【人工智能基础 复习笔记】

【注】

①内容概要：对每一章进行算法梳理、案例梳理，章末附慕课习题及答案解析。

②个人认为，考纲排除的一整个章节仍有考查选择题的可能，对这些章节也进行了一些简单的梳理。

③根据课堂内容，对一些章节可能考查的填空题进行了预测，在各章结束后呈现。

第一章 绪论

1.二十世纪初涌现的可计算思想的提出推动了原始递归函数、λ－演算和图灵机等“计算载体”的出现。

2.人工智能的手段与方法

手段与方法	特点
以符号主义为核心的逻辑推理	将概念（如命题等）符号化，从若干判断（前提）出发得到新判断（结论）
以问题求解为核心的探寻搜索	探寻搜索依据已有信息来寻找满足约束条件的待求解问题的答案
以数据驱动为核心的机器学习	从数据中发现数据所承载语义（如概念）的内在模式
以行为主义为核心的强化学习	根据环境所提供的奖罚反馈来学习所处状态可施加的最佳行动，在“探索（未知空间）-利用（已有经验）（exploration vs. exploitation）”之间寻找平衡，完成某个序列化任务，具备自我学习能力
以博弈对抗为核心的群体智能（两人及以上）	从“数据拟合”优化解的求取向“均衡解”的求取迈进

3.有效的形式化系统具有如下特性：完备性、一致性、可判定性。

4.推理、搜索和约束满足是人工智能问题求解中的三大方法。

5.不同推理方法及其差异

①归纳推理

定义：如果Ai(i为若干取值)，那么B。

说明：从若干事实出发，推理出一般性规律。

②演绎推理

定义：如果A，那么B。

说明：A是B的前提，但不一定是唯一前提。因此A是B的充分条件。在特殊情况下，A也可以是B的充分必要条件。

③因果推理

定义：因为A，所以B。

说明：A是B的唯一前提，“如果没有A，那么没有B”也成立。

【填空题】以下为填空题，涉及后续章节的基本内容。

4.搜索算法的分类：

①无信息搜索、有信息搜索和对抗搜索。

②有信息搜索又称启发式搜索，可利用与搜索求解相关的辅助信息。代表算法是：贪婪最佳优先搜索、A*搜索。

③对抗搜索又称博弈搜索，指在一个竞争环境中，智能体之间通过竞争实现相反的利益，一方最大化，另一方最小化这个利益。代表性算法：最大最小搜索、Alpha-Beta剪枝搜索和蒙特卡洛树搜索。

④无信息搜索是一种盲目搜索算法，按照扩展被搜索空间中的节点次序可以分为广度优先搜索、深度优先搜索等。

5.卷积神经网络通过卷积层、池化层、激活函数和全连接层来构造深度模型，辅以误差后向传播等机制，学习模型参数，得到输入数据的更具区别力的特征表达。

6.监督学习算法可以分为判别式、生成式。

①判别式监督学习算法有：回归分析、提升算法、支持向量机、决策树等。

②生成式监督学习算法有：隐狄利克雷分布（LDA）、隐马尔可夫链等。

7.无监督学习包括聚类、降维、期望极大等算法。

第一章 MOOC测验题

1.下面哪一句话准确描述了摩尔定律（ A ）

A.摩尔定律描述了计算机的计算速度每一年半增长一倍的规律

B.摩尔定律描述了互联网所链接节点随时间不断增长的规律

C.摩尔定律描述了计算机的体积大小随时间不断减少的规律

D.摩尔定律描述了计算机内存大小随时间不断增长的规律

2.下面哪个方法于20世纪被提出来，用来描述对计算机智能水平进行测试（ C ）

A.摩尔定律

B.费马定理

C.图灵测试

D.香农定律

3.1955年，麦卡锡、明斯基、香农和诺切斯特四位学者首次提出“artificial intelligence（人工智能）”这个概念时，希望人工智能研究的主题是（ D ）

A.避免计算机控制人类

B.全力研究人类大脑

C.人工智能伦理

D.用计算机来模拟人类智能

4.下面哪个描述不属于对人工智能的分类（ A ）

A.生物智能

B.混合增强人工智能

C.通用人工智能/跨域人工智能

D.领域人工智能/弱人工智能

5.以逻辑规则为核心的逻辑推理、以数据驱动为核心的机器学习和以问题引导为核心的强化学习是三种人工智能的方法，下面哪一句话的描述是不正确的（ D ）

A.强化学习的基本特征是智能体与环境不断进行交互，在交互过程不断学习来完成特定任务

B.以逻辑规则为核心的逻辑推理方法解释性强

C.目前以数据驱动为核心的机器学习方法需要从具有标签的大数据中来学习数据模式，完成给定任务（正确）

D.目前以数据驱动为核心的机器学习方法可从任意大数据（无论数据是具备标签还是不具备标签）中来学习数据模式，完成给定任务（错误）

6.我们常说“人类是智能回路的总开关”，即人类智能决定着任何智能的高度、广度和深度，下面哪一句话对这个观点的描述不正确（ D ）

A.机器智能和人类智能相互协同所产生的智能能力可超越人类智能或机器智能

B.人类智能是机器智能的设计者

C.机器智能目前无法完全模拟人类所有智能

D.机器智能目前已经超越了人类智能

7.下面哪句话描述了现有深度学习这一种人工智能方法的特点（ B ）

A.小数据，大任务

B.大数据，小任务

C.大数据，大任务

D.小数据，小任务

8.20世纪30年代，围绕可计算这一重要思想，一些模型被提出。下述哪个模型不是于这个时期在可计算思想推动下产生的（ A ）

A.冯诺依曼模型

B.原始递归函数

C.lambda 演算

D.图灵机

9.德国著名数学家希尔伯特在1900年举办的国际数学家大会中所提出的“算术公理的相容性 (the compatibility of the arithmetical axioms)”这一问题推动了可计算思想研究的深入。在希尔伯特所提出的这个问题中，一个算术公理系统是相容的需要满足三个特点。下面哪个描述不属于这三个特点之一（ A ）

A.复杂性，即算法性能与输入数据大小相关

B.一致性，即一个命题不可能同时为真或为假

C.可判定性，即算法在有限步内判定命题的真伪

D.完备性，即所有能够从该形式化系统推导出来的命题，都可以从这个形式化系统推导出来。

10.下面哪个描述不属于邱奇 - 图灵论题所包含的意思（ A ）

A.任何表达力足够强的(递归可枚举)形式系统同时满足一致性和完备性

B.有些数学问题是不可求解的(图灵不可停机的)

C.任何计算，如果存在一个有效过程，它就能被图灵机实现

D.凡是可计算的函数都可以用图灵机计算

第二章 逻辑与推理

一、命题逻辑

1.符号主义人工智能方法基于如下假设：可通过逻辑方法来对符号及其关系进行计算，实现逻辑推理，辨析符号所描述内容是否正确。

2.在符号主义人工智能中，所有概念均可通过人类可理解的“符号”及符号之间的关系来表示。

3.命题逻辑

①命题逻辑(proposition logic)是应用一套形式化规则对以符号表示的描述性陈述进行推理的系统。

②在命题逻辑中，一个或真或假的描述性陈述被称为原子命题，对原子命题的内部结构不做任何解析。

③若干原子命题可通过逻辑运算符来构成复合命题。

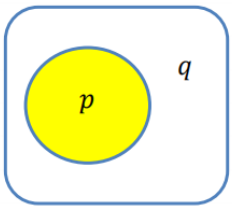
4.命题联结词

命题连接符号	表示形式	意义
与(and)	$p \wedge q$	命题合取(conjunction)，即“p且q”
或(or)	$p \vee q$	命题析取(disjunction)，即“p或 q”
非（not）	$\neg p$	命题否定(negation)，即“非 p ”
条件(conditional)	$p \rightarrow q$	命题蕴含(implication)，即“如果p 则 q”
双向条件(bi-conditional)	$p \leftrightarrow q$	命题双向蕴含(bi-implication)，即“p 当且仅当 q”

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

“条件”命题联结词中前提为假时命题真假取值

- “如果p那么q $(p \longrightarrow q)$”定义的是一种蕴涵关系（即充分条件），也就是命题q 包含着命题p （ p是q的子集）
- p不成立相当于p是一个空集，空集可被其他所有集合所包含，因此当p不成立时，“如果p那么q”永远为真。

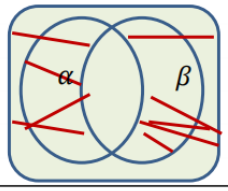
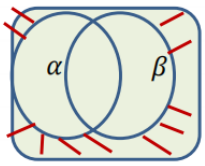


5.逻辑等价

逻辑等价：给定命题p和命题q，如果p和q在所有情况下都具有同样真假结果，那么p和q在逻辑上等价，一般用≡来表示，即$p \equiv q$。

逻辑等价为命题进行形式转换带来了可能，基于这些转换不再需要逐一列出p和q的真值表来判断两者是否在逻辑上等价，而是可直接根据已有逻辑等价公式来判断p和q在逻辑上是否等价。

$\alpha \wedge \beta \equiv \beta \wedge \alpha$ (∧的交互律)	$(\alpha \Rightarrow \beta) \equiv \neg\alpha \vee \beta$ (蕴涵消除)
$\alpha \vee \beta \equiv \beta \vee \alpha$ (∨的交互律)	$(\alpha \Leftrightarrow \beta) \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$(双向消除)
$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ (∧的结合律)	$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ (De Morgan)
$(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$ (∨的结合律)	$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ (De Morgan)
$\neg(\neg\alpha) \equiv \alpha$ (双重否定)	$(\alpha \wedge (\beta \vee \gamma)) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ (∧ 对 ∨的分配律)
$(\alpha \Rightarrow \beta) \equiv \neg\beta \Rightarrow \neg\alpha$(逆否命题)	$(\alpha \vee (\beta \wedge \gamma)) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ (∨ 对 ∧的分配律)

$(\alpha \Rightarrow \beta) \equiv \neg\beta \Rightarrow \neg\alpha$(逆否命题)	秋天天气变凉⇒大雁南飞越冬≡大雁没有南飞越冬⇒秋天天气没有变凉
	$x \geq 0 \Rightarrow x^2 \geq 0 \equiv x^2 < 0 \Rightarrow x < 0$
$(\alpha \Rightarrow \beta) \equiv \neg\alpha \vee \beta$ (蕴涵消除)	α为假、则命题恒为真；α为真、则β须为真
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ (De Morgan)	
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ (De Morgan)	

【解答题】解答题考点：逻辑与推理

以下是与本考点相关的基础知识、方法、例题。

6.逻辑推理涉及的一些基本规则（题目可能要求注明逻辑推理过程中每一步涉及的规则）

①假言推理：就是三段论。

②与消解：从一堆条件的与为真，推出各个条件分别为真。

③与导入：从各个条件分别为真，推出它们的与为真。

假言推理 (Modus Ponens)	$\frac{\alpha\Rightarrow\beta,\ \alpha}{\beta}$
与消解 (And-Elimination)	$\frac{\alpha_1\wedge\alpha_2\wedge\cdots\wedge\alpha_n}{\alpha_i(1\le i\le n)}$
与导入 (And-Introduction)	$\frac{\alpha_1,\alpha_2,\ldots,\alpha_n}{\alpha_1\wedge\alpha_2\wedge\cdots\wedge\alpha_n}$

④双重否定

⑤单项消解（单项归结）：已知两个命题中至少有一个为真，且其中一个确定为假，推出另一个为真。

⑥消解或归结：单项消解或单项归结的一般情况。

双重否定 (Double-Negation Elimination)	$\frac{\neg\neg\alpha}{\alpha}$
单项消解或单项归结 (Unit Resolution)	$\frac{\alpha \vee \beta, \neg\beta}{\alpha}$
消解或归结 (Resolution)	$\frac{\alpha\vee\beta,\neg\beta\vee\gamma}{\alpha\vee\gamma}, \frac{\alpha_1\vee\alpha_2\vee\cdots\vee\alpha_m,\neg\beta}{\alpha_1\vee\alpha_2\vee\cdots\vee\alpha_{k-1}\vee\alpha_{k+1}\vee\cdots\vee\alpha_m}(\neg\alpha_k = \neg\beta)$

7.逻辑推理三道例题

【例1】

1	$\alpha \vee \beta$
2	$\alpha \rightarrow \gamma$
3	$\beta \rightarrow \gamma$

例2.3 已知如上命题成立，请证明命题γ是成立的

1	$\alpha \vee \beta$	已知
2	$\neg\alpha \vee \gamma$	2进行蕴涵消除
3	$\neg\beta \vee \gamma$	3进行蕴涵消除
4	$\neg\gamma$	假设命题γ不成立
5	$\beta \vee \gamma$	1和2进行归结
6	$\neg\alpha$	2和4进行归结
7	$\neg\beta$	3和4进行归结
8	γ	5和7进行归结
9	假设不成立，命题γ成立	

【例2】

1	$\alpha \vee \beta$
2	$\neg\alpha \vee \beta$
3	$\alpha \vee \neg\beta$
4	$\neg\alpha \vee \neg\beta$

例2.4证明如上命题集是不可满足的

1	$\alpha \vee \beta$	已知
2	$\neg\alpha \vee \beta$	已知
3	$\alpha \vee \neg\beta$	已知
4	$\neg\alpha \vee \neg\beta$	已知
5	β	1和2进行归结
6	$\neg\beta$	3和4进行归结
因此，从该命题集中同时推出命题β和命题$\neg\beta$，因此原命题集是不可满足的		

【例3】

1	$\alpha \vee \gamma$
2	$\neg\beta \vee \gamma$
3	$\neg\gamma \vee \alpha$
4	$\neg\alpha \vee \beta$
5	$\neg\alpha \vee \neg\gamma$

例2.5证明如上命题集是不可满足的

1	$\alpha \vee \gamma$	已知
2	$\neg\beta \vee \gamma$	已知
3	$\neg\gamma \vee \alpha$	已知
4	$\neg\alpha \vee \beta$	已知
5	$\neg\alpha \vee \neg\gamma$	已知
6	α	1和3进行归结
7	β	4和6进行归结
8	γ	2和7进行归结
9	$\neg\alpha$	5和8进行归结
从该命题集合中可同时推出α和$\neg\alpha$两个命题，因此原命题集合是不可满足的		

8.合取范式与析取范式

①合取范式：若干个析取式的交。

②析取范式：若干个合取式的并。

范式（normal form)是命题逻辑中的重要概念。范式是把命题公式化归为一种标准的形式。范式最大的作用是可以进行两个命题的等价判定。

- 有限个简单合取式构成的析取式称为析取范式
- 由有限个简单析取式构成的合取式称为合取范式
- 析取范式与合取范式统称为范式 (normal form)

◆ 假设$\alpha_i(i=1,2,\dots,k)$为简单的合取式，则$\alpha=\alpha_1\vee\alpha_2\vee\cdots\vee\alpha_k$为析取范式

例如：$(\neg\alpha_1\wedge\alpha_2)\vee\alpha_3$, $\neg\alpha_1\vee\alpha_3\vee\alpha_2$ 等

◆ 假设$\alpha_i(i=1,2,\dots,k)$为简单的析取式，则$\alpha=\alpha_1\wedge\alpha_2\wedge\cdots\wedge\alpha_k$为合取范式

例如：$(\alpha_1\vee\alpha_2)\wedge\neg\alpha_3$, $\neg\alpha_1\wedge\alpha_3\wedge(\neg\alpha_2\vee\alpha_4)$等

9.合取范式与析取范式的性质

①一个析取范式是不成立的，当且仅当它的每个简单合取式都不成立。

②一个合取范式是成立的，当且仅当它的每个简单析取式都是成立的。

③任一命题公式都存在着与之等值的析取范式与合取范式(注意：命题公式的析取范式与合取范式都不是唯一的)

【例4】求解合取范式、析取范式。

【思路】首先把->化为标准的或、非形式，根据是否使用分配律，可以分别获得合取范式、析取范式。

例2.6：求$\neg(\alpha\to\beta)\vee\neg\gamma$的析取范式与合取范式

$$
\begin{aligned}
&\neg(\alpha\to\beta)\vee\neg\gamma\\
\Leftrightarrow\ &\neg(\neg\alpha\vee\beta)\vee\neg\gamma\\
\Leftrightarrow\ &(\alpha\wedge\neg\beta)\vee\neg\gamma\ (\text{析取范式})\\
\Leftrightarrow\ &(\alpha\vee\neg\gamma)\wedge(\neg\beta\vee\neg\gamma)\ (\text{合取范式})
\end{aligned}
$$

二、谓词逻辑

1.在谓词逻辑中，将原子命题进一步细化，分解出个体、谓词和量词，来表达个体与总体的内在联系和数量关系，这就是谓词逻辑研究内容。

2.谓词逻辑中三个核心概念：

个体、谓词（predicate）和量词（quantifier）

【例5】事件符号化。

例 2.8 将下列客观事实符号化。

1. Richard是国王。
2. Lucy和Lily是姐妹。
3. 北京是中国的首都。

解 (1) $King$(Richard)。其中，Richard是一个个体常量，$King$是一个描述“国王”这个一元关系的谓词。

(2) $Sister$(Lucy, Lily)。其中，Lucy和Lily是两个个体常量，$Sister$是一个描述“姐妹”这个二元关系的谓词。

(3) $Capital$(北京, 中国)。其中，北京和中国是两个个体常量，$Capital$是一个描述“首都”这个二元关系的谓词。

虽然，使用个体和谓词的组合可以描述简单的客观事实。但是在某些复杂的情况下，需要对某一对象集合的共同属性进行描述。而对每个对象的属性特征进行一一列举太过冗余，量词的引入可以有效地解决这一问题。

