

Python及AI算法简介



Python的用途和人工智能算法入门

Huang Gang

1. Python 是一门什么 语言?

<https://docs.python.org/3/tutorial/index.html>

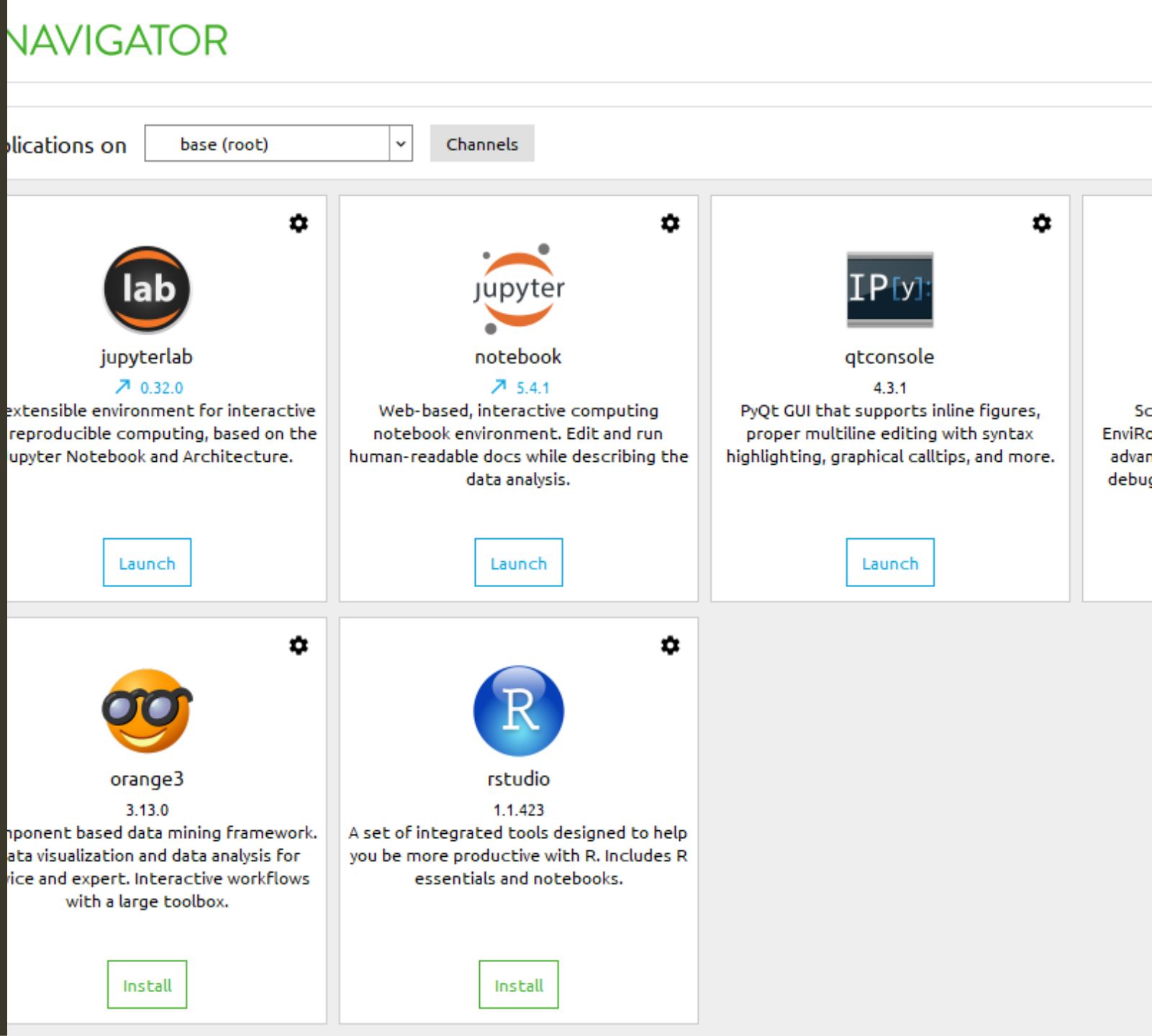
“Python is a programming language that lets you work quickly and integrate systems more effectively. ” ----<https://www.python.org>

开发环境: Anaconda, Enthought, ...

用途: 科学计算, 数据分析, 网页编程, GUI开发, 软件开发,

Python环境: Anaconda

- Anaconda 使Python的环境创建与管理变得极其简单.





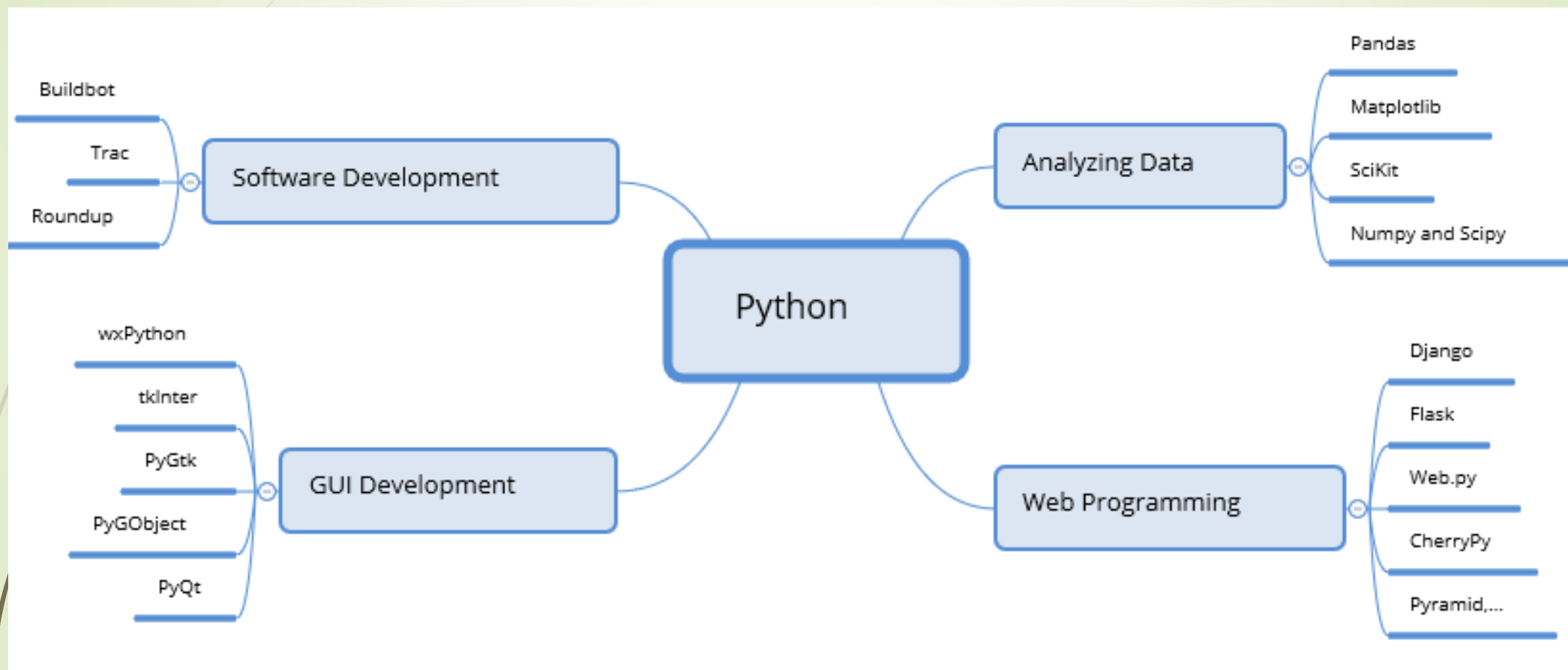
Smarter Together

We transform the business of science with artificial intelligence. Let's empower your best thinkers to move faster and discover more.

