doi: 10.7690/bgzdh.2015.11.013

基于双概率相似关系的不完备决策系统数据补齐算法

阎桂林[1,2]，徐廷学[1]，袁有宏[3]

（1. 海军航空工程学院兵器科学与技术系，山东 烟台 264001；2. 中国人民解放军 92514 部队，山东 烟台 264001；
3. 海军驻兰州地区军代室，兰州 730070）

摘要：针对某型导弹故障测试数据的部分缺失问题，提出一种基于双概率相似关系的不完备决策系统数据补齐算法。在 ROUSTIDA 算法的基础上，根据决策属性值将不完备决策系统划分为各个决策子系统，考虑不同决策子系统中相似对象对决策规则的影响，选择同一决策子系统内最相似的对象来填补条件属性缺失值，并通过实例进行验证分析。结果表明：该算法在一定程度上能降低计算的复杂性、提高填补率，并能有效解决 ROUSTIDA 算法可能存在的决策规则冲突问题。

关键词：双概率相似关系；不完备决策系统；决策规则；数据补齐算法

中图分类号：TP301.6 **文献标志码**：A

Completing Data Algorithm Based on Double Probability Similarity in Incomplete Decision System

Yan Guilin[1,2], Xu Tingxue[1], Yuan Youhong[3]

(1. *Department of Ordnance Science & Technology*, *Naval Aeronautical Engineering Academy*, *Yantai* 264001, *China*;
2. *No*. 92514 *Unit of PLA*, *Yantai* 264001, *China*;
3. *Navy Representative Office in Lanzhou District*, *Lanzhou* 730070, *China*)

Abstract: According to the problem of the lack of fault test data of a certain type of missile, the paper puts forward a completing data algorithm based on double probability similarity in incomplete decision system. In accordance with the decision attribute values, the incomplete decision system was divided into various decision subsystem by the algorithm based on ROUSTIDA algorithm. Considering that similar objects in different decision subsystem may affect the decision-making, the algorithm of this paper completed the losing value of condition attributes with the most similar objects in the same decision subsystem, which was analyzed with example. The results prove that the algorithm can partly reduce the computational complexity, improve the completing rate and effectively solve the problem that decision-making principle may exist conflict by ROUSTIDA algorithm.

Keywords: double probability similarity; incomplete decision system; decision-making principle; completing data algorithm

0 引言

导弹故障测试数据是导弹故障的外在表现。通过导弹故障测试数据来研究导弹故障诊断方法是一种行之有效的途径，然而由于测试设备以及操作人员记录等原因，导弹故障测试数据往往存在缺失数据，这不利于导弹故障诊断方法的研究。

波兰科学家 Pawlak 在 1982 年提出了处理不精确知识和不完备信息的粗糙集理论。该理论能够根据数据的不可分辨关系对缺失数据进行补齐处理，对于后续基于粗糙集的导弹故障诊断方法研究具有重要作用。基于粗糙集的数据填补算法能尽可能地表现出信息系统基本特征的隐含规律，具有较好的填补效果。在基于粗糙集的数据填补算法中，最具有代表性的填补算法是 ROUSTIDA 算法。该算法以扩充的可辨识矩阵为基础，以产生尽可能高支持度的分类规则为目标，并尽可能使缺失数据的对象与信息系统其他相似对象属性值保持一致[1]。但是在属性值具有不唯一的填补对象时，该算法将无法补齐缺失属性，只能采用均值法或频率法[2]，没有考虑不同决策属性对象的条件属性均值或频率的区别，填补后也比较容易出现决策规则冲突问题。针对 ROUSTIDA 算法存在的问题，许多科研工作者对该算法进行了研究和改进：Kryszkiewicz[3]于 1998 年提出粗糙集中对象间存在相似关系，指出相似关系也是一种容差关系；文献[4−5]利用量化相似关系的理论，对样本间的相似关系进行了定量分析，并把量化引入 ROUSTIDA 算法中，从而提高了填补效率；文献[6]研究了在决策属性无缺失情况下，对存在缺失条件属性的对象，仅在与该对象的决策属性值相同对象中选择属性值对其填补，在一定程度上避免决策规则冲突情况；文献[7]提出了以样本间概

