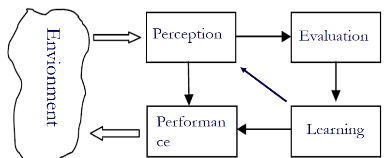
Behavior/action (weak AI )

*Can the machine act intelligently?* Turing test.

Thought process/reasoning (strong AI )

*Are machines actually thinking?* Chinese Room

the idea of a non-biological machine being intelligent is incoherent

The Architecture of intelligent system with learning capability

Reinforcement Learning choose actions to obtain a lot of reward.

Supervised Learning Given example input/output pairs(answer)

Unsupervised Learning Given examples, no labeling(no answer)

Overfitting learns not only the underlying patterns in the training data but also the *noise* and outliers. (generalization)

Bias: errors introduced by the model due to *oversimplified assumptions*.

Robustness: how does the training data influence the learning result? Data Scale, Change, Noise, and Imbalance

Computation Complexity: efficiency of the learning algorithms? Time, Memory, Scalability, convergency

Transparency: interpretability and *whether humans can understand how it makes decisions.*

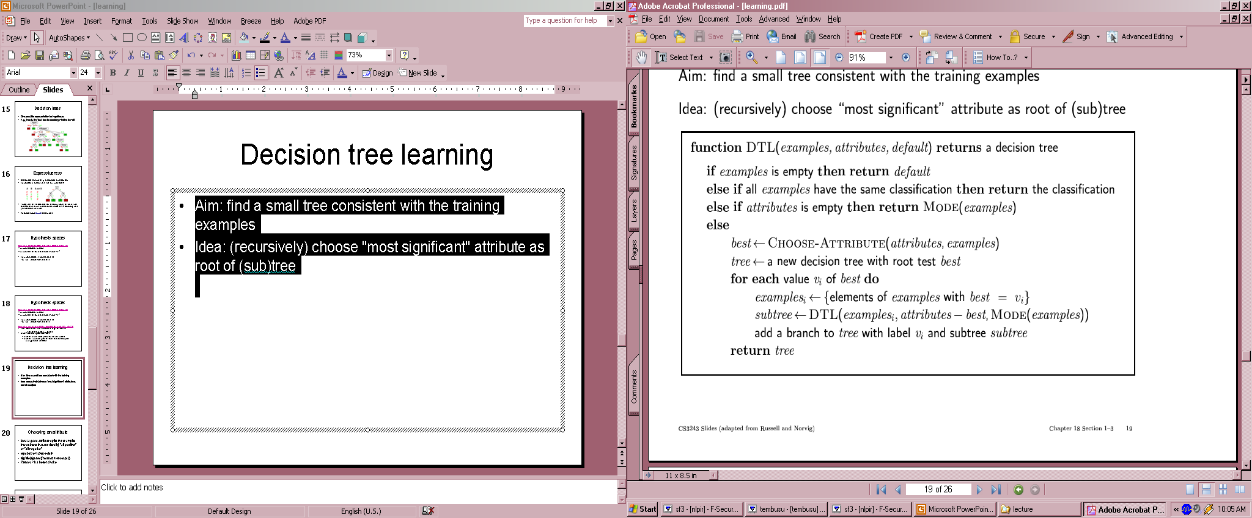
Rote Learning Save solutions ,retrieve when encountering same problem.

Ockham’s razor: prefer the simplest hypothesis consistent with data

Decision tree用于决策的递归recursive、分层hierarchical模型

Examples described by attribute values

Aim Find a small tree consistent with the training examples

Idea (recursively) Choose "most significant" attribute as root of (sub)tree

Information Content (Entropy):I(P(v1), … , P(vn)) = Σi -P(vi) log2 P(vi)

Information Gain (IG) or reduction in entropy from the attribute test:

Choose the attribute with the largest IG

Description Length(DL):用DFS遍历决策树进行编码

1表示下一个节点非叶并记录当前节点的属性 0表示下一个节点是叶节点并用Y N记录节点类属性

*“1 喜欢程度 1 替代物 0 N 0 Y 0 Y 1 急用 0 Y 1 价格 0 N 0 N 0 Y 0 N”*，

其编码长度为24.585。编码长度的计算过程：由于在根节点上有4种属性可供选择，因此表示根节点上的属性“喜欢程度”需要log2 4比特。由于“喜欢程度”已经作为根节点的属性，因此下面可供选择的属性只有三种。这说明表示下一层的属性“替代物”最少需要log2 3比特。依次类推，表示属性“急用”需要1比特；表示属性“价格”需要0比特。这样表示属性的比特数总共为4.585。而表示“0/1”，“N/P”各自需要1比特，则上述编码中表示属性之外的其他数据共需要20比特。编码例外数据只需指定它在所有n个可能数据(n=∑属性种类×属性个数)中的位置：log2 n比特

贝叶斯学习

h:hypothesis D:data P(h|D)后验概率

Maximum A Posterior (MAP)极大后验概率 知道了类条件概率P(D|h)和先验概率P(h)，就可以进行贝叶斯决策

H是结论的集合 贝叶斯学习的主要目标是类条件概率和类先验概率核心的学习任务是如何学习类条件概率

Naive Bayesian Classifiers 假设数据向量中各分量相互独立，从而将高维数据分布转化为若干一维数据分布进行处理

(决策公式)

文档分类 19个字 每个字有5000种可能的取值 1000份文档700不喜欢300喜欢 先验概率P(like)=0.3 P(不喜欢)=0.7

类条件概率m估计 h表示类别 w表示字 表示h类文档中w出现的次数 nj表示用于训练的hj类文档中字的总数

v表示字的可能取值数(5000)

Bayesian Belief Networks 数据向量中各分量之间的局部依赖关系

随机向量中某一分量所依赖的分量集合称为它的父节点parents(x) 联合概率分布可以用有向无环图来表示，图中顶点为随机向量中的各个分量，有向边表示分量之间的局部依赖关系.将图中顶点看作相对独立的事物，则贝叶斯信念网也表示了事物之间的因果关系

A→B A为B的父节点 节点边有条件概率表显示所有可能的条件概率值的集合

Objective 极大似然估计 所有观测数据同时存在的可能性最大化

设w表示由所有条件概率值构成的向量,D表示由所有观测数据构成的向量，则所有观测数据同时存在的概率Pw(D),学习目标：

Optimization 梯度上升 迭代更新贝叶斯信念网中各节点的条件概率表，直到各节点条件概率表不再发生改变或达到最大迭代次数为止。

xij节点表示i的第j种取值

uik表示节点的父节点集对应的第k种取值组合 Wijk表示父节点取uik时该节点取xij的条件概率，η为学习率

Unsupervised Learning: Clustering *Discovering groups and Identifying interesting distributions in the underlying data.*划分聚类方法直接将数据集划分为相应的子集 层次聚类方法通过分解与合并数据实现聚类，反映了类别的层次关系（层次聚类的过程是合并或者分解的过程）

连续数据的相似性度量

距离公理 d(x,y)≥0 d(z,z)=0 d(x,y)=d(y,x) d(x,y)≤d(x,z)+d(y,z)

欧式距离 sqrt()

城区距离

切比雪夫max

闵可夫斯基(

k-means

Step1 从数据集中随机选择k个数据，作为初始簇的均值向量。

Step2 对剩余的每个数据根据其与各个簇均值向量的距离将它划分给最近的簇。

Step3 在新产生的k个簇的基础上更新各个簇的均值向量。

Step4 重复Step2-3，直到停止条件满足，如均值向量不再发生变化，或者达到最大迭代次数等等。

优点是简单易实现、可以收敛到局部最优、具有多项式时间复杂度。缺点在于：需要事先给定聚类的个数即k值。不够鲁棒，易受噪声和初始均值向量影响。

k-medoids Partitioning Around Medoids

替换代价 用h替换i后，j对应的替换代价

j本来属于i的类别，用h替换i后，j被划分至h的类别

C=d(j,h)-d(j,i)

j本来属于i的类别。用h替换i后j被划分至t类别

C=d(j,t)-d(j,i)

j本来属于t类别。用h替换i后j仍然属于t类别 C=0

j本来属于t类别。用h替换i后j被划分至h的类别

C=d(j,h)-d(j,t)

用非中心点h替换中心点i后，用上述方法计算出每个非中心点所对应的代价并求和得总代价，总代价小于0则应替换

Step1 随机选择k个中心点数据

Step2 对每一对非中心点h和中心点i计算替换总代价

Step3 如果最小C<0，则相应的中心点i被非中心点h替代

Step4 重复步骤Step2-3，直到最小C≥0.

