1 命题逻辑

1.1 命题，命题公式

命题：能判断真假（不是既真又假）的陈述句为命题。如：“雪是黑色的”，一般由小写英文字母p,q,r,s等表示。
单个常量或变量的命题称作合式公式;联结词联接的合式公式的组合也是合式公式；合式公式有限次的组合所构成的字符串成为命题公式.习惯上用大写英文字母A,B,C,D等表示

lnot否定（非），∧合取（与）,∨析取（或）,→ 蕴含（IF...THEN）,↔等价（当且仅当）
命题公式的联结词组合仍然是命题公式,$A \rightarrow B$只有A取T，B取F时结果才为F
给命题中的各变量赋值成为对该命题的一个解释,若公式中命题变量由p,q,r给出，则顺序由英文字母顺序给出,真值表：设公式 A含$n(n \geq 1)$ 个命题变量，将 A在2^n个取值下的取值情况列成表，称为 A的真值表。公式的分类:永真式（全1）、永假式（全0）、可满足式（可1）、非重言式的可满足式（可1可0）
$A \rightarrow B \Leftrightarrow \neg A \vee B$，置换规则可以充分使用
$A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$,
$A \rightarrow B \Leftrightarrow \neg B \rightarrow \neg A$,
$A \leftrightarrow B \Leftrightarrow \neg A \leftrightarrow \neg B$,
$(A \rightarrow B) \wedge (A \rightarrow \neg B) \Leftrightarrow \neg A$,
$A \Rightarrow (A \vee B)$，$(A \wedge B) \Rightarrow A$,
$((A \rightarrow B) \wedge A) \Rightarrow B$，$((A \rightarrow B) \wedge \neg B) \Rightarrow \neg A$,
$((A \vee B) \wedge \neg A) \Rightarrow B$,
$((A \rightarrow B) \wedge (B \rightarrow C)) \Rightarrow (A \rightarrow C)$,
$((A \leftrightarrow B) \wedge (B \leftrightarrow C)) \Rightarrow (A \leftrightarrow C)$,
$(A \rightarrow B) \wedge (C \rightarrow D) \wedge (A \vee C) \Rightarrow (B \vee D)$
优先顺序 ¬高于∧∨高于↔→，括号优先，同级从左向右
任意命题公式都存在着与之等值的析取范式和合取范式。
求合取范式的基本原则：利用∨对∧的分配。(将∧移动到外面来）,基本步骤：

- 削去对于¬, ∨, ∧来说冗余的联结词；
- 内移或削去否定号；
- 利用分配率。

归结原理:$p \vee q \wedge \neg q \vee r \Rightarrow p \vee r$.
子句：子句是变量（文字）的集合，各个项之间被析取分隔
子句集：逻辑公式的子句集是合取范式形式下的所有子句（元素）的集合。
归结式：有子句$C_1 = P \vee C_1'$,$C_2 = P \vee C_2'$,存在互补对P和~P，则可得归结式:$C_{12} = C_1' \vee C_2'$归结法基本原理证明$A \wedge A \wedge A \wedge \neg B$是矛盾式