收稿日期：2015-07-16；**修回日期**：2015-08-20
作者简介：阎桂林(1986—)，男，广西人，硕士，助理工程师，从事武器装备综合保障理论与技术研究。

率最大值作为相似度最大来进行样本填补的思路。

笔者在 ROUSTIDA 算法的基础上提出了一种基于双概率相似度的改进不完备数据补齐方法。该方法针对导弹故障历史测试数据存在的部分缺失问题，在不完备数据决策属性无缺失情况下对条件属性进行填补。

1　粗糙集相关理论

定义 1　对于信息系统 $S=(U,A,V,f)$，非空有限集 $U=\{x_1,x_2,\cdots,x_n\}$ 称为论域，$A=C\bigcup D$ 是属性集合，并且 $C\bigcap D=\varnothing$，其中 $C=\{c_1,c_2,\cdots,c_m\}$ 是条件属性集，$D=\{d\}$ 是决策属性集，$V=\bigcup_{a\in A}V_a$，V_a 是属性 a 的值域，$f:U\times A\to V$ 是一个信息函数，为每个对象的每个属性赋予一个信息值。设 $c_k\in C$，$c_k(x_i)$ 为属性 c_k 的属性值，若 $c_k(x_i)$ 取值为空，且当决策属性集 $D\neq\varnothing$ 时，该信息系统又称之为不完备决策系统。

定义 2　对于不完备决策系统 $S=(U,A,V,f)$，依据决策属性值 $V_d=\{d_1,d_2,\cdots,d_l\}$ 可将系统划分为不同的子系统，$S=S_1\bigcup S_2\bigcup\cdots\bigcup S_l$，其中 S_l 为决策系统中的某个决策子系统[8]。给定 2 个对象 $x_i,x_j\in S_l$，则同一决策子系统内对象 x_i 在条件属性 c_k 上与对象 x_j 近似的概率定义为

$$R_k^+(x_i,x_j)=\begin{cases}0 & c_k(x_i)\neq c_k(x_j)\wedge c_k(x_i)\neq *\wedge c_k(x_j)\neq * \\ \sum\left(\left|E_{c_k}\right|/\left|V_{c_k}\right|\right)^2 & c_k(x_i)=*\wedge c_k(x_j)=* \\ \left|c_k(x_i)\right|/\left|V_{c_k}\right| & c_k(x_i)\neq *\wedge c_k(x_j)=* \\ \left|c_k(x_j)\right|/\left|V_{c_k}\right| & c_k(x_i)=*\wedge c_k(x_j)\neq * \\ 1 & c_k(x_i)=c_k(x_j)\wedge c_k(x_i)\neq *\wedge c_k(x_j)\neq *\end{cases}。\tag{1}$$

其中：$\left|E_{c_k}\right|$ 表示两对象在缺失条件属性 c_k 上取值相同时，该取值对应的对象个数；$\left|V_{c_k}\right|$ 表示条件属性 c_k 所具有非空取值的对象的个数；$\left|c_k(x_i)\right|$ 表示在条件属性 c_k 上取值为 $c_k(x_i)$ 的对象的个数。

则对象 x_i 与 x_j 的相似度可表示为

$$R^+(x_i,x_j)=\prod_{c_k\in C}R_k^+(x_i,x_j)。\tag{2}$$

在填补缺失的条件属性时，同一决策子系统内两对象相似度高能提高缺失属性填补的有效性，但如果仅考虑同一决策子系统两对象的相似度而不考虑分属于不同决策子系统的两对象的相似度，则可能导致填补后分属于不同决策子系统的两对象在条件属性上的取值都相同，从而产生规则冲突；因此，在填补时要考虑不同决策子系统的对象对该填补对象的影响。实际上，对于任意 2 个对象 $x_i\in S_l,x_j\notin S_l$，假设对象 x_i 的条件属性缺失，则仅需考虑该对象与条件属性无缺失的对象 x_j 的相似度，若对象 x_j 存在缺失属性，则与对象 x_i 的相似度存在不确定性，故此时不需要先考虑不同决策子系统的对象对填补对象的影响，而是先进行同一决策子系统内的填补。因此，不同决策子系统之间对象 x_i 与 x_j 相似度可定义为

$$R^-(x_i,x_j)=\begin{cases}-1 & \forall c_k(x_j)\neq *,\left(c_k(x_i)=c_k(x_j)\right)\wedge\left(c_k(x_i)\neq *\right), \\ 0 & \text{其他}\end{cases}$$
$$k=1,2,\cdots,m。\tag{3}$$

