

基于改进差别矩阵的核增量式更新算法

赖桃桃, 冯少荣, 张东站

(厦门大学 计算机科学系, 厦门 361005)

(laitt_ct@163.com)

摘要: 分析发现杨明教授给出的改进的差别矩阵中存在不必要的计算, 为此提出了改进的差别矩阵定义和求核方法; 在此基础上提出一种基于改进差别矩阵的核增量式更新算法, 主要考虑对象动态增加情况下核的更新问题。理论分析表明改进的核增量式更新算法具有近线性时间和空间复杂度。实验结果显示算法有效可行。

关键词: 粗糙集; 差别矩阵; 核; 增量式更新

中图分类号: TP311 **文献标志码:** A

Incremental updating algorithm for computing core based on improved discernibility matrix

LA I Tao-tao, FENG Shao-rong, ZHANG Dong-zhan

(Department of Computer Science, Xiamen University, Xiamen Fujian 361005, China)

Abstract: Through analysis, it was found out that the improved discernibility matrix presented by Professor Yang Ming had unnecessary calculations. Therefore, an improved discernibility matrix definition together with a method for computing core was introduced. The authors introduced an incremental updating algorithm for computing core based on improved discernibility matrix, which mainly considered core updating when objects dynamically increased. Theoretical analyses show that incremental updating algorithm for computing core has nearly linear time and space complexity; and the experimental results show that the algorithm is efficient and effective.

Key words: rough set; discernibility matrix; core; incremental updating

0 引言

由波兰数学家 Z. Pawlak 1982 年提出的粗糙集理论^[1], 已经在机器学习、数据挖掘等领域中得到了较为广泛的应用。决策表信息系统是粗糙集理论的主要研究对象, 决策表信息系统的约简是所有粗糙集理论和应用研究的焦点问题之一。决策表核属性的确定往往是信息约简的基础, 因而探索研究求核的有效方法具有重要的实用价值。

目前有不少求解核的方法, 其中 Hu 提出的基于差别矩阵的求解核方法是经典的核求解方法之一^[2]。Hu 方法在某些情况下不能得到正确的核, 叶东毅等人在 Hu 的差别矩阵定义基础上, 提出新的差别矩阵并证明其求核方法是正确的^[3], 但计算量大, 为此杨明等人^[4]提出了一种新的改进的差别矩阵及其求核方法, 该方法在纠正 Hu 方法的错误的同时, 可有效地降低计算代价, 对它进行分析之后, 提出更有效的方法。在文献 [4] 的基础上, 杨明等人^[5]提出计算核的增量式更新算法, 但是算法的时间和空间复杂度较高, 本文提出了改进的增量式更新算法, 有效地降低时间和空间复杂度。

1 粗糙集理论的相关概念和理论

为了便于叙述, 先对信息系统的属性约简以及属性核的概念进行简单介绍。

定义 1 一个决策表信息系统 $S = (U, A, V, f)$, 其中, U 为论域, $A = C \cup D$, $C \cap D = \emptyset$, C 为条件属性集, D 为结果属性集, 一般只含有一个属性, $V = \{V_r \mid r \in A\}$ 是属性值的集合, V_r 表示 $r \in A$ 的值域, $f: U \times A \rightarrow V$ 是一个信息函数, 它指定 U 中每个对象 x 的属性值。

定义 2 给定一个决策表信息系统 $S = (U, C \cup D, V, f)$, 对于 $X \subseteq U$ 和 $R \subseteq C$, X 的下近似集与 X 的上近似集分别定义为 $R_-(X) = \{Y \mid Y \subseteq U/R \text{ 且 } Y \subseteq X\}$ 和 $R_+(X) = \{Y \mid Y \subseteq U/R \text{ 且 } Y \cap X \neq \emptyset\}$ 。

在信息系统 (或决策表) 中, 若一些数据具有相同的条件属性而具有不同的分类, 则称这类数据是不一致的, 否则为一致的。

不失一般性, 假设仅有一个决策属性 D , 其取值范围是 $1, 2, \dots, k$, 由 D 导出的等价类构成划分: $\{c_1, c_2, \dots, c_k\}$, 其中 $c_i = \{x \in U \mid f(x, D) = i\}$, $i = 1, 2, \dots, k$ 。