3.全称量词、存在量词的组合关系

全称量词与存在量词之间的组合

- $\forall x P(x) \equiv \neg \exists x \neg P(x)$
- $\forall x \neg P(x) \equiv \neg \exists x P(x)$
- $\neg \forall x P(x) \equiv \exists x \neg P(x)$
- $\exists x P(x) \equiv \neg \forall x \neg P(x)$

4.约束变元与自由变元

①约束变元：在全称量词或存在量词的约束条件下的变量符号，称为约束变元。

②自由变元：不受全称量词或存在量词约束的变量符号，称为自由变元。

③自由变元的一些相关等式（不受全称、存在量词约束，可以从中分离）：

设$A(x)$是包含变元x的公式，B是不包含变元x的谓词公式，则如下逻辑等价关系成立：

$$(\forall x)(A(x) \vee B) \equiv (\forall x)A(x) \vee B$$
$$(\forall x)(A(x) \wedge B) \equiv (\forall x)A(x) \wedge B$$
$$(\exists x)(A(x) \vee B) \equiv (\exists x)A(x) \vee B$$
$$(\exists x)(A(x) \wedge B) \equiv (\exists x)A(x) \wedge B$$

5.量词与分配律

①在约束变元相同的情况下，全称量词对合取具有分配律。

②在约束变元相同的情况下，存在量词对析取具有分配律。

$$(\forall x)(A(x) \wedge B(x)) \equiv (\forall x)A(x) \wedge (\forall x)B(x)$$

$$(\exists x)(A(x) \vee B(x)) \equiv (\exists x)A(x) \vee (\exists x)B(x)$$

【例6】事件符号化。

例 2.9 使用全称量词或存在量词描述下列事实。

（1）所有的国王都是人。

（2）所有的国王都头戴皇冠。

解 (1) “所有的国王都是人”表示的含义为“对于所有的x，如果x是国王，那么x是人”，其符号化表示为$(\forall x)(King(x) \rightarrow Person(x))$。其中$x$是变量符号，由于$x$受到全称量词的约束，因此 x 是约束变元；$King(x)$ 是一个一元谓词，表示 x 是国王，$Person(x)$是一个一元谓词，表示x是人。

(2) “所有的国王都头戴皇冠”表示的含义为“对于所有的x，如果x是国王，那么x头戴皇冠”，符号化表示为$(\forall x)(King(x) \rightarrow Head_On(\text{Crown}, x))$。其中$x$是变量符号，由于$x$受到全称量词的约束，因此$x$是约束变元；Crown是一个常量符号，表示皇冠；$King(x)$是一个一元谓词，表示$x$是国王，$Head_On(\text{Crown}, x)$是一个二元谓词，表示$x$头戴皇冠。

6.量词位置互换

①当公式中存在多个量词时，若多个量词都是全称量词或者都是存在量词，则量词的位置可以互换；若多个量词中既有全称量词又有存在量词，则量词的位

置不可以随意互换。

②举例：

$$(\forall x)(\forall y)A(x,y) \Leftrightarrow (\forall y)(\forall x)A(x,y)$$

$$(\exists x)(\exists y)A(x,y) \Leftrightarrow (\exists y)(\exists x)A(x,y)$$

$$(\forall x)(\forall y)A(x,y) \Rightarrow (\exists y)(\forall x)A(x,y)$$

$$(\forall x)(\forall y)A(x,y) \Rightarrow (\exists x)(\forall y)A(x,y)$$

$$(\exists y)(\forall x)A(x,y) \Rightarrow (\forall x)(\exists y)A(x,y)$$

$$(\exists x)(\forall y)A(x,y) \Rightarrow (\forall y)(\exists x)A(x,y)$$

$$(\forall x)(\exists y)A(x,y) \Rightarrow (\exists y)(\exists x)A(x,y)$$

$$(\forall y)(\exists x)A(x,y) \Rightarrow (\exists x)(\exists y)A(x,y)$$

7.项与原子谓词公式

①项是描述对象的逻辑表达式，被递归地定义为：

(1)常量符号和变量符号是项；

(2)若$f(x1, x2, \cdots, xn)$是n元函数符号，$t1,t2, \cdots ,tn$是项，则$f(t1,t2, \cdots ,tn)$是项；

(3)有限次数地使用上述规则产生的符号串是项。

②原子谓词公式

若$P(x1, x2, \cdots, xn)$是n元谓词，$t1,t2, \cdots ,tn$是项，则称$P(t1,t2, \cdots ,tn)$是原子谓词公式，简称原子公式。

8.合式公式

合式公式是由逻辑联结词和原子公式构成的用于陈述事实的复杂语句，又称谓词公式，由以下规则定义：

(1)命题常项、命题变项、原子谓词公式是合式公式；

(2)如果A是合式公式，则$\neg A$也是合式公式；

(3)如果A和B是合式公式，则$A \wedge B$、$A \vee B$、$A \rightarrow B$、$B \rightarrow A$、$A \leftrightarrow B$都是合式公式；

(4)如果A是合式公式，x是个体变项，则$(\exists x)A(x)$和$(\forall x)A\ x$也是合式公式；

(5)有限次数地使用上述规则构成的表达式是合式公式。

【例7】用合式公式表达事件。

例 2.10 使用合式公式描述下列事实。

（1）Tom不仅喜欢踢足球，还喜欢打篮球；（2）Tom的所有同学都喜欢他；（3）不是所有的男生都喜欢打篮球；（4）一个人是华侨，当且仅当他在国外定居并且具有中国国籍；（5）男生都爱看世界杯。

解 (1) $Like(\text{Tom}, \text{Football}) \wedge Like(\text{Tom}, \text{Basketball})$。其中，$Like(\text{Tom}, x)$表示Tom喜欢$x$，Football和Basketball分别表示踢足球和打篮球。

(2) $(\forall x)(Classmate(\text{Tom}, x) \rightarrow Like(x, \text{Tom}))$。其中，$Classmate(\text{Tom}, x)$表示$x$是Tom的同学，$Like(x, \text{Tom})$表示$x$喜欢Tom。

(3) $\neg(\forall x)(Boy(x) \rightarrow Like(x, \text{Basketball}))$。其中，$Boy(x)$表示$x$是男生，$Like(x, \text{Basketball})$表示$x$喜欢打篮球。

(4) $(\forall x)(Overseas_Chinese(x) \leftrightarrow (Overseas(x) \wedge Chinese(x)))$。其中，$Overseas_Chinese(x)$表示$x$是华侨，$Overseas(x)$表示$x$在国外定居，$Chinese(x)$表示$x$具有中国国籍。

(5) $(\forall x)(Boy(x) \rightarrow Like(x, \text{WorldCup}))$。其中，$Boy(x)$表示$x$是男生，$Like(x, \text{WorldCup})$表示$x$喜欢看世界杯。

【解答题考点】谓词逻辑与推理

9.全称量词、存在量词的消去与引入

设$A\ x$是谓词公式，x和y是变元，a是常量符号，则存在如下谓词逻辑中的推理规则：

①全称量词消去(Universal Instantiation, UI):$(\forall x)A\ x \rightarrow A(y)$

②全称量词引入(Universal Generalization, UG):$A\ y \rightarrow (\forall x)A\ x$

③存在量词消去(Existential Instantiation, EI):$(\exists x)A(x) \rightarrow A(c)$

④存在量词引入(Existential Generalization, EG):$A(a) \rightarrow (\exists x)A(x)$

【例8】全称量词消去与引入

例2.11 已知：

- $(\forall x)(P(x) \to Q(x))$
- $(\forall x)(Q(x) \to R(x))$

试证明：$(\forall x)(P(x) \to R(x))$

证明过程：

- $(\forall x)(P(x) \to Q(x))$
- $P(x) \to Q(x)$（消去全称量词）
- $(\forall x)(Q(x) \to R(x))$
- $Q(x) \to R(x)$（消去全称量词）
- $P(x) \to R(x)$
- $(\forall x)(P(x) \to R(x))$（引入$x$）

【例9】综合例题

例2.12 已知：

- $(\forall x)(F(x) \to (G(x) \wedge H(x))$
- $(\exists x)(F(x) \wedge P(x))$

试证明：$(\exists x)\ (P(x) \wedge H(x))$

证明过程：

1. $(\forall x)(F(x) \to (G(x) \wedge H(x))$ (已知)
2. $(\exists x)(F(x) \wedge P(x))$ (已知)
3. $F(a) \wedge P(a)$ (2的EI)
4. $F(a) \to (G(a) \wedge H(a))$ (1的UI)
5. $F(a)$ (由3知)
6. $G(a) \wedge H(a)$ (4和5的假言推理)
7. $P(a)$ (由3知)
8. $H(a)$ (由6知)
9. $P(a) \wedge H(a)$(7和8的合取)
10. $(\exists x)\ (P(x) \wedge H(x))$ (9的EG)

10.自然语言的形式化

每一个奇数均存在一个大于它的奇数：

- $\mathrm{odd}(x)$: x是奇数
- $\mathrm{Great}(\mathrm{x},\mathrm{y})$: x 大于y
- $(\forall x)\ \Big(odd(x) \to (\exists y)\big(odd(y) \wedge Great(y,x)\big)\Big)$

【例10】量词与证明

例 2.13 证明苏格拉底三段论“所有人都是要死的，苏格拉底是人，所以苏格拉底是要死的”。

解 设$F(x)$：x是人，$G(x)$：x是要死的，a是苏格拉底。

前提：$(\forall x)(F(x) \to G(x))$，$F(a)$

结论：$G(a)$

证明 (1) $(\forall x)(F(x) \to G(x))$

(2) $F(a) \to G(a)$ (1的全称量词消去)

(3) $F(a)$

(4) $G(a)$ (2和3的假言推理)

【例11】自然语言形式化、量词与证明

例2.14 前提： 1) 每驾飞机或者停在地面或者飞在天空；2) 并非每驾飞机都飞在天空

结论： 有些飞机停在地面

形式化： plane(x): x是飞机；in_ground(x): x停在地面；on_fly(x): x飞在天空

已知：$(\forall x)(plane(x) \to in_ground(x) \vee \text{on_fly}(x))$，$(\neg\forall x)(plane(x) \to on_fly(x))$

请证明：$(\exists x)(plane(x) \wedge in_ground(x))$

证明：

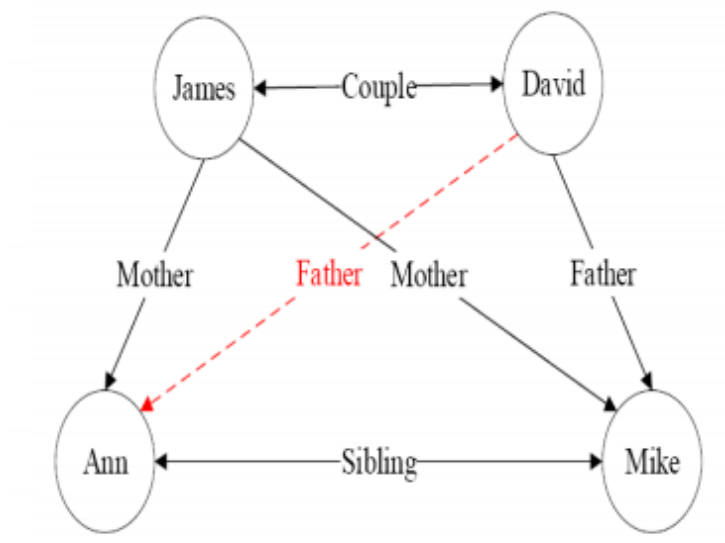
1. $(\neg\forall x)(plane(x) \to on_fly(x))$ (已知)
2. $(\exists x)\neg(plane(x) \to on_fly(x))$
3. $(\exists x)\neg(\neg plane(x) \vee on_fly(x))$
4. $(\exists x)(plane(x) \wedge \neg on_fly(x))$

1. $(\neg\forall x)(plane(x) \rightarrow on_fly(x))$
2. $(\exists x)\neg(plane(x) \rightarrow on_fly(x))$
3. $(\exists x)\neg(\neg plane(x) \vee on_fly(x))$
4. $(\exists x)(plane(x) \wedge \neg on_fly(x))$
5. $plane(a) \wedge \neg on_fly(a)$
6. $plane(a)$
7. $\neg on_fly(a)$
8. $(\forall x)(plane(x) \rightarrow in_ground(x) \vee \mathrm{on_fly}(x))$
9. $plane(a) \rightarrow in_ground(a) \vee \mathrm{on_fly}(a)$
10. $in_ground(a) \vee \mathrm{on_fly}(a)$
11. $in_ground(a)$
12. $plane(a) \wedge in_ground(a)$
13. $(\exists x)(plane(x) \wedge in_ground(x))$

三、知识图谱推理

1.知识图谱可视为包含多种关系的图。在图中，每个节点是一个实体（如人名、地名、事件和活动等），任意两个节点之间的边表示这两个节点之间存在的关系。

2.一般而言，可将知识图谱中任意两个相连节点及其连接边表示成一个三元组（*triplet*），即(left_*node*,*relation*,*right_node*)，例：(David,*Father*,*Mike*)。



3.知识图谱推理：

①知识图谱中存在连线的两个实体可表达

为形如<left_node, relation, right_node >的三元组形式，这种三元组也可以表示为一阶逻辑(first order logic, FOL)的形式，从而为基于知识图谱的推理创造了条件。

②例如：从<奥巴马，出生地，夏威夷>和<夏威夷，属于，美国>两个三元组，可推理得到<奥巴马，国籍，美国>。

③可利用一阶谓词来表达刻画知识图谱中节点之间存的关系，如图中形如<James,$Couple$,David>的关系可用一阶逻辑的形式来描述，即$Couple$(James, David)。

④$Couple(x, y)$是一阶谓词，$Couple$是图中实

体之间具有的关系，x和y是谓词变量

⑤从图中已有关系可推知David和Ann具有父

女关系，但这一关系在图中初始图(无红线)

中并不存在，是需要推理的目标。

4.从具体知识图谱中学习推理规则——归纳学习

归纳逻辑程序设计(inductive logic programming，ILP)算法，使用一阶谓词逻辑进行知识表示，通过修改和扩充逻辑表达式对现有知识归纳，完成推理任务。

5.**FOIL**(First Order Inductive Learner)算法

①推理思路：逐步给目标谓词添加前提约束谓词，直到所构成的推理规则不覆盖任何反例。

推理思路：从一般到特殊，逐步给目标谓词添加前提约束谓词，直到所构成的推理规则不覆盖任何反例。

从一般到特殊：对目标谓词或前提约束谓词中的变量赋予具体值，如将$(\forall x)(\forall y)(\forall z)\big(Mother(z,y) \wedge Couple(x,z) \rightarrow Father(x,y)\big)$这一推理规则所包含的目标谓词$Father(x,y)$中$x$和$y$分别赋值为David和Ann，进而进行推理。

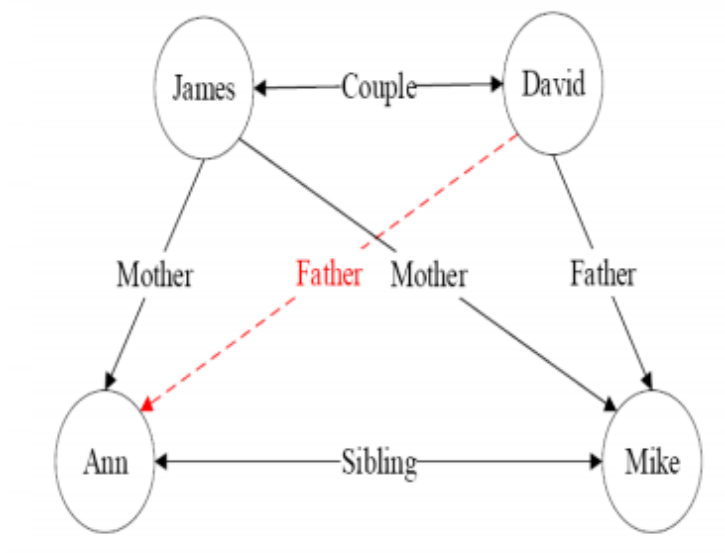
②FOIL信息增益值计算：

FOIL信息增益值计算方法如下：

$$FOIL_Gain = \widehat{m_+} \cdot \left(\log_2 \frac{\widehat{m_+}}{\widehat{m_+} + \widehat{m_-}} - \log_2 \frac{m_+}{m_+ + m_-} \right)$$

其中，$\widehat{m_+}$和$\widehat{m_-}$是增加前提约束谓词后所得新推理规则覆盖的正例和反例的数量，m_+和m_-是原推理规则所覆盖的正例和反例数量。

6.FOIL算法举例与分析



（1）前提与目标：根据知识图谱中的已知信息，学习得出如下隐藏信息。

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) ¬Father(David, James) ¬Father(James, Ann) ¬Father(James, Mike) ¬Father(Ann, Mike)

（2）从空集到第一条前提约束谓词

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x,y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL_Gain$
$Father(x,y) \leftarrow$	$Mother(x,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(x,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(y,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(y,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z,y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 3$	0.32
	$Sibling(x,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(x,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(y,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(y,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z,y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74
	$Couple(x,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$\boldsymbol{Couple(x,z)}$	$\boldsymbol{\widehat{m}_+ = 1}$	$\boldsymbol{\widehat{m}_- = 1}$	1.32
	$Couple(y,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(y,z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(z,x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(z,y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA

【理解】①上述表格中，如果一个正例都没有，那么信息增益为负无穷，记作NA.

②起始状态没有任何约束谓词，此时的正例、反例个数，分别是Father(x,y)和not Father(x,y)为真的个数，即目标谓词训练样例集合中，正反例的个数。根据下图可知，正例、反例数目分别是1和4.