Step5 将非中心点数据分配给距离其最近的中心点数据。

比k-均值算法健壮，但对于大数据集效率不高

CLARA(Clustering LARge Applications)算法在整个数据集上进行抽样，然后在抽样所获得的数据子集上利用k-中心点算法获得最佳的中心点。进一步利用多次不同抽样来改进CLARA则得到CLARANS(Clustering LARge Application based upon RANdomized Search)算法。

Hierarchical clustering 层次聚类

在自下而上的凝聚层次聚类方法中，首先将每个数据作为一类，然后逐步合并形成越来越大的类别，直到所希望的类别层次。在自上而下的分裂层次方法，首先将所有数据作为一类，然后逐步分解形成越来越小的类别，直到所希望的类别层次。凝聚层次聚类与分裂层次聚类是一对互逆过程 *简单的凝聚或分裂层次聚类算法效率不高，另外，一旦在聚类的某一层次上发生问题，则以后各层的聚类都是错误的，问题不可逆转*

BIRCH: treat the process of hierarchal clustering as a tree.

CURE: use a set of well-scattered points to represent each cluster用多个代表点表示一个簇，簇间距离为两个簇中代表点之间距离的最小值。开始时代表点为最分散的点（距离簇中心最远的点），然后随着聚类的进行，逐步使这些代表点向簇的中心方向收缩。在实际应用中为了解决大数据量问题采用随机抽样和分区方法提高计算效率。

Step1 抽样数据集，得到样本集，设样本集中数据个数为n

Step2 划分样本集为p个分区，每个分区规模为n/p

Step3 对每一分区进行CURE层次聚类，簇的数目为n/pq

Step4 当簇的数量下降缓慢时以及层次聚类结束时，删除异常点(非常小的簇)

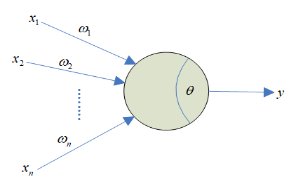
Step5 对形成的所有簇进行CURE层次聚类

Step6 将所有点分配到距离其最近的代表点所在的簇

CHAMELEON: use dynamic model 创建一个稀疏图(k-nearest neighbor graph)再把图分割，最后把分割的小块聚合

Artificial Neural Networks

人工神经元 f是激活函数(activation) g是整合函数combination function

是连接强度（连接权值）θ是输出阈值

整合函数：加权求和∑x -θ

径向距离函数 以输入向量与中心向量的欧式距离作为整合后的结果

ANN Architectures

Feedforward Architecture :without loops static不具有“记忆”能力

Feedback (Recurrent) Architecture :with loops dynamic (non-linear dynamical systems)

Learning Approaches to ANN

Error Correction *Supervised* 感知器BP 最小平方误差

Hebbrian Learning *Unsupervised* 强化连接

竞争型学习 非监督 获胜的神经元抑制失败的对刺激模式的响应

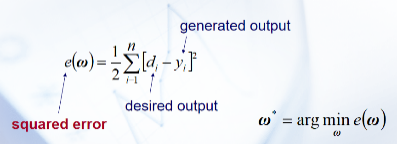
Single-layer Perceptron

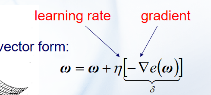
m个输入神经元 n个输出神经元有m(n+1)个权值

是输出阈值是第i个输入和第j个输出之间的连接权值

仅能解决一阶谓词逻辑问题 不能解决XOR

BP Network

BP学习中需逐层调整各个神经元的连接权值，学习过程由信号正向传播和误差反向传播两个阶段构成。信号正向传播用于获得在隐含层神经元和输出层神经元上的输出信号；误差反向传播用于将网络期望输出与实际输出的误差从输出层反向传播至输入层，并在此过程中更新各个神经元的连接权值 基于最小平方误差准则和梯度下降优化方法来确定权值调整法则



