集合覆盖问题的启发函数算法*

权光日 洪炳熔 叶 凤 任世军

(哈尔滨工业大学计算机科学与工程系 哈尔滨 150001)

摘要 本文给出了求解 NP 困难问题的完备策略的概念,在此基础上提出了一个求解集合覆盖问题的启发函数算法 SCHF(set-covering heuristic function),文中对该算法的合理性、时间复杂性以及解的精度进行了分析,本文的主要创新点是用已知的完备策略建立启发函数,并用该启发函数进行空间搜索求出优化解.该方法具有一定的普遍性,可以应用到其它的 NP 困难问题.它为求解 NP 困难问题的近似解提供了一种行之有效的方法.在规则学习中的应用结果表明,本文给出的 SCHF 算法是非常有效的.

关键词 启发函数,完备策略,示例学习,NP 困难问题.

中图法分类号 TP301.6

集合覆盖问题是 NP 困难问题中应用面最广的问题之一,它在模式识别、机器学习等领域中具有重要的应用.[1~9]目前已有许多比较有效的启发式算法[2~4],但是由于问题本身固有的难度,这些启发式算法在具有各自优点的同时也表露出它们各自的缺陷.本文在前人工作的基础上,提出了一种新的启发式算法——集合覆盖问题的启发函数算法 SCHF(set-covering heuristic function).通常的启发式算法是由一个或多个启发式策略组成.一般情况下,策略越多解的优化率越高,但是同时增加了计算时间复杂性.另外,启发式策略较多时,它们之间的优先级直接影响解的优化率和计算复杂性.如何确定启发式策略的优先级是在算法设计中比较棘手的问题.本文提出的启发函数方法比较成功地解决了这一难题.启发函数方法是根据已知的启发式策略合理地建立它们的优先级函数 $F(x)$ 的方法.算法根据启发函数的值进行相应的操作.

1 相关概念

为了讨论方便,我们首先给出一些将要使用的概念的定义.

最小集合覆盖问题:S 是一个集合,$S_1,S_2,\ldots,S_m$ 是 S 的子集,且构成 S 的覆盖,即 $\bigcup_{i=1}^{m} S_i=S$,求最小的覆盖.

定义 1. x 是属于 S 的任一取定的元素,如果 x 当且仅当只属于 $S_1,S_2,\ldots,S_m$ 的 K 个集合,则称 x 的频率为 K,用 $K(x)$ 表示.

例 1:设 $S=\{0,1,2,3,4,5,6,7,8,9\}$;$S_1=\{0,1,2,3,4\}$;$S_2=\{0,1,5,6,7\}$;$S_3=\{3,4,5,6,7\}$;$S_4=\{5,6,7,9\}$;$S_5=\{2,8,9\}$;$S_6=\{1,3,5,7\}$;$S_7=\{0,4,6,8\}$;$S_8=\{0,3,6\}$;$S_9=\{1,4,7\}$.

表 1 是例 1 的特征矩阵,最后一列表示 S 中每个元素的频率 $K(x)$,第 12 行表示 S_i 的基数.

定义 2. S_i 是 $S_1,S_2,\ldots,S_m$ 中的一个集合,令 $P(S_i)=\mathrm{Min}\{K(x);x\in S_i\}$,称 $P(S_i)$ 为 S_i 的覆盖度,空集的覆盖度定义为 $m+1$.

由例 1 来说,$P(S_1)=2,P(S_2)=4,P(S_3)=4,P(S_4)=2,P(S_5)=2,P(S_6)=4,P(S_7)=2,P(S_8)=4,P(S_9)=3$.

定义 3. 在 $S_1,S_2,\ldots,S_m$ 中如果排出一个集合 S_i 之后,出现 R 个必选的集合(覆盖度为 1 的集合),则称 S_i 的必选度为 R,用 $R(S_i)$ 表示.

由例 1 来说,$R(S_1)=1,R(S_2)=0,R(S_3)=0,R(S_4)=1,R(S_5)=3,R(S_6)=0,R(S_7)=1,R(S_8)=R(S_9)=0$.