1.2 谓词基本概念

个体词（小王），谓词（是个自然数）,个体常量（小写a,b,c,d）个体变量（小写x,y,z）个体域：个体变量的取值范围，用D表示
谓词常量（表示具体关系 P,Q,R,T）谓词变量（P(x),Q(x,y)...） n元谓词。一阶谓词：谓词中只含有个体词，不含有谓词。任意量词$\forall x P(x)$存在量词$\exists x P(x)$
设$P(x_1, x_2 ... x_n)$是任意的n元谓词,$t_1, t_2 ... t_n$ 是项，则称$P(t_1, t_2 ... t_n)$为原子公式。 原子公式是谓词公式,进行¬ ∨ ∧ →↔ $\forall x P(x) \exists x P(x)$后仍是谓词公式，有限次应用上述操作也是谓词公式.
换名规则：将量词辖域中出现的某个约束出现的个体变量及相应的指导变量，改成另一个此辖域中未曾出现过的个体变量符号，公式中其余部分不变。
替代规则：对某自由出现的个体变量用与原公式中所有个体变量符号不同的变量符号去替代，且处处替代
当被解释的谓词公式在特定解释下真值为真时，称这个解释满足这个谓词公式。满足一个谓词公式的解释就是这个谓词公式的模型。两个谓词公式是等价的，当且仅当在所有的解释下两个谓词公式都有相同值时。前束范式:把所有的量词都提到最左端去
$\neg(\exists x)P(x) \Leftrightarrow (\forall x)(\neg P(x))$ 含常值表达式
$\neg(\forall x)P(x) \Leftrightarrow (\exists x)(\neg P(x))$ 可以随便提
$(\exists x)(P(x) \wedge Q(x)) \neq (\exists x)P(x) \wedge (\exists x)Q(x)$,
$(\forall x)(P(x) \vee Q(x)) \neq (\forall x)P(x) \vee (\exists x)Q(x)$,
$(\exists x)(P(x) \vee Q(x)) = (\exists x)P(x) \vee (\exists x)Q(x)$,
$(\forall x)(P(x) \wedge Q(x)) = (\forall x)P(x) \wedge (\exists x)Q(x)$

1.3 谓词归结相关

Skolem标准型转换过程：

- 削去对于¬, ∨, ∧来说冗余的联结词；¬深入量词内部
- 任意量词左移，利用分配律
- 变量易名，存在量词左移，直至所有的量词移到前面，得：
- 消去存在量词∃，若存在量词左边没有任意量词，则只将其改写为常量，有就改写为任意量词的函数
- 略去任意量词∀，简单地省略掉该量词。

子句集：和命题部分基本相同，合取范式各个子式
置换是形如$\{t_1/x_1, t_2/x_2, ... t_n/x_n\}$的有限集合。(不能循环置换）
置换的合成：$\theta \cdot \lambda$ 表示先做λ 变换，再做θ变换

$$\theta = \{t_1/x_1, \ldots, t_n/x_n\}; \lambda = \{u_1/y_1, \ldots, u_n/y_n\}$$

$$\theta \cdot \lambda = \{t_1 \cdot \lambda/x_1, \ldots, t_n \cdot \lambda/x_n, u_1/y_1, \ldots, u_n/y_n\}$$

注意事项：

- $t_i \cdot \lambda = x_i$时，删去$t_i\lambda/x_i$一项（自己变成自己）
- 当$y_i \in \{x_1, x_2.x_3, ... x_n\}$时，删去$u_j/y_j$（前项已经提过）
- 剩下的元素构成新的集合

对于给定的谓词公式F_1, F_2 ,采用逐一比较找出不一致，并作相应的合一置换，则求得最一般合一置换。
归结过程（注意谓词的一致性，常量的一致性，变量与函数，不能同时消去两个互补对，使用合一来归结）

- 写出谓词关系公式,用反演法写出谓词表达式；
- 化为 Skolem 标准型求取子句集S ；
- 对 S 中可归结的子句做反复归结；
- 得到空子句,命题得证。

归结控制策略

- 宽度优先：优先合并有互补项的。
- 支持集优先策略：要归结的两个子句至少有一个是与目标公式的否定式有关的子句
- 单元子句优先每次归结时，优先选择单文字子句做归结。
- 删除策略：删除永真式/重复出现的子句/被归类的子句

2 机器学习

任务，性能标准，训练经验（已有数据集），目标函数（将当前情况映射为实数等，不是要求最小化的目标）
黑白棋中函数逼近法：后面步数的目标函数作为目前目标函数的估计，最小化估计和实际值的差别（LMS，梯度下降）$E = \sum (V - \hat{V})^2$
自学习黑白棋的组件（有种强化学习的意思）