权值迭代

Recurrent Network

Hopfield network Feedback (recurrent) architecture Dynamic System

通过能量函数最小化达到系统稳定状态。但在计算能量函数最小值时可能陷入局部最优

应用：联想记忆 优化计算

波尔兹曼机的结构与霍普菲尔德网络基本一致，关键区别在于激活函数不同。在波尔兹曼机中，神经元状态的改变不再是确定的，而是按照波尔兹曼-吉布斯分布进行随机的改变。随温度降低，激活概率逐渐趋于确定值的特性正是波尔兹曼机实现全局最优的基础

Long Short-Term Memory

自组织特征映射网是一种非监督学习型网络，通过神经元之间的相互竞争和协同，以及在此基础上的自组织过程，使得可以在一维或二维空间中用有限点表达高维空间中众多的甚至无限的数据，并且可以刻画数据的统计分布和拓扑特性，表现其内在特征

搜索与问题求解

状态空间（S,F,C,I,G）其中S是状态的集合，F是用于状态转换的操作符的集合，C是状态转换代价的集合，I是初始状态的集合，G是目标状态的集合。

与或图 对问题进行分解、变换的过程

分解：所有子问题都有解时原问题有解(与节点) 变换：一个子问题有解原问题就有解（或节点）直到以下两种情况发生：（1）不能再进行分解或变换，（2）子问题可以直接解答，不需要再进行分解或变换。那些不需要再进行分解或变换，可以直接解答的子问题被称为本原问题。

可解顶点:1.该顶点是终止顶点。2.该顶点是“与”节点中的父顶点，并且其全部子顶点都是可解的。3.该顶点是“或”节点中的父顶点，并且其子顶点中至少有一个是可解的。

Step1 确定单个问题的描述形式，可采用状态空间表示法。

Step2 从原始问题开始，逐步进行分解和变换，直到本原问题。然后将全部分解和变换过程表示为与或图。

（同样，对于复杂问题，与或图一般不是一次生成，而是逐步展开的，具体方法详见下一节。）

Step3 从端顶点开始，逐级向上回溯，标注各顶点为可解顶点或不可解顶点，直到标注原始顶点为可解顶点或不可解顶点为止。

Step4 当原始顶点被确定为可解顶点时，输出相应解图为问题的解

图搜索基本算法。

Step1 把初始顶点放入Open表，建立仅包含初始顶点的图G；

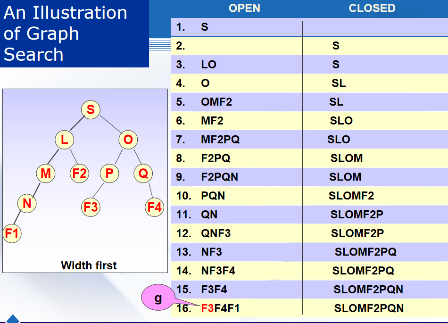
Step2 按一定策略，从Open表中取出待扩展顶点，放入Closed表；

Step3 扩展从Open表中取出的待扩展顶点，分为“该顶点可扩展”和“该顶点不可扩 展”两种情况进行处理：

如果该顶点可扩展，则对于状态空间，将扩展得到的、未在G中出现过的子顶点放入Open表。然后根据需要修改G中顶点之间的父子关系；对于与或图，将扩展得到的子顶点放入Open表；对于子顶点中的终止顶点，标注其可解，并逐步回溯，确定其祖先顶点是否可解。

如果该顶点不可扩展，则对于状态空间，不做任何处理；对于与或图，标注顶点为不可解，并逐步回溯，确定其祖先节点是否不可解。

Step4 重复Step2-3直到结束。对于状态空间，结束条件是待扩展顶点为目标顶点，或者Open表为空；对于与或图，结束条件是初始顶点被标注为可解或不可解，或者Open表为空。



启发式图搜索

A\*算法 估价函数f(n) = g(n) + h(n)

g(n) = cost so far to reach n

h(n) = estimated cost from n to goal (heuristic function)

f(n) = estimated total cost of path through n to goal

A heuristic h(n) is admissible if for every node n, h(n) ≤ h\*(n), where h\*(n) is the true cost to reach the goal state from n.