定义 4. 设 NPH(NP-hard)是一个给定的 NP 困难问题(例如,集合覆盖问题、TSP 等),ST(strategy)是一个求解 NPH 的策略,如果 ST 的前提条件的验证以及策略的操作皆可在多项式时间内完成,则称 ST 为多项式时间策略.

---

* 本文研究得到国防科工委"九五"攻关项目基金资助.
作者权光日,1962 年生,博士生,副教授,主要研究领域为人工智能,NP 完全问题.洪炳熔,1937 年生,博士,教授,博士导师,主要研究领域为智能机器人.叶凤,1960 年生,博士生,工程师,主要研究领域为人工智能.任世军,1962 年生,博士生,副教授,主要研究领域为人工智能,图论.

本文通讯联系人:权光日,哈尔滨 150001,哈尔滨工业大学计算机科学与工程系智能机器人研究室

本文 1996-10-03 收到原稿,1997-11-20 收到修改稿

一般的启发式策略都是多项式时间策略.例如,就集合覆盖问题来说,选取最大基数的集合作为覆盖中的一员(贪心法);就TSP来说,选择最邻近的点构成回路等都是多项式策略.

表1　例1的特征矩阵

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	$K(x)$
0	1	1	0	0	0	0	1	1	0	4
1	1	1	0	0	0	1	0	0	1	4
2	1	0	0	0	1	0	0	0	0	2
3	1	0	1	0	0	1	0	1	0	4
4	1	0	1	0	0	0	1	0	1	4
5	0	1	1	1	0	1	0	0	0	4
6	0	1	1	1	0	0	1	1	0	5
7	0	1	1	1	0	1	0	0	1	5
8	0	0	0	0	1	0	1	0	0	2
9	0	0	0	1	1	0	0	0	0	2
$\|S_i\|$	5	5	5	4	3	4	4	3	3	
$P(S_i)$	2	4	4	2	2	4	2	4	3	
$R(S_i)$	1	0	0	1	3	0	1	0	0	

定义5. NPH是一个NP困难问题,ST是求解NPH的一个多项式时间策略,如果NPH的实例满足ST条件时,经过ST操作之后问题的实例规模比原来的小,并且问题的新实例的最优化解都是原问题实例的优化解,则称ST为完备策略.

通俗地讲,一个完备策略是,当问题的实例满足策略条件时,把该实例化简为更易解的实例,并且新实例的最优化解都是原实例的最优化解,即不增加最优化解的策略.

在求解NP困难问题时,用完备策略既能降低问题的求解难度,又能保留一些(不增加)最优解,所以它是求解NP困难问题的最重要的策略(如果有的话),因此用启发式算法求解NP困难问题时,我们首先应该考虑寻找完备策略.

集合覆盖问题的完备策略:

完备策略1. 如果$S_1,S_2,\ldots,S_m$中的一个集合$S_i=S$,则选择S_i作为最优化覆盖中的唯一一个集合S_i.

完备策略2. 如果存在$x\in S$,x只属于$S_1,S_2,\ldots,S_m$中的一个集合S_i,即$P(S_i)=1$,则选择S_i作为最优化覆盖中的一个集合S_i.

完备策略3. 如果$S_i\subseteq S_j$,则排出S_i.

不难看出以上3个策略都是多项式时间策略(具体的时间复杂性估计在第3节中给出).

下面我们证明这3个策略满足定义5.

策略1显然满足定义5.

考虑策略2.当问题的实例满足策略2的条件时,即存在$x\in S$只属于覆盖中的一个集合,不妨设$x\in S_i$,则S_i一定在任何一个覆盖中(否则不能形成一个覆盖),于是S_i属于任何一个最优化覆盖中,这就是说,选择S_i之后求解难度降低了,但还保留着所有最优化覆盖,故策略2是完备策略.

考虑策略3.问题的实例满足策略3的条件时,即存在两个集合S_i,S_j满足$S_i\subseteq S_j$,如果一个最优覆盖含有S_i,则在该覆盖中,用S_j来代替S_i后仍然是一个最优化覆盖,故排除S_i之后问题的难度降低了,但不增加最优化覆盖,因此策略3是完备策略.

