个点，DBSCAN；
- ④ 基于模型的方法：为每簇假定了一个模型，寻找数据对给定模型的最佳拟合，GMM；
- ⑤ 基于模糊的聚类：在聚类汇总引入隶属度概念。

无监督学习：降维 |



为什么要降维？

- ① 维数少可以使算法有更快的计算速度，减少机器内存占用等
- ② 多个特征携带的“信息”有相同或类似的情况（冗余）
- ③ 用于数据可视化。

无监督学习：降维 II

常用的降维算法有：

- ① **PCA**: 最近重构性、最大可分性,

$$\arg \min_W - \text{tr}(W^T X X^T W)$$

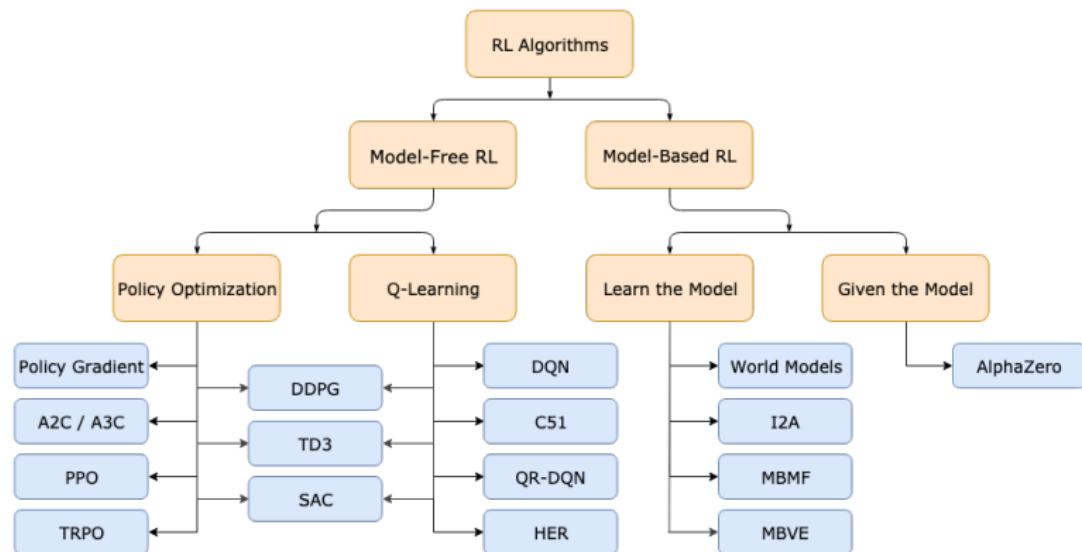
$$\text{s.t. } W^T W = I$$

- ② **SVD**: 一个 $m \times n$ 的实数矩阵 A 可分解为 $A = U \Sigma V^T$, $U \in R^{m \times m}$, $\Sigma \in R^{m \times n}$, $V \in R^{n \times n}$

- ③ **FFT, DCT, WT**: 主要用于时序数据的分析。

- ④ **t-SNE**

强化学习

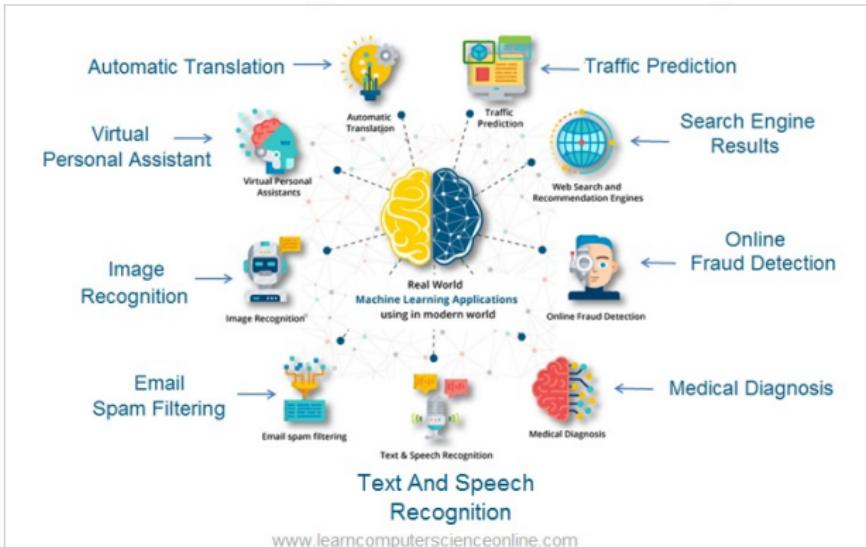


Section 3

机器学习算法应用场景

应用场景

Real World Applications Of Machine Learning



NLP技术应用



(a) 垃圾邮件过滤



(b) 输入法



(c) 机器翻译



(d) 理解未知语言

语音识别技术应用



(e) 智能客服



(f) 语音输入法



(g) 智能音箱



(h) 听书

机器视觉应用



(i) 人脸识别



(j) OCR



(k) 车牌识别



(l) 步态识别



(m) 红外温测

推荐系统应用



(n) 搜索引擎



(o) 电商



(p) 新闻推送



网易云音乐

(q) 音乐推送

谢谢聆听!