- 执行系统：利用目标函数进行决策
- 鉴定器：对局面进行评价生成训练样本
- 泛化器：利用训练样本拟合目标函数
- 实验生成器：生成新的局面供系统搜索

3 概念学习

3.1 定义

给出某一类别的正例和反例，获得类别的一般定义（从bool函数的输入输出推出bool函数）
实例集：实际例子的集合。可能的种类数就是每个属性对应个数的乘积(D)。
概念：定义在实例集上的bool函数(c),假设(h)定义在实例集上
假设集：目标概念的集合，假设目标表现为合取形式，则每个属性对应取值增加了？(全接受）和$\oslash$（全不接受）两种取值(H)。当假设的表示形式选定后，那么也就隐含地为学习算法确定了所有假设的空间
训练样本：可以用$< x, c(x) >$描述训练样本，正例，反例等
归纳学习假设：任一假设如果在足够大的训练样本集中很好地逼近目标函数，它也能在未来的测试实例中很好地逼近目标函数。
在统计机器学习中，要求训练数据和测试数据同分布，就可以满足归纳学习假设。
假设空间的元素数量？
$h_j \geq_g h_k$，前者比后者更一般（$h_k(x) = 1 \rightarrow h_j(x) = 1$)。
$h_j >_g h_k$严格一般。
假设集元素之间用→连接，树根为(?,...?)，形成偏序关系。

3.2 FIND-S算法（寻找极大特殊假设）

思路：从最特殊假设开始，尝试覆盖正例，并在失败的时候一般化。
将h初始化为H中最特殊假设
对每个正例x
对h的每个属性约束α
- 如果x满足约束α那么不做任何操作
- 否则将h中α替换为能够令x满足的下一个更一般约束。
输出假设

利用$\geq_g$搜索，保证得到和正例一致的最特殊空间。训练数据均正确且假设正确的情况下得到结果与反例一致。
缺点：

- 不能确定学习过程收敛，不能说明为什么要用最特殊的假设
- 训练样本不一致导致完全失效
- 可能存在多个最特殊假设（2*2卡诺图1*2空间）

3.3 变形空间和候选消除算法

输出是与训练样本一致的所有假设构成的集合。
一个假设h与训练样本集合D一致， 当且仅当对中每一个样本$< x, c(x) >$都有$h(x) = c(x)$（同时包括正负样本）
变形空间：$VS_{H,D}$，H中与训练样本D一致的假设构成的子集。
列表后消除算法（太复杂）：找到所有的遍历
关于假设空间H和训练数据D的一般边界G，是在H中与D相一致的最一般成员的集合。 特殊边界S是最特殊成员的集合。（两者都是假设的集合，而不仅仅是假设）
变型空间表示定理$VS_{H,D} = \{h \in H | (\exists s \in S)(\exists s \in G)(g \geq_g h \geq_g s)\}$（需要自行判断是否存在这种夹逼）
算法执行过程

- G初始化为H中最一般假设¡?¿
- S初始化最特殊假设$< \oslash >$
- 如果d是一个正例
  从G中移去所有与d不一致的假设
  对S中每个与d不一致的假设s
  从S中移除s
  将与d一致的比s稍微一般的假设h加入S，且G的**每个**成员比h更一般
  从S中移除比其他假设更一般的假设
- 如果d是一个反例
  从S中移去所有与d不一致的假设
  对G中每个与d不一致的假设g
  从G中移除g
  将与d一致的比g稍微特殊的假设h加入G，且S的**某个**成员比h更一般
  从S中移除比其他假设更一般的假设

收敛到正确定的条件：1. 训练样例中没有错误。2. H中确实包含目标概念的正确假设。
否则可能：变形空间为空。
下一步训练样本：被变形空间中的一些假设一半分成正例，一半分成反例。不完全学习应用：一致同意，一致反对和投票置信。