一般情况下,策略的个数越多优化率越高,但同时带来了时间复杂性.即使同一组策略来说,策略的次序不同,解的精度和时间复杂性也不同.对于给定的一组策略,怎样构造优化率高、时间复杂性低的算法呢?在下一节中给出的启发函数算法,在集合覆盖问题上很好地解决了这方面的问题.启发函数的构造以及启发函数算法的构造具有一定的普遍性,它为求解NP困难问题的近似解,提供了一种新的行之有效的方法.

2　启发函数与启发函数算法

上节给出了3个完备策略和3个启发式策略.本节先给出构造启发函数的基本准则,然后给出启发函数算法,最后对算法进行合理性分析.

构造启发函数的基本准则.

(1) 给出一个 $S_1,S_2,\ldots,S_m$ 到实数轴 R 上的映射 $F(X)$,当 S_i 的排除优先级比 S_j 的排除优先级高时,$F(S_i)<F(S_j)$.

(2) 当满足完备策略 1 或 2 时,$F(S_i)$最大,并且 $F(S_i>L)$. L 是一个给定的常数.

(3) 当$|S_i\bigcup S_j-S_i\bigcap S_j|<<|S_i\bigcap S_j|$时,$F(S_i)\cong F(S_j)$.

(4) 当 $P(S_i)\cong P(S_j)$,$R(S_i)\cong R(S_j)$,$|S_i|\cong|S_j|$ 时,$F(S_i)\cong F(S_j)$.

根据以上的 4 条构造准则,我们构造如下启发函数.

$$F(S_i)=a\frac{1+R(S_i)}{P(S_i)-1}+b\frac{1}{N-|S_i|} \tag{1}$$

其中 a 和 b 是待定的参数,$N=|S|$,$L=a(1+N)+b$.

启发函数的性质.

性质 1. 如果覆盖中的集合 S_i 满足完备策略 1 和 2,则 $F(S_i)=\infty$. 事实上,S_i 满足完备策略 1 时,$N=|S_i|$,故 $F(S_i)=\infty$. S_j 满足完备策略 2 时,$P(S_j)=1$,故 $F(S_j)=\infty$.

性质 2. 如果 $S_i\subseteq S_j$,则 $F(S_i)\geq F(S_j)$. 事实上,$S_i\subseteq S_j$ 时,$P(S_i)\leq P(S_j)$,$|S_i|\geq|S_j|$,$R(S_i)\geq R(S_j)$,故 $F(S_i)\geq F(S_j)$.

性质 3. 如果$|S_i|=|S_j|$并且 $1<P(S_i)<P(S_j)$,则 $F(S_i)\geq F(S_j)$. 事实上,$1<F(S_i)<P(S_j)$时,$P(S_j)\geq 3$,$R(S_i)\geq R(S_j)=0$,故必有 $F(S_i)\geq F(S_j)$.

性质 4. 如果 $P(S_i)=P(S_j)$,$|S_i|=|S_j|$,$R(S_i)>R(S_j)$,则 $F(S_i)>F(S_j)$. 这是显然的.

性质 5. 如果 $P(S_i)=P(S_j)$,$R(S_i)=R(S_j)$,$|S_i|>|S_j|$,则 $F(S_i)<F(S_j)$. 事实上,这也是显然的.

性质 6. 启发函数(1)还满足构造启发函数的基本准则(3)和(4). 即启发函数(1)具有"连续性".

总之,启发函数(1)在包含了上述 3 个完备策略的基础上"连续"地扩展了完备策略的操作范围. 图 1 表示文献[3]中给出的基于上述 3 个完备策略的启发式算法 HCV[2,3]与本文给出的启发函数算法 SCHF 之间的关系.

(a) 启发式算法HVC的示意图 (b) 启发函数算法SCHF的示意图

图1 启发式算法HVC与启发函数算法SCHF的关系

图 1(a)中 if A then $U(\ )$表示完备策略 1;if B then $V(\ )$表示完备策略 2;if C then $W(\ )$表示完备策略 3;而 if D then $X(\ )$表示"贪心"策略(其中 $D=S-A-B-C$).

下面我们用启发函数(1),建立集合覆盖问题的启发函数算法 SCHF.