Python环境: Canopy

Enthought Canopy: 用于科学计算的强大Python环境

Python的用途





Python用于数据分析

- Numpy : <http://www.numpy.org/> (高维数组)
- Matplotlib: <https://matplotlib.org/> (数据分析作图)
- Pandas : <http://pandas.pydata.org/> (数据分析工具)
- Scipy
- Scikit

Python用途

- 数据分析: 介绍NumPy, Pandas等在数据分析中的应用

O'REILLY®

2nd Edition

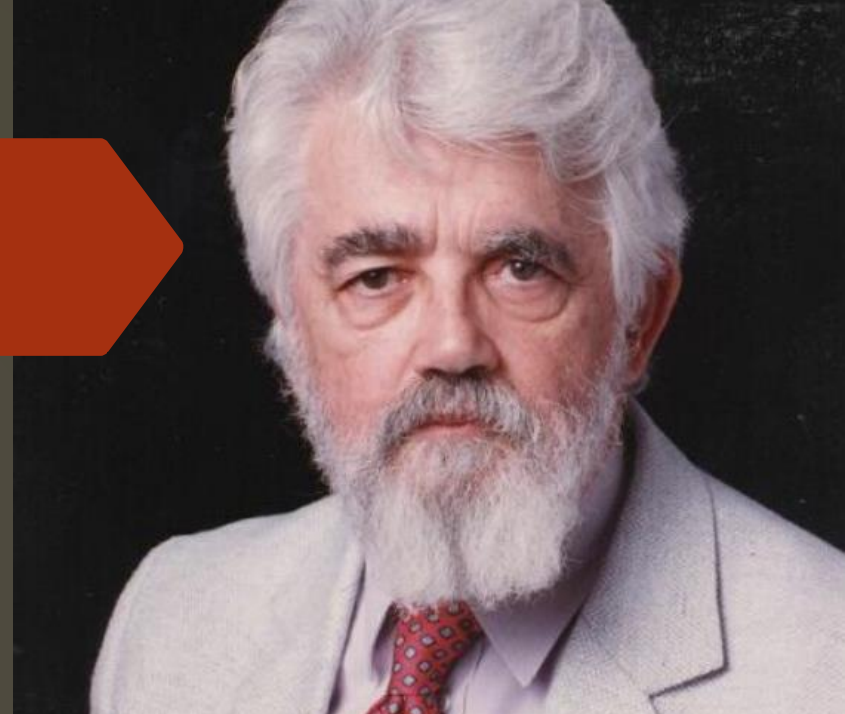
Python for Data Analysis

DATA WRANGLING WITH PANDAS,
NUMPY, AND IPYTHON



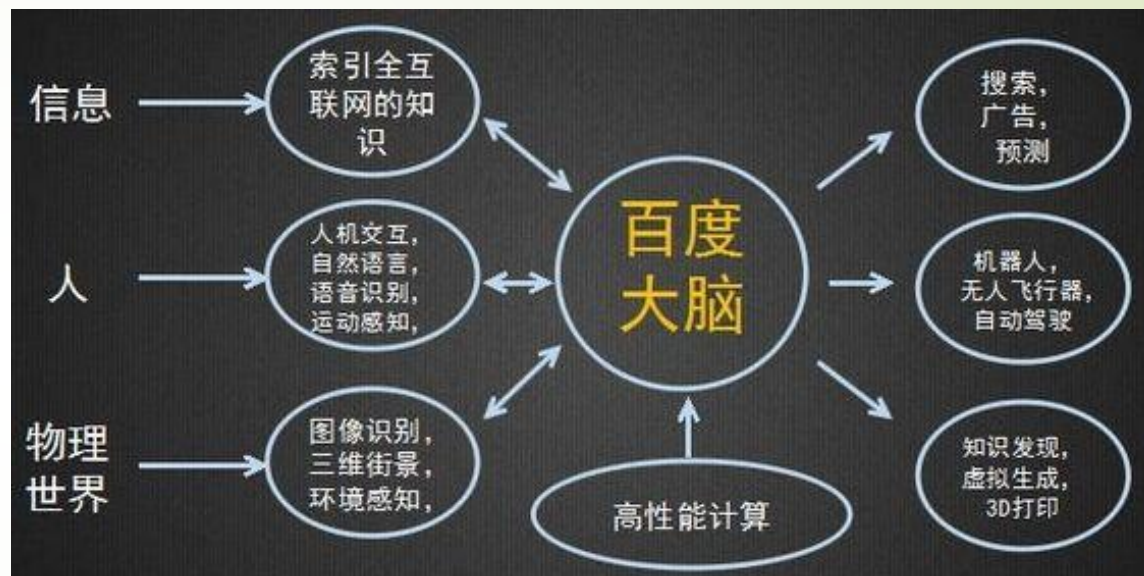
2.人工智能算法入门

- 1956年8月，在美国汉诺斯小镇的达特茅斯学院，约翰·麦卡锡（McCarthy）、马文·闵斯基(Minsky, 人工智能与认知学专家)、克劳德·香农（Shannon, 信息论的创始人）、艾伦·纽厄尔（Newell, 计算机科学家）、赫伯特·西蒙（Simon, 诺贝尔经济学奖得主）等科学家正聚在一起，讨论着一个完全新奇的主题：用机器来模仿人类学习以及其他方面的智能。此次会议开了两个月的时间，大家为会议讨论的内容起了一个名字：人工智能 (Artificial Intelligence). 1956, 人工智能元年.



2.0 人工智能的应用范围

- 人工智能(Artificial Intellegence, AI)可以做什么？
- 人工智能的**应用范围很广泛**,从人工智能的历史, 搜索算法的建立, 设计游戏, 解决游戏难题, 到限制条件问题都值得学习. 机器学习算法是人工智能里的核心.
- 人工智能可广泛应用在自然语言处理, 机器人学, 机器视觉, 机器翻译, 语音分析, 量化交易等领域. 用Python语言编程来解决人工智能问题是一个值得学习的技术.