定义 3　对于信息系统 S，$C=\{c_1,c_2,\cdots,c_m\}$ 是条件属性集，设 $x_i\in U$，对象 x_i 的缺失属性集 MAS_i 和信息系统 S 的缺失对象集 MOS 定义如下：

$$\mathrm{MAS}_i=\left\{c_k\middle|c_k(x_i)=*,k=1,2,\cdots,m\right\}；\tag{4}$$

$$\mathrm{MOS}=\left\{i\middle|\mathrm{MAS}_i\neq\varnothing,i=1,2,\cdots,n\right\}。\tag{5}$$

对于任意 2 个对象 x_i 与 x_j，若 $x_i,x_j\in S_l$，则对象 x_i 的最大相似对象集 LS_{i+} 定义如下：

$$\mathrm{LS}_{i+}=\left\{x_j\middle|\max R^+(x_i,x_j),i\neq j,j=1,2,\cdots,n\right\}。\tag{6}$$

若 $x_i\in S_l,x_j\notin S_l$，则对象 x_i 的相似对象集 LS_{i-} 定义如下：

$$\mathrm{LS}_{i-}=\left\{x_j\middle|R^-(x_i,x_j)=-1,i\neq j,j=1,2,\cdots,n\right\}。\tag{7}$$

定义 4　设双概率相似矩阵 $\boldsymbol{M}=\left(M(i,j)_{n\times n}\right)$，其矩阵中的元素 $M(i,j)$ 可定义为

$$M(i.j)=\begin{cases}R^+(x_i,x_j) & x_i,x_j\in S_l,i\neq j \\ R^-(x_i,x_j) & x_i\in S_l,x_j\notin S_l,i\neq j \\ 1 & i=j\end{cases}。\tag{8}$$

其中 $i,j=1,2,\cdots,n$。双概率相似矩阵是关于主对角线对称的矩阵，该矩阵中每个元素表示相应的 2 个对象的相似度。

2　基于双概率相似关系的 ROUSTIDA 算法

通常情况下，由于不完备决策系统中缺失值的数量及分布不同，可能需要对双概率相似矩阵 $\boldsymbol{M}$ 进行多次计算才能补齐数据。因此，可设第 r 次填补后决策系统为 S^r，相应的缺失属性集为 MAS_i^r，缺失对象集为 MOS^r，同一决策子系统内最大相似集 LS_{i+}^r 和不同决策子系统相似集 LS_{i-}^r，其中 $r=0$ 表示不完备决策表的初始状态。ROUSTIDA 算法在计算

无差别矩阵时需要将矩阵的所有元素重新计算，计算较复杂，而文中定义的双概率相似矩阵只需要计算该矩阵中不为零的元素，这是因为 $\boldsymbol{M}$ 中为零的元素表示的是该两对象之间相似度为零，而仅当相似度不为零时才能为缺失属性的填补提供支持。因此，双概率相似矩阵的定义方式能降低计算的复杂性。

基于双概率相似关系的改进 ROUSTIDA 算法如下：

Input：不完备信息系统 $S=\{U,C\cup D,V,f\}$；

Output：完备信息系统 $S=\{U^r,C\cup D,V,f^r\}$。

Step1：根据决策属性值 $V_d=\{d_1,d_2,\cdots,d_l\}$ 将不完备信息系统分割成多个决策子系统 $S=S_1\cup S_2\cup\cdots\cup S_l$，计算 MAS_i^0、MOS^0、M^0、LS_{i+}^0 和 LS_{i-}^0，令 $r=0$。

Step2：令 $k=|C|,k=1,2,\cdots,m$，产生 S^{r+1}。

1) 对于所有 $x_i\notin\mathrm{MOS}^r$，有 $c_k\left(x_i^{r+1}\right)=c_k\left(x_i^r\right)$；