初值,$COVER=\{\Phi\}$,$COVER0=\{S_1,S_2,\ldots,S_m\}$;

第 1 步. 求出使 $F(S_i)$最大的 S_{i0}和使 $F(S_i)$最小的 S_{j0}(其中 $S_i,S_{i0},S_{j0}\in COVER0$);

第 2 步. 如果 $F(S_{i0})>L$,则 $COVER=COVER+\{S_{i0}\}$,$S=S-S_{i0}$,$COVER0=COVER0-\{S_{i0}\}$,
否则 $COVER0=COVER0-\{S_{j0}\}$;

第 3 步. 如果 $S=\Phi$,则输出 $COVER$ 之后停机,否则返回第 1 步.

以例 1 为例说明 SCHF 的求解过程. 首先取定参数 $a=b=1$,$L=12$. 第 1 步,经计算得 $F(S_1)=2.5$,$F(S_2)=0.53$,$F(S_3)=0.53$,$F(S_4)=2.17$,$F(S_5)=4.14$,$F(S_6)=0.5$,$F(S_7)=2.17$,$F(S_8)=0.48$,$F(S_9)=0.64$. 算法 SCHF 删除 S_8. 第 2 步,删除 S_6;第 3 步,删除 S_9;第 4 步,删除 S_2;第 5 步,选择 S_1;第 6 步,删除 S_3;第 7 步,选择 S_4;第 8 步,选择 S_5 结束. 最后得到的最小覆盖是 S_1,S_4,S_5. 不难看出 S_1,S_4,S_5 是例 1 的一个最优解.

下面分析 SCHF 算法的合理性. NP 完全理论中有一个著名猜想——任何一个 NP 困难问题都没有多项式时间算法. 虽然,这一猜想尚未得到证明,但学术界大多数学者认为该猜想是成立的. 集合覆盖问题是 NP 困难问题,所以我们不能追求总能求出最优解的多项式时间算法,这是不现实的. 评价一个启发式算法(求解 NP 困难问题近似解的)的优劣主要看它的计算复杂性(时间复杂性与空间复杂性)与优化能力. 估计优化能力的常用方法有两种:①用数学方法严格地进行估计解的范围;②通过随机实验进行分析解的优化率. 下面我们给出 3 种算法(SCHF,HVC 和"贪心"算法)同时进行的随机实验结果,而有关 SCHF 算法的计算复杂性估计的结果在下一节中给出.

集合{0,1,...,98,99}中随机抽出 m 个子集合 $S_1,S_2,\ldots,S_m$,它们的并 $S=\bigcup_{i=1}^{m}S_i$ 作为被覆盖的集合,这样得到的

最小集合覆盖问题的实例记作 $INSTANCE(S_1,S_2,\ldots,S_m)$,为了方便起见,简记为 $INSTANCE(m)$. 图 2 表示对于每个取定的 $m(m=10,20,\ldots,90,100)$,我们从集合{0,1,...,98,99}中随机地抽出 100 个实例 $INSTANCE(m)$进行实验得到的结果. 其中"□"表示 100 次随机实验中"贪心"算法的解不次于 SCHF 和 HCV 算法的解的次数(即"贪心"算法选定的集合个数不多于 SCHF 和 HCV 选定的集合个数);"△"表示 100 次随机实验中 HCV 算法的解不次于 SCHF 和"贪心"算法的解的次数(即 HCV 算法选定的集合个数不多于 SCHF 和"贪心"算法选定的集合个数);"○"表示 100 次随机实验中 SCHF 算法的不次于 HCV 和"贪心"算法的解的次数(即 SCHF 选定的集合个数不多于 HCV 和"贪心"算法选定的集合个数).