目标谓词 训练样例集合	Father(David, Mike) ¬Father(David, James) ¬Father(James, Ann) ¬Father(James, Mike) ¬Father(Ann, Mike)

③Mother(x,y)：图谱中有2个实例。

对于Mother(James,Ann),Father(James,Ann)是反例。

对于Mother(James,Mike),Father(James,Mike)是反例。

所以有0个整理，2个反例。计算得到的信息增益为NA。

④Couple(x,z)：图谱中有1个实例。

对于Couple(David,James): x = David, z = James.

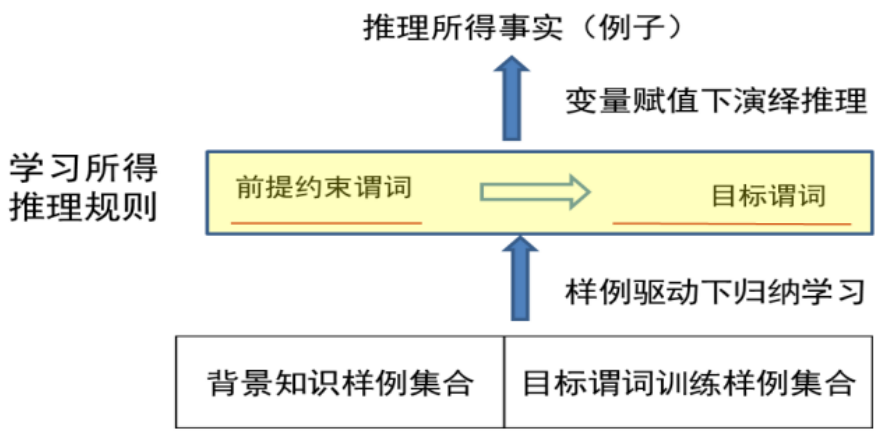
在目标谓词训练样例集合中，x = David 对应的是前两个实例，其中1个正例，1个反例。从而可以计算信息增益为1.32.

⑤类似地，可以计算其他所有前提约束谓词带来的信息增益。

⑥选择带来信息增益最大的前提约束谓词，加入推理规则。

⑦迭代上述步骤，不断选择带来信息增益最大的前提约束谓词，加入推理规则，直到没有任何反例出现，迭代终止。

7.FOIL算法总结：给定目标谓词，FOIL算法从实例（正例、反例、背景知识样例）出发，不断测试所得推理规则是否还包含反例，一旦不包含，则学习结束，由此充分展示了“归纳学习”的能力。在学得推理规则后，再给推理规则中的变量赋予具体例子，经过“演绎”得到新的知识。



8.路径排序推理——算法

(1)特征抽取：生成并选择路径特征集合。生成路径的方式有随机游走（random walk）、广度优先搜索、深度优先搜索等。

(2)特征计算：计算每个训练样例的特征值$P(s \rightarrow t; \pi j)$。该特征值可以表示从实体节点$s$出发，通过关系路径$\pi j$到达实体节点$t$的概率；也可以表示为布尔值，表示实体$s$

到实体t之间是否存在路径πj；还可以是实体s和实体t之间路径出现频次、频率等。

(3)分类器训练：根据训练样例的特征值，为目标关系训练分类器。当训练好分类器后，即可将该分类器用于推理两个实体之间是否存在目标关系。

9.路径排序推理——举例

例 2.16 对图2.1中的知识图谱采用路径排序算法。

(1) 目标关系：$Father$

(2) 对于目标关系$Father$，生成四组训练样例，一个为正例、三个为负例：

正例：(David, Mike)

负例：(David, James)，(James, Ann)，(James, Mike)

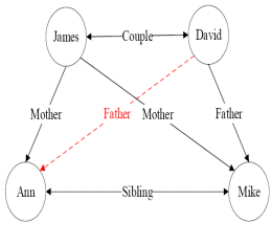
(3) 从知识图谱采样得到路径，每一路径链接上述每个训练样例中两个实体：

(David, Mike)对应路径：$Couple \rightarrow Mother$

(David, James)对应路径：$Father \rightarrow Mother^{-1}$（$Mother^{-1}$与$Mother$为相反关系）

(James, Ann)对应路径：$Mother \rightarrow Sibling$

(James, Mike)对应路径：$Couple \rightarrow Father$



(4) 对于每一个正例/负例，判断上述四条路径可否链接其包含的两个实体，将可链接（记为1）和不可链接（记为0）作为特征，于是每一个正例/负例得到一个四维特征向量：

(David, Mike)：$\{[1,0,0,0],1\}$

(David, James)：$\{[0,1,0,0],-1\}$

(James, Ann)：$\{[0,0,1,0],-1\}$

(James, Mike)：$\{[0,0,1,1],-1\}$（$Mother \rightarrow Sibling$ 和$Couple \rightarrow Father$ 两条路径均包含了James和Mike）

(5) 依据(4)中的训练样本，训练分类器M。

(6) 预测。对于图2.1中形如(David, Ann)的样例，得到其特征值为$[1,0,0,0]$（$Couple \rightarrow Mother$ 这一条路径包含了David和Ann），将特征向量输入到分类器M中，如果分类器M给出分类结果为1，则$Father$(David, Ann)成立。

10.**基于分布式表示的知识推理**

①随着深度学习应用的进展，在自然语言处理（Natural Language Processing，NLP）中，通过无监督学习将一个单词表示成一个隐向量(latent vector)的分布式表示方法取得了良好效果。研究者将知识图谱中的节点和节点之间的关系均映射到连续数值空间中（即分布式表示），同时对映射过程进行约束，使得知识图谱中节点之间的拓扑结构得以保留，从而有效地对知识图谱进行表示和推理。

②如给定2.1中$Father$(David, Mike)这一个样例知识，可首先学得David和Mike的分布式表达（向量表达）**David**和**Mike**以及谓词$Father$的分布式表达$\boldsymbol{Father}$，然后再学习一个度量函数ψ使得**David+**$\boldsymbol{Father}$的结果与**Mike**很接近，即ψ(**david** + **Father**, **Mike**)取值最小（也即**david** + **Father** ≈ **Mike**）。于是，给定所有

<left_node，relation，right_node >形式的样例知识，就可去学习度量函数ψ，使得ψ(**left_node**, **relation**, **right_node**)取值最小。

③一旦学习得到度量函数ψ，就可用其去判断任意两个节点之间是否具有某一关系。

第二章 MOOC习题

1.如果命题p为真、命题q为假，则下述哪个复合命题为真命题（B）

A.如果p则q

B.如果q则p

C.非p

D.p且q

2.p和q均是原子命题，“如果p那么q”是由p和q组合得到的复合命题。下面对“如果p那么q”这一复合命题描述不正确的是（ D ）

A.“如果p那么q”意味着命题q包含着命题p，即p是q的子集

B.当p不成立时，“如果p那么q”恒为真

C.“如果p那么q”定义的是一种蕴涵关系（即充分条件）

D.无法用真值表来判断“如果p那么q”的真假

3.

下面哪个复合命题与“如果秋天天气变凉，那么大雁南飞越冬”是逻辑等价的

（ A ）

A.如果大雁不南飞越冬，那么秋天天气没有变凉

B.如果大雁不南飞越冬，那么秋天天气变凉

C.如果秋天天气没有变凉，那么大雁不南飞越冬

D.如果秋天天气变凉，那么大雁不南飞越冬

4.下面哪一句话对命题逻辑中的归结(resolution)规则的描述是不正确的（ C ）

A.对命题q及其反命题应用归结法，所得到的命题为空命题(非真非假)

B.在两个析取复合命题中，如果命题q及其反命题分别出现在这两个析取复合命题中，则通过归结法可得到一个新的析取复合命题，只是在析取复合命题中要去除命题q及其反命题。

C.对命题q及其反命题应用归结法，所得到的命题为假命题

D.如果命题q出现在一个析取复合命题中，命题q的反命题单独存在，则通过归结法可得到一个新的析取复合命题，只是在析取复合命题中要去除命题q及其反命题。

5.下面哪一句话对命题范式的描述是不正确的（ B ）

A.有限个简单析取式构成的合取式称为合取范式

B.一个析取范式是不成立的，当且仅当它包含一个不成立的简单合取式

C.有限个简单合取式构成的析取式称为析取范式

D.一个合取范式是成立的，当且仅当它的每个简单析取式都是成立的

8.知识图谱可视为包含多种关系的图。在图中，每个节点是一个实体（如人名、地名、事件和活动等），任意两个节点之间的边表示这两个节点之间存在的关系。下面对知识图谱的描述，哪一句话的描述不正确（C）

A.知识图谱中的节点可以是实体或概念

B.知识图谱中一条边连接了两个节点，可以用来表示这两个节点存在某一关系

C.知识图谱中两个节点之间仅能存在一条边(错误，可以有不止一条边，代表同时具有多种关系，比如brother&classmate)

D.知识图谱中一条边可以用一个三元组来表示

9.如果知识图谱中有David和Mike两个节点，他们之间具有classmate和brother关系。在知识图谱中还存在其他丰富节点和丰富关系（如couple, parent等）前提下，下面描述不正确的是（ C ）

A.可以从知识图谱中形成classmate<David, Mike>的表达，这里classmate是谓词。

B.可以从知识图谱中形成brother<David, Mike>的表达，这里brother是谓词。

C.仅可从知识图谱中找到classmate和brother这个谓词的正例，无法找到这两个谓词的反例。

D.可从知识图谱中找到classmate和brother这个谓词的正例和反例。

10.下面对一阶归纳推理（FOIL）中信息增益值(information gain)阐释不正确的是（ C ）

A.在算法结束前，每次向推理规则中加入一个前提约束谓词，该前提约束谓词得到的新推理规则具有最大的信息增益值。

B.信息增益值用来判断向推理规则中所加入前提约束谓词的质量。

C.信息增益值大小与背景知识样例数目有关。(错误)

D.在计算信息增益值过程中，需要利用所得到的新推理规则和旧推理规则分别涵盖的正例和反例数目。

第三章 搜索求解

一、搜索算法基础

1.搜索树：用一棵树来记录算法探索过的路径

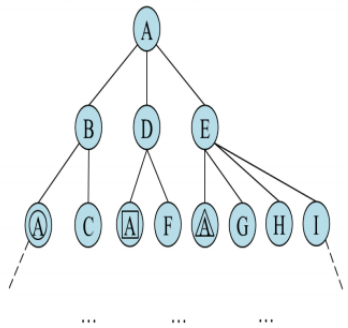


图3.2 搜索树示例（对应图3.1中路径问题。搜索树第三层中标号为*A*的三个节点由于具有从初始结点到达其的不同路径而属于不同结点）

- 搜索树第三层中有三个标号均为A的结点，它们在图中分别被圆圈、正方形和三角形框住。虽然这三个标号均为A的结点对应同一个城市，即三个标号为A的结点所对应状态相同，但是这三个节点在搜索树中却是不同结点，因为它们分别代表了从初始状态出发到达城市 A 的三条不同路径。
- 这三个结点表示的路径分别为：A → B → A、A → D → A和A → E → A。因此需要注意的是，在搜索树中，同一个标号一定表示相同的状态，其含义为智能体当前所在的城市，但是一个标号可能有多个的结点与之对应，不同的结点对应了从初始状态出发的不同路径。
- 搜索算法可以被看成是一个构建搜索树的过程，从根结点（初始状态）开始，不断展开每个结点的后继结点，直到某个结点通过了目标测试。

2.搜索算法可以被看成是一个构建搜索树的过程，根结点（初始状态）开始，不断展开每个结点的后继结点，直到某个结点通过了目标测试。

3.搜索算法的评价指标

完备性	当问题存在解时，算法是否能保证找到一个解。
最优性	搜索算法是否能保证找到的第一个解是最优解。
时间复杂度	找到一个解所需时间。
空间复杂度	在算法的运行过程中需要消耗的内存量。

完备性和最优性刻画了算法找到解的能力以及所求的解的质量，时间复杂度和空间复杂度衡量了算法的资源消耗，它们通常用O符号（big O notation）来描述。

表 3.1 搜索树中用于估计复杂度的变量含义

符号	含义
b	分支因子，即搜索树中每个节点最大的分支数目
d	根节点到最浅的目标结点的路径长度
m	搜索树中路径的最大可能长度
n	状态空间中状态的数量

4.树搜索的一般算法流程

函数： TreeSearch

输入： 节点选择函数 pick_from，后继节点计算函数 successor_nodes

输出： 从初始状态到终止状态的路径

```
F ← {根节点}
while F ≠ ∅ do
    n ← pick_from(F)
    F ← F − {n}
    if goal_test(n) then
        return n.path
    end
    F ← F ∪ successor_nodes(n)
end
```

集合$\mathcal{F}$用于保存搜索树中可用于下一步探索的所有候选结点，这个集合被称为**边缘（fringe）集合**，有时也被叫做**开表（open list）**。

5.剪枝搜索：在某些情况下，主动放弃一些后继结点能够提高搜索效率而不会影响最终搜索结果，甚至能解决无限循环（即算法不停机）问题。

6.图搜索——不允许环路存在，对所有环路节点进行剪枝。

【理解】某个结点被搜索过后，它的状态就放入了集合C。再次遇到它的时候，由于它的状态在C中，所以判定扩展该结点会产生环路，故剪枝。

函数： GraphSearch

输入： 节点选择函数 pick_from，后继节点计算函数 successor_nodes

输出： 从初始状态到终止状态的路径

```
F ← {根节点}
C ← ∅
while F ≠ ∅ do
    n ← pick_from(F)
    F ← F − {n}
    if goal_test(n) then
        return n.path
    end
    if n.state ∉ C then
        C ← C ∪ {n.state}
        F ← F ∪ successor_nodes(n)
    end
end
```

删除引起
环路的节点

二、启发式搜索（有信息搜索）

1.【填空题】启发式搜索的代表算法有：贪婪最佳优先搜索、A*搜索。

2.辅助信息——评价函数与启发函数

①评价函数：选择下一个扩展的节点。

②启发函数：计算完成任务还需要多少代价。

辅助信息	所求解问题之外、与所求解问题相关的特定信息或知识。	
评价函数（evaluation function）f(n)	从当前节点n出发，根据评价函数来选择后续结点。	下一个结点是谁?
启发函数（heuristic function）h(n)	计算从结点n到目标结点之间所形成路径的最小代价值，这里将两点之间的直线距离作为启发函数。	完成任务还需要多少代价?

3.贪婪最佳优先搜索

①评价函数=启发函数

②思想：优先扩展距离近（启发函数取值小）的结点。

③具有完备性，但是不具有最优性。

④问题：总是扩展与目标结点之间代价最小的结点，而不考虑从初始结点到达该结点所需付出的代价。

4.贪婪最佳优先搜索-举例

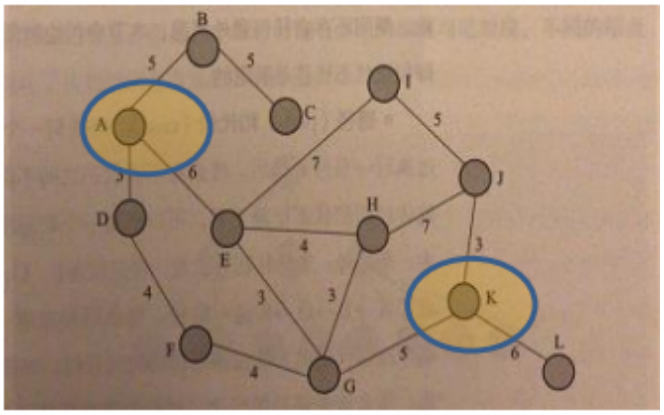
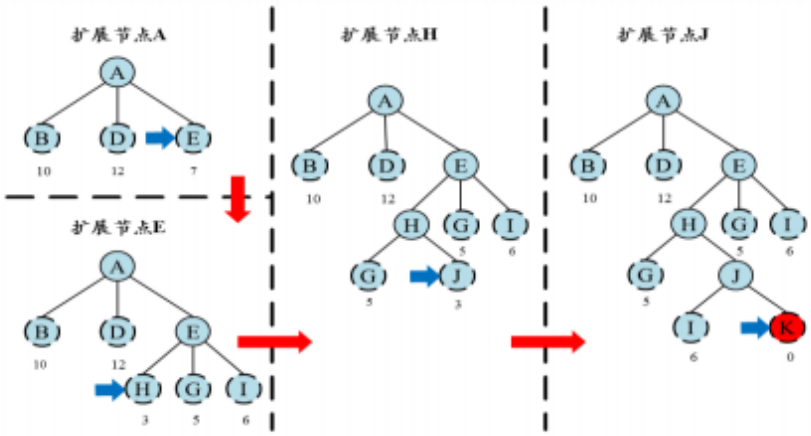


表3.2 每个状态（城市）所对应启发函数取值

状态	A	B	C	D	E	F	G	H	I	J	K	L
h(n)	13	10	6	12	7	8	5	3	6	3	0	6

辅助信息：任意一个城市与终点城市K之间的直线距离



①如图，各个结点标注的数字，代表当前结点到目标结点的距离。已扩展的结点，不进行标注；未扩展的结点，才标明它们到目标结点的

距离。

②每一步都是选择距离目标最近，且未被扩展的结点进行扩展，故选择序列为E、H、J、K.

③如此得到的路径是A->E->H->J->K，但是最短路径是A->E->G->K.

