基于相对决策熵与加权相似性的粗糙集数据补齐方法

王莎莎[1] 江 峰[1] 王文鹏[2]
(青岛科技大学信息科学与技术学院 青岛 266061)[1] (青岛科技大学经济与管理学院 青岛 266061)[2]

摘 要 现有的基于粗糙集的数据补齐方法在计算任意两个对象之间的相似性时并没有考虑不同条件属性之间的差异性。针对这一问题,引入一种新的加权相似性的概念,并提出一种基于相对决策熵与加权相似性的粗糙集数据补齐算法RDNAWS。RDNAWS算法采用相对决策熵的概念来度量每个条件属性的重要性,并通过计算每个条件属性的重要性以及决策属性集对其的依赖性来为每个条件属性提供一个权值,从而将不同的条件属性有效地区分开来。在真实数据集上的实验表明,与现有的算法相比,所提算法能够获得更好的分类性能。

关键词 不完备数据,粗糙集,数据补齐,相对决策熵,加权相似性
中图法分类号 TP39 **文献标识码** A

Rough Set Approach to Data Completion Based on Relative Decision Entropy and Weighted Similarity

WANG Sha-sha[1] JIANG Feng[1] WANG Wen-peng[2]
(College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China)[1]
(College of Economics and Management, Qingdao University of Science and Technology, Qingdao 266061, China)[2]

Abstract The current data completion methods based on rough sets do not consider the differences between different condition attributes when calculating the similarities between any two objects. To solve this problem, this paper introduced a new notion of weighted similarity, and proposed a rough set data completion algorithm called RDNAWS based on relative decision entropy and weighted similarity. RDNAWS algorithm adopts the concept of relative decision entropy to measure the significance of each condition attribute. Through calculating the significance of each condition attribute and the dependence of the set of decision attributes on it, RDNAWS provides a weight for each condition attribute, which can efficiently distinguish various condition attributes. The experimental results on real data sets demonstrate that our algorithm can obtain better classification performance than the current algorithms.

Keywords Incomplete data, Rough set, Data completion, Relative decision entropy, Weighted similarity

1 前言

在现实生活中,由于数据获取的限制、数据测量时的误差等原因,使得大部分数据集都是不完备的,即存在缺失值。例如,在市场调查中被调查人拒绝透露相关问题的答案,或者答案是无效的,数据录入人员的疏忽遗漏了某些数据等。由于数据集的不完备性会影响后续的数据分析,并导致分析结果的不精确性,因此,我们需要采取相应的办法来处理不完备数据。而数据补齐,即对缺失值进行补齐,是一种可行并且经常被采用的方法。

目前,有很多种数据补齐方法被提出。这些方法大致可分为两类[1]:一类是基于统计的方法,包括平均值补齐、组合化补齐、条件均值补齐、条件组合化补齐等;另一类是基于粗糙集的方法,例如ROUSTIDA算法[1]以及针对该算法的改进算法[6-8,14,15]等。基于统计的方法可以对连续型数据和离散型数据进行补齐操作,保证了数据的分布规律,但它没有考虑对象之间以及数据属性之间的相互关系;而基于粗糙集的方法则充分考虑了对象之间和属性之间的相互关系,这样就使得补齐之后的数据更加接近现实数据,从而更具有现实意义[4]。

自1982年被提出以来,粗糙集理论已经在数据挖掘等领域发挥着重要的作用[1,5]。近年来,越来越多基于粗糙集的数据补齐方法被提出[3,4,6-8,13-15],其中ROUSTIDA算法[1]及其相应的改进算法[6-8,14,15]被广泛使用。这类算法的主要思想是:缺失值的补齐主要通过计算决策表中含有缺失值的对象与其他不含缺失值的对象之间的相似性来实现,把最相似对象的相应取值作为最终的填充值。但是,这类算法普遍存在一个问题,即在计算决策表中任意两个对象之间的相似性时,假设所有条件属性具有相同的重要性,而且决策属性集对每个条件属性的依赖度也是一样的,并没有考虑不同属性之间所存在的差异对最终补齐结果的影响。在实际应用中,不