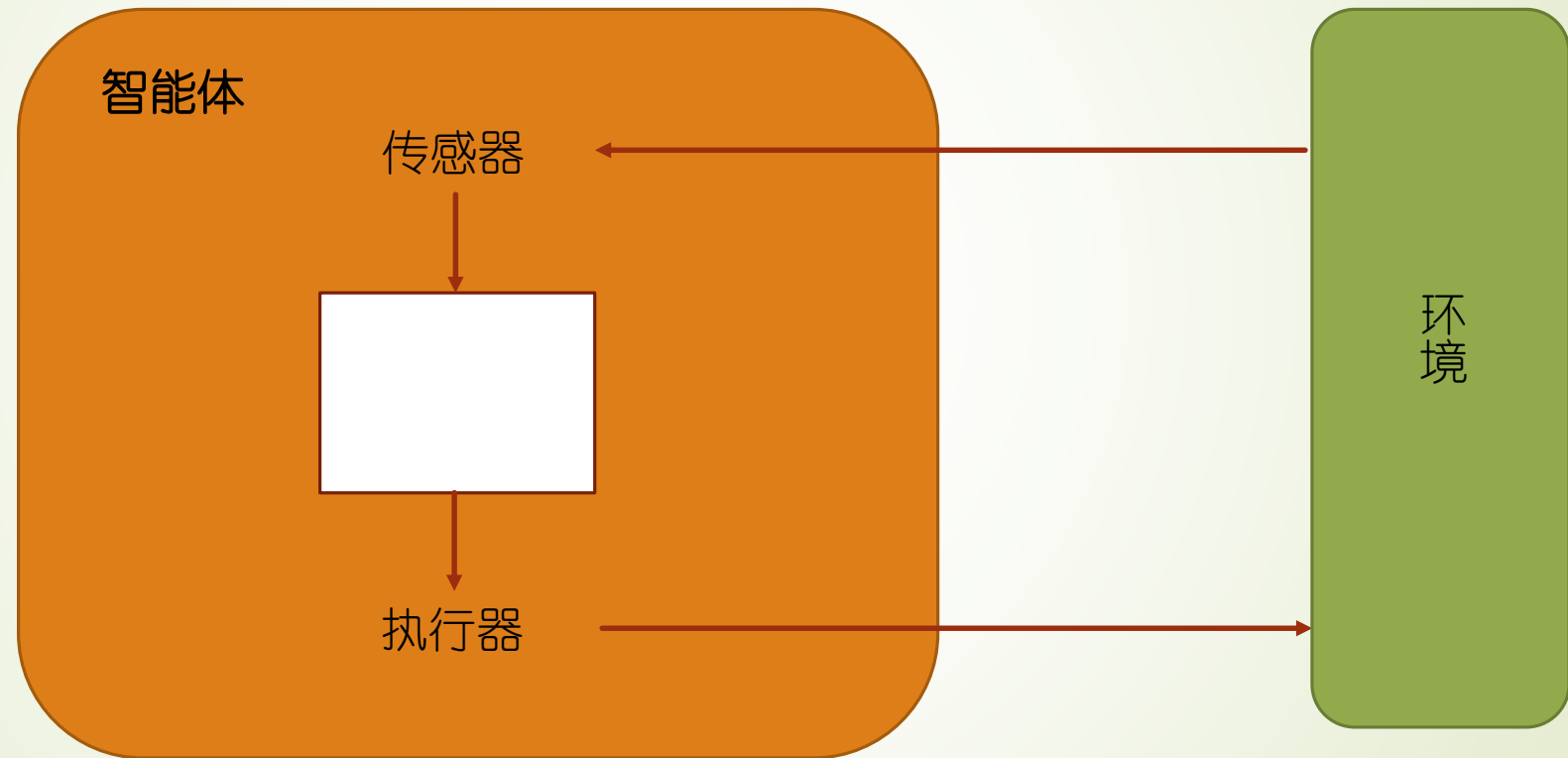
2.1 AI中的基本算法

- 最基本的算法就是搜索. 有许多中搜索方法可以使用比如**盲目搜索** (暴力搜索, 无脑搜索, uninformed search), 提示性搜索(又叫**启发性搜索**), 对抗搜索(游戏)等.
- 第二类话题就是马科夫决策过程和强化学习. 有一系列的应用, 如自然语言处理, 机器人, 机器视觉等.

理想智能体

- 先来看理性智能体. 我们研究人工智能的目的是设计智能体, 它们可以感知其环境并且作用到环境上, 从而实现其目标或者任务. 一个智能体可以视为一个函数 $F(x)$, 该函数从感知到的环境映射到一个作用在环境上的动作. 理性智能体, 就是做正确的事情的智能体.
- 何为正确的事呢? 就是智能体的表现达到最优, 即所谓性能度量最大化. 人工智能(AI)在给定的计算条件下, 使得性能度量达到最大化. 这就是AI的目的. 要使得性能度量最大, 可以从硬件和软件两方面优化改进, 我们这里只讨论软件方面.

智能体(Agent)



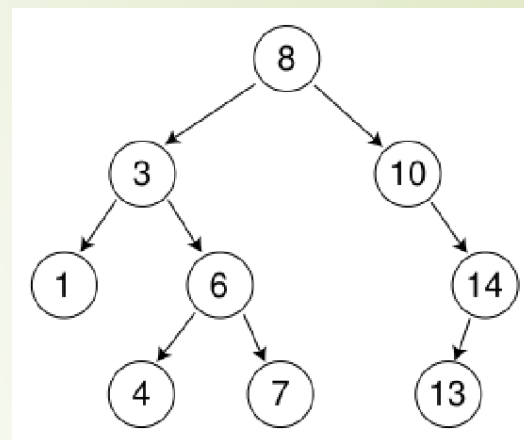
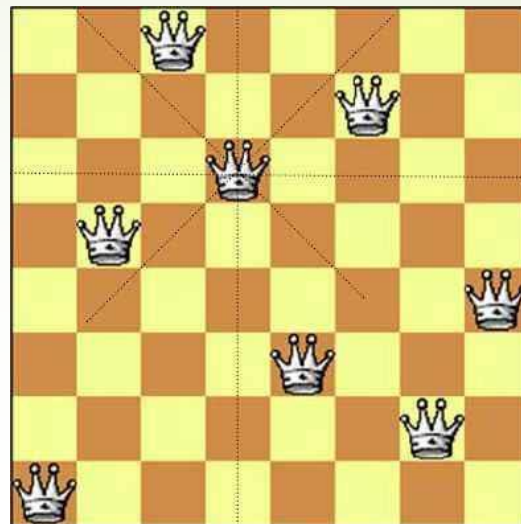
2.2 搜索算法

- 搜索智能体可以帮助我们从已知点出发找到目标点.
- 典型的例子是走迷宫, 从某个给定起点和终点, 找出一条路线使得我们能从起点到达终点. 智能体会思考为了达到目的该如何做.
- 智能体要做的就是定义出到达目标点的动作或动作序列 (路径). 一条路径会有不同的代价和深度 (此处指的是通过该路径找到的解在搜索树中的深度).
- 最常见搜索方法可分为有两大类: **盲目搜索**, 它包括广度优先搜索, 深度优先搜索, 均匀代价搜索等. **启发式搜索**运用了一些如何更快地到达目标的经验法则或启发式信息, 这类搜索法包括贪婪搜索法, A*搜索法, 等等.



搜索算法的例子

- 搜索算法的例子包括八皇后问题。八皇后问题是指，我们在64个格子的国际象棋棋盘上适当地放置8个皇后，使得它们横向，纵向，对角向都不"共线"。这就是要从约百万亿种可能的状态中，搜索出满足以上约束条件的状态来。
- 另一个典型的搜索算法的例子就是路线搜索。给定包含一些城市的地图，地图可以用图结构来表示：城市用结点表示，城市之间的可能的路线用线表示。



搜索问题的定义 (以八皇后问题为例)

- **目标**: 在 8×8 格的国际象棋上摆放八个皇后, 使其不能互相攻击, 即任意两个皇后都不能处于同一行、同一列或同一斜线上, 求出所有的摆法.
- **状态**: 棋盘上从0到8个皇后的位置的任意组合.
- **初始状态**: 空棋盘.
- **动作**: 一个棋手增加一个皇后到棋盘上的空格子.
- **转变模型**: 更新后的棋盘.
- **目标测试**: 是否棋盘上的8个皇后满足目标, 互不攻击.