5.A*搜索

①评价函数 = 从起始结点到当前结点的实际最小代价(已知)+从当前结点到目标结点的后续估计最小代价(估计)

评价函数：$f(n) = g(n) + h(n)$

- $g(n)$表示从起始结点到结点n的开销代价值，$h(n)$表示从结点n到目标结点路径中所估算的最小开销代价值。
- $f(n)$可视为经过结点n、具有最小开销代价值的路径。

$$\underbrace{f(n)}_{\text{评价函数}} = \underbrace{g(n)}_{\substack{\text{起始结点到结点}n\text{代价}\\ \text{(当前最小代价)}}} + \underbrace{h(n)}_{\substack{\text{结点}n\text{到目标结点代价}\\ \text{(后续估计最小代价)}}}$$

6.A*搜索-举例

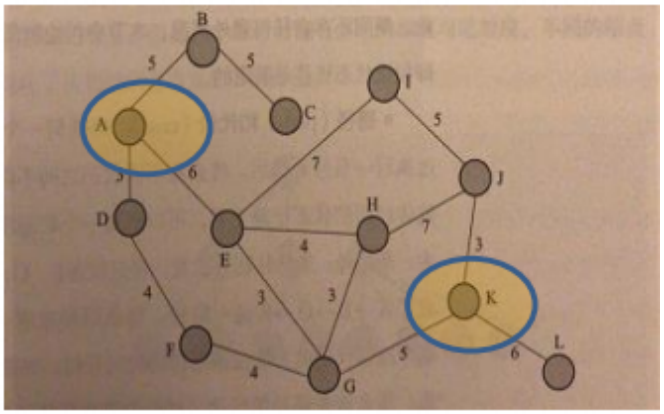
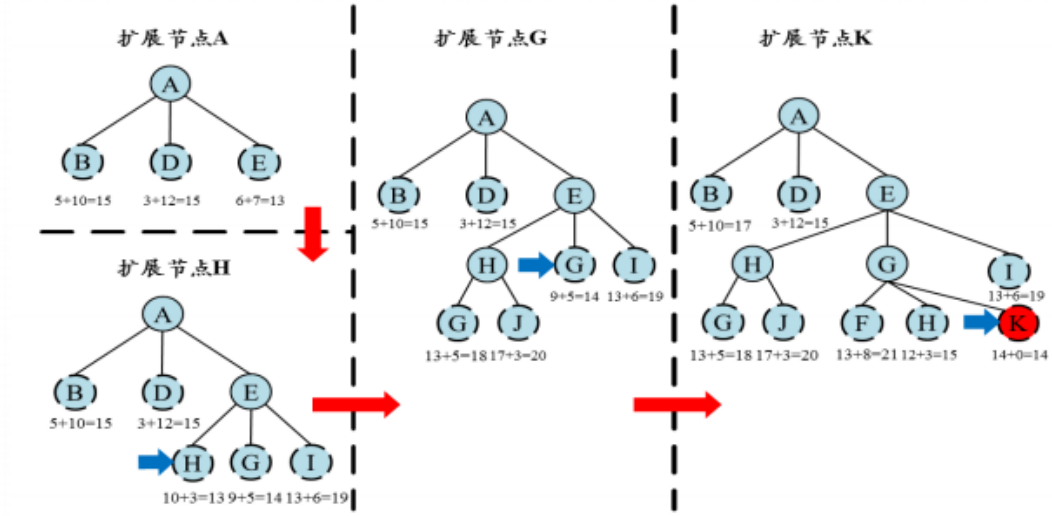


表3.2 每个状态（城市）所对应启发函数取值

状态	A	B	C	D	E	F	G	H	I	J	K	L
h(n)	13	10	6	12	7	8	5	3	6	3	0	6

辅助信息：任意一个城市与终点城市K之间的直线距离



①如图所示：评价函数 = 启发函数（本例中为当前结点到目标结点的直线距离）+从起点到当前结点所需的最小代价（已知）

②A*算法选择评价函数值最小的结点进行扩展，故选择的结点序列是：E->H->G->K。

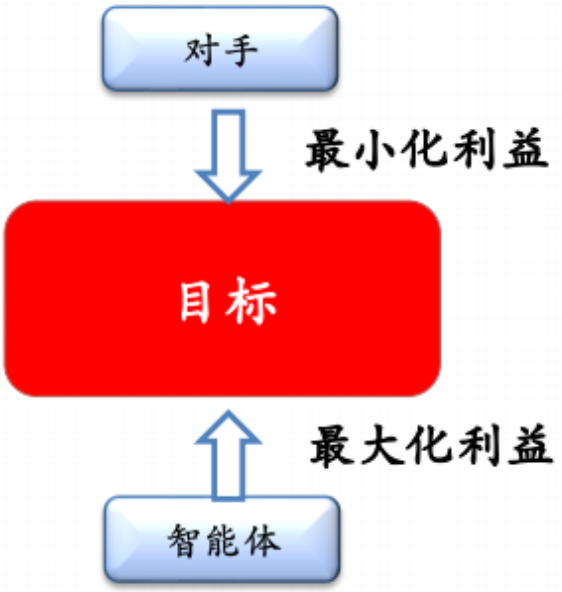
③A*算法具有可容性、完备性、一致性、最优性。

三、对抗搜索

1.【填空题】常见的对抗搜索有：最小最大搜索、Alpha-Beta剪枝搜索、蒙特卡洛树搜索。

2.对抗搜索(Adversarial Search)也称为博弈搜索(Game Search)。在一个竞争的环境中，智能体(agents)之间通过竞争实现相反的利益，

一方最大化这个利益，另外一方最小化这个利益。



3.三种对抗搜索方法简介：

■**最小最大搜索(Minimax Search):**最小最大搜索是在对抗搜索中最为基本的

一种让玩家来计算最优策略的方法.

■**Alpha-Beta剪枝搜索(Pruning Search)**:一种对最小最大搜索进行改进的算

法，即在搜索过程中可剪除无需搜索的分支节点，且不影响搜索结果。.

■**蒙特卡洛树搜索(Monte-Carlo Tree Search)**:通过采样而非穷举方法来实现

搜索

4.对抗搜索基本概念

- 本书主要讨论在确定的、全局可观察的、竞争对手轮流行动、零和游戏（zero-sum）下的对抗搜索
- 两人对决游戏 (MAX and MIN, MAX先走) 可如下形式化描述，从而将其转换为对抗搜索问题

状态	状态s包括当前的游戏局面和当前行动的智能体。函数$player(s)$给出状态s下行动的智能体。
动作	给定状态s，动作指的是player(s)在当前局面下可以采取的操作a，记动作集合为actions(s)。
状态转移	给定状态s和动作$a \in actions(s)$，状态转移函数result(s,a)决定了在s状态采取a动作后所得后继状态。
终局状态检测	终止状态检测函数terminal_test(s)用于测试游戏是否在状态s结束。
终局得分	终局得分utility(s,p)表示在终局状态s时玩家p的得分。

注：所谓零和博弈是博弈论的一个概念，属非合作博弈。指参与博弈的各方，在严格竞争下，一方的收益必然意味着另一方的损失，博弈各方的收益和损失相加总和永远为“零”，双方不存在合作的可能。与“零和”对应，“双赢博弈”的基本理论就是“利己”不“损人”，通过谈判、合作达到皆大欢喜的结果。

5.最小最大搜索

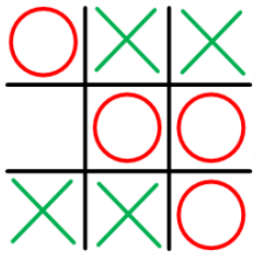
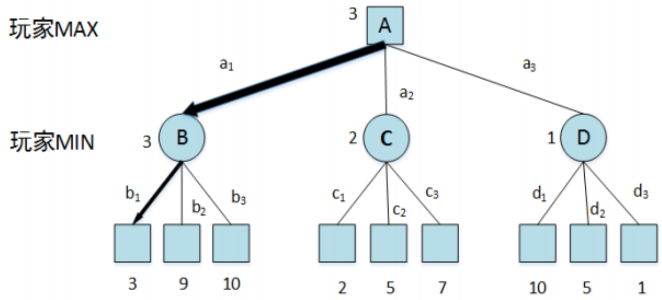


图 3.9 井字棋的盘面，此时执“○”棋子的玩家获胜



$$\text{minimax}(s) = \begin{cases} \text{utility}(s), & \text{if terminal_test}(s) \\ \max_{a \in \text{actions}(s)} \text{minimax}(\text{result}(s,a)), & \text{if player}(s) = MAX \\ \min_{a \in \text{actions}(s)} \text{minimax}(\text{result}(s,a)), & \text{if player}(s) = MIN \end{cases}$$

如图所示：MAX玩家选择下一层结点中收益最大者，即B结点；然后MIN玩家选择下一层结点中收益最小者，即收益为3的方形结点。

函数：MinimaxDecision

输入： 当前的盘面状态s

输出： 玩家 MAX 行动下，当前最优动作a^*

```
a* ← arg max_{a∈action(s)} MinValue(result(s, a))
```

函数：MaxValue

输入： 当前的盘面状态s

输出： 玩家 MAX 行动下，当前状态的得分v = minimax(s, MAX)

```
if terminal_test(s) then return utility(s)
v ← −∞
foreach a ∈ action(s) do
    v ← max(v, MinValue(result(s, a)))
end
```

函数：MinValue

输入： 当前的盘面状态s

输出： 玩家 MIN 行动下，当前状态的得分v = minimax(s, MIN)

```
if terminal_test(s) then return utility(s)
v ← +∞
foreach a ∈ action(s) do
    v ← min(v, MaxValue(result(s, a)))
end
```

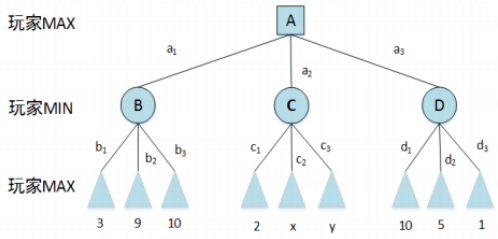
6.Alpha-Beta剪枝搜索

①对最小最大搜索的改进，在剪枝同时，不影响算法性能。

②剪枝举例

Alpha-Beta 剪枝搜索算法在Minimax算法中可减少被搜索的结点数，即在保证得到与原Minimax算法同样的搜索结果时，剪去了不影响最终结果的搜索分枝。

$$\begin{aligned}&\text{minimax}(A)\\&=\max(\min(3,9,10),\min(2,x,y),\min(10,5,1))\\&=\max(3,\min(2,x,y),1)\end{aligned}$$



由于min(2, x, y)的值小于等于2，所以即使不知道x和y的值，根结点的分数也可以算出来等于3。也就是说，算法没有必要计算动作c_2和c_3对应的两个子树在游戏结束时所得分数，就能决定在根结点采取动作a_1从而在游戏结束时可获得收益3，因此动作c_2和c_3所对应子树可以被剪枝。

图中MIN选手所在的结点C下属动作c_2和c_3所对应分支与根结点最终优化决策的取值无关，可不被访问。

剪枝结点：x,y。无需探索这两个结点，对最终结果不会产生影响。

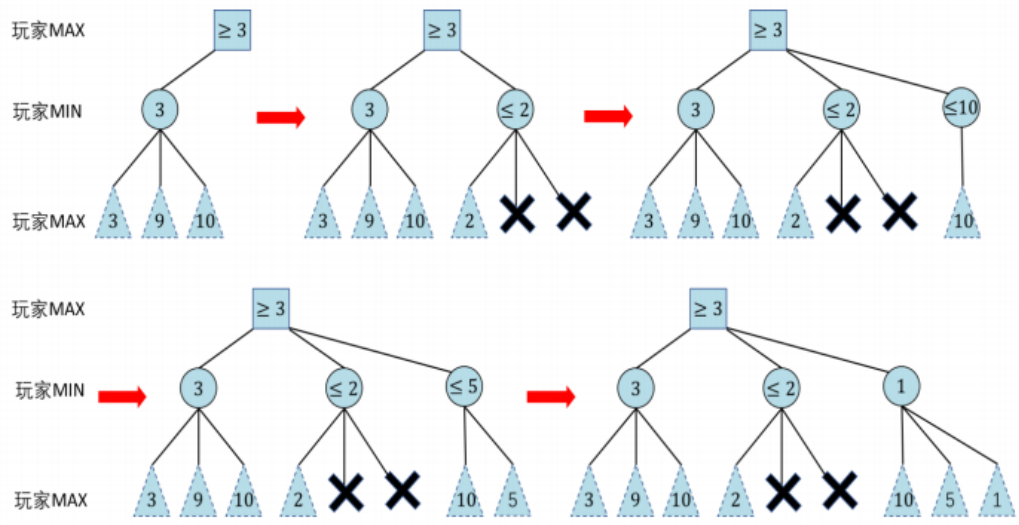
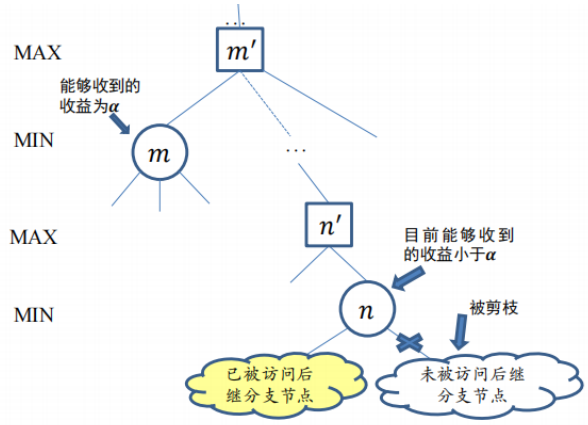


图 3.12 存在剪枝的搜索树部分扩展过程

③针对MIN层的剪枝原则：假设有一个位于MIN层的结点m，已知该结点能够向其上MAX结点反馈的收益为α（alpha）。n是与结点m位于同一层的某个兄弟（sibling）结点的后代结点。如果在结点n的后代结点被访问一部分后，知道结点n能够向其上一层MAX结点反馈收益小于α，则结点n的未被访问孩子结点将被剪枝。



假设有一个位于MIN层的结点m，已知该结点能够向其上MAX结点反馈的收益为α（alpha）。n是与结点m位于同一层的某个兄弟（sibling）结点的后代结点。如果在结点n的后代结点被访问一部分后，知道结点n能够向其上一层MAX结点反馈收益小于α，则结点n的未被访问孩子结点将被剪枝。

图3.14 基于MIN结点反馈收益进行剪枝(alpha剪枝)

④针对MAX层的剪枝原则：考虑位于MAX层的结点m，已知结点m能够从其下MIN层结点收到的收益为β（beta）。结点n是结点m上层结点m'的位于MAX层的后代结点，如果目前已知结点n能够收到的收益

大于β，则不再扩展结点n的未被访问后继结点，因为位于MIN层的结点m'只会选择收益小于或等于β的结点来采取行动。

⑤α和β的初始化与更新：对于MAX节点，如果其孩子结点（MIN结点）的收益大于当前的α值，则将α值更新为该收益；对于MIN结点，如果其孩子结点（MAX结点）的收益小于当前的β值，则将β值更新为该收益。根结点（MAX结点）的α值和β值分别被初始化为$-\infty$和$+\infty$。

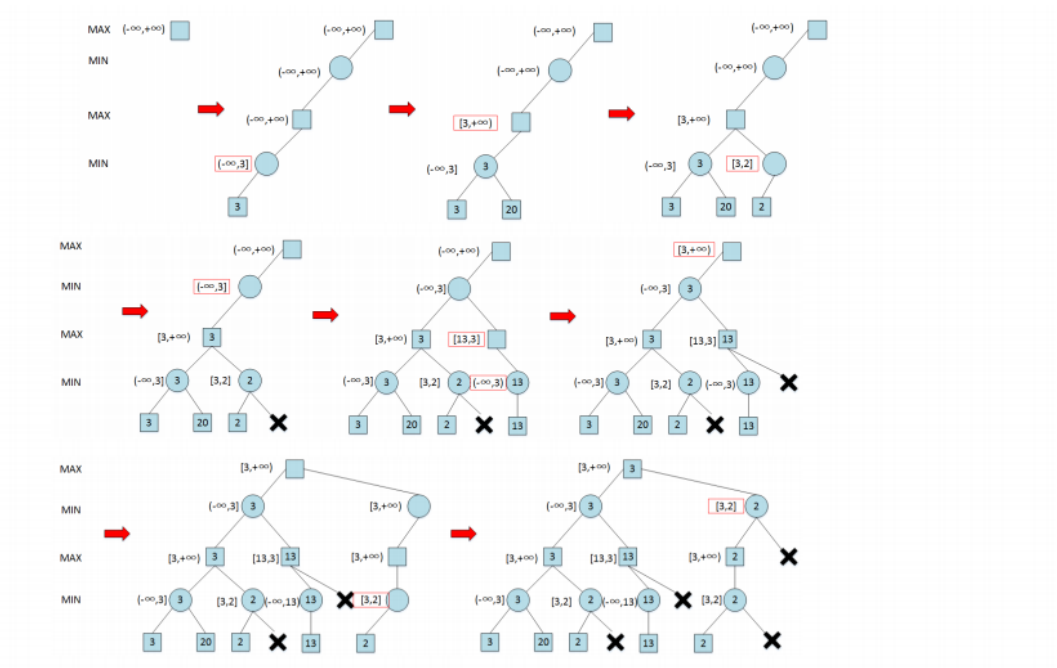
随着搜索算法不断被执行，每个结点的α值和β值不断被更新。大体来说，每个

结点的$[\alpha, \beta]$从其父结点提供的初始值开始，取值按照如下形式变化：α逐渐增

加、β逐渐减少。不难验证，如果一个结点的α值和β值满足$\alpha > \beta$的条件，则该

结点尚未被访问的后续结点就会被剪枝，因而不会被智能体访问。

⑥剪枝示意图：



⑦算法伪代码

函数：AlphaBetaDecision

输入： 当前的盘面状态s

输出： 玩家 MAX 行动下，当前最优动作a^*

```
1 v, a* ← MaxValue(s, −∞, +∞)
```

函数：MinValue

输入： 当前的盘面状态s，当前节点的下界α和上界β

输出： 玩家 MIN 行动下，当前状态的得分$v = \text{minimax}(s, \text{MIN})$和最优动作$a^*$

前置条件： $\alpha \le \beta$

```
1  if terminal_test(s) then return utility(s), null
2  v ← +∞
3  a* ← null
4  foreach a ∈ action(s) do
5      v', a' ← MaxValue(result(s, a), α, β)
6      if v' < v† then
7          v ← v'
8          a* ← a
9      end
10     β ← min(β, v)
11     if α ≥ β then return v, a*
12 end
```