到稿日期:2013-04-24 返修日期:2013-09-18 本文受国家自然科学基金项目(60802042,61273180),山东省自然科学基金项目(ZR2011FQ005,ZR2010FQ027),山东省优秀中青年科学家科研奖励基金项目(BS2012ZZ003),山东省高等学校科技计划项目(J11LG05)资助。
王莎莎 硕士生,主要研究方向为数据挖掘、粗糙集;江 峰 博士,副教授,主要研究方向为人工智能、粗糙集等,E-mail:jiangkong@163.net(通信作者)。

同条件属性对最终决策的贡献往往是不同的，即每个条件属性的重要度是不同的，并且决策属性集对不同条件属性的依赖度也是不一样的。因此，在计算不同对象之间的相似性时，充分考虑上述两个因素的影响对于最终的补齐结果是有帮助的。

针对上述问题，我们曾经在文献[13]中提出过一种基于加权相似性的数据补齐算法 WSDCA，该算法采用一种新的机制来计算对象之间的相似性，即加权相似性。加权相似性充分考虑决策表中不同条件属性的差异性，按照决策属性集对每个条件属性的依赖度以及该属性自身的重要性将不同的属性严格区分开来。本文将在文献[13]的基础上，进一步提出一种基于相对决策熵和加权相似性的数据补齐算法 RDNAWS。与 WSDCA 算法不同，本文将采用相对决策熵的概念[2]来计算每个条件属性的重要性，而不是基于正区域来计算。另外，在 WSDCA 算法中，对任意条件属性 a，我们将 a 的重要性与决策属性集对 a 的依赖度这两者之和作为 a 的权重，并没有考虑这两种度量在计算 a 的权重时是否应该发挥不同的作用。而在本文中，在计算 a 的权重时，我们将采用一种新的策略来动态地调整属性重要性与属性依赖度这两者所发挥的作用(即这两种度量在计算 a 的权重时所发挥的作用可以根据实际情况进行动态调整)，而不仅仅只是将这两者之和作为 a 的权重。

与粗糙集中现有的信息熵模型不同，相对决策熵采用 Pawlak 所提出的粗糙度[5]进行定义。粗糙度是一种刻画集合不确定性的有效方式，广泛应用于诸多领域。本文基于相对决策熵来度量条件属性的重要性，把每个条件属性的重要性以及决策属性集对其的依赖度作为该属性的权值来计算对象之间的相似性，从而得到一种新的加权相似性的概念。另外，本文所提出的加权相似性不仅考虑到决策表中不同条件属性之间的差异性，而且在计算属性的权重时也把属性重要性与属性依赖度这两种度量方式区别开来，使得它们在计算属性权重时可以发挥不同的作用，从而更加贴近实际情况。

为了验证 RDNAWS 算法的有效性，我们在真实数据集上开展了实验。实验结果表明，RDNAWS 的性能要优于现有的基于统计的方法。另外，相对于文献[13]中所提出的算法 WSDCA，RDNAWS 也具有更好的性能，因此，本文所提方法是对文献[13]中所提方法的一种有效改进。

2 粗糙集理论相关概念

定义 1(信息表) 信息表是一个四元组 $IS=(U,A,V,f)$，其中[5]：

(1)U 和 A 分别表示一个非空、有限的对象集和属性集；

(2)V 是所有属性论域的并集，即 $V=\bigcup_{a\in A}V_a$，其中 V_a 为属性 a 的值域；

(3)$f:U\times A\to V$ 是一个函数，使得对任意 $a\in A$ 和 $x\in U$，$f(x,a)\in V_a$。

如果将 IS 中的 A 进一步划分为两个不相交的子集：条件属性集 C 和决策属性集 D，那么这种特殊的信息表又称为决策表，简记 $DT=(U,C,D,V,f)$。

定义 2(不可分辨关系) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $B\subseteq C\cup D$，定义由 B 所决定的一个不可分辨关系为：$IND(B)=\{(x,y)\in U\times U:\forall a\in B(f(x,a)=f(y,a))\}$[5]。

可以证明，$IND(B)$ 是 U 上的一个等价关系，并且 $IND(B)=\bigcap_{a\in B}IND(\{a\})$。

定义 3(相似关系) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $B\subseteq C$，定义由 B 所决定的一个相似关系 $N(B)$ 为：$N(B)=\{(x,y)\in U\times U:\exists a\in B(f(x,a)=f(y,a))\}$[1]。