2) 对于所有 $x_i\in\mathrm{MOS}^r$，$c_k\in\mathrm{MAS}_i^r$，执行如下操作：

(1) 若 $\left|\mathrm{LS}_{i-}^r\right|=0$，则

(a) 若 $\left|\mathrm{LS}_{i+}^r\right|=1$，设 $x_j\in\mathrm{LS}_{i+}^r$，若 $c_k\left(x_j^r\right)=*$，则 $c_k\left(x_i^{r+1}\right)=*$；否则 $c_k\left(x_i^{r+1}\right)=c_k\left(x_j^r\right)$；

(b) 若 $\left|\mathrm{LS}_{i+}^r\right|\geqslant 2$，对于任意 2 个对象 $x_{j_1},x_{j_2}\in\mathrm{LS}_{i+}^r$，执行如下操作：

① 若 $c_k\left(x_{j_1}^r\right)\neq *\wedge c_k\left(x_{j_2}^r\right)\neq *\wedge c_k\left(x_{j_1}^r\right)\neq c_k\left(x_{j_2}^r\right)$，则 $c_k\left(x_i^{r+1}\right)=*$；

② 若 $c_k\left(x_{j_1}^r\right)\neq *\wedge c_k\left(x_{j_2}^r\right)\neq *\wedge c_k\left(x_{j_1}^r\right)=c_k\left(x_{j_2}^r\right)$，或唯一存在 $x_{j_1}\in\mathrm{LS}_{i+}^r$，满足 $c_k\left(x_{j_1}^r\right)\neq *$，则 $c_k\left(x_i^{r+1}\right)=c_k\left(x_{j_1}^r\right)$；

③ 否则，$c_k\left(x_i^{r+1}\right)=*$。

(2) 若 $\left|\mathrm{LS}_{i-}^r\right|\geqslant 1$，令 $\left|\mathrm{LS}_{i-}^r\right|=t$，对于任意 $x_{j_0}\in\mathrm{LS}_{i-}^r$ 做循环，则

(a) 若 $\left|\mathrm{LS}_{i+}^r\right|=1$，设 $x_{j_1}\in\mathrm{LS}_{i+}^r$，则执行如下操作：

① 若 $c_k\left(x_{j_1}^r\right)=*$，则 $c_k\left(x_i^{r+1}\right)=*\wedge c_k\left(x_i^{r+1}\right)\neq c_k\left(x_{j_0}^r\right)$；

② 否则，若 $c_k\left(x_{j_0}^r\right)\neq c_k\left(x_{j_1}^r\right)$，则 $c_k\left(x_i^{r+1}\right)=c_k\left(x_{j_1}^r\right)$；若 $c_k\left(x_{j_0}^r\right)=c_k\left(x_{j_1}^r\right)$，则重新计算 LS_{i+}^r 且 $i\neq j_1$，若 $\left|\mathrm{LS}_{i+}^r\right|=1$，则转到(2)～(a)，否则，转到(2)～(b)。

(b) 若 $\left|\mathrm{LS}_{i+}^r\right|\geqslant 2$，对于任意 2 个对象 $x_{j_1},x_{j2}\in\mathrm{LS}_{i+}^r$，执行如下操作：

① 若 $c_k\left(x_{j_1}^r\right)\neq *\wedge c_k\left(x_{j_2}^r\right)\neq *\wedge c_k\left(x_{j_1}^r\right)\neq c_k\left(x_{j_2}^r\right)$，则 $c_k\left(x_i^{r+1}\right)=*$；

② 若 $c_k\left(x_{j_1}^r\right)\neq *\wedge c_k\left(x_{j_2}^r\right)\neq *\wedge c_k\left(x_{j_1}^r\right)=c_k\left(x_{j_2}^r\right)$，且 $c_k\left(x_{j_0}^r\right)\neq c_k\left(x_{j_1}^r\right)$，则 $c_k\left(x_i^{r+1}\right)=c_k\left(x_{j_1}^r\right)=c_k\left(x_{j_2}^r\right)$；若 $c_k\left(x_{j_0}^r\right)=c_k\left(x_{j_1}^r\right)$，则重新计算 LS_{i+}^r 且 $i\neq j_1$，转到(2)；

③ 当仅存在 $x_{j_1}\in\mathrm{LS}_{i+}^r$，$c_k\left(x_{j_1}^r\right)\neq *$，若 $c_k\left(x_{j_0}^r\right)\neq c_k\left(x_{j_1}^r\right)$，则 $c_k\left(x_i^{r+1}\right)=c_k\left(x_{j_1}^r\right)$；若 $c_k\left(x_{j_0}^r\right)=c_k\left(x_{j_1}^r\right)$，则重新计算 LS_{i+}^r 且 $i\neq j_1$，转到(2)；