注： †第 6 行和第 11 行中对分数相同情况的处理方法与正文中不同，此处只保留一个可行解，而放弃所有分数相同的解

7.蒙特卡洛树搜索

（1）**探索**（exploration）和**利用**（exploitation）之间存在对立关系。贪心

算法基本上是**利用**从已有尝试结果中所得估计来指导后续动作，但问题是所得估计往往不能准确反映未被（大量）探索过的动作。因此，需要在贪心算法中增加一个能够改变其“惯性”的内在动力，以使得贪心算法能够访问那些尚未被（充分）访问过的空间。

（2）ε-贪心算法

ϵ-贪心算法： 在探索与利用之间进行平衡的搜索算法。在第t步，ϵ-贪心算法按照如下机制来选择摇动赌博机：

$$l_t = \begin{cases} \text{argmax}_i\ \bar{x}_{i,T_{(i,t-1)}}, & \text{以}1-\epsilon\text{的概率} \\ \text{随机的}i \in \{1,2,\cdots,K\}, & \text{以}\epsilon\text{的概率} \end{cases}$$

即以$1-\epsilon$的概率选择在过去$t-1$次摇动赌博机臂膀行动中所得平均收益分数最高的赌博机进行摇动；以ϵ的概率随机选择一个赌博机进行摇动。

不足： 与被探索的次数无关。可能存在一个给出更好奖励期望的动作，但因为智能体对其探索次数少而认为其期望奖励小。因此，需要对那些探索次数少或几乎没有被探索过的动作赋予更高的优先级。没有将每个动作被探索的次数纳入考虑，这似乎是不合理的。另一方面，简单地优先去尝试探索次数少的动作也未必合理，因为只需要少量的探索次数，就能够较好估计方差较小的动作的奖励期望。

问题：①没有将每个动作被探索的次数纳入考虑，这似乎是不合理的。

②另一方面，简单地优先去尝试探索次数少的动作也未必合理，因为只需要少量的探索次数，就能够较好估计方差较小的动作的奖励期望。

（3）上限置信区间算法（UCB1）

①思想：为每个动作的奖励期望计算一个估计范围，优先采用估计范围上限较高的动作。

②举例：动作1的奖励期望取值的不确定度（估计范围）虽然最大，但是因为其均值太小，因此UCB1算法不优先考虑探索动作1。动作2和3的奖励期望的均值相同，但是动作2的奖励期望取值的不确定度（估计范围）更大，于是因为置信上限更大，动作2会被UCB1算法优先考虑。

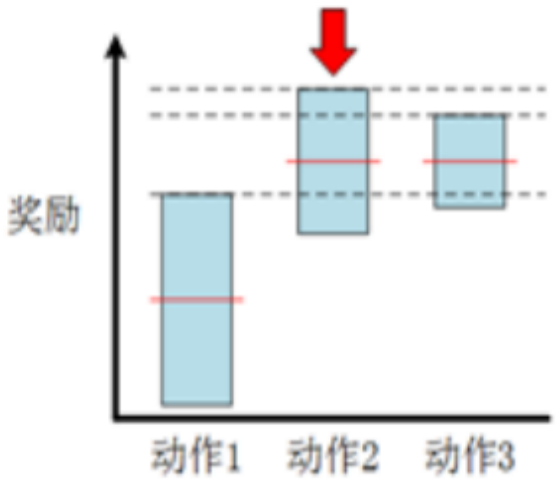


图 3.23 UCB1算法策略示意图

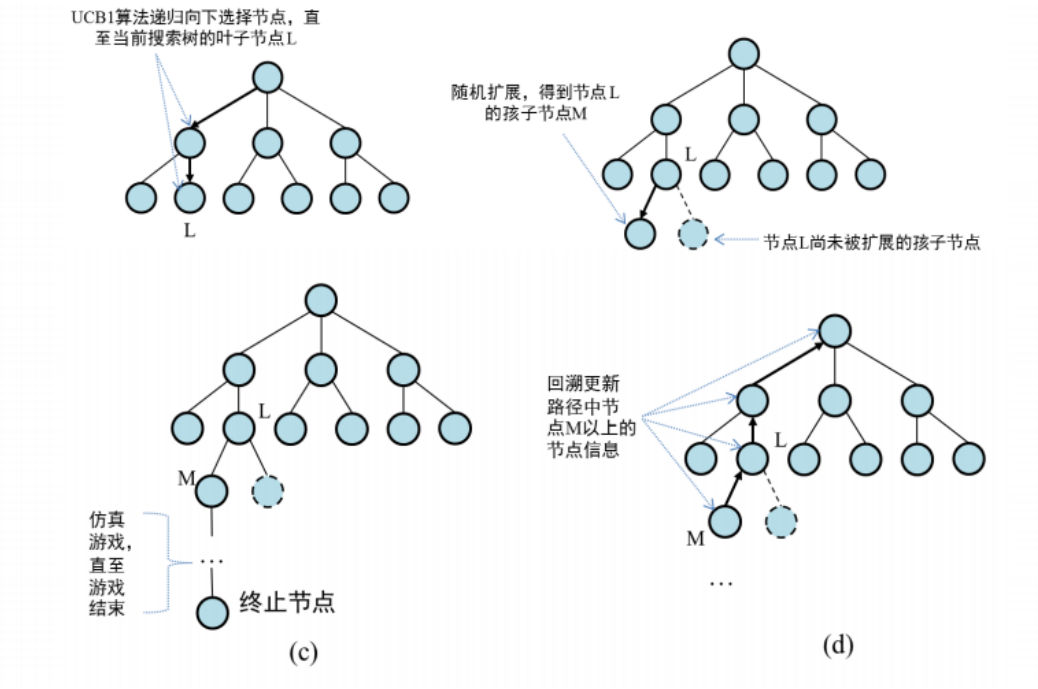
③公式

UCB1算法的策略可以描述为，在第t次时选择使得（3.5）式子取值最大的动作a_{l_t}，其中l_t由如下式子计算得到：

$$l_t = \mathrm{argmax}_i \, \bar{x}_{i,T_{(i,t-1)}} + C\sqrt{\frac{2\ln t}{T_{(i,t-1)}}} \qquad (3.5)$$

在过去第t次已经对动作a_i探索了$T_{(i,t-1)}$次，在当前问题中对应摇动了第i个赌博机的臂膀$T_{(i,t-1)}$次，执行动作a_i所收到收益分数的均值为$\bar{x}_{i,T_{(i,t-1)}}$

8.【填空题】蒙特卡洛树搜索的基本步骤：选择、扩展、模拟、反向传播。



9.蒙特卡洛树搜索举例一

【注意事项】为了统一使用UCB1算法求解，算法将MIN层的子结点（即MAX层结点）的总分记为其相反数。

【例1】

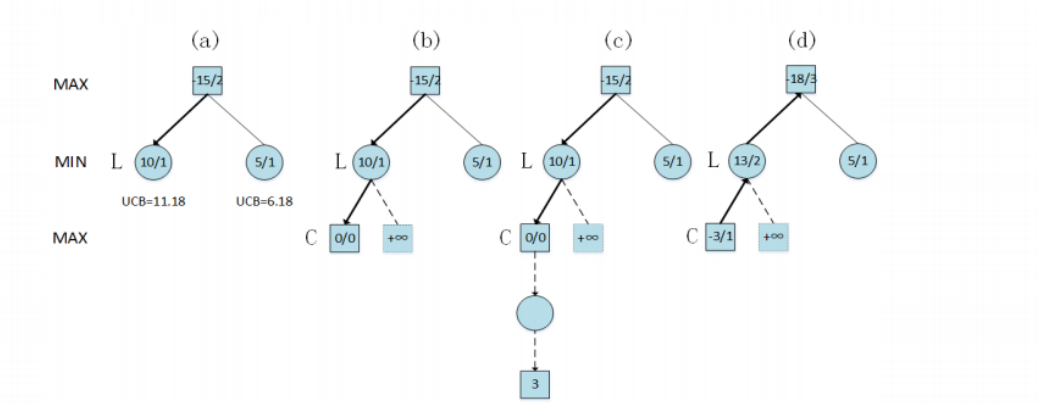


图 3.25 蒙特卡洛树搜索算法。结点中数字代表总收益分数/被访问次数。

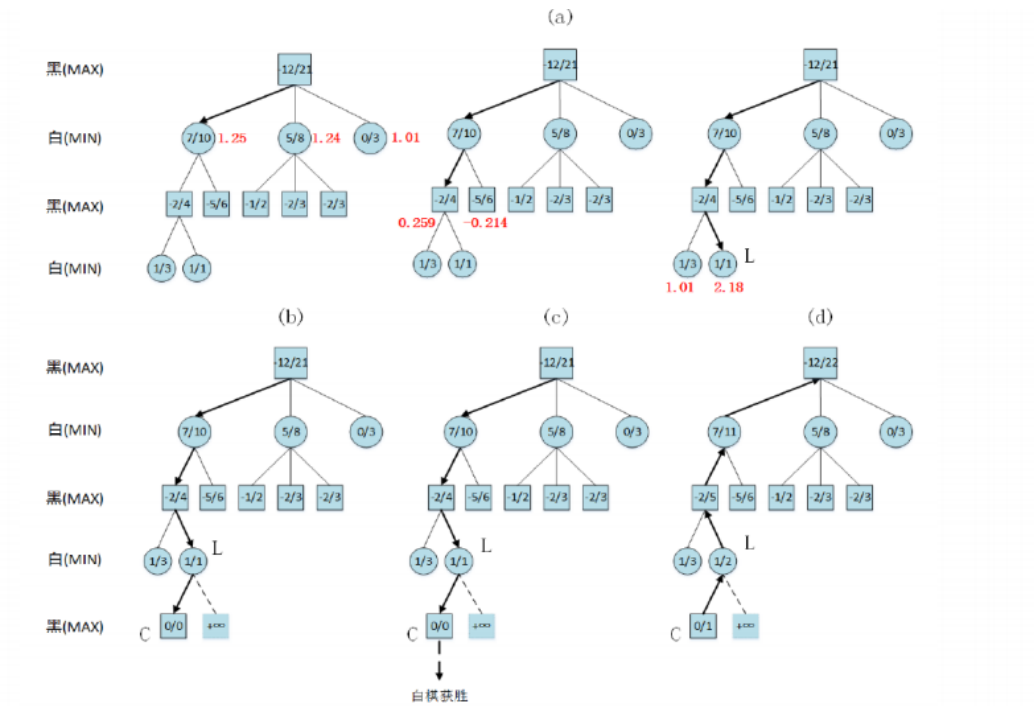
计算第二层节点的UCB值：左侧结点为$\frac{10}{1}+\sqrt{\frac{2\ln 2}{1}}=11.18$，右侧结点为$\frac{5}{1}+\sqrt{\frac{2\ln 2}{1}}=6.18$，因此算法选择第二层左侧的结点L，由于该节点有尚未扩展的子结点，因此选择阶段结束。

①在图3.25(b)中，算法随机扩展了L的子结点C，将其总分数和被访问次数均初始化为0。注意，为了清晰地展示算法选择扩展的结点，图3.25(b)画出了L的其他未被扩展的子结点，并标记其UCB值为正无穷大，以表示算法下次访问到L时必然扩展这些未被扩展的结点。图3.4.6(c)中采用随机策略模拟游戏直至完成游戏。当游戏完成时，终局得分为3。

②在图3.4.6(d)中C结点的总分被更新为-3，被访问次数被更新为1；L结点的总分被更新为13，被访问次数被更新为2；根结点的总分被更新为-18，被访问次数被更新为3。在更新时，会将MIN层结点现有总分加上终局得分分数， MAX层结点现有总分减去终局得分分数。这是因为在对抗搜索中，玩家MIN总是期望最小化终局得分，因此在MIN层选择其子结点时，其目标并非选取奖励最大化的子结点，而是选择奖励最小化的结点，**为了统一使用UCB1算法求解，算法将MIN层的子结点（即MAX层结点）的总分记为其相反数。**

10.蒙特卡洛树搜索举例二

【例2】



11.蒙特卡洛树搜索算法伪代码

函数：UCTSearch
输入： 当前状态s_0
输出： 玩家 MAX 行动下，当前最优动作a^*

```
v_0 ← create_node(s_0)
while 未达到最大迭代次数 do
    v_l ← SelectPolicy(v_0)
    s_t ← SimulatePolicy(v_l.state)
    BackPropagate(v_l, s_t)
end
a* ← UCB1(v_0, 0)
```

函数：SelectPolicy
输入： 选择的起始节点v_0
输出： 选择步骤的结束节点v

```
v ← v_0
while not terminal_test(v.state) do
    if v 存在未被扩展的子节点 then
        return Expand(v)
    else
        v ← UCB1(v, C_p)
    end
end
```

函数：SimulatePolicy
输入： 状态s_0
输出： 模拟的终止状态s

```
s ← s_0
while not terminal_test(s) do
    a ← 从 action(s) 中随机采样
    s ← result(s, a)
end
```

函数：BackPropagate
输入： 反向传播更新的起始节点v，终局状态s_t

```
while v is not null do
    v.N ← v.N + 1
    v.Q ← v.Q − utility(s_t, player(v.state))
    v ← v.parent
end
```

函数：Expand
输入： 节点v
输出： 未被扩展的后继节点v'

```
a ← action(v.state) 中随机的未探索动作
v' ← create_node(result(v.state, a))
v'.N ← 0
v'.Q ← 0
v'.parent ← v
v'.children ← ∅
v.children ← v.children ∪ {v'}
```

函数：UCB1
输入： 节点v，超参数c
输出： 置信上限最大的动作a^*

$$a^* \leftarrow \arg\max_{v' \in v.children} \frac{v'.Q}{v'.N} + c\sqrt{\frac{2\ln v.N}{v'.N}}$$

第三章 MOOC习题

1.在启发式搜索（有信息搜索）中，评价函数的作用是（B）

A.判断搜索算法的空间复杂度

B.从当前节点出发来选择后续节点

C.计算从当前节点到目标节点之间的最小代价值

D.判断搜索算法的时间复杂度

2.在启发式搜索（有信息搜索）中，启发函数的作用是（C）

A.判断搜索算法的空间复杂度

B.从当前节点出发来选择后续节点

C.计算从当前节点到目标节点之间的最小代价值

D.判断搜索算法的时间复杂度

3.在贪婪最佳优先搜索中，评价函数取值和启发函数取值之间的关系是（A）

A.等于

B.不等于

C.小于

D.大于

4.在A*搜索算法中，评价函数可以如下定义（D）

A.评价函数之值=(从起始节点出发到当前节点最小开销代价)*(从当前节点出发到目标结点最小开销代价)

B.评价函数之值=(从起始节点出发到当前节点最小开销代价)/(从当前节点出发到目标结点最小开销代价)

C.评价函数之值=(从起始节点出发到当前节点最小开销代价)-(从当前节点出发到目标结点最小开销代价)

D.评价函数之值=(从起始节点出发到当前节点最小开销代价)+(从当前节点出发到目标结点最小开销代价)

5.A*算法是一种有信息搜索算法，在最短路径搜索中引入的辅助信息是（D）

A.任意一个城市到起始城市之间直线距离

B.路途中天气和交通状况等信息

C.旅行者兴趣偏好信息

D.任意一个城市到目标城市之间直线距离

6.为了保证A*算法是最优的，需要启发函数具有可容（admissible）和一致（consistency）的特点,下面对启发函数具有可容性这一特点

的解释正确的是

（C）

A.启发函数是递减的

B.启发函数是递增的

C.启发函数不会过高估计从当前节点到目标结点之间的实际开销代价

D.启发函数不会过高估计从起始节点到目标结点之间的实际开销代价

7.下面对minimax搜索算法描述中，哪句描述是不正确的（ C ）

A.MAX节点希望自己收益最大化

B.给定一个游戏搜索树，minimax算法通过每个节点的minimax值来决定最优策略

C.minimax搜索不需要遍历游戏树中所有节点

D.MIN节点希望对方收益最小化

8.Alpha和Beta两个值在Alpha-Beta剪枝搜索中被用来判断某个节点的后续节点是否可被剪枝，下面对Alpha和Beta的初始化取值描述正确的是（ C ）