定义 3 设 $P \subseteq C$, 对划分 $\{c_1, c_2, \dots, c_k\}$ 的 P -近似精度为 $\rho_P = \frac{\sum_{i=1}^k |P_+(c_i)|}{|U|}$ 。

定义 4 设 $P \subseteq C$, 若 $\rho_P = \rho_C$, 且不存在 $R \subset P$, $\rho_R = \rho_C$, 则称 P 为 C 的一个 (相对于决策属性 D 的) 属性约简。所有 C 的属性约简的交称为 C 的核, 记为 $Core(C)$ 。

定义 5 如果属性 $a \in C$ 满足 $\rho_{C-\{a\}} < \rho_C$, 则称属性 a 为不可缺少的, 否则, 称属性为冗余的。

性质 1 属性 $a \in Core(C)$ 当且仅当 a 是不可缺少的属性。

2 已有的差别矩阵定义及其改进

为有效求核, HU 等人提出一种简洁的利用改进差别矩阵来确定核的方法, 但得出的结论在某些情况下是错误的, 如该方法不能得到例 1 的核, 却可得到例 2 的核, 详见文献 [3]。

例 1 表 1 为二值数据表, 其中共有 5 个元素和 4 个属性, $C = \{C_1, C_2, C_3\}$ 为条件属性集, D 为决策属性。

收稿日期: 2009 - 03 - 24; 修回日期: 2009 - 05 - 17。 基金项目: 国家自然科学基金资助项目 (50604012)。

作者简介: 赖桃桃 (1985 -), 男, 福建长汀人, 硕士研究生, 主要研究方向: 数据挖掘; 冯少荣 (1964 -), 男, 河北南宫人, 副教授, 博士, 主要研究方向: 数据挖掘; 张东站 (1974 -), 男, 江苏新沂人, 副教授, 博士, 主要研究方向: 数据挖掘。

例 2 在例 1 中删除第 1 个对象 (即第 1 条记录) 即得到表 2。

表 1 数据表 (I)

| 元素 | 属性 | | | |
|-------|----|----|----|---|
| | C1 | C2 | C3 | D |
| x_1 | 1 | 0 | 1 | 1 |
| x_2 | 1 | 0 | 1 | 0 |
| x_3 | 0 | 0 | 1 | 1 |
| x_4 | 0 | 0 | 1 | 0 |
| x_5 | 1 | 1 | 1 | 1 |

表 2 数据表 (II)

| 元素 | 属性 | | | |
|-------|----|----|----|---|
| | C1 | C2 | C3 | D |
| x_1 | 1 | 0 | 1 | 0 |
| x_2 | 0 | 0 | 1 | 1 |
| x_3 | 0 | 0 | 1 | 0 |
| x_4 | 1 | 1 | 1 | 1 |

针对 HU 方法的缺陷,叶东毅等人^[3]提出了新的差别矩阵定义并给出求核方法,但计算量大。为改进文献 [3] 的不足,杨明等人^[4]提出了改进的差别矩阵定义以及求解核方法,详细内容可参见文献 [4]。杨明等人^[5]在文献 [4] 的基础上,提出了基于差别矩阵的核增量式更新算法。

定义 6^[5/409] 对给定的信息系统 IS, 定义差别矩阵 $M_2 = \{m_{ij}\}$ 为:

$$m_{ij} = \begin{cases} \{a \in C: f(x_i, a) = f(x_j, a)\}, \\ f(x_i, D) = f(x_j, D) \text{ 且 } x_i \in U_1, x_j \in U_1 \\ \{a \in C: f(x_i, a) = f(x_j, a)\}, \\ x_i \in U_1, x_j \in U_2 \\ \emptyset, \text{ 其他} \end{cases} \quad (1)$$