搜索的例子：路线问题



- 两座城市之间的距离标记在它们的连线上。假设你想要从北京到马尔代夫。要达到此目的，你有不同的路线可走。搜索智能体的目的就是为了探索这些可能性，并找出最好的路线。不同的路线需要不同的花费，花费可以用这些连线的长度(比如，单位为千米的数) 这里，搜索智能体的角色就是找到从北京到马尔代夫的路线，同时找出最好的路线。



搜索的例子: 路线问题

例如: 想从北京到马尔代夫,如何确定路线.

- 状态: 处于地图上任意一个城市.
- 初始状态: 北京
- 动作: 北京→ 香港; 北京→ 曼谷; 北京→ 新加坡
- 目标测试: 到马尔代夫了吗?
- 路径成本: 路径长度(km) 或 路费(\$)

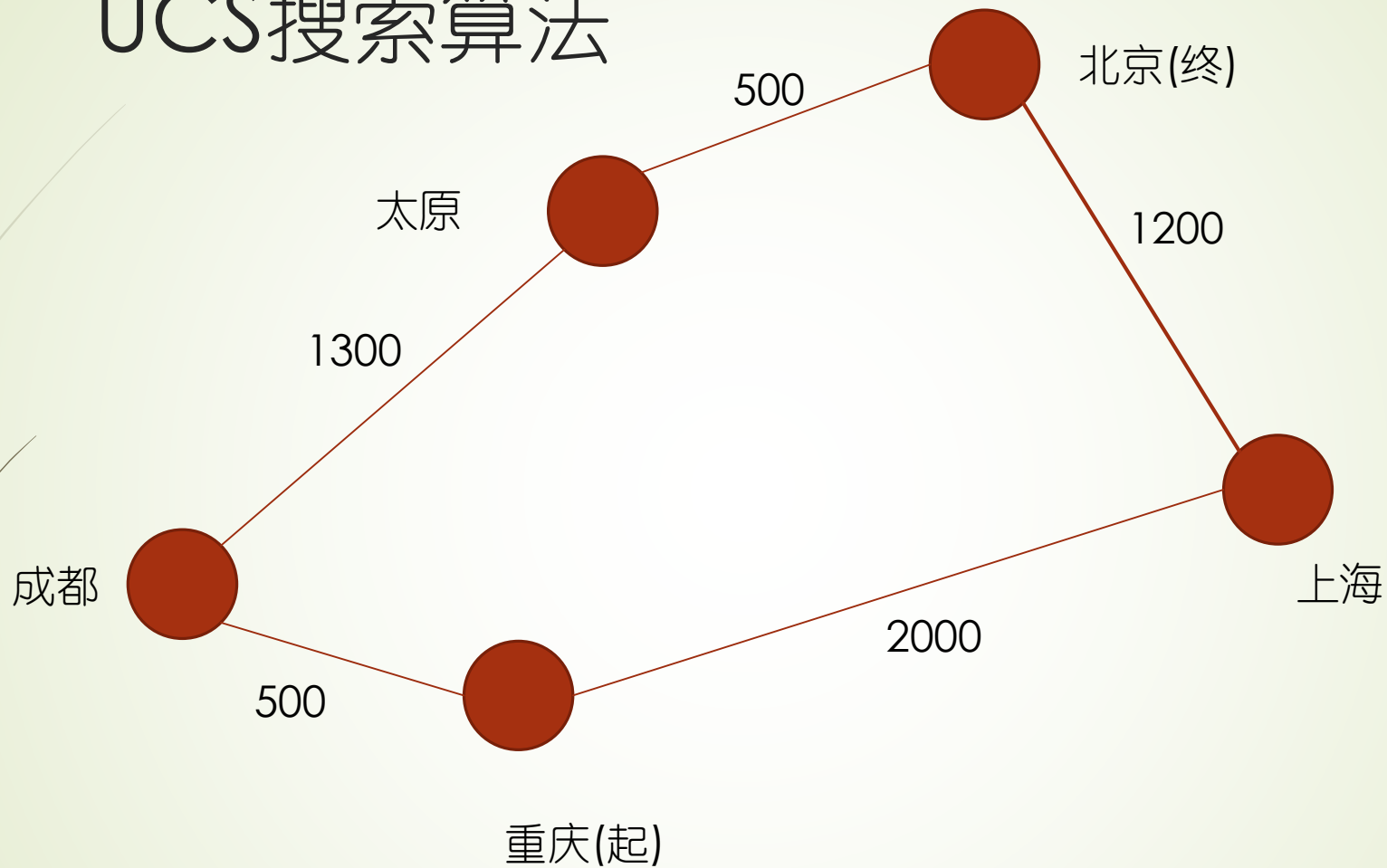
搜索策略

- 搜索空间: 抽象的空间, 往往用**搜索树**表示, 此搜索树是动作序列的模型.
- 搜索策略: 定义为搜索树上的结点**展开**的顺序.
- 评估一个策略的好坏, 可以从4个维度:
 - 1. 完备性: 如果有一个解, 该策略是否总能找到该解?
 - 2. 时间复杂度: 展开的结点的数目
 - 3. 空间复杂度: 内存中结点的最大数目
 - 4. 最优解: 该算法是否总能找到最优解?

搜索的分类

- 盲目搜索: 包括广度优先搜索(BFS), 深度优先搜索(DFS),
- 受限深度搜索(depth-limited search, DLS), 均匀成本搜索(Uniform-cost search, UCS), 迭代加深搜索(Iterative deepening search)
- 启发式搜索: 贪婪搜索, A*搜索.

UCS搜索算法

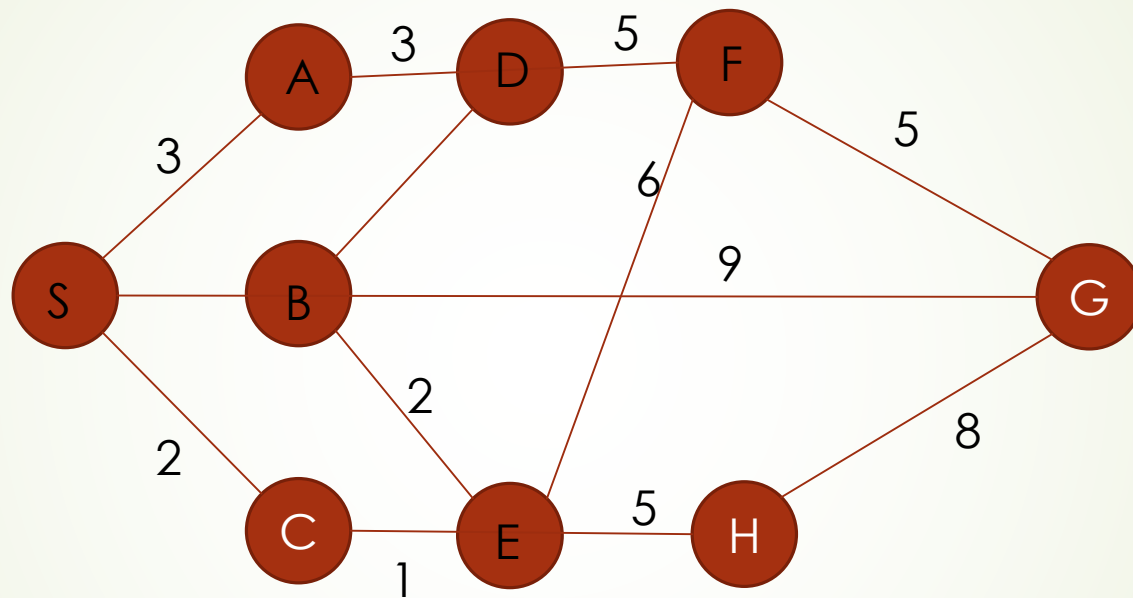


第一条路径 (BFS) : 重庆—上海—北京 (cost = 3200 km)

第二条路径: 重庆—成都—太原—北京 (cost = 500+1300+500=2300 km)

UCS将选择第二条路径,因为它比第一条路径更"便宜"! (UCS: 将成本函数最低者先选出来.)