A.Alpha和Beta可随机初始化

B.Alpha的初始值大于Beta的初始值

C.Alpha和Beta初始值分别为负无穷大和正无穷大

D.Alpha和Beta初始值分别为正无穷大和负无穷大

9.下面对上限置信区间 (Upper Confidence Bound,UCB)算法在多臂赌博机中的描述，哪句描述是不正确的（ A ）

A.UCB算法每次随机选择一个赌博机来拉动其臂膀

B.UCB算法既考虑拉动在过去时间内获得最大平均奖赏的赌博机，又希望去选择那些拉动臂膀次数最少的赌博机

C.UCB算法在探索-利用（exploration-exploitation）之间寻找平衡

D.UCB算法是优化序列决策问题的方法

10.下面对minimax搜索、alpha-beta剪枝搜索和蒙特卡洛树搜索的描述中，哪一句描述是不正确的（ B ）

A.minimax是穷举式搜索

B.对于一个规模较小的游戏树，alpha-beta剪枝搜索和minimax搜索的结果会不同

C.alpha-beta剪枝搜索和蒙特卡洛树搜索都是非穷举式搜索

D.三种搜索算法中，只有蒙特卡洛树搜索是采样搜索

第四章 监督学习

一、机器学习基本概念

1.机器学习基本目标

①从原始数据中提取特征。

②学习映射函数f。

③通过映射函数f将原始数据映射到语义任务空间，即寻找数据和任务目标之间的关系。

2.【填空题】

①机器学习一般可以分成三类：监督学习、无监督学习、强化学习。

②监督学习(supervised learning)：数据有标签、一般为回归或分类等任务。

③无监督学习(un-supervised learning)：数据无标签、一般为聚类或若干降维任务

④强化学习(reinforcement learning)：序列数据决策学习，一般为从与环境交互中学习

3.损失函数

①训练集中一共有n个标注数据，第i个标注数据记为(xi, yi),其中第i个样本数据为xi，yi是xi的标注信息。

②从训练数据中学习得到的映射函数记为f，f对xi的预测结果记为$f(xi)$。损失函数就是用来计算xi真实值yi与预测值$f(xi)$之间差值的函数。

③很显然，在训练过程中希望映射函数在训练数据集上

得到“损失”之和最小，即：

$$\min \sum_{i=1}^{n} Loss(f(x_i), y_i)$$

4.常见的损失函数

损失函数名称	损失函数定义
0-1损失函数	$Loss(y_i, f(x_i) = \begin{cases} 1, f(x_i) \neq y_i \\ 0, f(x_i) = y_i \end{cases}$
平方损失函数	$Loss(y_i, f(x_i)) = (y_i - f(x_i))^2$
绝对损失函数	$Loss(y_i, f(x_i)) = \lvert y_i - f(x_i) \rvert$
对数损失函数/对数似然损失函数	$Loss(y_i, P(y_i \lvert x_i)) = -logP((y_i \lvert x_i)$

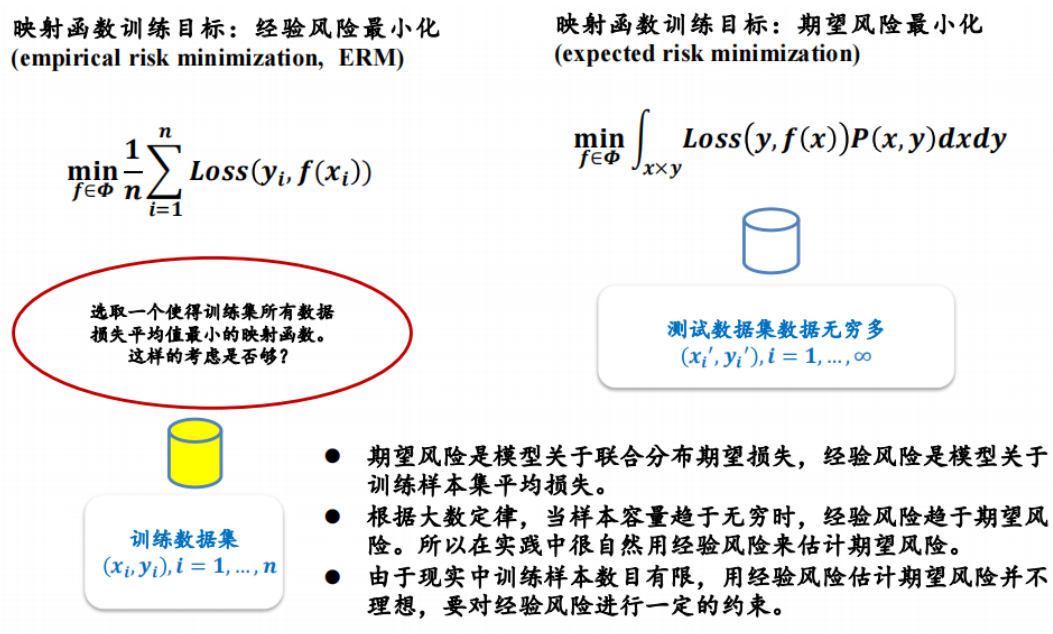
5.经验风险与期望风险

①经验风险(empirical risk)：训练集中数据产生的损失。经验风险越小说明学习模型对训练数据拟合程度越好。

②期望风险(expected risk)：当测试集中存在无穷多数据时产生的损失。期望风

险越小，学习所得模型越好。

6.映射函数训练目标：最小化经验风险、期望风险



①期望风险是模型关于联合分布期望损失，经验风险是模型关于训练样本集平均损失。

②根据大数定律，当样本容量趋于无穷时，经验风险趋于期望风险。所以在实践中很自然用经验风险来估计期望风险。

③由于现实中训练样本数目有限，用经验风险估计期望风险并不理想，要对经验风险进行一定的约束。

7.经验风险、期望风险与泛化能力的关系

经验风险最小化

$$\min_{f\in\Phi}\frac{1}{n}\sum_{i=1}^{n}Loss(y_i,f(x_i))$$

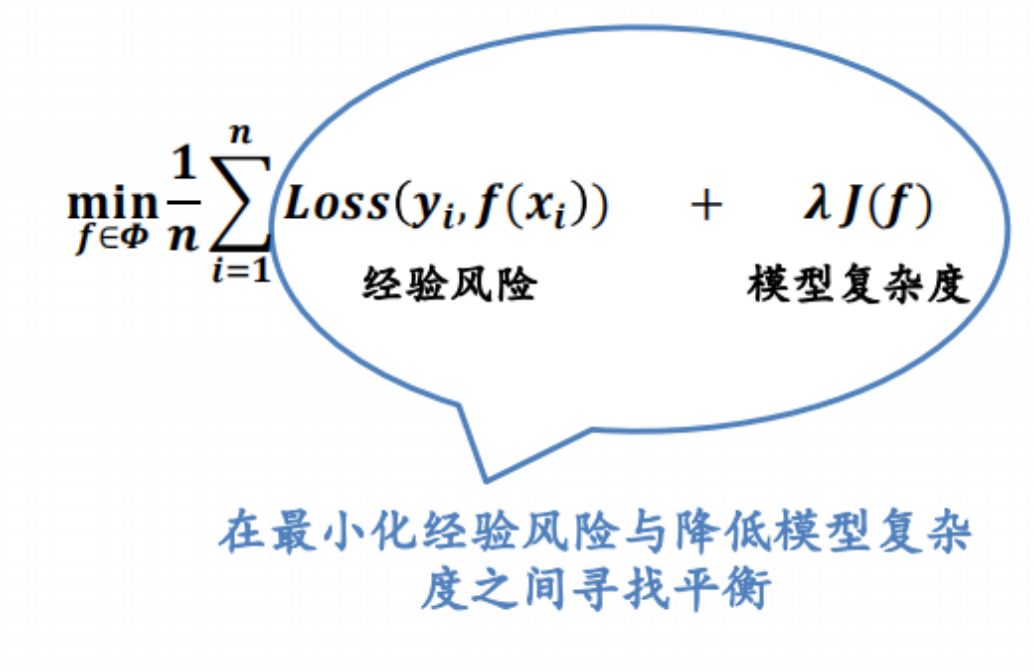
期望风险最小化

$$\min_{f\in\Phi}\int_{x\times y}Loss(y,f(x))P(x,y)dxdy$$

经验风险小（训练集上表现好）	期望风险小（测试集上表现好）	泛化能力强
经验风险小（训练集上表现好）	期望风险大（测试集上表现不好）	过学习（模型过于复杂）
经验风险大（训练集上表现不好）	期望风险大（测试集上表现不好）	欠学习
经验风险大（训练集上表现不好）	期望风险小（测试集上表现好）	“神仙算法”或“黄粱美梦”

表4.3 模型泛化能力与经验风险、期望风险的关系

8.由于经验风险最小化，仅反映了针对局部数据的表现；期望风险最小化因为无法得到全量数据，也具有局限性。故提出结构风险，结构风险等于经验风险+表示模型复杂度的正则化项（惩罚项）。



9.【填空题】

①监督学习方法可以分成判别方法、生成方法两类。所得模型分别称为判别模型、生成模型。

②典型判别模型有：回归模型、神经网络、支持向量机和Ada Boosting等。

③典型生成模型有：贝叶斯方法、隐马尔可夫链等。

10.判别方法

①判别方法直接学习判别函数$f(X)$或者条件概率分布$P(Y|X)$作为预测的模型，即判别模型。

②判别模型关心在给定输入数据下，预测该数据的输出是什么。

③典型判别模型包括回归模型、神经网络、支持向量机和Ada boosting等。

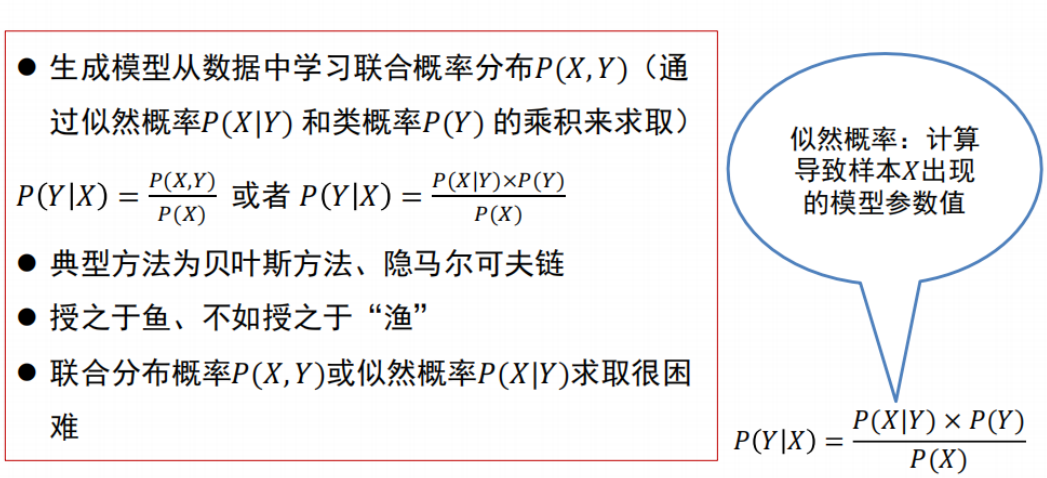
11.生成方法

①生成模型从数据中学习联合概率分布$P(X, Y)$（通过似然概率$P(X|Y)$和类概率$P(Y)$的乘积来求取）

②典型方法为贝叶斯方法、隐马尔可夫链

③授之于鱼、不如授之于“渔”

④联合分布概率$P(X, Y)$或似然概率$P(X|Y)$求取很困难



二、回归分析

1.一元线性回归——利用最小二乘法求解参数

回归模型参数求取：$y_i = ax_i + b\ (1 \le i \le n)$

- 记在当前参数下第i个训练样本x_i的预测值为$\hat{y}_i$
- x_i的标注值（实际值）y_i与预测值$\hat{y}_i$之差记为$(y_i - \hat{y}_i)^2$
- 训练集中n个样本所产生误差总和为：$L(a,b) = \sum_{i=1}^{n}(y_i - a \times x_i - b\ )^2$

目标：寻找一组a和b，使得误差总和$L(a,b)$值最小。在线性回归中，解决如此目标的方法叫最小二乘法。

一般而言，要使函数具有最小值，可对$L(a,b)$ 参数a和b分别求导，令其导数值为零，再求取参数a和b的取值。

回归模型参数求取：$y_i = ax_i + b\ (1 \le i \le n)$ $\quad \min\limits_{a,b} \quad L(a,b) = \sum\limits_{i=1}^{n}(y_i - a \times x_i - b\ )^2$

对b求偏导

$$\frac{\partial L(a,b)}{\partial b} = \sum_{i=1}^{n} 2(y_i - ax_i - b)(-1) = 0$$

$$\rightarrow \sum_{i=1}^{n}(y_i - ax_i - b) = 0$$

$$\rightarrow \sum_{i=1}^{n}(y_i) - a\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} b = 0$$

$$\rightarrow n\bar{y} - an\bar{x} - nb = 0 \quad \Rightarrow \quad b = \bar{y} - a\bar{x}$$

回归模型参数求取：$y_i = ax_i + b\ (1 \le i \le n)$ $\quad \min\limits_{a,b} \quad L(a,b) = \sum\limits_{i=1}^{n}(y_i - a \times x_i - b)^2$

$$\frac{\partial L(a,b)}{\partial a} = \sum_{i=1}^{n} 2(y_i - ax_i - b)(-x_i) = 0$$

将$b = \bar{y} - a\bar{x}\ (\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n},\ \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n})$代入上式

$$\rightarrow \sum_{i=1}^{n}(y_i - ax_i - \bar{y} + a\bar{x})(x_i) = 0$$

$$\rightarrow \sum_{i=1}^{n}(y_i x_i - ax_i x_i - \bar{y}x_i + a\bar{x}x_i) = 0$$

$$\rightarrow \sum_{i=1}^{n}(y_i x_i - \bar{y}x_i) - a\sum_{i=1}^{n}(x_i x_i - \bar{x}x_i) = 0$$

$$\rightarrow (\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}) - a(\sum_{i=1}^{n} x_i x_i - n\bar{x}^2) = 0$$

$$a = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\,\bar{y}}{\sum_{i=1}^{n} x_i x_i - n\bar{x}^2}$$

2.多元线性回归

①参数求解目标：

接下来扩展到数据特征的维度是多维的情况，在上述数据中增加一个影响火灾影响面积的潜在因素—风力。

气温x	5.1	8.2	11.5	13.9	15.1	16.2	19.6	23.3
风力z	4.5	5.8	4	6.3	4	7.2	6.3	8.5
火灾影响面积y	2.14	4.62	8.24	11.24	13.99	16.33	19.23	28.74

多维数据特征中线性回归的问题定义如下：假设总共有m个训练数据$\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{m}$，其中$\boldsymbol{x}_i = [\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \dots, \boldsymbol{x}_{i,D}] \in \mathbb{R}^D$，$D$为数据特征的维度，线性回归就是要找到一组参数$\boldsymbol{a} = [\boldsymbol{a}_0, \boldsymbol{a}_1, \dots, \boldsymbol{a}_D]$，使得线性函数：

$$f(\boldsymbol{x}_i) = a_0 + \sum_{j=1}^{D} a_j\, x_{i,j} = a_0 + \boldsymbol{a}^T \boldsymbol{x}_i$$

②参数求解方法：

最小化均方误差函数：

$$J_m = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - f(\boldsymbol{x_i})\right)^2$$

为了方便，使用矩阵来表示所有的训练数据和数据标签。

$$X = [x_1, \dots, x_m], \qquad \boldsymbol{y} = [y_1, \dots, y_m]$$

其中每一个数据x_i会扩展一个维度，其值为1，对应参数a_0。均方误差函数可以表示为：

$$J_m(\boldsymbol{a}) = \left(\boldsymbol{y} - X^T\boldsymbol{a}\right)^T\left(\boldsymbol{y} - X^T\boldsymbol{a}\right)$$

均方误差函数$J_n(\boldsymbol{a})$对所有参数$\boldsymbol{a}$求导可得：

$$\nabla J(\boldsymbol{a}) = -2X(y - X^T\boldsymbol{a})$$

因为均方误差函数$J_n(\boldsymbol{a})$是一个二次的凸函数，所以函数只存在一个极小值点，也同样是最小值点，所以令$\nabla J(\boldsymbol{a}) = 0$可得

$$XX^T\boldsymbol{a} = X\boldsymbol{y}$$

$$\boldsymbol{a} = \left(XX^T\right)^{-1}X\boldsymbol{y}$$

3.对数几率回归（逻辑斯蒂回归）

（1）思想：线性回归对离群点很敏感，对数几率回归可以用来减少离群点的影响。

（2）逻辑斯蒂回归引入Sigmoid函数进行拟合。其中，z 应该被替换为wTx+b。w向量表示各个维度的自变量对应的参数，b表示偏置。wTx得到的是一个数。

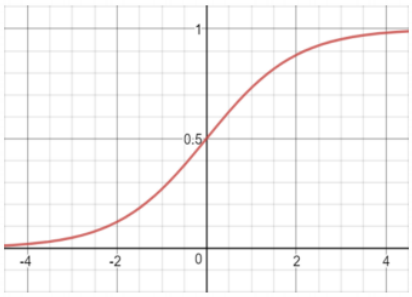


图4.6 Sigmoid函数

逻辑斯蒂回归(logistic regression)就是在回归模型中引入sigmoid函数的一种非线性回归模型。Logistic回归模型可如下表示：

$$y = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x}+b)}} \quad ，其中 y \in (0,1), z = \boldsymbol{w}^T\boldsymbol{x} + \mathrm{b}$$

这里$\frac{1}{1+e^{-z}}$是sigmoid函数、$\boldsymbol{x} \in \mathbb{R}^d$是输入数据、$\boldsymbol{w} \in \mathbb{R}^d$和$b \in \mathbb{R}$是回归函数的参数。

（3）特点：

①sigmoid函数是单调递增的，其值域为(0，1)， 函数输出可作为概率值。在前面介绍的线性回归中，回归函数的值域一般为(−∞, +∞)。

②对输入z取值范围没有限制，但当z大于一定数值后，函数输出无限趋近于1，而小于一定数值后，函数输出无限趋近于0。特别地，当$z = 0$时，函数输出为0.5这里z是输入数据x和回归函数的参数w内积结果（可视为$\boldsymbol{x}$各维度进行加权叠加）