其中: $U_1 = \bigcup_{i=1}^k C_i$, $U_2 = U - U_1$, $U_2 = delrep(U_2)$ 。函数 $delrep(U_2)$ 描述如下:

```

begin
U2 = ∅;
for 任意 x ∈ U2 do
if 不存在 y ∈ U2 使得 a ∈ C, f(x, a) = f(y, a) then
U2 = U2 ∪ {x};
return U2;
end
    
```

定理 1^[5/401] 对于信息系统 IS, 若记 $DM(C, M_2) = \{m_{ij} / m_{ij} \in M_2 \text{ 且 } m_{ij} \text{ 为单个属性}\}$, 则有 $DM(C, M_2) = Core(C)$, 即当且仅当某个 m_{ij} 为单个属性时, 该属性属于核 $Core(C)$ 。

对实例 2, $U_1 = \{x_1, x_4\}$, $U_2 = \{x_2, x_3\}$, $U_2 = \{x_2\}$ 。依据定义 6 建立的差别矩阵为:

$$\begin{matrix} & x_1 & x_4 & x_2 \\ x_1 & \emptyset & \{C2\} & \{C1\} \\ x_4 & \{C2\} & \emptyset & \{C1, C2\} \end{matrix}$$

由这个例子可以看出, 当 $x_i \in U_1, x_j \in U_1$ 时 $m_{ij} = m_{ji}$ 的, 式 (1) 计算了 m_{ij} 及 m_{ji} , 我们只需要计算 m_{ij} 就能正确的求核, 所以对 m_{ji} 的计算是多余的。当 $|U_1|$ 较大时, 这种多余的计算量就很大, 特别当决策表一致时, 增加的不必要计算量是求核实际所需计算量的一倍。

故当 $x_i \in U_1, x_j \in U_1$ 时, 通过限定 $i > j$, 只计算 m_{ij} , 而不计算与 m_{ij} 对称且相等的 m_{ji} , 给出定义 6 的改进且与其等价的定义如下。

定义 7 对给定的信息系统 IS, 定义差别矩阵 $M_3 = \{m_{ij}\}$ 为:

$$m_{ij} = \begin{cases} \{a \in C: f(x_i, a) = f(x_j, a)\}, \\ f(x_i, D) = f(x_j, D) \text{ 且 } x_i \in U_1, x_j \in U_1, i > j \\ \{a \in C: f(x_i, a) = f(x_j, a)\}, x_i \in U_1, x_j \in U_2 \\ \emptyset, \text{ 其他} \end{cases} \quad (2)$$

其中 U_1, U_2, U_2 同定义 6。

定理 2 上对于信息系统 IS, 若记 $DM(C, M_3) = \{m_{ij} / m_{ij} \in M_3 \text{ 且 } m_{ij} \text{ 为单个属性}\}$, 则有 $DM(C, M_3) = Core(C)$, 即当且仅当某个 m_{ij} 为单个属性时, 该属性属于核 $Core(C)$ 。

证明 由 $DM(C, M_3)$ 与 $DM(C, M_2)$ 相等。证毕。

3 已有的核增量式计算方法

定理 3^[5/411] 对于信息系统 IS, 若记 $DMSC(C) = \{(m, count) : m \in M_2 \text{ 且为单个属性}, count(1) \text{ 为 } m \text{ 在 } M_2 \text{ 中出现的次数}\}$, $DM(C, M_2) = \{m : (m, count) \in DMSC(C)\}$, 其中, M_2 为由定义 6 得到的差别矩阵, 则有 $DM(C, M_2) = Core(C)$, 即当且仅当某个 m 为单个属性时, 该属性属于核 $Core(C)$ 。

当信息系统 (或决策表) 动态增加对象时, 动态更新差别矩阵并对 $DMSC(C)$ 中各属性出现次数进行修改。文献 [5] 把它分为三种情况分别进行讨论, 并提出了两个算法, 分别为算法 1 和算法 2, 算法 2 优于算法 1, 本文对算法 2 进行改进, 两算法的具体情况参见文献 [5]。