例: 下图中的结点的访问顺序分别是什么?



BFS: S-A-B-C-D-G Queue: SABCDEGFH

DFS: S-A-D-F-E-H-G Stack: SCBADFGEH

UCS: S-C-E-B-G Cost: $2+1+2+9 = 14$

- D6 - H8 - F9 - G14

Priority Queue: S0 - C2 - A3 - E3 - B5

贪婪搜索和A*搜索

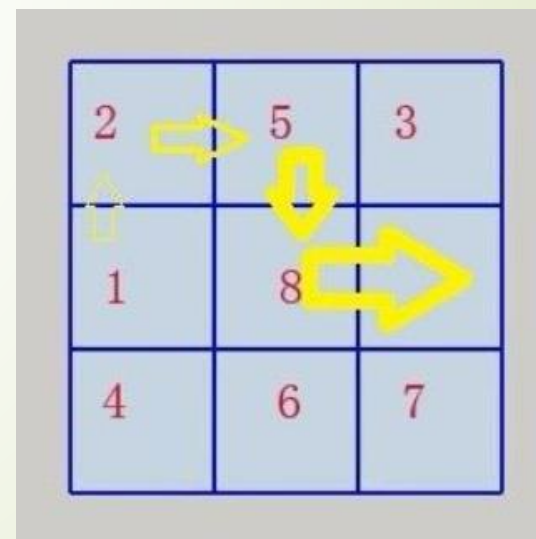
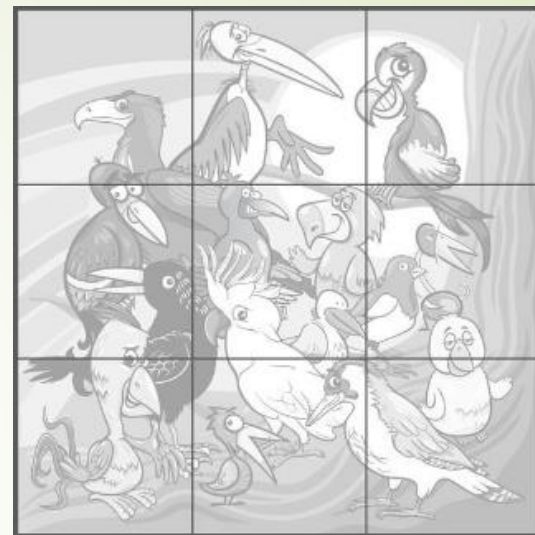
- 贪婪搜索: 评估函数 $h(n)$ 估算从结点 n 到最近的目标的成本(cost),并之最小
- A*搜索: 最小化总评估函数 $f(n) = g(n) + h(n)$. 其中, $h(n)$ 表示由 n 到目标所用的成本;
 $g(n)$ 表示到达结点 n 所用的成本.
- 总结: A*搜索总能找到最优解, 而贪婪搜索算法有时找到最优解.

对抗搜索

- AI的另一个重要话题是**对抗搜索**。实际上，已经有一些游戏可以用AI解决。我们看到国际象棋，象棋以及智力问答竞赛等等。基本思想就是设计智能体来玩游戏，并且与对手对抗。这里面的重要概念包括mini max算法, alpha beta pruning 和随机游戏。这里略过。在下棋这一领域，计算机(或者Chinook)能打败任何人类。
- 对抗搜索的另一个名字就是游戏(games)。对抗(adversarial)意味着存在一个我们无法控制的对手。

对比:游戏与搜索

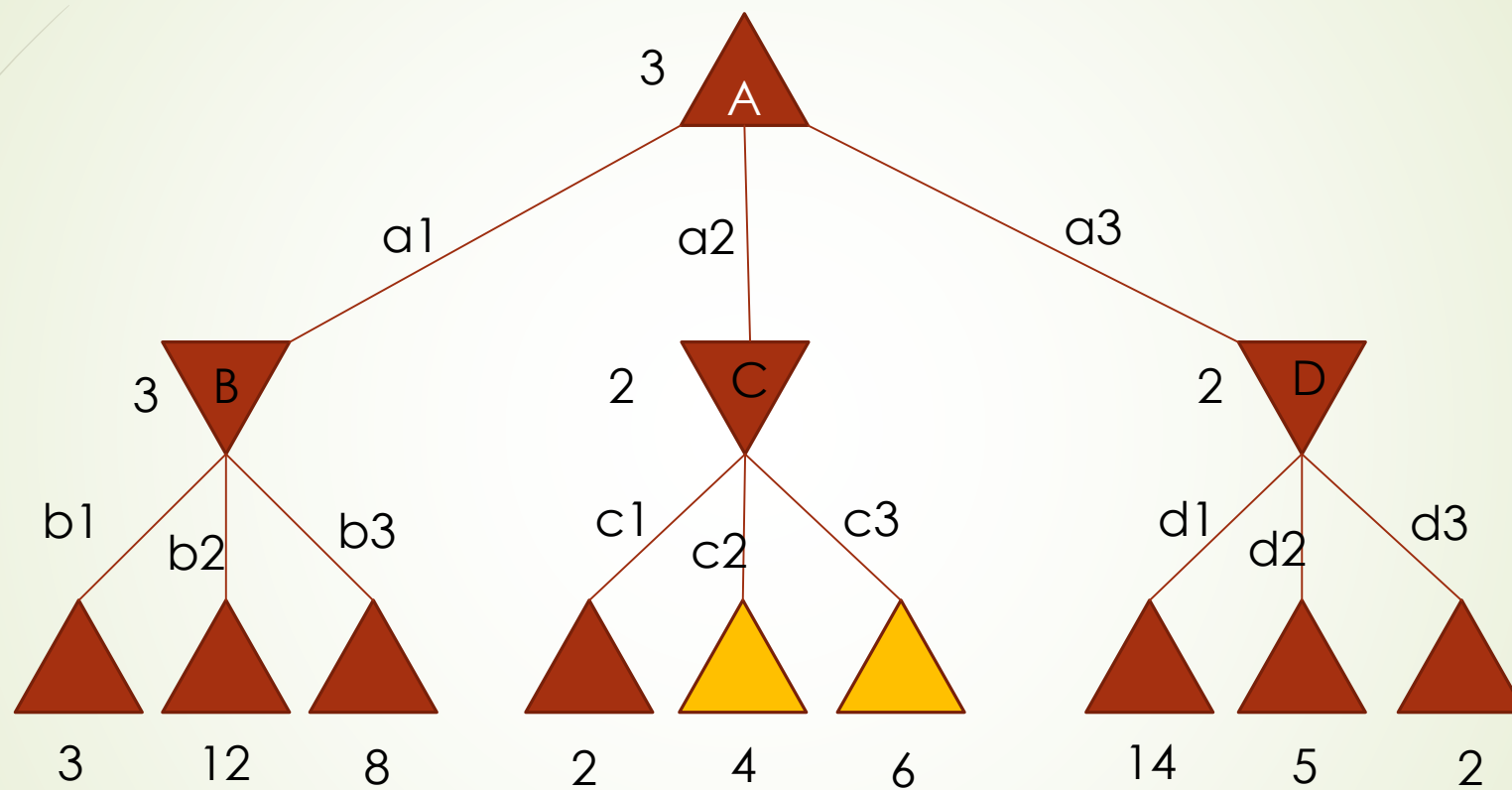
- 游戏也是一个**搜索问题**，但是最优解不是一个引向目标的动作序列。
- 游戏的最优解是一个帮助我们赢得比赛的策略. 也就是说，如果对手做A操作，那么我们就做A'操作，对手做B操作，我们就相应地施行B'操作,如果对手做C操作，我们就做C'，等等。
- 这种策略可以写成规则，好比下棋时候的**口诀**，但是它非常地枯燥。