③$\boldsymbol{x}$各维度加权叠加之和结果取值在0附近时，函数输出值的变化幅度比较大（函数值变化陡峭），且是非线性变化。但是，各维度加权叠加之和结果取值很大或很小时，函数输出值几乎不变化，这是基于概率的一种认识与需要。

4.逻辑斯蒂回归与二分类问题

逻辑斯蒂回归虽可用于对输入数据和输出结果之间复杂关系进行建模，但由于逻辑斯蒂回归函数的输出具有概率意义，使得逻辑斯蒂回归函数更多用于二分类问题（y = 1表示输入数据$\boldsymbol{x}$属于正例，y = 0表示输入数据$\boldsymbol{x}$属于负例）。

$y = \frac{1}{1+e^{-(\boldsymbol{w}^T x+b)}}$可用来计算输入数据$\boldsymbol{x}$属于正例概率，这里y理解为输入数据$\boldsymbol{x}$为正例的概率、1 − y理解为输入数据x为负例的概率，即$p(y=1|\boldsymbol{x})$。我们现在对比值$\frac{p}{1-p}$取对数(即$\log\left(\frac{p}{1-p}\right)$)来表示输入数据$\boldsymbol{x}$属于正例概率。$\frac{p}{1-p}$被称为几率(odds)，反映了输入数据$\boldsymbol{x}$作为正例的相对可能性。$\frac{p}{1-p}$的对数几率(log odds)或logit函数可表示为$\log\left(\frac{p}{1-p}\right)$。

显然，可以得到 $p(y=1|\boldsymbol{x}) = h_\theta(x) = \frac{1}{1+e^{-(\boldsymbol{w}^T x+b)}}$ 和 $p(y=0|\boldsymbol{x}) = 1 - h_\theta(x) = \frac{e^{-(\boldsymbol{w}^T x+\mathrm{b})}}{1+e^{-(\boldsymbol{w}^T x+\mathrm{b})}}$。$\theta$表示模型参数（$\theta = \{\boldsymbol{w}, b\}$）。于是有：

$$\mathrm{logit}(p(y=1|\boldsymbol{x})) = \log\left(\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}\right) = \log\left(\frac{p}{1-p}\right) = \boldsymbol{w}^T\boldsymbol{x} + \mathrm{b}$$

如果输入数据$\boldsymbol{x}$属于正例的概率大于其属于负例的概率，即$p(y=1|\boldsymbol{x})>0.5$，则输入数据$\boldsymbol{x}$可被判断属于正例。这一结果等价于$\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}>1$，即$\log\left(\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}\right)>\log 1=0$，也就是$\boldsymbol{w}^T\boldsymbol{x}+\mathrm{b}>0$成立。

从这里可以看出，logistic回归是一个线性模型。在预测时，可以计算线性函数$\boldsymbol{w}^T\boldsymbol{x}+\mathrm{b}$取值是否大于0来判断输入数据$\boldsymbol{x}$的类别归属。

5.基于似然函数的参数优化

①最大化对数似然函数：用于求解逻辑斯蒂回归模型中的参数。

模型参数的似然函数被定义为$\mathcal{L}(\theta|\mathcal{D})=\mathrm{p}(\mathcal{D}|\theta)$，其中$\mathcal{D}=\{(x_i,y_i\}(1\le \mathrm{i}\le \mathrm{n})$表示所有观测数据（或训练数据），$\theta$表示模型参数（$\theta=\{\boldsymbol{w},b\}$）。在最大化对数似然函数过程中，一般假设观测所得每一个样本数据是独立同分布(independent and identically distributed, i.i.d），于是可得：

$$\mathcal{L}(\theta|\mathcal{D})=\mathrm{p}(\mathcal{D}|\theta)=\prod_{i=1}^{n}\mathrm{p}(y_i|\mathrm{x},\theta)=\prod_{i=1}^{n}\left(h_\theta(x_i)\right)^{y_i}\left(1-h_\theta(x_i)\right)^{1-y_i}$$

对上述公式取对数：

$$\mathrm{l}(\theta)=\log\left(L(\theta|\mathcal{D})\right)=\sum_{i=1}^{n}y_i\log\left(h_\theta(x_i)\right)+(1-y_i)\log(1-h_\theta(x_i))$$

最大似然估计目的是计算似然函数的最大值，而分类过程是需要损失函数最小化。因此，在上式前加一个负号得到损失函数(交叉熵)：

$$\begin{aligned}\mathcal{T}(\theta)&=-\mathrm{l}(\theta)=-\log\left(L(\theta|\mathcal{D})\right)\\&=-(\sum_{i=1}^{n}y_i\log\left(h_\theta(x_i)\right)+(1-y_i)\log(1-h_\theta(x_i)))\end{aligned}$$

$\mathcal{T}(\theta)$等价于：$\mathcal{T}(\theta)=\begin{cases}-\log\left(h_\theta(x_i)\right) & if\ y=1\\-\log\left(1-h_\theta(x_i)\right) & if\ y=0\end{cases}$

求导过程：

需要最小化损失函数来求解参数。损失函数对参数θ的偏导如下（其中，$h'_\theta(x) = h_\theta(x)(1-h_\theta(x)), \log' x = \frac{1}{x}$）

$$
\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_j} &= -\sum_{i=1}^{n}\left(y_i \frac{1}{h_\theta(x_i)}\frac{\partial h_\theta(x_i)}{\partial \theta_j} + (1-y_i)\frac{1}{1-h_\theta(x_i)}\frac{\partial(1-h_\theta(x_i))}{\partial \theta_j}\right) \\
&= -\sum_{i=1}^{n}\frac{1}{h_\theta(x_i)}\left(\frac{y_i}{h_\theta(x_i)} - \frac{1-y_i}{1-h_\theta(x_i)}\right) \\
&= -\sum_{i=1}^{n} x_i h_\theta(x_i)(1-h_\theta(x_i))\left(\frac{y_i}{h_\theta(x_i)} - \frac{1-y_i}{1-h_\theta(x_i)}\right) \\
&= -\sum_{i=1}^{n} x_i(y_i(1-h_\theta(x_i)) - (1-y_i)h_\theta(x_i)) \\
&= \sum_{i=1}^{n}(y_i - h_\theta(x_i))x_i
\end{aligned}
$$

最大似然估计
（maximum likelihood estimation, MLE）

将求导结果代入梯度下降迭代公式得：

$$\theta_j = \theta_j - \eta\sum_{i=1}^{n}(y_i - h_\theta(x_i))x_i$$

6.最大似然估计、最大后验概率估计

最大似然估计
（maximum likelihood estimation, MLE）

$$\mathcal{L}(\theta|\mathcal{D}) = \mathrm{p}(\mathcal{D}|\theta) = \prod_{i=1}^{n}\mathrm{p}(y_i|\mathrm{x},\theta)$$

最大后验概率估计
（Maximum a posteriori estimation, MAP）

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

2001年，著名统计学家、机器学习先驱 Leo Breiman 发表了著名文章《统计建模：两种文化》（Statistical Modeling: The Two Cultures）

统计建模一个重要矛盾是：人类对解决复杂现实问题的向往与传统数据模型孱弱的预测力之间的矛盾。

在从数据到结论的过程中，有两种统计建模文化。

- 数据模型：假设数据是通过给定的随机数据模型生成的
- 算法模型：数据生成机制视为未知。
- 一直以来，统计界几乎完全使用数据模型。这种情况造就了无关紧要的理论、有问题的结论，并使统计学家无法研究广阔、有趣的现实问题。算法建模，无论在理论上还是实践上，都在统计学之外的领域飞速发展。它既可以用于大型复杂的数据集，也可以用于小型数据集。在小型数据集的处理上，算法模型也比数据模型更准确，能产生更丰富的信息。如果我们的目标是让统计学能够使用数据来解决问题，那么我们需要摆脱对数据模型的完全依赖，采用更多样化的工具。

6.决策树

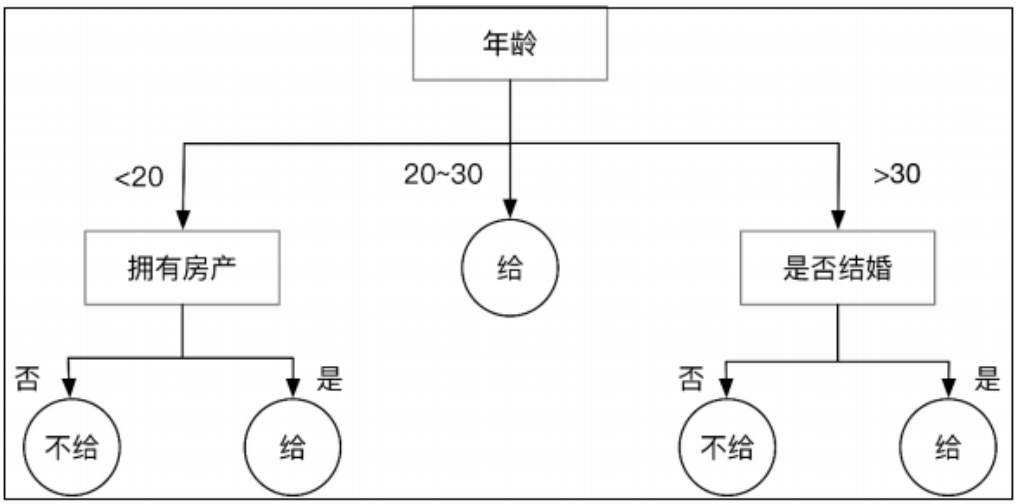
决策树是一种通过树形结构来进行分类的方法。在决策树中，树形结构

中每个非叶子节点表示对分类目标在某个属性上的一个判断，每个分支

代表基于该属性做出的一个判断，最后树形结构中每个叶子节点代表一

种分类结果，所以决策树可以看作是一系列以叶子节点为输出的决策规

则。



7.信息熵

信息熵（entropy）

假设有K个信息，其组成了集合样本D，记第k个信息发生的概率为$p_k(1 \le k \le K)$”。如下定义这K个信息的信息熵：

$$E(D) = -\sum_{k=1}^{K} p_k \log_2 p_k$$

$E(D)$值越小，表示D包含的信息越确定，也称D的纯度越高。需要指出，所有p_k累加起来的和为1。

要点：构建决策树时划分属性的顺序选择是重要的。性能好的决策树随着划分不断进行，决策树分支结点样本集的“纯度”会越来越高，即其所包含样本尽可能属于相同类别。

表 4.7 年龄属性划分后子样本集情况统计

年龄属性取值a_i	“>30”	“20~30”	“<20”
对应样本数$\|D_i\|$	5	4	5
正负样本数量	(2+, 3-)	(4+，0-)	(3+，2-)

(1)信息熵越小，表示信息越确定，纯度越高。

表 4.7 年龄属性划分后子样本集情况统计

年龄属性取值 a_i	“>30”	“20~30”	“<20”
对应样本数 $\lvert D_i \rvert$	5	4	5
正负样本数量	(2+, 3-)	(4+，0-)	(3+，2-)

$$“年龄 > 30”:\quad Ent(D_0) = -\left(\frac{2}{5}\times\log_2\frac{2}{5}+\frac{3}{5}\times\log_2\frac{3}{5}\right) = 0.971$$

$$“年龄20\sim30”:\quad Ent(D_1) = -\left(\frac{4}{4}\times\log_2\frac{4}{4}+0\right) = 0$$

$$“年龄 < 20”:\quad Ent(D_2) = -\left(\frac{3}{5}\times\log_2\frac{3}{5}+\frac{2}{5}\times\log_2\frac{2}{5}\right) = 0.971$$

（2）如上图所示，20-30的年龄段没有反例，表明纯度很高；另外两个年龄段的纯度有待提升，需要作进一步优化。

（3）信息熵增益计算

得到上述三个的信息熵后，可进一步计算使用年龄属性对原样本集进行划分后的信息增益，计算公式如下：

$$Gain(D,A) = Ent(D) - \sum_{i=1}^{n}\frac{|D_i|}{|D|}Ent(D_i)$$

将 $A =$ 年龄 代入。于是选择年龄这一属性划分后的信息增益为：

$$Gain(D,年龄) = 0.940 - \left(\frac{5}{14}\times 0.971 + \frac{4}{14}\times 0 + \frac{5}{14}\times 0.971\right) = 0.246$$

同理，可以计算银行流水、是否结婚、是否拥有房产三个人物属性的信息增益。通过比较四种属性信息增益的高低来选择最佳属性对原样本集进行划分，得到最大的“纯度”。如果划分后的不同子样本集都只存在同类样本，那么停止划分。

①如图，各个分支结点计算出信息熵之后，它们父结点的信息熵，应该是它们信息熵的加权和。加权的依据是各个叶子结点包含实例的个数。

②由于信息熵是正增益，因此纯度降低。

（4）信息增益率、Gini系数

引入信息量概念，通过信息增益/信息量，即得到信息增益率。

$$info = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$$

$$Gain - ratio = Gain(D, A)/info$$

另一种计算更简的度量指标是如下的Gini系数：

$$\mathrm{Gini(D)} = 1 - \sum_{k=1}^{K} p_k^2$$

相对于信息熵的计算$E(D) = -\sum_{k=1}^{K} p_k \log_2 p_k$，不用计算对数log，计算更为简易。

四、线性判别分析（LDA）

1.线性判别分析(linear discriminant analysis，LDA)是一种基于监督学习的降维方法，也称为Fisher线性判别分析（fisher's discriminant analysis，FDA) [Fisher 1936]。对于一组具有标签信息的高维数据样本，LDA利用其类别信息，将其线性投影到一个低维空间上，在低维空间中同一类别样本尽可能靠近，不同类别样本尽可能彼此远离。

2.符号定义

假设样本集为$\mathrm{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_N, y_N)\}$，样本$\boldsymbol{x}_i \in \mathbb{R}^d$的类别标签为$y_i$。其中，$y_i$的取值范围是$\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$，即一共有$K$类样本。

定义$\boldsymbol{X}$为所有样本构成集合、N_i为第i个类别所包含样本个数、X_i为第i类样本的集合、$\boldsymbol{m}$为所有样本的均值向量、$\boldsymbol{m}_i$为第i类样本的均值向量。Σ_i为第i类样本的协方差矩阵，其定义为：

$$\Sigma_i = \sum_{\boldsymbol{x} \in \boldsymbol{X}_i} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T$$

3.线性投影举例

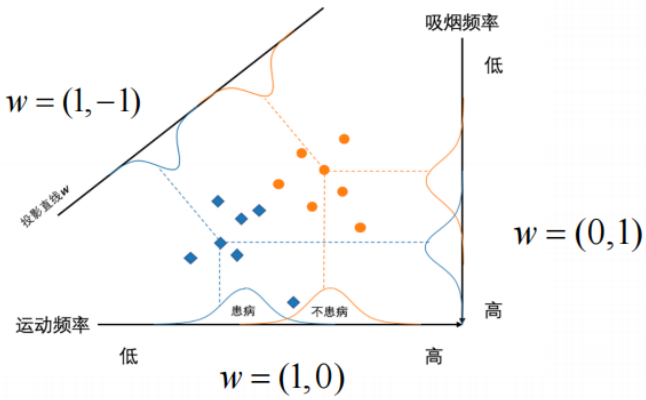
先来看$K=2$的情况，即二分类问题。在二分类问题中，训练样本归属于$\mathcal{C}_1$或$\mathcal{C}_2$两个类别，并通过如下的线性函数投影到一维空间上：

$$y(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{x}\ (\boldsymbol{w}\in\mathbb{R}^n)$$

正方形表示患病个体、圆形表示不患病个体

节点$(1,1),(2,2),(3,3),(4,4)\ldots$

都会投影到同一个点。



如图所示：投影到直线w上，可以使得类内方差最小，类间方差最大；而投影到横纵轴，两个类别都有一定重叠部分，类间方差不够大。

4.线性判别分析——二分类问题

先来看$K=2$的情况，即二分类问题。在二分类问题中，训练样本归属于$\mathcal{C}_1$或$\mathcal{C}_2$两个类别，并通过如下的线性函数投影到一维空间上：

$$y(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{x}\ (\boldsymbol{w}\in\mathbb{R}^n)$$

投影之后类别$\mathcal{C}_1$的协方差矩阵s_1为：

$$s_1=\sum_{x\in\mathcal{C}_1}\left(\boldsymbol{w}^T\boldsymbol{x}-\boldsymbol{w}^T\boldsymbol{m}_1\right)^2=\boldsymbol{w}^T\sum_{x\in\mathcal{C}_1}[(\boldsymbol{x}-\boldsymbol{m}_1)(\boldsymbol{x}-\boldsymbol{m}_1)^T]\boldsymbol{w}$$

同理可得到投影之后类别$\mathcal{C}_2$的协方差矩阵s_2。

（1）最大化目标函数

投影后两个协方差矩阵s_1和s_2分别为$\boldsymbol{w}^T\Sigma_1\boldsymbol{w}$和$\boldsymbol{w}^T\Sigma_2\boldsymbol{w}$。$s_1$和$s_2$可用来衡量同一类别数据样本之间“分散程度”。为了使得归属于同一类别的样本数据在投影后的空间中尽可能靠近，需要最小化s_1+s_2取值。

在投影之后的空间中，归属于两个类别的数据样本中心可分别如下计算：

$$m_1 = \boldsymbol{w}^T\boldsymbol{m}_1, \quad m_2 = \boldsymbol{w}^T\boldsymbol{m}_2$$

这样，就可以通过$\|m_2 - m_1\|_2^2$来衡量不同类别之间的距离。为了使得归属于不同类别的样本数据在投影后空间中尽可能彼此远离，需要最大化$\|m_2 - m_1\|_2^2$取值。