定义 4(上、下近似) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $B\subseteq C\cup D$ 和 $X\subseteq U$，X 的 B-上近似与 B-下近似分别定义为[5]：

$$\overline{X}_B=\bigcup\{[x]_B\in U/IND(B):[x]_B\cap X\neq\varnothing\}$$

$$\underline{X}_B=\bigcup\{[x]_B\in U/IND(B):[x]_B\subseteq X\}$$

定义 5(粗糙度) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $B\subseteq C\cup D$ 和 $X\subseteq U(X\neq\varnothing)$，集合 X 的 B-粗糙度定义为[5]：

$$\rho_B(X)=|\overline{X}_B-\underline{X}_B|/|\overline{X}_B|$$

定义 6(相对正区域) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $B\subseteq C$，定义决策属性集 D 相对于条件属性集 B 的正区域为[1,5]：

$$Pos_B(D)=\bigcup\{Y\mid Y\subseteq X,X\in U/IND(D),Y\in U/IND(B)\}$$

定义 7(属性依赖度) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $a\in C$，定义决策属性集 D 相对于 a 的依赖度为[5]：

$$\gamma_a(D)=|Pos_{\{a\}}(D)|/|U|$$

3 基于相对决策熵与加权相似性的数据补齐算法

目前，基于粗糙集的数据补齐方法在计算对象的相似性时，通常假设每个条件属性的重要性都是一样的，并且决策属性集对每个条件属性的依赖度也是一样的。通过分析，我们不难发现，在实际应用中不同条件属性之间有着显著的差异，而且不同条件属性对最终的决策所起的作用也是不一样的。因此，在计算对象之间的相似性时，应该将不同的条件属性区别开来。

针对上述问题，我们在文献[13]中提出了一种基于加权相似性的数据补齐方法，该方法在计算对象的相似性时，为每个条件属性设置一个权重并根据权重来区分不同的属性。在文献[13]的基础上，本文进一步提出一种基于相对决策熵与加权相似性的数据补齐方法。我们引入一种相对决策熵的概念来计算任意条件属性 a 的重要性，并且采用粗糙集中相对正区域的概念来计算决策属性集对 a 的依赖度。在计算对象之间的相似性时，属性 a 的重要性以及决策属性集对 a 的依赖度这两种度量被分别用来计算 a 的权重，而且这两种度量在计算 a 的权重时将发挥不同的作用，这样我们就得到了一种新的加权相似性的概念。

本文所提出的加权相似性概念是对文献[13]中所提出的加权相似性的有效改进。首先，我们采用相对决策熵而不是

基于正区域来计算每个条件属性的重要性；其次，在计算条件属性 a 的权重时，我们采用一种新的策略来动态地调整属性重要性与属性依赖度这两者所发挥的作用，即这两种度量在计算 a 的权重时所发挥的作用可以根据实际情况进行动态地调整，而不仅仅只是简单地将这两者之和作为 a 的权重。

下面，我们首先给出相对决策熵的定义并引入基于相对决策熵的属性重要性，然后给出一种新的加权相似性的定义。

定义 8(相对决策熵)　给定决策表 $DT=(U,C,D,V,f)$，令 $U/IND(D)=\{Y_1,\cdots,Y_m\}$ 为 $IND(D)$ 对 U 的划分，对任意 $B\subseteq C$，D 在 $IND(B)$ 下的相对决策熵定义为[2]：

$$RDE(D,B)=\sum_{i=1}^{m}\rho_B(Y_i)\log_2(\rho_B(Y_i)+1)$$

其中，$\rho_B(Y_i)$ 为集合 Y_i 的 B-粗糙度，$1\leqslant i\leqslant m$。

定义 9(基于相对决策熵的属性重要性)　给定决策表 $DT=(U,C,D,V,f)$，对任意 $a\in C$，我们将属性 a 在 C 中相对于 D 的重要性定义为[2]：

$$SGF(a,C,D)=RDE(D,C-\{a\})-RDE(D,C)$$

定义 10(加权相似性)　给定决策表 $DT=(U,C,D,V,f)$，对任意 $a\in C$，令 $\gamma_a(D)$ 和 $SGF(a,C,D)$ 分别表示 D 对 a 的依赖度以及 a 的重要性。对任意 $u_1,u_2\in U$，u_1 与 u_2 的加权相似性定义为：

$$WS(u_1,u_2)=\sum_{a\in C}(1+p\times\gamma_a(D)+(1-p)\times SGF(a,C,D))\times h_a(u_1,u_2)$$