An admissible heuristic never overestimates the cost to reach the goal, i.e., it is optimistic

Step1. Put the initial node into open table

Step2. Select a node from open table and put it into closed table according to heuristic function.

Step3. Expand the selected node. If the node can be expanded, put all the expanded nodes into open table.

Step4. Repeat Step2-3 until the goal node is selected in step2, or open table is empty.

Adversarial Search 博弈搜索

Game tree 在“双人、零和、信息完备、非偶然性”博弈过程中，一般假设博弈中的己方为MAX，另一方为MIN。用博弈树表示双方你来我往的博弈过程。博弈树中的顶点表示当某一方准备采取行动时所面对的博弈状态，边表示可以采取的行动，通过一条边连接的两个顶点表示当某一方采取行动后博弈状态的变化。博弈树在结构上就是一棵与或树。其特点是：

1)博弈的初始状态是初始顶点；

2)博弈树中的“或”顶点和“与”顶点逐层交替出现；

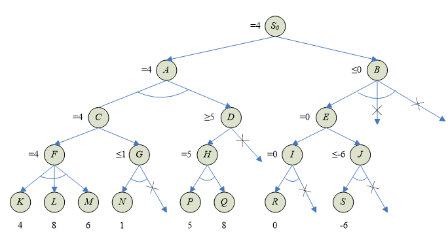
3)整个博弈过程始终站在某一方的立场上，所有能使自己获胜的叶顶点是本原问题，能使自己获胜的顶点是可解顶点；所有使对方获胜的顶点都是不可解顶点。

极大极小搜索与α-β剪枝

有界深度优先策略生成博弈子树。首先沿着某一分枝一直向下扩展顶点，直到所规定的深度。在获得相应叶顶点后，计算叶顶点的静态估值，进而倒推非叶顶点的估值。根据倒推结果，在非叶顶点的向下分枝中，剪掉那些目前尚未扩展，但无论其扩展与否，都不会改变非叶顶点倒推值的分枝

α is the value of the best (i.e., highest-value) choice found so far at any choice point along the path for MAX .If v is worse than α, MAX will avoid it, prune that branch. Define β similarly for MIN

α剪枝 任何MIN顶点Sn的β值小于或等于它的某一祖先MAX顶点的值，则Sn以下的分枝可停止搜索，并令顶点的倒推值为β

β剪枝 任何MAX顶点Sn的α值大于或等于它的某一个祖先MIN顶点的β值，则Sn以下的分枝可停止搜索，并令顶点的倒推值为α。

Knowledge and reasoning

一阶谓词逻辑 first order logic

表示知识时首先定义原子谓词，然后用连接词或量词把这些谓词连结起来，形成谓词公式。对于事实性知识通常用否定、析取或合取符号连接起来的谓词公式表示。对于过程性知识通常用蕴含式表示。

产生式规则 production system

If p(前提，前件，事实的逻辑组合) then q（后件）

对于确定性事实用三元组表示，分为两种情况：用三元组（对象，属性，值）表示事物的属性。例如（雪，颜色，白）表示“雪的颜色是白的”。用三元组（关系，对象1，对象2）表示事物之间的相互关系。例如（热爱，王峰，祖国）表示“王峰热爱祖国”不确定性事实（雪，颜色，白，0.8）表示“雪的颜色是白的”这一事实可以相信的程度为0.8。