A-B剪枝

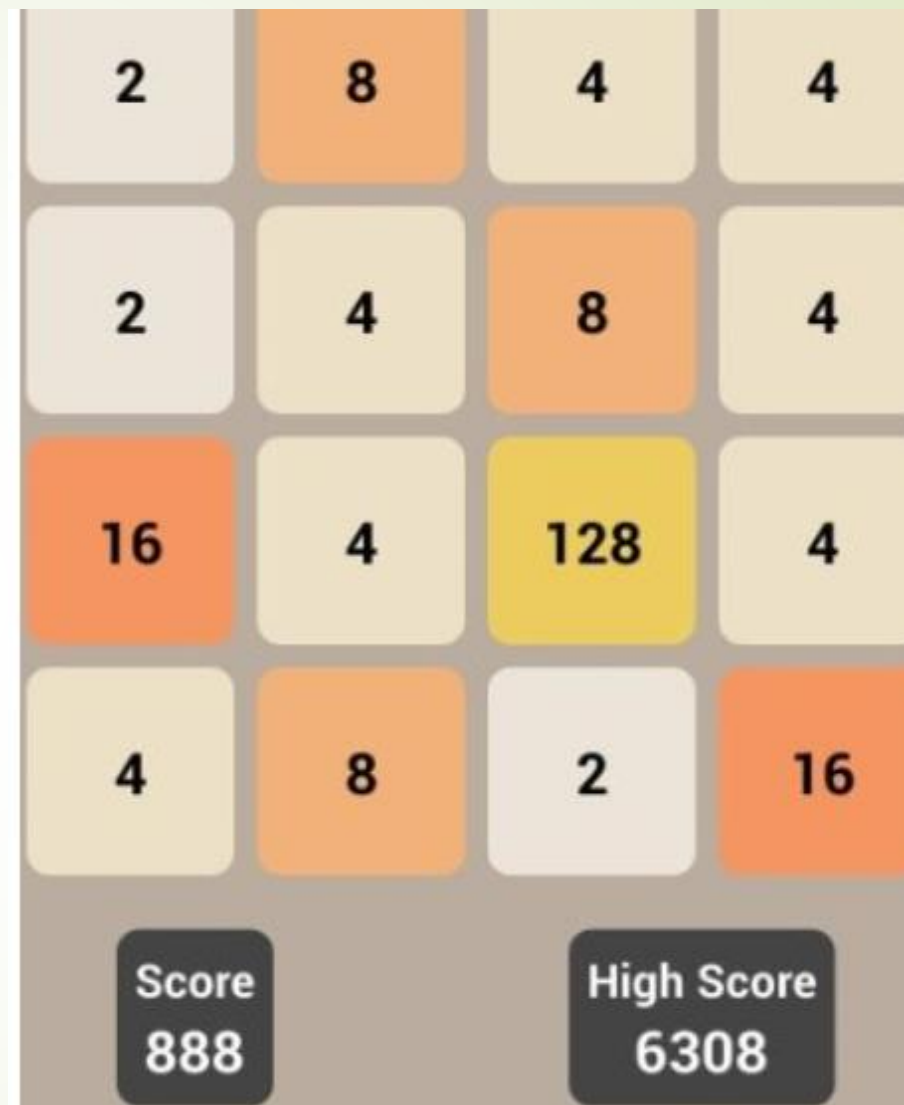
MAX

MIN



对抗搜索的例子: 2048-Puzzle

➡ <http://gabrielecirulli.github.io/2048>



2.3 机器学习

- 机器学习是一门关于从数据中提取知识的学问 (统计学习). 它是当今AI的必然趋势.
- 机器学习是关于如何创建能从数据和观测中提高我们的经验的计算机程序 (Tom Riccio).
- 你想要能教计算机如何学习并如何提高经验. 这是机器学习之核心.



机器学习的用途(一)

- 电影的自动推荐
- 商品(如食物的自动推荐)
- 个性化电台
- 在你的照片中识别出你认识的人

机器学习的用途(二)

- 人们把机器学习算法应用到材料科学来研究材料的"结构--功能"对应关系;
- 在分子物理化学中, 研究多尺度下分子的结构和力学机制;
- 在粒子物理中,发现新的粒子
- 在天文学中,发现新的天体
- 在化学中寻找最高效的化学反应,预测化学反应的产物以及预测晶体制备策略[4,5]
- 生物信息学中的基因序列分析 (DNA序列编码特征分析, 发现模式—功能对应关系等)

机器学习分类

- 机器学习主要可分为监督学习，非监督学习和强化学习等。
- 第一类学习，是**监督学习**。当你有标签时，那么你在做的就是监督学习。这些标签可以是任意连续的数值，也可以是离散的值（如文本）。如果标签只能取两个值，那么我们就在讨论二进制分类。我们想要建立的是一个函数，给定一个输入集合或是实例的描述，得出输出集合。
- 例如，假设我们有银行顾客的信息：他们的年龄，性别，职位，工资等等。对每个顾客，我们有一个标签：是否有信用卡。
- 电子邮件是否是垃圾邮件也属于标签的例子。这样例子还有很多。因此，我们有所谓的正例子和负例子。它们可以分别用1, 0表示。
- 如何找出分开这两类特征的边界往往是这类问题的目标。这是监督学习的典型例子。因为已经告诉了我们谁是正例子，谁是负例子了，所以称之为监督学习。**监督学习**包括分类，k近邻预测，神经网络，线性回归，boosting等等方法。

机器学习: 非监督学习

- 第二类学习, 就是非监督学习(unsupervised learning).
- 非监督学习中, 我们不必为实例做标签. 也就是说, 我们的**数据集**中的实例是没有标签的.
- 例如, 我们有顾客的数据, 却没有任何类型的标签与之相联系. 要解决的问题就是: 给定这些数据点, 我们能找出这些实例的数据点的聚类(clusters)吗? 我们要寻找一个函数 F 用以把输入集合 X 映射到聚类的集合. 这完全是**非监督**的算法.
- 关于非监督学习, 我们有很多不同的方法来实现. 最主要的为**聚类, 降维算法**.