其中，参数 p 的取值为 0 和 1 之间的实数，$h_a:U\times U\to\{0,1\}$ 是一个从 $U\times U$ 到 $\{0,1\}$ 的函数，使得对任意 $(u_1,u_2)\in U\times U$，如果 $f(u_1,a)=f(u_2,a)$，则 $h_a(u_1,u_2)=1$；否则 $h_a(u_1,u_2)=0$。

上述公式重新定义了加权相似性，通过相对决策熵来计算条件属性的重要性，并且利用每个属性的重要性和决策属性集对其的依赖度来区分不同的条件属性。具体来说，对任意条件属性 a，将 a 的重要性与 D 对 a 的依赖度这两者按照一定的比例(具体的比例值由参数 p 动态地确定)累加起来，从而得到 a 的权重。这样，在计算对象的相似性时，不同的条件属性就可以根据其权重的大小被严格区分开来。

算法 1　RDNAWS

输入：不完备决策表 DT=(U,C,D,V,f)，其中|U|=n，C={a_1,a_2,…,a_m}，D={d}

输出：完备决策表 DT′=(U,C,D,V′,f′)

(0)将 DT 中的每个空缺值采用特殊值 * 进行替换。

(1)对条件属性集 C 和决策属性集 D，执行如下操作：

(1.1)根据 U 中对象在 C 上的取值，按照值域 V_C 上的一个给定次序对 U 中对象进行基数排序[9]；

(1.2)求出划分 U/IND(C)；

(1.3)根据 U 中对象在 D 上的取值，按照值域 V_D 上的一个给定次序对 U 中对象进行基数排序；

(1.4)求出划分 U/IND(D)={Y_1,…,Y_s}；

(1.5)对任意 $Y_i\in$U/IND(D)，计算 Y_i 的 C-上近似与 C-下近似，并得到 Y_i 的 C-粗糙度 $\rho_C(Y_i)$，$1\leqslant i\leqslant s$；

(1.6)根据定义 8，计算 D 在 IND(C)下的相对决策熵 RDE(D,C)。

(2)对任意 a∈C，循环执行如下操作：

(2.1)根据 U 中对象在 C−{a}上的取值，按照值域 $V_{C-\{a\}}$ 上的一个给定次序对 U 中对象进行基数排序；

(2.2)求出划分 U/IND(C−{a})；

(2.3)对任意 $Y_i\in$U/IND(D)，计算 Y_i 的(C−{a})-上近似与(C−{a})-下近似，并得到 Y_i 的(C−{a})-粗糙度 $\rho_{C-\{a\}}(Y_i)$，$1\leqslant i\leqslant s$；

(2.4)计算 D 在 IND(C−{a})下的相对决策熵 RDE(D,C−{a})；

(2.5)计算 a 在 C 中相对于 D 的重要性 SGF(a,C,D)；

(2.6)根据 U 中对象在 a 上的取值，按照值域 V_a 上的一个给定次序，对 U 中对象进行基数排序；

(2.7)求出划分 U/IND({a})；

(2.8)计算正区域 $Pos_{\{a\}}(D)$，并得到 D 相对于 a 的依赖度 $\gamma_a(D)$。

(3)对任意 u∈U，循环执行如下操作：

对任意 a∈C，如果 f(u,a)=*，则

(3.1)在 U 中找出所有在 a 上的取值不等于 * 并且在 d 上的取值等于 f(u,d)的对象。把所有满足上述条件的对象都放到集合 S 中；

(3.2)根据定义 10，计算 u 与 S 中每个对象的加权相似性；

(3.3)在 S 中找出与 u 的相似性最大的对象 Max；

(3.4)令 f(u,a)=f(Max,a)。

(4)算法结束，返回补齐后的决策表 DT′。

在算法 1 中，对任意 $B\subseteq C$，我们采用基数排序的方法对 U 进行排序，然后再求划分 $U/IND(B)$[9]，使得计算 $U/IND(B)$ 的时间复杂度为 $O(|B|\times n)$。基数排序可以有效降低计算划分的复杂度，从而可以降低整个算法的复杂度。