同时考虑上面两点，就得到了需要最大化的目标$J(\boldsymbol{w})$，定义如下：

$$J(\boldsymbol{w}) = \frac{\|m_2 - m_1\|_2^2}{s_1 + s_2}$$

（2）类间散度矩阵与类内散度矩阵的定义

$$J(\boldsymbol{w}) = \frac{\left\|\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)\right\|_2^2}{\boldsymbol{w}^T\Sigma_1\boldsymbol{w} + \boldsymbol{w}^T\Sigma_2\boldsymbol{w}} = \frac{\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T\boldsymbol{w}}{\boldsymbol{w}^T(\Sigma_1 + \Sigma_2)\boldsymbol{w}} = \frac{\boldsymbol{w}^T\boldsymbol{S}_b\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w}}$$

其中，$\boldsymbol{S}_b$称为类间散度矩阵(between-class scatter matrix)，即衡量两个类别均值点之间的“分离”程度，可定义如下:

$$\boldsymbol{S}_b = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T$$

$\boldsymbol{S}_w$则称为类内散度矩阵(within-class scatter matrix)，即衡量每个类别中数据点的“分离”程度，可定义如下：

$$\boldsymbol{S}_w = \Sigma_1 + \Sigma_2$$

由于$J(\boldsymbol{w})$的分子和分母都是关于$\boldsymbol{w}$的二项式，因此最后的解只与$\boldsymbol{w}$的方向有关，与$\boldsymbol{w}$的长度无关（即便扩展或缩减$\boldsymbol{w}$，$\boldsymbol{w}\boldsymbol{w}$所得长度也可被约分掉），因此可令分母$\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w} = 1$，然后用拉格朗日乘子法来求解这个问题。

（3）Fisher线性判别

对应拉格朗日函数为：

$$L(\boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{S}_b \boldsymbol{w} - \lambda(\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w} - 1)$$

对$\boldsymbol{w}$求偏导并使其求导结果为零，可得$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b\boldsymbol{w} = \lambda\boldsymbol{w}$。由此可见，λ和$\boldsymbol{w}$分别是$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b$的特征根和特征向量，$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b\boldsymbol{w} = \lambda\boldsymbol{w}$也被称为Fisher线性判别（Fisher linear discrimination）。

$$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b\boldsymbol{w} = \boldsymbol{S}_w{}^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1) \times \lambda_w = \lambda\boldsymbol{w}$$

由于对$\boldsymbol{w}$的放大和缩小操作不影响结果，因此可约去上式中的未知数λ和λ_w，得到：$\boldsymbol{w} = \boldsymbol{S}_w{}^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$

$$\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \left(\frac{\partial}{\partial x_1}\sum_{i,j=1}^{n} a_{ij}x_ix_j, \cdots, \frac{\partial}{\partial x_n}\sum_{i,j=1}^{n} a_{ij}x_ix_j\right)^{\mathrm{T}}$$

$$= \begin{bmatrix} 2a_{11}x_1 + (a_{12}+a_{21})x_2 + \cdots + (a_{1n}+a_{n1})x_n \\ \vdots \\ (a_{1n}+a_{n1})x_1 + (a_{n2}+a_{2n})x_2 + \cdots + 2a_{nn}x_n \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{bmatrix} + \begin{bmatrix} a_{11}x_1 + a_{21}x_2 + \cdots + a_{n1}x_n \\ \vdots \\ a_{1n}x_1 + a_{2n}x_2 + \cdots + a_{nn}x_n \end{bmatrix}$$

$$= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{x} = (\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}.$$

当 $\boldsymbol{A}$ 是实对称矩阵时，二次型 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}$ 对 $\boldsymbol{x}$ 的导数为

$$\frac{\mathrm{d}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\mathrm{d}\boldsymbol{x}} = 2\boldsymbol{A}\boldsymbol{x};$$

5.线性判别举例——多分类问题

（1）类间散度矩阵、类内散度矩阵的重定义

假设n个原始高维数据所构成的类别种类为K、每个原始数据被投影映射到低维空间中的维度为r。

令投影矩阵$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_r)$，可知$\boldsymbol{W}$是一个$n \times r$矩阵。于是，$\boldsymbol{W}^T\boldsymbol{m}_i$为第$i$类样本数据中心在低维空间的投影结果，$\boldsymbol{W}^T\Sigma_i\boldsymbol{W}$为第$i$类样本数据协方差在低维空间的投影结果。

类内散度矩阵$\boldsymbol{S}_w$重新定义如下：

$$\boldsymbol{S}_w = \sum_{i=1}^{K}\Sigma_i，其中\Sigma_i = \sum_{x \in class\ i}(\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T$$

在上式中，$\boldsymbol{m}_i$是第i个类别中所包含样本数据的均值。

类间散度矩阵$\boldsymbol{S}_b$重新定义如下：

$$\boldsymbol{S}_b = \sum_{i=1}^{K}\frac{N_i}{N}(\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

（2）目标函数

将多类LDA映射投影方向的优化目标$J(\boldsymbol{W})$改为：

$$J(\boldsymbol{W})=\frac{\prod_{diag}\boldsymbol{W}^T\boldsymbol{S}_b\boldsymbol{W}}{\prod_{diag}\boldsymbol{W}^T\boldsymbol{S}_w\boldsymbol{W}}$$

其中，$\prod_{diag}\mathbb{A}$为矩阵$\mathbb{A}$主对角元素的乘积。

继续对$J(\boldsymbol{W})$进行变形：

$$J(\boldsymbol{W})=\frac{\prod_{diag}\boldsymbol{W}^T\boldsymbol{S}_b\boldsymbol{W}}{\prod_{diag}\boldsymbol{W}^T\boldsymbol{S}_w\boldsymbol{W}}=\frac{\prod_{i=1}^{r}\boldsymbol{w}_i^T\boldsymbol{S}_b\boldsymbol{w}_i}{\prod_{i=1}^{r}\boldsymbol{w}_i^T\boldsymbol{S}_w\boldsymbol{w}_i}=\prod_{i=1}^{r}\frac{\boldsymbol{w}_i^T\boldsymbol{S}_b\boldsymbol{w}_i}{\boldsymbol{w}_i^T\boldsymbol{S}_w\boldsymbol{w}_i}$$

显然需要使乘积式子中每个$\frac{\boldsymbol{w}_i^T\boldsymbol{S}_b\boldsymbol{w}_i}{\boldsymbol{w}_i^T\boldsymbol{S}_w\boldsymbol{w}_i}$取值最大，这就是二分类问题的求解目标，即每一个$\boldsymbol{w}_i$都是$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b\boldsymbol{W}=\lambda\boldsymbol{W}$的一个解。

（3）算法步骤：

对线性判别分析的降维步骤描述如下：

1. **计算数据样本集中每个类别样本的均值**
2. **计算类内散度矩阵S_w和类间散度矩阵S_b**
3. **根据$S_w{}^{-1}S_bW=\lambda W$来求解$S_w{}^{-1}S_b$所对应前r个最大特征值所对应特征向量$(w_1,w_2,\dots,w_r)$，构成矩阵W**
4. **通过矩阵W将每个样本映射到低维空间，实现特征降维。**

（4）线性判别分析、主成分分析比较

	线性判别分析	主成分分析
是否需要样本标签	监督学习	无监督学习
降维方法	优化寻找特征向量$\boldsymbol{w}$	优化寻找特征向量$\boldsymbol{w}$
目标	类内方差小、类间距离大	寻找投影后数据之间方差最大的投影方向
维度	LDA降维后所得到维度是与数据样本的类别个数K有关	PCA对高维数据降维后的维数是与原始数据特征维度相关

五、Ada Boosting

1.基本思想：

①对于一个复杂的分类任务，可以将其分解为若干子任务，然后将若干子任务完成方法综合，最终完成该复杂任务。

②将若干个弱分类器(weak classifiers)组合起来，形成一个强分类器(strong classifier)。

2.霍夫丁不等式

- 霍夫丁不等式：全国人口中看该电视节目的人口比例（记作x）与抽样人口中观看该电视节目的人口比例（记作y）满足如下关系：

$P(|x-y| \geq \epsilon) \leq 2e^{-2N\epsilon^2}$($N$是采样人口总数、$\epsilon \in (0,1)$是所设定的可容忍误差范围)

当N足够大时，“全国人口中电视节目收视率”与“样本人口中电视节目收视率”差值超过误差范围ϵ的概率非常小。

3.强可学习、弱可学习概念

强可学习与弱可学习等价，发现弱学习算法后，可将其提升为强学习算法。

> 在概率近似正确背景下，有“强可学习模型”和“弱可学习模型”

强可学习 (strongly learnable)	● 学习模型能够以较高精度对绝大多数样本完成识别分类任务
弱可学习 (weakly learnable)	学习模型仅能完成若干部分样本识别与分类，其精度略高于随机猜测。(不可学习指的是学习模型所获得精度仅为50%，即相当于随机猜测)
强可学习和弱可学习是等价的，也就是说，如果已经发现了“弱学习算法”，可将其提升（boosting）为“强学习算法”。Ada Boosting算法就是这样的方法。具体而言，Ada Boosting将一系列弱分类器组合起来，构成一个强分类器。	

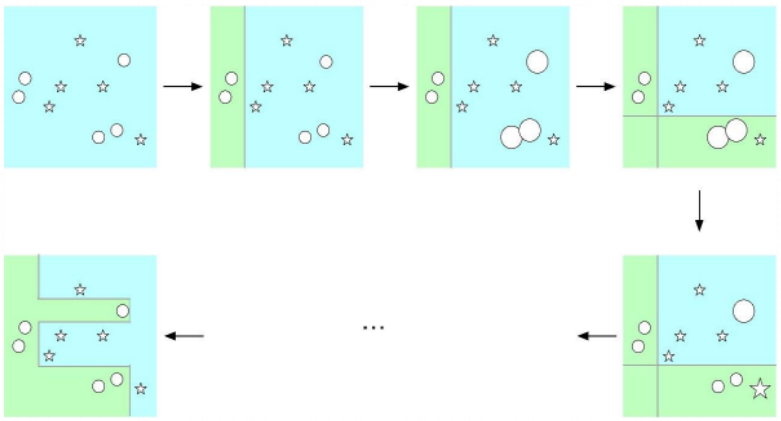


图4.9 Ada Boosting算法学习过程示意图

4.算法

关键想法：弱分类器线性组合，形成强分类器。

- **给定包含N个标注数据的训练集合Γ，$\Gamma = \{(x_1, y_1), \dots, (x_N, y_N)\}$。**

  $x_i (1 \le i \le N) \in X \subseteq R^n, y_i \in Y = \{-1, 1\}$

- **Ada Boosting算法将从这些标注数据出发，训练得到一系列弱分类器，并将这些弱分类器线性组合得到一个强分类器。**

  1. **初始化每个训练样本的权重**

  $D_1 = (w_{11}, \dots, w_{1i}, \dots, w_{1N})$，其中 $w_{1i} = \frac{1}{N} (1 \le i \le N)$

2. 对$m = 1, 2, \dots, M$

a) 使用具有分布权重D_m的训练数据来学习得到第m个基分类器（弱分类器）G_m：

$$G_m(x): X \to \{-1, 1\}$$

b) 计算$G_m(x)$在训练数据集上的分类误差

$err_m = \sum_{i=1}^{N} w_{mi} I(G_m(x_i) \neq y_i)$ 这里：$I(\cdot) = 1$，如果$G_m(x_i) \neq y_i$；否则为0

c) 计算弱分类器$G_m(x)$的权重：$\alpha_m = \frac{1}{2} \ln \frac{1 - err_m}{err_m}$

d) 更新训练样本数据的分布权重：$D_{m+1} = w_{m+1,i} = \frac{w_{m,i}}{Z_m} e^{-\alpha_m y_i G_m(x_i)}$，其中$Z_m$是归一化因子以使得$D_{m+1}$为概率分布，$Z_m = \sum_{i=1}^{N} w_{m,i} e^{-\alpha_m y_i G_m(x_i)}$

3. 以线性加权形式来组合弱分类器$f(x)$

$$f(x) = \sum_{i=1}^{M} \alpha_m G_m(x)$$

得到强分类器$G(x)$

$$G(x) = sign(f(x)) = sign(\sum_{i=1}^{M} \alpha_m G_m(x))$$

五、支持向量机

1.【填空题】支持向量机 （support vector machine,SVM）通过结构风险最小化（structural risk minimization）来解决过学习问题。

2.在理论上，支持向量机认为：分类器对未知数据（即测试数据）进行分类时所产生的期望风险（expected risk，即真实风险）不是由经验风险单独决定的，而是由两部分组成：1）从训练集合数据所得经验风险（如果经验风险小、期望风险很大，则是过学习）；2）置信

风险（confidence risk），它与分类器的VC维及训练样本数目有关。

六、生成学习模型

【填空题】**常见的生成学习模型有隐马尔可夫模型、隐狄利克雷分布(latent dirichlet allocation，LDA)等。**

第四章 MOOC习题答案

1.下面哪一种机器学习方法没有利用标注数据（ B ）

A.有监督学习

B.无监督学习

C.回归分析

D.半监督学习

2.下面对经验风险和期望风险的描述中，哪一个属于欠学习（ C ）

A.经验风险大、期望风险小

B.经验风险小、期望风险小

C.经验风险大、期望风险大

D.经验风险小、期望风险大

3.下面对结构风险最小化的描述中，哪一个描述是不正确的（ D ）

A.在结构风险最小化中，优化求解目标为使得经验风险与模型复杂度之和最小

B.结构风险最小化与经验风险最小化的目标是不同的（正确）

C.结构风险最小化在最小化经验风险与降低模型复杂度之间寻找一种平衡

D.为了更好保证结构风险最小化，可适当减少标注数据(错误)

4.监督学习方法又可以分为生成方法(generative approach)和判别方法(discriminative approach)。下面哪个方法不属于判别方法（ A ）

A.贝叶斯方法

B.神经网络

C.回归模型

D.Ada boosting

5.下面哪句话语较为恰当刻画了监督学习方法中生成方法的特点（ A ）

A.授之于鱼、不如授之于“渔”

B.三个臭皮匠、抵一个诸葛亮

C.毕其功于一役

D.屡败屡战、屡战屡败、最后成功

6.线性回归中优化目标函数的求取过程与下面哪一种方法是相同的（ D ）

A.最大似然估计

B.最小化后验概率

C.最大后验概率

D.最小二乘法

7.下面哪一句话对Ada Boosting的描述是不正确的（ A ）

A.在所构成的强分类器中，每个弱分类器的权重累加起来等于1（错误）

B.在所构成的强分类器中，每个弱分类器的权重是不一样的

C.该算法将若干弱分类器线性加权组合起来，形成一个强分类器

D.在每一次训练弱分类器中，每个样本的权重累加起来等于1

8.下面哪一句话对概率近似正确 (probably approximately correct, PAC)的描述是不正确的 （ C ）

A.强可学习和弱可学习是等价的，即如果已经发现了“弱学习算法”，可将其提升（boosting）为“强学习算法”

B.强可学习模型指学习模型能够以较高精度对绝大多数样本完成识别分类任务

C.强可学习模型指学习模型仅能完成若干部分样本识别与分类，其精度略高于随机猜测

D.在概率近似正确背景下，有“强可学习模型”和“弱可学习模型”

9.Ada Boosting在最小化如下目标函数（ B ）

A.0-1损失函数

B.指数损失函数

C.平方损失函数

D.绝对损失函数

10.下面对回归和分类的描述不正确的是（ D ）

A.在回归分析中，学习得到一个函数将输入变量映射到连续输出空间

B.在分类模型中，学习得到一个函数将输入变量映射到离散输出空间

C.两者均是学习输入变量和输出变量之间潜在关系模型

D.回归是一种无监督学习、分类学习是有监督学习

1.下面对逻辑斯蒂回归(logistic regression)描述不正确的是（ A ）

A.在逻辑斯蒂回归中，输入数据特征加权累加值在接近−∞或+∞附近时，模型输出的概率值变化很大

B.逻辑斯蒂回归是一种非线性回归模型

C.在逻辑斯蒂回归中，能够实现数据特征加权累加

D.逻辑斯蒂回归中所使用Sigmoid函数的输出形式是概率输出

2.逻辑斯蒂回归模型（logistic regression）属于（ A ）

A.监督学习

B.以上都不是

C.半监督学习

D.无监督学习

3.潜在语义分析属于（ D ）

A.半监督学习

B.以上都不是

C.监督学习

D.无监督学习

4.线性区别分析属于（ B ）

A.无监督学习

B.监督学习

C.以上都不是

D.半监督学习

5.逻辑斯蒂回归函数也可被称为（ A ）

A.对数几率回归(log-odds regression)

B.最大后验估计回归模型

C.最大似然估计回归模型

D.交叉熵回归模型

6.在隐性语义分析中，给定M个单词和N个文档所构成的单词-文档矩阵(term-document)矩阵，对其进行分解，将单词或文档映射到一个R维的隐性空间。下面描述不正确的是（ A ）

A.这一映射过程中需要利用文档的类别信息

B.通过矩阵分解可重建原始单词-文档矩阵，所得到的重建矩阵结果比原始单词-文档矩阵更好捕获了单词-单词、单词-文档、文档-文档之间的隐性关系

C.单词和文档映射到R维隐性空间后，单词和文档具有相同的维度

D.隐性空间维度的大小由分解过程中所得对角矩阵中对角线上不为零的系数个数所决定

7.线性区别分析(linear discriminant analysis， LDA)在进行数据降维时，原始高维数据被映射到低维空间中后需要达到的优化的目标是（ C ）

A.只要保证不同类别样本尽可能彼此远离

B.既要保证不同类别样本尽可能靠近，也要保证同一类别样本尽可能彼此远离