新增对象时如何修改差别矩阵, 文献 [5] 举了以下三个例子, 如下所示。

1) 对实例 2, 若新增对象 x 为 $(1, 0, 0, 0)$, 则由算法 1 可得 $U_1 = \{x_1, x_4, x\}$, $U_2 = \{x_2\}$, $M_2(x)$ 为:

$$\begin{matrix} & x_1 & x_4 & x & x_2 \\ x_1 & \emptyset & \{C2\} & \emptyset & \{C1\} \\ x_4 & \{C2\} & \emptyset & \{C2, C3\} & \{C1, C2\} \\ x & \emptyset & \{C2, C3\} & \emptyset & \{C1, C3\} \end{matrix}$$

2) 对实例 2, 若新增对象 x 为 $(1, 0, 1, 1)$, 则由算法 1 可得 $U_1 = \{x_4\}$, $U_2 = \{x_1, x_2\}$, $M_2(x)$ 为:

$$\begin{matrix} & x_1 & x_4 & x_2 \\ x_4 & \{C2\} & \emptyset & \{C1, C2\} \end{matrix}$$

3) 对实例 2, 若新增对象 x 为 $(0, 0, 1, 1)$, 则由算法 1 可得 $U_1 = \{x_1, x_4\}$, $U_2 = \{x_2\}$, $M_2(x)$ 不变。

由于本文提出的算法不存储差别矩阵, 针对文献 [5] 的三种情况本文分别做如下处理。

x 与 $(U_1 \cup U_2)$ 一致, 计算 x 与 $U_1 \cup U_2$ 间的 m_{ij} , 并按规则增加至 $DMSC(C)$, $U_1 = U_1 \cup \{x\}$ 。具体的规则见算法 2。

x 与 U_1 不一致时, 在 U_1 中找出与 x 不一致的 y , 计算 y 与 $U_1 \cup U_2$ 之间的 m_{ij} , 并把 $DMSC(C)$ 中的相应的 m_{ij} 按规则删除; $U_2 = U_2 \cup \{y\}$, $U_1 = U_1 - \{y\}$; 计算 y 与 U_1 之间的 m_{ij} , 按规则增加至 $DMSC(C)$ 。

x 与 U_2 不一致时, $DMSC(C)$ 保持不变。

4 新的求核方法及核增量式算法

由上面的准备工作我们给出新的新的求核方法及核增量式算法。

算法 1 求核算法

输入: U_1, U_2 (其中 U_1, U_2 同定义 4)

输出: $DMSC(C)$

```

begin
  for each  $x_i \in U1$  do {
    for each  $x_j \in U1 \cup U2$  do {
       $m_{ij} = \emptyset$ ;
       $flage = 0$ ;
      if ( $(x_j \in U1, f(x_i, D)) \neq f(x_j, D)$  and  $i > j$ )  $x_j \in U2$ )
        for each  $a \in C$  {
           $flage ++$ ;
          if ( $f(x_i, a) \neq f(x_j, a)$ ) then  $m_{ij} = m_{ij} \cup a$ ;
          if ( $flage > 1$ ) break;
        }
      if ( $flage = 1$ ) then {
        if 存在 ( $m_{ij}, count$ )  $DMSC(C)$  then
          从  $DMSC(C)$  删除 ( $m_{ij}, count$ ) 后插入 ( $m_{ij}, count + 1$ );
        else 插入 ( $m_{ij}, 1$ ) 到  $DMSC(C)$  中;
      }
    }
  }
end

```

算法 2 改进的核增量式算法 (Improved Optimization for Incremental Updating Algorithm of a Core, DUAC)