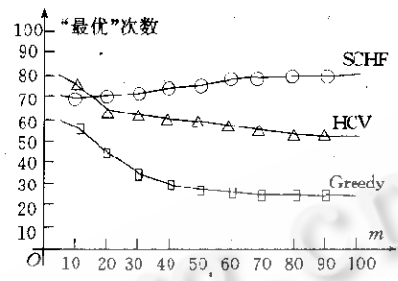

图2　表示SCHF,HCV和"贪心"算法的解的优化率

结果表明,随着 m 的增大,SCHF 算法得到的解最优(最优解是指上述 3 个算法得到的 3 个解中集合个数最少的解)的"概率"接近 78%,而 HCV 和"贪心"算法的解最优的"概率"分别趋向于 51%和 18%. 其中 78%−51%+18%=147%>100%是因为对于有些实例,上述 3 个算法得到的 3 个解中有 1 个以上"最优"解的缘故.

3　算法的计算复杂性

下面分析 SCHF 算法的计算复杂性.

(1) 对于取定的 x,$K(x)$的计算需要 m 次(m 是集合的个数),故所有 $K(x)$的计算需要 nm 次.

(2) 集合 S_i 的基数最大为 n,故 $P(S_i)$计算最多是 n 次. $|S_i|$的计算最多需要 n 次,故计算所有的 $P(S_i)$和$|S_i|$最多需要 $2nm$ 次.

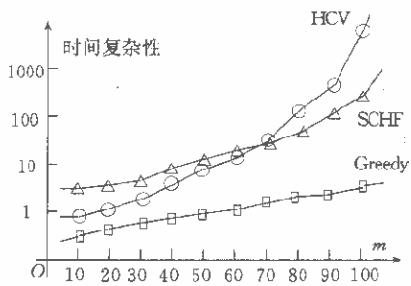

图3　表示SCHF,HCV和"贪心"算法的计算复杂性

(3) $R(S_i)$的计算可以按下面的方法进行. 先求出含有元素 x 的所有集合,构成一个集合号组成的集合记作 $Sn(x)$,计算所有的 $Sn(x)$和$|Sn(x)|$需要 $2nm$ 次的计算. 用 $Sn(x)$计算 $R(S_i)$,对于取定的 S_i,$R(S_i)$的计算最多需要 $O(n+m)$次,计算所有的 $R(S_i)$最多需要 $2nm+O(m(n+m))$次(实际上,$R(S_i)$是只有 $P(S_i)=2$ 时才计算的,所以实际计算量比理论上的小得多).

(4) 对于取定的 S_i,函数 $F(S_i)$的计算只需要 7 次算术运算,求所有的 $F(S_i)$最多需要 $O(m)$次.

(5) 求最大值和最小值的计算需要 $2m-2$ 次.

(6) 选择或者排除操作最多需要 $O(mn)$.

现在我们可以计算出从 $K(x)$的计算开始一次选择或者排除结束为止的计算量

$$\begin{aligned}T(m,n)&=nm+2nm+2nm+O(m(m+n))+O(m)+2(m-1)+O(nm)\\&=O(m(n+m))\end{aligned}$$

总的计算量

$$\begin{aligned}T&\leq T(m,n)+T(m-1,n)+\ldots+T(2,n)+T(1,n)\\&\leq O(m(n+m))+O((m-1)(n+m-1))+\ldots-O(n)\\&\leq O(nm^2+m^3)\end{aligned}$$

因此,SCHF 算法的计算复杂性上界为 $O(nm^2+m^3)$,这比文献[2,3]中的 HCV 算法降低了一个关于 m 的阶. 图 3 给出了 SCHF,HVC 和"贪心"算法的计算时间复杂性曲线.

4　SCHF 算法在规则学习中的应用

在把集合覆盖问题的启发函数算法应用到规则学习算法的选择子的优化方面,我们提出了一种新的规则学习算法——基于扩张矩阵的启发函数学习算法 AEF. 目前,规则学习方面生成的公式数较少的算法是 AE9.[7]AEF 算法中正例子的复合策略采用了 AE9 中的策略. AEF 算法的具体描述如下:

(1) 按文献[7]中的方法从正例子集 PE 中选择一个种子 e, $F=\{e\}$, $path=\Phi$, $CPE=\Phi$.

(2) 作 F 的扩张矩阵 $EM(F)$, 若有必选元素则放入 $path$ 中, 删去 NE 中出现该必选元素的行(反例). 若 NE 空, 则终止; 若非空, 则删去 PE 和 CPE 中出现该必选元数的对应行. 重复执行 2, 直至 $EM(F)$ 中不存在必选元数为止.