④ 否则，$c_k\left(x_i^{r+1}\right)=*$。

(c) 令 $t=t-1$，若 $t=0$ 或 $c_k\left(x_i^{r+1}\right)\neq *$，则结束循环。

(3) 若 $S^{r+1}=S^r$，则结束循环，转到 Step3；否则，令 $r=r+1$，重新计算 M^r、MAS_i^r、MOS^r，LS_{i+}^r 和 LS_{i-}^r，转到 Step2−2)。

Step3：如果决策表中对象 x_i 仍有缺失属性值 $c_k(x_i)$，对象 x_i 决策属性值为 d_l，其决策子系统为 S_l，则执行如下操作：

1) 若 $\left|\mathrm{LS}_{i-}^r\right|=0$，则对象 x_i 应在决策子系统 S_l 中取出现次数最多的非空属性值进行补齐；

2) 若 $\left|\mathrm{LS}_{i-}^r\right|\geqslant 1$，设 $x_j\in\mathrm{LS}_{i-}^r$，则对象 x_i 应在决策子系统 S_l 中取满足 $c_k(x_i)\neq c_k(x_j)$ 且出现次数最多的非空属性值进行补齐。

Step4：输出结果，算法结束。

3　实例分析

3.1　实例验证

以某型导弹常见故障 d_1,d_2,d_3 为例，当这些故障出现时，涉及到的导弹相关测试项为 c_1,c_2,c_3,c_4,c_5,c_6。表 1 给出了多次测试提取出的 15 次故障测试数据的不完备决策，根据厂家说明书给出的测试项正常取值范围，得到精确的离散化区间，测试数据正常用“0”表示，超差则用“1”来表示。

该不完备决策表经 ROUSTIDA 算法第 1 次填补后，对象 x_2 的条件属性 c_4、c_5 和 c_6，对象 x_5、x_{13} 的条件属性 c_5，以及对象 x_{11} 的条件属性 c_6 无法填补。由于 $S^1\neq S^0$，重新计算无差别矩阵，并计算 S^2，得 S^2=S^1，第一次填补后未能填补的条件属性仍不能填补，故需采用概率法填补，填补结果如表 2 所示。

表 1　某型导弹故障数据不完备决策

S	c_1	c_2	c_3	c_4	c_5	c_6	d
x_1	1	0	1	0	1	1	1
x_2	1	0	1	*	*	*	1
x_3	1	0	0	0	1	1	1
x_4	0	0	0	1	0	1	1
x_5	0	0	0	1	*	*	1
x_6	0	0	0	1	1	1	2
x_7	1	0	0	0	1	*	2
x_8	1	0	0	1	1	0	2
x_9	1	0	0	1	*	*	2
x_{10}	1	0	1	1	1	0	2
x_{11}	0	1	1	*	*	*	3
x_{12}	0	1	1	0	0	1	3
x_{13}	1	0	1	1	*	*	3
x_{14}	0	1	1	0	0	0	3
x_{15}	1	0	1	1	0	0	3

表 2　ROUSTIDA 算法补齐后决策

S	c_1	c_2	c_3	c_4	c_5	c_6	d
x_1	1	0	1	0	1	1	1
x_2	1	0	1	0	(*,1)	1	1
x_3	1	0	0	0	1	1	1
x_4	0	0	0	1	0	1	1
x_5	0	0	0	1	0	1	1
x_6	0	0	0	1	1	1	2
x_7	1	0	0	0	1	(*,1)	2
x_8	1	0	0	1	1	0	2
x_9	1	0	0	1	1	0	2
x_{10}	1	0	1	1	1	0	2
x_{11}	0	1	1	0	0	0	3
x_{12}	0	1	1	0	0	1	3
x_{13}	1	0	1	1	0	0	3
x_{14}	0	1	1	0	0	0	3
x_{15}	1	0	1	1	0	0	3

文中算法在第 1 次填补后仅有对象 x_2 的条件属性 c_5 和对象 x_7 的条件属性 c_6 无法填补，重新计算双概率相似矩阵 $\boldsymbol{M}^0$ 中不为零的元素对应的相似度，并计算 S^2，得 $S^2=S^1$，对未填补的条件属性使用本文算法中 Step3 的方法填补，填补结果如表 3。