输入: 1) $U1, U2$, 由算法 1 计算得到的 $DMSC(C)$; 2) 新增对象为 x 。

输出: $Core(C)$ 。

```

begin
  if  $x$  与 ( $U1 \cup U2$ ) 一致 then {
    for each  $x_i \in U1 \cup U2$  do {
       $m_{ij} = \emptyset$ ;  $flage = 0$ ;
      if ( $(f(x_i, D)) \neq f(x, D)$  and  $x_i \in U1 \cup U2$ )
        for each  $a \in C$  {
           $flage ++$ ;
          if ( $f(x_i, a) \neq f(x, a)$ ) then  $m_{ij} = m_{ij} \cup a$ ;
          if ( $flage > 1$ ) break;
        }
      if ( $flage = 1$ ) then {
        if 存在 ( $m_{ij}, count$ )  $DMSC(C)$  then
          从  $DMSC(C)$  删除 ( $m_{ij}, count$ ) 后插入 ( $m_{ij}, count + 1$ );
        else 插入 ( $m_{ij}, 1$ ) 到  $DMSC(C)$  中;
      }
    }
     $U1 = U1 \cup \{x\}$ ;
  }
  else if  $x$  与  $U1$  不一致 then {
    在  $U1$  找到与  $x$  不一致的对象  $y$ ;
    for each  $x_i \in U1 \cup U2$  do { // 此时的  $y \in U1$ 
       $m_{ij} = \emptyset$ ;  $flage = 0$ ;
      if ( $(f(x_i, D)) \neq f(y, D)$  and  $x_i \in U1 \cup U2$ )
        for each  $a \in C$  {
           $flage ++$ ;
          if ( $f(x_i, a) \neq f(y, a)$ ) then  $m_{ij} = m_{ij} \cup a$ ;
          if ( $flage > 1$ ) break;
        }
      if ( $flage = 1$ ) then {
        if 相应的  $count > 1$  then 从  $DMSC(C)$  删除 ( $m_{ij}, count$ )
          后插入 ( $m_{ij}, count - 1$ );
        else 从  $DMSC(C)$  中删除 ( $m_{ij}, count$ );
      }
    }
     $U2 = U2 \cup \{y\}$ ,  $U1 = U1 - \{y\}$ ;
    for each  $x_i \in U1$  do { // 此时的  $y \in U2$ 
       $m_{ij} = \emptyset$ ;  $flage = 0$ ;
      for each  $a \in C$  {

```

```

           $flage ++$ ;
          if ( $f(x_i, a) \neq f(y, a)$ ) then  $m_{ij} = m_{ij} \cup a$ ;
          if ( $flage > 1$ ) break;
        }
      if ( $flage = 1$ ) then {
        if 存在 ( $m_{ij}, count$ )  $DMSC(C)$  then
          从  $DMSC(C)$  删除 ( $m_{ij}, count$ ) 后插入 ( $m_{ij}, count + 1$ );
        else 插入 ( $m_{ij}, 1$ ) 到  $DMSC(C)$  中;
      }
    }
    由  $DMSC(C)$  得到核  $Core(C)$ ;
  end

```

分析算法 1 可知: 由于当且仅当某个 m_{ij} 为单个属性时, 该属性属于核 $Core(C)$ 。所以当 m_{ij} 有两个或两个以上属性时, 停止此 m_{ij} 的计算, 它必不是核。把单个属性的 m_{ij} 保存至 $DMSC(C)$ 。

分析算法 2 可知, 如下。

1) x 与 ($U1 \cup U2$) 一致时, x 将加入 $U1$, 文献 [5] 对差别矩阵增加了一行一列, 由上面的分析可知 $m_{ij} = m_{ji}$, 当 $x_i \in U1, x_j \in U1$, 增加的列, 对于核计算来说是多余的, 所以只需计算一行; 其计算量 $|U1| + |U2|$ 。判断 x 与 ($U1 \cup U2$) 是否一致的时间为 $|U1| + |U2|$ 。故总的的时间为 $2 \times (|U1| + |U2|)$ 。

2) x 与 $U1$ 不一致时, 计算 $U1$ 中与 x 不一致的对象 y , 删除 $DMSC(C)$ 中 y 所在行的单个属性; 其计算量为 $|U1| + |U2|$ 。然后 y 作为 $U2$ 中的对象计算相应的 $DMSC(C)$; 其计算量为 $|U1|$ 。判断一致性的时间为 $|U1| + |U2|$, 所以 x 与 $U1$ 不一致时总的的时间为 $3 \times |U1| + 2 \times |U2|$ 。

3) 故本文的算法 2 的时间复杂度为 $O(3 \times |U1| + 2 \times |U2|)$ 小于文献 [5] 算法 2 的时间复杂度 $O(5 \times |U1| + 3 \times |U2|)$ 。