3.4 归纳偏置

有偏的概念：H不包含所有的概念（如仅有合取）
无偏学习器：假设集是实例集的幂集（共有假设$2^{|X|}$种）变形空间S边界变为所有正例的析取式，G边界变为反例的析取的否定式。
无偏学习的无用性：每一个新样本都会被变型空间中刚好半数的假设划分为正例， 而被另一半划分为反例。 原因如下，若H是X的幂集，而x是某个新样本，则对于变型空间中一覆盖x的假设h，必然存在另一假设h'，它与h几乎相等，只不过对x的分类不同。而且，如果h在变型空间中，那么h'也在，因为它对于已往训练样本的划分与h完全一样。
归纳推理：考虑一般情况下任意的学习算法L以及为任意目标概念c提供的任意训练数据$D = \{< x, c(x) >\}$，训练过程结束后，L需要对新的样本进行分类。令$L(x_i, D_c)$表示在对训练数据D_c学习后L赋予w_i的类别(正例或反例)， 我们可以如下描述这一归纳推理过程：$(D_c \wedge x_i) f L(x_i, D_c)$这里的记号y f z表示z从y归纳推理得到
L的归纳偏置为这样的前提集合： 为使所有的新实例x_i满足：$(B \wedge D_c \wedge x_i) a L(x_i, D_c)$这里的记y a z表示z从y演绎派生
候选消除算法的归纳偏置： 目标概念c包含在给定的假设空间H中。偏置强度越大，泛化能力越强

- 机械式学习器(ROTE-LEARNER)： 简单地将每个观察到的训练样本存储下来。 新样本的分类通过在内存中匹配进行。 如果实例在内存中找到了， 存储的类别结果被输出。否则系统拒绝进行分类。
- 候选消除算法：新的实例只在变型空间所有成员都进行同样分类时才输出分类结果,否则系统拒绝分类。

4 其他部分

4.1 推理

消解方法的缺点：1. 推理效率低.2. 化为子句时，可能丢失原蕴涵形表达式中的控制信息。如$\neg A \rightarrow (B \vee C)$与$\neg B \rightarrow (A \vee C)$都等价于$A \vee B \vee C/$。优点：用规则进行推理，便于理解

规则系统： If Then 推理规则

- If if1
  if2
- ...

- Then then1
  then2
- ...

优点：直观，易于理解

4.2 正向演绎系统（事实驱动系统）

综合数据库：用谓词公式表示综合数据库中的知识，变成非蕴涵形的与或形表达式 $(\exists u)(\forall v)\{Q(v, u) \wedge \neg[(R(v) \vee P(v)) \vee S(u, v)]\}$
求子句步骤：消去→符号，谓词深入到每一个文字，化为斯柯林范式，删去全称量词，变量更名
一个事实表达式可由与或图表示:由变换该公式得到的子句集可作为此与或图的解图的集合（终止与叶节点）读出：也就是说，所得到的每个子句是作为解图的各个叶节点上文字的析取规则库

- 正向演绎系统使用的规则为F(forward)规则，　形如$L \to W$，L：文字，W：是公式

- 因此$L \to W$是单文字的前项规则，可用上大下小的与或图表示

复杂规则的简化

- $X \wedge Y \to Z$化成$X \to \neg Y \vee Z$ 或$Y \to \neg X \vee Z$

- $X \wedge Y \to Z$化成$X \to Z$并且$Y \to Z$

4.3　逆向演绎系统

规则：1.　事实表示为文字的合取式（数据库）.2.　规则库为B(backward)规则，$W \to L$,L为单文字.3.　单文字后向规则

复杂规则的简化:$Z \to X \wedge Y$：$Z \to X$并且$Z \to Y$
$Z \to X \vee Y$:$Z \wedge \neg X \to Y$或$Z \wedge \neg Y \to X$

任意形式的目标表达式与与或图表示,综合正向和逆向的演绎推理等

4.4　基于规则的系统

Rule-based system,也叫产生式系统（Production System）,系统构成：控制策略、产生式规则、总数据库

总数据库：综合数据库，上下文，黑板.存放初始状态，事实或证据，中间推理结果和推理的最后结果。

产生式规则:规则库:存放与求解问题有关的领域的知识

- 完整性：在任何情况下有规则可用

- 一致性：各种规则不能矛盾

- 准确性：给出比较具体的结果

控制策略:推理机构，控制系统的运行,从规则库中选择规则的策略,多条规则适合时，选择哪一条规则,把中间结论放进数据库,满足目标时结束推理,记住规则的应用过程，给出问题的解决路径