(3) 若 PE 非空, 检查 PE 中的每一正例, 看它与 F 的合并是否一致公式. 若不是, 则从 PE 中删除该正例; 若是, 则保留覆盖正例数目最多的一个合并取代 F, 将 PE 中被 F 覆盖的正例放入 CPE 中. 重复步骤 2、3, 直至 PE 变空.

(4) 如果 PE 空而 CPE 非空, 则检查 CPE 中的每一个正例, 看它与 F 合并后是否为一致公式. 若不是, 则从 CPE 中删去该正例; 若是, 则生成合并公式, 保留一个覆盖最多的正例的合并并取代 F, 从 CPE 中删去被新的 F 覆盖的正例. 重复 2~4, 直至 CPE 变空.

(5) 此时 PE 和 CPE 均空, 但 NE 非空. 作 $EM(F)$, 用集合覆盖问题的启发函数算法 SCHF 求优化公式.

AE9 算法和 AEF 算法之间的区别在于第 5 个策略. 我们从睡眠例子[7]中随机抽出一些, 进行了选择子数和正确识别率方面的比较实验, 结果如表 2 所示.

表 2 睡眠例子的实验结果

抽取的例子数		500	600	700	800	900	1 000	1 100	1 200	1 236
AE9	选择子数	81	92	103	124	146	189	234	257	1 263
	识别率%	76	80	83	85	89	93	97	99	100
AEF	选择子数	74	80	93	109	121	168	211	221	1 236
	识别率%	84	87	89	93	95	97	98	99	100

表 2 中的识别率是对总的样本(1 236 个)的正确识别率. 选择子数是所有公式中的选择子的总和.

上述结果表明, 通过少量例子的训练建立的识别系统, AEF 算法的正确识别率明显高于 AE9. 这表明本文给出的 SCHF 算法是非常有效的. SCHF 算法还可以应用到最小特征提取、模式匹配等许多方面, 在这里不一一列举. 有关启发函数方法在其他 NP 困难问题中的应用(如图着色问题、任务分配问题、背包问题等)将另文发表.

参考文献

1 Li Guo-jie. Research on computational complexity of artificial intelligence. Pattern Recognition and Artificial Intelligence, 1992,5(3):1~9

2 Wu X. Optimization problems in extension matrixes. Science in China(Series A), 1992,35(3):363~373

3 Wu X. Inductive learning: algorithms and frontiers. Artificial Intelligence, 1993,7:93~108

4 Chvatal V. A greedy heuristic for the set-covering problem. Mathematics of Operations Research, August 1979,4(3):123~129

5 Hong Jia-rong. Learning from examples and multifunctional learning system AE5. Chinese Journal of Computers, 1989,12(2):78~82

6 Hong Jia-rong. Theory of extension matrixes in learning from examples. Chinese Journal of Computers, 1991,14(6):37~42

7 Zhao Mei-de, Hong Jia-rong. An algorithm of generalized extension matrixes in learning from examples and implementation. Chinese Journal of Computers, 1994,17(9):83~88

8 Chen Bin, Hong Jia-rong, Wang Ya-dong. Minimum feature subset selection problem. Journal of Computer Science Technology, 1997,12(2):63~67

9 Chen Bin, Hong Jia-rong. Maximum composition problem in learning from examples and algorithm. Chinese Journal of Computers, 1997,20(2):87~91

A Heuristic Function Algorithm for Minimum Set-covering Problem

QUAN Guang-ri HONG Bing-rong YE feng REN Shi-jun

(*Department of Computer Science and Engineering Harbin Institute of Technology Harbin* 150001)

Abstract In this paper, an algorithm based on heuristic function for minimum set-covering problem is presented, as well as the definition of complete strategy concept and the method to structure heuristic function. The rationality, the time complexity and the precision of the solution are discussed for the algorithm. The basic idea is to structure heuristic function with the given heuristic strategies. The method can apply to other NP hard problems. As application of the algorithm, this paper presents a new algorithm of learning rule from the examples based on heuristic function.

Key words Heuristic function, complete strategy, learning from examples, NP hard problem.

Class number TP301.6