4) 算法 2 空间复杂度为 $O(|C|)$, C 为决策属性数, $|C| \ll |U1| + |U2|$ 。文献 [5] 空间复杂度为 $O(|U1| \times (|U1| + |U2|))$ 。故本文较文献 [5] 空间复杂度有显著的改善。

5 实验

在内存为 1 024 MB, CPU 为 P4 2.9 GHz, 操作系统为 Windows XP 的联想 PC 上, Eclipse 下 Java 实现了文献 [5] 的算法 2 及本文中的算法 2。标记文献 [5] 的算法 2 为 O UAC, 本文算法 2 为 D UAC。利用 UC1 上所提供的蘑菇数据库 (mushroom) 来进行实验, 该数据库有 8 124 个对象。将蘑菇数据库看作决策表, 并进行以下两组实验。

1) 从 8 124 个对象中选择 7 000 个对象作为基准决策表 (基准决策表的含义是指该决策表生成的差别矩阵作为算法 O UAC 和 D UAC 的输入), 从剩下的 1 124 个对象中依次选择 200, 500, 800, 1 124 个对象作为增量, 实验结果如图 1。

2) 由蘑菇数据库生成 8 000 个对象, 其中不一致对象数为 1 000, 从生成的 8 000 个对象中选择 7 500 条作为基准决策表, 从剩下的 500 个对象中依次选择 100, 200, 300, 500 个对象作为增量, 实验结果如图 2 所示。

分析实验可得如下结果。

第一组实验, 8 124 个对象为一致性对象。算法 O UAC 计算及遍历差别矩阵相应的行和列, 而 D UAC 只需计算一行且我们优化了核属性计算算法, 所以 D UAC 的计算时间

比较少。

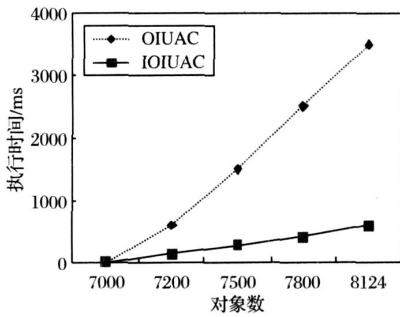


图1 第1组实验算法的执行时间

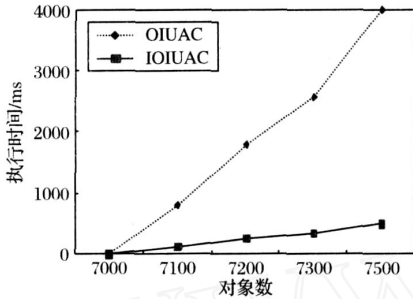


图2 第2组实验算法的执行时间

第二组实验,当 x 与 $U1$ 不一致时, OIUAC 得先遍历差别矩阵中与 x 相应的行和列,删除 $DMSC(C)$ 中相关的核属性,然后把该对象插入 $U2$,计算 $U2$ 中相应的差别矩阵,把核属

性插入 $DMSC(C)^{15,141}$ 。此时 DUAC 的计算量明显少于 OIUAC。

实验过程监测显示,因为 OIUAC 存储差别矩阵,随着对象数从 7 200 个增加到 8 124 个,内存的使用增加很快,从 225MB 至 285MB;而 DUAC 一直保持稳定且比较小的内存增加,都只增加 20MB 左右。所以 DUAC 较 OIUAC 的另一重要优势为 DUAC 可以应对大数据集的挑战。

6 结语

本文提出一种基于改进差别矩阵的核增量更新算法,主要考虑对象动态增加情况下核的更新问题。证明了该算法的正确性及它具有近线性时间和空间复杂度,本文算法可以有效应对大数据集的挑战,是对文献 [5] 算法的有效改进。

参考文献:

- [1] PAWLAK Z. Rough sets [J]. International Journal of Information and Computer Science, 1982, 11(5): 341 - 356.
- [2] HU X H, CERCONE N. Learning in relational databases: A rough set approach [J]. Computational Intelligence: An International Journal, 1995, 11(2): 323 - 338.
- [3] 叶东毅,陈昭炯. 一个新的差别矩阵及其求核方法 [J]. 电子学报, 2002, 30(7): 1086 - 1088.
- [4] 杨明,孙志挥. 改进的差别矩阵及其求核方法 [J]. 复旦学报:自然科学版, 2004, 43(5): 865 - 868.
- [5] 杨明. 一种基于改进差别矩阵的核增量式更新算法 [J]. 计算机学报, 2006, 29(3): 407 - 413.