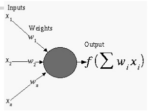
规则的冲突和解决:从匹配的几条规则中选择一条

4.5　神经网络

Total-Sum_Squared_Error(TSSE)=

$$\frac{1}{2}\sum_{pattern}\sum_{outputs}(desired - actual)^2, E = \frac{1}{2}\sum_{i=1}^{m}(d_i - y_i)^2$$

Backpropagation Preparation



- $e_j(n) = d_j(n) - y_j(n)$

$$\mathrm{E(n)} = \frac{1}{2}\sum_{j\in c} e_j^2(n) \qquad E_{AV} = \frac{1}{N}\sum_{n=1}^{N}\mathrm{E(n)}$$

$$v_j(n) = \sum_{i=0}^{p} w_{ji}(n) y_i(n) \qquad y_j(n) = \varphi_j(v_j(n))$$

$$\frac{\partial \mathrm{E(n)}}{\partial w_{ji}(n)} = \frac{\partial \mathrm{E(n)}}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)}\frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

$$\frac{\partial \mathrm{E(n)}}{\partial e_j(n)} = e_j(n),\ \frac{\partial e_j(n)}{\partial y_j(n)} = -1,\ \frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j'(v_j(n)),\ \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_j(n)$$

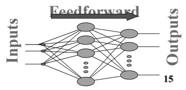
$$\frac{\partial \mathrm{E(n)}}{\partial w_{ji}(n)} = -e_j(n)\varphi_j'(v_j(n))y_j(n)$$

14

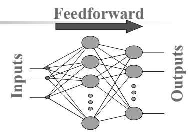
Backpropagation Preparation

- k is an output unit, j is an inner unit
- $\mathrm{E(n)} = \frac{1}{2}\sum_{k\in c} e_k^2(n)$
- $\frac{\partial \mathrm{E(n)}}{\partial y_j(n)} = \sum_k e_k(n)\frac{\partial e_k(n)}{\partial y_j(n)} = \sum_k e_k(n)\frac{\partial e_k(n)}{\partial v_k(n)}\frac{\partial v_k(n)}{\partial y_j(n)}$
- $e_k(n) = d_k(n) - y_k(n) = d_k(n) - \varphi_k(v_k(n))$
- $\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi'(v_k(n))$

$v_j - y_j - v_k - y_k - e_k$



15



- $v_k(n) = \sum_{j=0}^{q} w_{kj}(n) y_j(n)$
- $\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n)$
- $\frac{\partial \mathrm{E(n)}}{\partial y_j(n)} = -\sum_k e_k(n)\varphi_k'(v_k(n))\,w_{kj}(n) = -\sum \delta_k(n) w_{kj}(n)$

- $w_i(n+1) = w_i(n) - \eta\nabla E(w_i)$

$v_j - y_j - v_k - y_k - e_k$

16

BP应用:采用Re_LU激活函数，交叉熵。统计batch梯度下降，随机抽取训练集（非常重要），输入标准化（零均值1方差），调整学习步长，使用dropout降低过拟合

训练过程：随机初始化权重，当训练误差大于目标时：
训练样本，得到输出和输出误差，反向传播等
最后测试神经网络性能。

CNN　结构：卷积，池化，全连接.Le-Net5结构,【输入】1@32*32【卷积】6@28*28【池化】6@14*14【卷积】12@10*10【池化】12@5*5【全连】100@1*1【全连】10【输出】

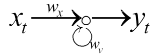
CNN适用于：信号以数组方式接受。信号有强的局部关联性，信号特征可能出现在任何位置。

1D：时间序列分析，文本。2D：音频，图片。3D：视频

Recurrent Neural Networks(RNN)