下面分析算法 1 在最坏的情况下的时间复杂度。首先，步骤(0)的复杂度为 $O(m\times n)$，步骤(1)的复杂度为 $O(m\times n)$，步骤(2)的复杂度为 $O(m^2\times n)$，步骤(3)的复杂度为 $O(m\times n\times k)$，其中 m 和 n 分别表示 C 与 U 的势，k 表示 DT 中空缺值的个数。因此，当 $k<m$ 时，算法 1 的时间复杂度为 $O(m^2\times n)$；当 $k>m$ 时，算法 1 的时间复杂度为 $O(m\times n\times k)$。另外，算法 1 的空间复杂度为 $O(m+n)$。

据统计，在实际的数据集中，空缺值所占比例一般在 4%～5%左右[6]，因此，算法 1 的时间复杂度是可以接受的。

4　实验

我们基于 Java 语言实现了 RDNAWS 算法，并在 2 个不完备的 UCI 数据集上测试了其性能：Voting 和 Roth[10]。在 Voting 上，我们分别比较 RDNAWS、WSDCA、Mean-Completer 和 Conditioned-Mean-Completer 这 4 种算法的性能，其中 WSDCA 来自于文献[13]，Mean-Completer 和 Conditioned-Mean-Completer 都来自于 ROSETTA 软件[11]。在 Roth 上，我们分别比较 RDNAWS、WSDCA、Conditioned-Combinatorial-Completer 和 Conditioned-Mean-Completer 这 4 种算法的性能，其中 Conditioned-Combinatorial-Completer 也来自 ROSETTA[11]。在实验中，为了计算对象之间的加权相似性，我们对参数 p 设置了多个不同的值进行测试，最终得出在 p 的取值为 0.3 的条件下，本文所提算法能够得到较好的实验结果。

我们首先采用不同的补齐算法对 Voting 和 Roth 中的缺失值进行补齐操作，然后针对补齐之后的数据集采用 WEKA 中提供的分类算法(包括 Id3、J48 和 J48graft)进行分类训练

和测试[12]，从而分析出不同的补齐算法对数据集分类效果的影响。实验的具体步骤如下：

(1)利用 RDNAWS 对 Voting 和 Roth 中的缺失值进行补齐，从而得到补齐后的 Voting 和 Roth 数据集；

(2)分别利用 WSDCA、Mean-Completer 和 Conditioned-Mean-Completer 算法来处理 Voting，并得到 3 组补齐后的 Voting；

(3)分别利用 WSDCA、Conditioned-Mean-Completer 和 Conditioned-Combinatorial-Completer 算法来处理 Roth，并得到 3 组补齐后的 Roth；

(4)针对前面 3 步中所产生的由不同算法补齐之后的 Voting 和 Roth 数据集，分别利用 WEKA 中的分类器 Id3、J48 和 J48graft 进行分类训练和测试，从而比较不同算法的性能(其中测试模式为：Percentage split(66% for training)，即从相应的数据集中随机抽取出 66%的数据作为训练集，剩下的作为测试集)。

4.1 Voting 上的实验结果

Voting 数据集由 435 个对象组成，包括 16 个条件属性和 1 个决策属性，该数据集中含有 392 个缺失值[10]。我们采用不同的算法来处理 Voting 中的缺失值，实验结果如表 1 所列。

表 1 Voting 上的结果

采用不同算法补齐之后的数据集	精度(%)		
	Id3	J48	J48graft
RDNAWS-Completed-Voting	97.3125	98.0952	98.0952
WSDCA-Completed-Voting	97.2973	97.973	97.973
MC-Completed-Voting	94.596	96.6216	95.9459
CMC-Completed-Voting	95.2703	95.2703	95.2703

在表 1 中，第 1 列第 2—5 行分别表示由 RDNAWS、WSDCA、Mean-Completer 和 Conditioned-Mean-Completer 这 4 种算法补齐之后所得到的数据集。从表 1 可以看出，采用 RDNAWS 补齐之后的 Voting 数据集在 Id3、J48 和 J48graft 这 3 个分类器下的分类精度都要高于其他算法。因此，对于 Voting 而言，采用 RDNAWS 进行补齐，可以获得更好的分类性能。