进化计算 进化算法本质上是启发式随机搜索算法

Step1 随机产生解的一个初始群体。

Step2 根据问题目标，评价当前群体中每个个体（解）的适应度。

Step3 根据个体适应度，对当前群体执行自然选择和遗传操作（基因重组、突变等），获得迭代后的下一代解。

Step4 如果所获得的个体（解）已满足要求，则算法停止，否则返回到Step2。

Genetic algorithms 遗传算法

Step1 算法初始化：确定编码方法、适应度函数、群体规模、选择算子、交叉算子、突变算子、交叉概率、突变概率等算法参数。

Step2 随机产生解的初始群体

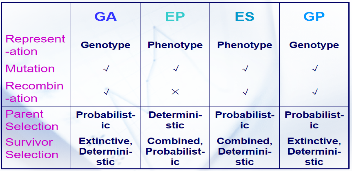
Step3 计算群体中每个个体的适应度

Step4 执行以下遗传操作形成新群体

Step4.1 选择：按适应度高的原则，从旧群体中选出个体组成群体A，群体A的规模与旧群体相当，旧群体中的个体可能在群体A中多次出现，也可能不出现。

Step4.2 交叉：根据交叉概率，在群体A上反复随机选择两个个体发生交叉，获得规模与群体A相同的群体B

Step4.3 突变：根据突变概率，群体B中每个个体发生突变，获得新群体。

Step5 如果群体性能已满足要求，或者已达到最大迭代次数，则算法停止，输出群体中适应度最高的个体作为结果；否则返回Step3。

Genetic programming (tree based)→选择1个个体突变(random change in tree)或者2个个体重组（交换子树）

Evolution strategies

self-adaptation of (mutation) parameters standard Consist of three parts:

Object variables: x1,…,xn

Strategy parameters:

Mutation step sizes: δ1,…,δn

Rotation angles

Evolutionary programming

no recombination!

self-adaptation of parameters standard

Swarm Intelligence

Simple behaviours of individual agents+ Communication between a group of agents = Emergent complex behaviour of the group of agents

*Distributed, no central control or data source*

*Limited communication*

*No (explicit) model of the environment*

*Perception of environment (sensing)*

*Ability to react to environment changes.*

Ant Colony Optimization (ACO)

Each artificial ant is a probabilistic mechanism that constructs a solution to the problem, using:

Artificial pheromone deposition

Heuristic information: pheromone trails, and others

Particle Swarm Optimization (PSO)

Population initialized by assigning *random positions and velocities*; potential solutions are then flown through hyperspace.

Each particle keeps track of its “best” (highest fitness) position in hyperspace. This is called “pBest” for an individual particle. It is called “gBest” for the best in the population. It is called “lBest” for the best in a defined neighborhood

At each time step, each particle *stochastically accelerates* toward its pBest and gBest (or lBest).

Step1. Initialize population in hyperspace.

Step2. Evaluate fitness of individual particles.

Step3. Modify velocities based on previous best and global (or neighborhood) best.

Step4. Terminate on some condition.

Step5. Go to step 2.

Adaptation operates on velocities *Most other methods operate on positions*Effect: PSO has a built-in momentumParticles tend to hurdle past optima – an advantage, since the best positions are remembered anywayAutonomous Agents

Weak notion

Situatedness: *situated in the world, not in disembodied systems.情景*

Autonomous: *function without intervention*

Proactive: *goal-directed behaviour*

Reactive: *perceive and respond to changing environment*

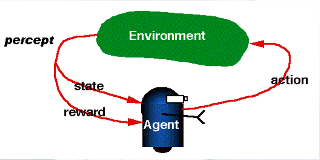
Social ability: *interaction with others*

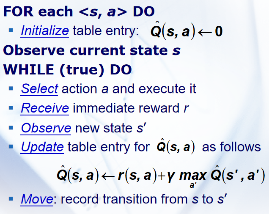
Strong notion

Belief Desire Intention knowledge（mental components）BDI

BDI models

Three categories: - informative (knowledge, Belief, assumptions) - motivational (Desires, motivations, goals) - deliberative (Intentions, plans)Reinforcement Learning Based on trial-error interactions



Q-Learning

Deep Reinforcement Learning Value-based Policy-Based