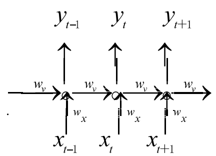
- $y_t = \phi(v_t)$

$v_t = w_v y_{t-1} + w_x x_t$

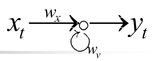


$$\frac{\partial \mathrm{E}}{\partial \mathrm{w}_v} = \sum_{t=1}^{s}\frac{\partial \mathrm{E}_t}{\partial \mathrm{w}_v}$$

$$\frac{\partial \mathrm{E}}{\partial \mathrm{w}_v} = \sum_{k=1}^{t}\frac{\partial \mathrm{E}_t}{\partial y_t}\cdot\frac{\partial y_t}{\partial v_t}\cdot\frac{\partial v_t}{\partial v_k}\cdot\frac{\partial v_k}{\partial \theta}$$



Recurrent Neural Networks(RNN)



$$\frac{\partial \mathrm{E}}{\partial \mathrm{w}_v} = \sum_{k=1}^{t}\frac{\partial \mathrm{E}_t}{\partial y_t}\cdot\frac{\partial y_t}{\partial v_t}\cdot\frac{\partial v_t}{\partial v_k}\cdot\frac{\partial v_k}{\partial \theta}, \qquad \frac{\partial \mathrm{E}_t}{\partial y_t} = \sum_{t=1}^{s}\frac{\partial(d_t - y_t)}{\partial y_t}$$

$$\frac{\partial v_t}{\partial v_k} = \prod_{i=k+1}^{t}\frac{\partial v_i}{\partial v_{i-1}} = \prod_{i=k+1}^{t}\frac{\partial v_i}{\partial y_{i-1}}\cdot\frac{\partial y_{i-1}}{\partial v_{i-1}} = \prod_{i=k+1}^{t} w_v\cdot\phi'(v_t)$$

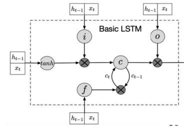
$$\frac{\partial y_t}{\partial v_t} = \phi'(v_t)$$

- $y_t = \phi(v_t)$, $v_t = w_v y_{t-1} + w_x x_t$

LSTM

$$\mathrm{i_t} = sigm(\boldsymbol{w}_{xi}x_t + \boldsymbol{w}_{hi}h_{t-1} + b_i)$$
$$f_t = sigm(\boldsymbol{w}_{xf}x_t + \boldsymbol{w}_{hi}h_{t-1} + b_f)$$
$$o_t = sigm(\boldsymbol{w}_{xo}x_t + \boldsymbol{w}_{ho}h_{t-1} + b_o)$$
$$g_t = \tanh(\boldsymbol{w}_{xg}x_t + \boldsymbol{w}_{hg}h_{t-1} + b_g)$$
$$c_t = f_t\odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t\odot\tanh(c_t)$$

$$\tanh(x) = \frac{1-\exp(-2x)}{1+\exp(-2x)}$$
双曲正切



$$\binom{x_1}{x_2}\odot\binom{y_1}{y_2} = \binom{x_1y_1}{x_2y_2}$$

LSTM

- $z = f(x_1, x_2)$
- $\frac{\partial \mathrm{E}_t}{\partial x_1} = \frac{\partial \mathrm{E}_t}{\partial z}\cdot\frac{\partial z}{\partial x_1} = \frac{\partial \mathrm{E}_t}{\partial z}\odot x_2$
- $\binom{x_1}{x_2}\odot\binom{y_1}{y_2} = \binom{x_1y_1}{x_2y_2}$

37

感知机(Perceptron)

$$x = [x_1 x_2 \ldots x_p]^T, x_j \in R, w = [w_1 w_2 \ldots w_p]^T, w_j \in R$$

$$v = \sum_{i=i}^{p} w_i x_i - \theta = W^T X - \theta, y = sgn(v)$$

输出:Sigmoid Function

$$f(x) = \frac{1}{1+e^{-\lambda x}}$$

Rectification,Piecewise linear, linear function, perceptron, work flow
多层神经网络的任意逼近能力
Classification:Binary classification, 1 output unit: y=0 or 1, Multi-class classification, K output units