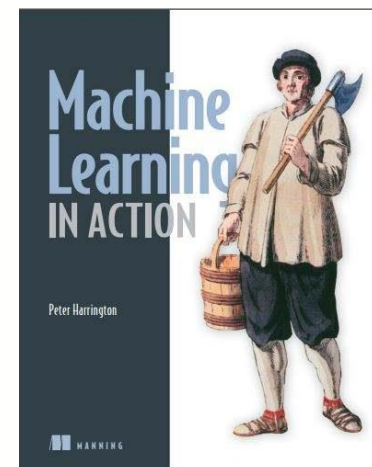
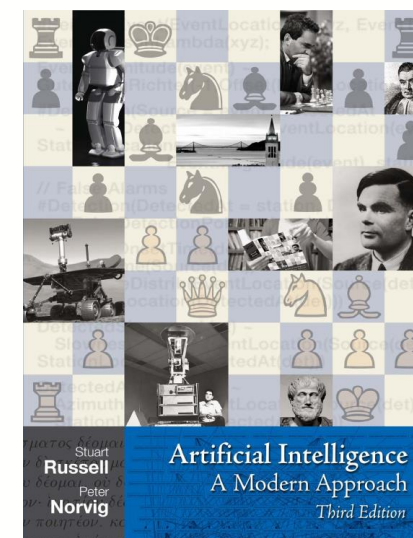
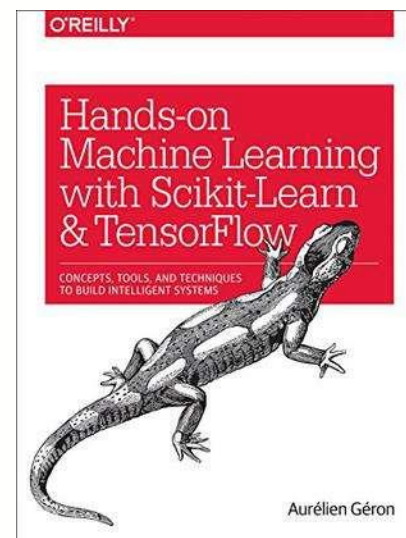


机器学习: 强化学习

- 在强化学习中, 我们设计智能体在随机或特定**环境**中演化.
- 智能体从强化或延迟奖励中学习. 它是一类用于在输出结果为随机值的决策问题中的学习方法, 它包含了一个能连续计划, **学习并影响其环境**的智能体. 强化学习的驱动力是最大化奖励.
- 强化学习在**对抗, 博弈, 游戏, 投资, 交易**等等领域都可以应用.

机器学习

- <http://aima.cs.berkeley.edu> (<人工智能:一种现代方法>) [1]
- live.vhall.com/192630502 (机器学习与自然科学)



约束问题(constraint satisfaction problem)

- 约束问题，属搜索问题，它不关注搜索的路径，只关心目标。例如解一个游戏。一般地，问题用变量来表示，而不是用态来表示。约束问题的典型例子是数独游戏(Sudoku)。
- 在9行9列的格子盘中，横向一排格子我们称为一行，纵向的一排格子称为一列。要达到的目标是各行的数字都只能取从1到9的数字，各列亦如此，且在9个3X3的小九格的数字也只能是从1到9的不重复数字。你可表示这个问题为一个CSP。每个格子可视为一个变量。加在这些变量上的约束条件就是所有3X3格子内，所有行，所有列中的数字都只能是从1到9的不重复数字。问题的解就是去寻找为变量赋值的方法，以满足这些约束条件。
- <http://www.cn.sudokupuzzle.org/>

7	6	4			2			3
	1	9	3			4	6	
2				6		7		
				8	4	1	2	
			1		5			
	7	8	2	9				
		7		5				4
	3	1			7	5	8	
6			8			9	3	7

2.5 自然语言处理和操纵物理世界

- 人工智能可以用于**自然语言处理**，在这个领域我们关注计算机和人类语言之间的相互作用，我们关注视觉，感知于图像处理，并致力于建立AI机器视觉。这里的任务就是从任务中提取信息。这样的任务包括操纵，谈判，模式识别等。
- NLP中用到概率相关概念：连接概率，语言模型(如N-gram 模型)，训练corpus，测试corpus，语言模型的平滑化，...
- 主要进展：文本分类；解析句子的语法结构：文本→树；情感分析；机器翻译；文章摘要
[MIT: **AI's Language problem**, MIT Technology Review]
- 机器人学关注如何让人工智能体**操纵物理世界**。比如，实现**自动驾驶**，**识别人脸**，机器人的自我运动等。

关于人工智能的评论

- “人工智能(AI)与100年前的电气革命一样,将会是一场新的革命。” 吴恩达
- “对于AI, 我们必须十分小心,或许它比核武器更危险。” 埃隆·马斯克
- “人工智能（我指的是不是狭隘的AI）的发展速度之快令人难以置信。除非你对Deepmind这样的项目有直接的接触，否则你不知道它的发展速度有多快它以接近指数的速度发展。”埃隆·马斯克
- “到2035年，人类的思维不可能，也不应该可以继续跟上人工智能机器的步伐了。” 格雷·斯科特
- “我设想在未来，我们可能就相当于机器人的宠物狗狗，到那时我也会支持机器人的。”克劳德·香农

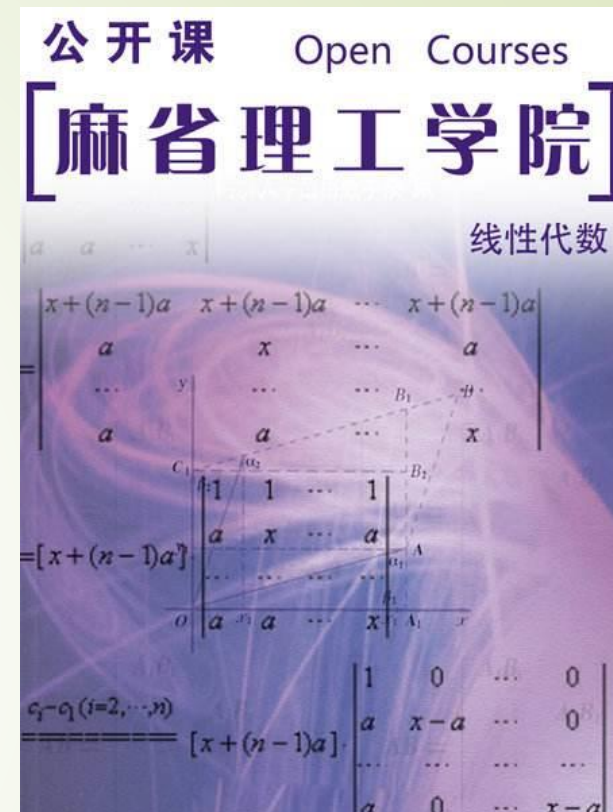
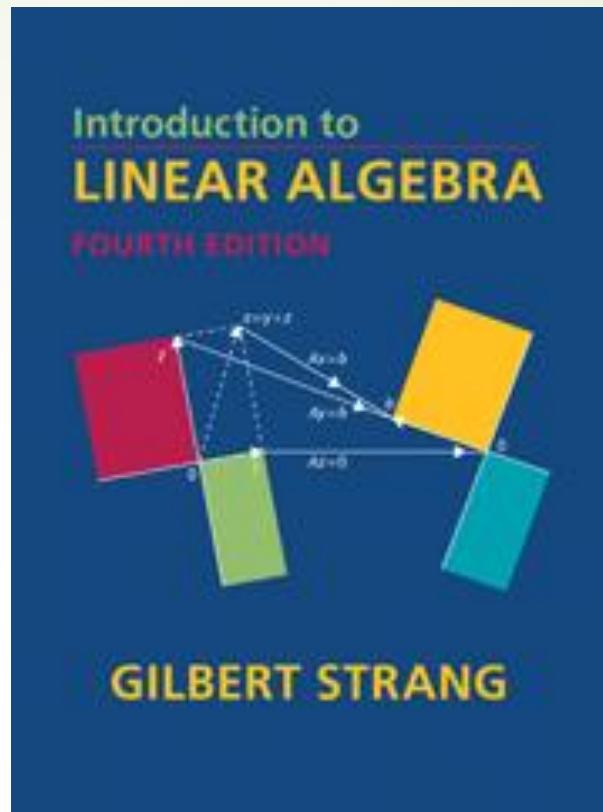
'AI IS THE NEW ELECTRICITY'



"Just as electricity transformed almost everything 100 years ago, today I actually have a hard time thinking of an industry that I don't think AI will transform in the next several years."