4.2 Roth 上的实验结果

Roth 中共有 160 个对象，包括 4 个条件属性和 1 个决策属性。由于 Roth 本身是完备的，我们人为地从该数据集中随机删除部分条件属性值(约占整个数据集的 10%)，这样就得到了一个不完备的 Roth。同样，我们采用不同的算法来处理这个不完备的 Roth，实验结果如表 2 所列。

表 2 Roth 上的结果

采用不同算法补齐之后的数据集	精度(%)		
	Id3	J48	J48graft
RDNAWS-Completed-Roth	77.0492	77.0492	77.0492
WSDCA-Completed-Roth	77.778	75.9259	75.9259
CMC-Completed-Roth	61.111	64.8148	64.8148
CCC-Completed-Roth	60	68	68.8

在表 2 中，第 1 列第 2—5 行分别表示由 RDNAWS、WSDCA、Conditioned-Mean-Completer 和 Conditioned-Combinatorial-Completer 这 4 种算法补齐之后的数据集。从表 2 可以看出，采用 RDNAWS 补齐之后的 Roth 数据集，除了在 Id3 上的精度略低于 WSDCA 之外，在 J48 和 J48graft 上的精度都明显高于其他算法。因此，从整体上看，对于 Roth 而言，采用 RDNAWS 进行补齐，同样可以获得更好的分类性能。

结束语 本文通过引入相对决策熵的概念，对文献[13]中提出的加权相似性进行了改进，并提出一种基于相对决策熵与加权相似性的不完备数据补齐算法。该算法不仅采用相对决策熵来计算属性重要性，并且引入一种新的策略来动态地调整属性重要性与属性依赖度这两种度量在计算属性权重时所发挥的作用，从而可以最大程度地区分不同的条件属性。最后，我们通过在 UCI 数据集上的实验验证了该算法的有效性。

参 考 文 献

[1] 王国胤. Rough 集理论与知识获取[M]. 西安：西安交通大学出版社，2001

[2] 江峰，王春平，曾惠芬. 基于相对决策熵的决策树算法及其在入侵检测中的应用[J]. 计算机科学，2012，39(4)：223-226

[3] 焦娜，苗夺谦，张红云. 多决策表缺失属性补齐算法的研究[J]. 计算机科学，2009，36(1)：142-145

[4] 潘巍，王阳生，杨宏戟. 粗糙集理论中新的针对不完备信息系统的处理方法研究[J]. 计算机科学，2007，134(16)：158-161

[5] Pawlak Z. Rough Sets [J]. International Journal of Computer and Information Sciences，1982，11：341-356

[6] 孟军，刘永超，莫海波. 基于粗糙集理论的不完备数据填补方法[J]. 计算机工程与应用，2008，44(6)：175-177

[7] 王国胤. Rough 集理论在不完备信息系统中的扩充[J]. 计算机研究与发展，2002，39(10)：1238-1243

[8] Kryszkiewicz M. Rough set approach to incomplete information system [J]. Information Sciences，1998，112(14)：39-49

[9] 徐章艳，刘作鹏，杨炳儒，等. 一个复杂度为 max(O(|C||U|)，O(|C|²|U/C|)) 的快速属性约简算法[J]. 计算机学报，2006，29(3)：391-399

[10] Bay S D. The UCI KDD repository[OL]. http://kdd.ics.uci.edu，1999

[11] Øhrn A. Rosetta Technical Reference Manual[R]. http://www.idi.ntnu.no/aleks/rosetta，1999

[12] Hall M，Frank E，Pfahringer H G，et al. The WEKA data mining software：an update[J]. SIGKDD Explor. News.，2009，11(1)：10-18

[13] 赵洪波，江峰，曾惠芬，等. 一种基于加权相似性的粗糙集数据补齐方法[J]. 计算机科学，2011，38(11)：167-170

[14] 田树新，吴晓平，王红霞. 一种基于改进的 ROUSTIDA 算法的数据补齐方法[J]. 海军工程大学学报，2011，23(5)：11-15

[15] 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