(上接第 2472 页)

距离为 108 cm。与实际的声源目标 (-30°, 28°) 方位之间的误差很小。相比用耳廓装置进行声源方位的评估,结合听觉

和视觉传感系统具有更高的定位效果。见图 5,结果证明,提出的由粗到精的定位策略具有较高的定位精度。

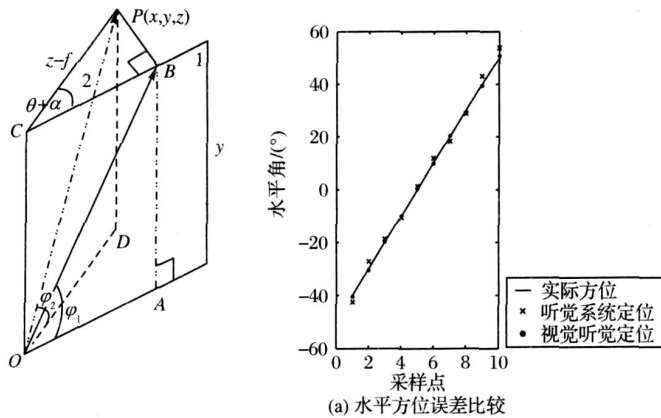


图 4 声源标高角评估模型

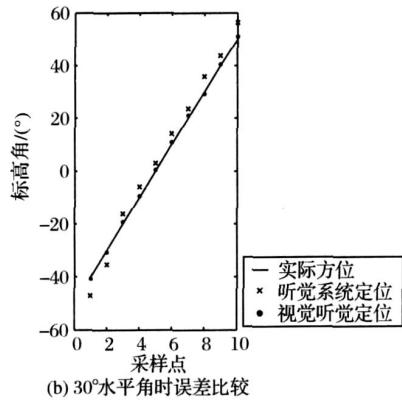


图 5 实验结果比较

4 结语

本文结合机器人系统中的听觉系统和双目立体视觉系统,利用多种定位指标,对空间声源目标进行全方位的综合定位。利用由粗到精的定位策略,定位的结果具有较高的精确性,能够满足移动机器人的作业要求。提出的定位思路,可以为其他视觉和听觉融合系统提供解决方案。

参考文献:

- [1] RAO DAN, XIE BO-SUN. Head rotation and sound image localization in the median plane [J]. Chinese Science Bulletin, 2005, 50(2): 412 - 416.
- [2] FRANCART T, BROKX J, WOUTERS J. Sensitivity to interaural level difference and loudness growth with bilateral bimodal stimulation [J]. Audiology & Neuro-Otology, 2008, 13(5): 309 - 319.

- [3] HWANG S, PARK Y, PARK Y. Sound direction estimation using artificial ear [C]// ICAS07: International Conference on Control, Automation and Systems Seoul, Korea: [s n], 2007: 1906 - 1910.
- [4] RAYKAR V C, DURA ISWAMI R, YEGNANARA YANA B. Extracting the frequencies of the pinna spectral notches in measured head related impulse responses [J]. Acoustical Society of America Journal, 2005, 118(1): 364 - 374.
- [5] OGAWA Y, YAMAOKA D, YAMADA H, et al. Binocular stereo vision processing based on pulse coupled neural networks [C]// SICE 2004 Annual Conference Tokyo: [s n], 2004: 311 - 316.
- [6] DAVISON A J. Real-time simultaneous localization and mapping with a single camera [C]// Proceedings of the 9th IEEE International Conference on Computer Vision Washington, DC: IEEE Computer Society, 2003: 1403 - 1410.