Andrew Ng

Former chief scientist at Baidu, Co-founder at Coursera

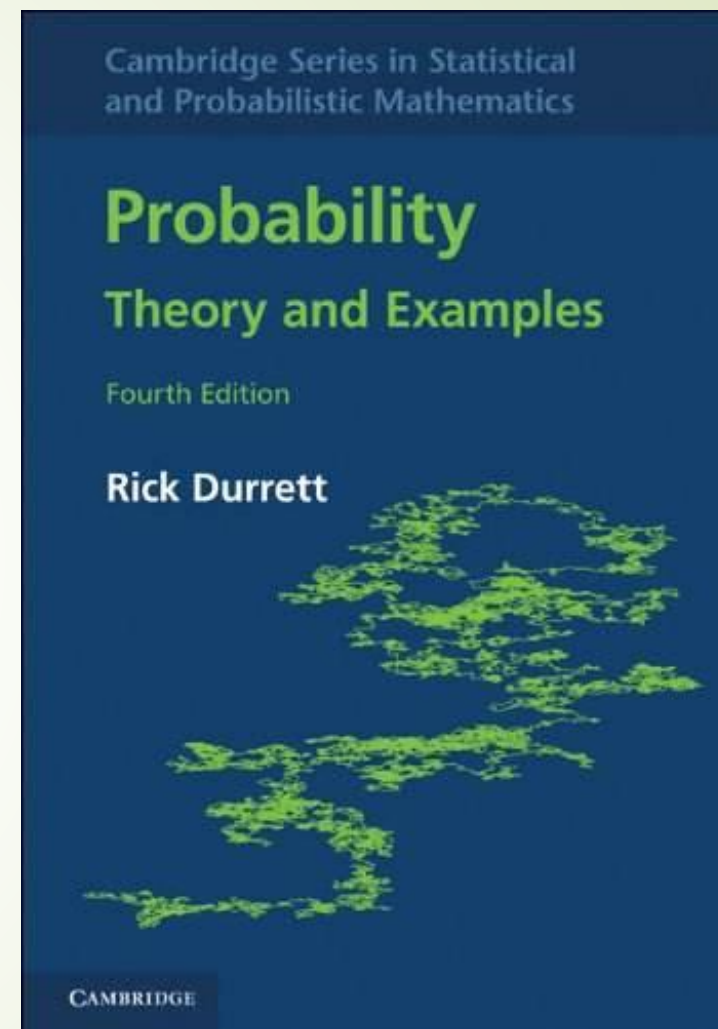
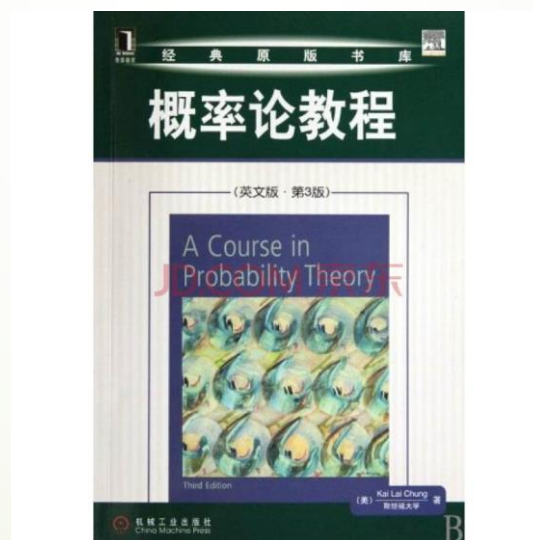


需要的准备知识

如果你想学习人工智能算法，那么你的准备知识应该包括一些编程知识，线性代数和对概率的理解。

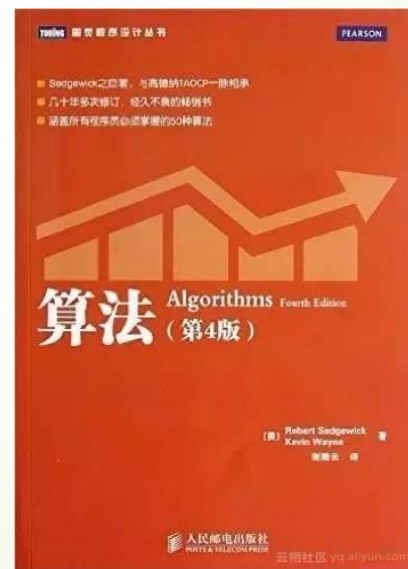
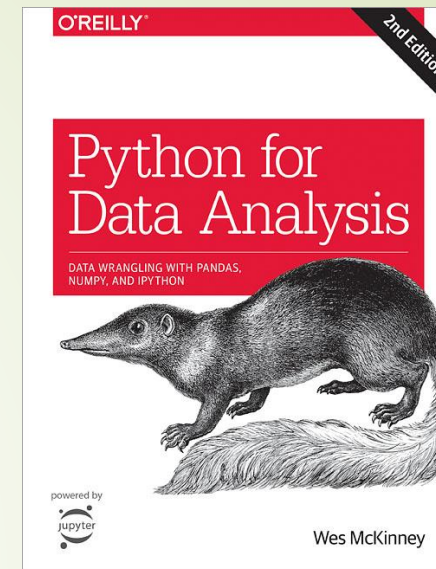
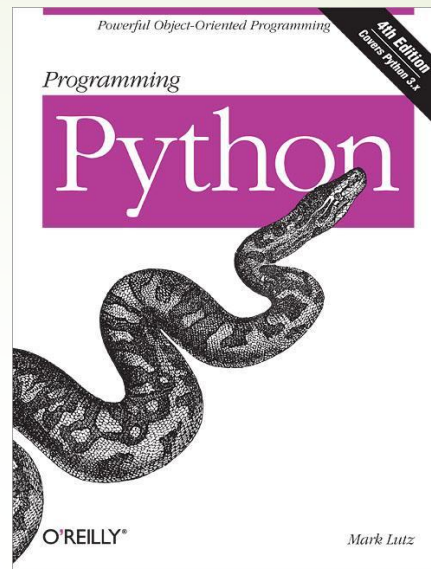
准备知识

- 概率论教材 (钟开莱)
- 概率:理论和实际例子 (Durrett)



准备知识: 编程

1. 一门编程语言: **Python, Java, C++**等;
2. 算法与数据结构.



参考文献

- [1] Russel, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1995
- [2] Luger, G. F. and Stubblefield, W. A., *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Benjamin/Cummings, Menlo Park, CA, 1993.
- [3] A. C. Mueller, S. Guido, *Python机器学习入门*, O' Reilly, 2016
- [4] P. Raccuglia, K. C. Elbert, P. D. F. Adler, C. Falk and M. B. Wenny, *Machine-learning-assisted materials discovery using failed experiments*. *Nature*, 2016 , 533 (7601) :73
DOI: 10.1038/nature17439
- [5] <http://darkreactions.haverford.edu/>