表 3　文中算法补齐后决策

S	c_1	c_2	c_3	c_4	c_5	c_6	d
x_1	1	0	1	0	1	1	1
x_2	1	0	1	(*,1)	(*,1)	(*,1)	1
x_3	1	0	0	0	1	1	1
x_4	0	0	0	1	0	1	1
x_5	0	0	0	1	(*,1)	1	1
x_6	0	0	0	1	1	1	2
x_7	1	0	0	0	1	1	2
x_8	1	0	0	1	1	0	2
x_9	1	0	0	1	1	0	2
x_{10}	1	0	1	1	1	0	2
x_{11}	0	1	1	0	0	(*,1)	3
x_{12}	0	1	1	0	0	1	3
x_{13}	1	0	1	1	(*,1)	0	3
x_{14}	0	1	1	0	0	0	3
x_{15}	1	0	1	1	0	0	3

3.2　结果分析

ROUSTIDA 算法出现无法填补的条件属性较多，填补效率低，且重新计算无差别矩阵时计算较复杂，同时还出现了 x_3 与 x_7，x_5 与 x_6 决策规则冲突的情况。决策规则冲突的原因是因为在补齐 x_7 时，选择的补齐对象是从整个不完备决策表中去寻找无差别对象，而无差别对象又与 x_7 的决策属性值不同，从而造成填补后的决策规则冲突；而 x_5 的填补在完成决策表的循环计算后条件属性 c_5 仍无法补齐，采用概率法进行补齐从而导致决策规则冲突。

文中所用的基于双概率相似关系的改进补齐算法由于分别考虑了同一决策子系统和不同决策子系统中最相似的对象对填补的不同影响，只有对象 x_2 的条件属性 c_5 和对象 x_7 的条件属性 c_6 无法填补，同时在对这些无法填补的条件属性用概率法进行补齐时，也考虑了同一决策子系统和不同决策子系统对填补的不同影响，更好地避免了冲突。同时在计算双概率相似矩阵时，由于每次循环只需要计算该矩阵中不为零的元素，相比 ROUSTIDA 算法的无差别矩阵的计算更为简单。

4　结束语

笔者针对导弹故障测试数据不完备特点，提出了一种基于双概率相似度的改进不完备数据补齐方法。该方法在不完备数据决策属性无缺失的情况下对条件属性进行填补，在考虑不同决策子系统中相似对象信息对填补后的决策规则产生影响的前提下，在同一决策子系统内通过计算双概率相似矩阵找到最相似的对象来填补缺失属性值。同时，对存在无法补齐的条件属性的情况，采用概率法进行补齐的同时也考虑同一决策子系统和不同决策子系统对填补结果的影响。实例结果说明：改进的算法更加符合实际，不仅有效地降低了计算的复杂性，而且能更好地避免决策规则冲突的情况发生。

参考文献：

[1] 丁春荣，李龙澍．基于相似关系向量的不完备数据补齐算法[J]．计算机应用研究, 2013, 30(2): 383-385.

[2] 王清晖，刘文奇．基于粗糙集理论缺省数据的改进算法[J]．昆明理工大学学报, 2004, 29(2): 148-150.

[3] Kryszkiewicz M. Rough set approach to incomplete information system[J]. Information Sciences, 1998, 113(3/4): 274-292.

[4] 李阳锋．一种不完备决策表的数据补齐方法[J]．科技信息, 2008, 15(20): 176-177.

[5] 吕召勇，张立权，陈志新．一种基于双重决策矩阵的不完备信息处理方法[J]．北京建筑工程学院学报, 2010, 26(4): 53-57.

[6] 刘文军．基于粗糙集理论的不完备决策表的完备化方法[J]．长沙电力学院学报(自然科学版), 2006, 21(4): 60-62.

[7] 秦华妮．基于量化相似关系的不完备决策信息系统的完备化算法[J]．湖南工程学院学报, 2007, 17(1): 65-67.

[8] 李萍，吴祈宗．基于概率相似度的不完备信息系统数据补齐算法[J]．计算机应用研究, 2009, 26(3): 881-883.