机器学习及其应用研讨会
南京大学，2006

次属性原理及在特征选择上的应用

王 珏 韩素青 梁洪力

中国科学院自动化研究所

Machine Learning and Data Mining 2006

模型多样性

Leo Breiman认为机器学习的三个基本问题是：
(1)**Rashomon**问题：模型多样性。
(2)**Occam**问题：简单与精度的冲突。
(3)**Bellman**问题：维数灾难。
Statistical modeling: The two cultures, Statistical Science, 2001, Vol. 16, No.3, 199-231.

Rashomon问题产生的原因是：数据集合的获得是不可控的。

Leo Breiman对**Rashomon**问题的解释是：数据集合中存在多个好模型，哪个模型是对数据的最好的解释。

Rashomon问题广泛地存在于我们面临的的数据集合之中。

Rashomon

Rashomon(罗生门)是黑泽明根据芥川龙之介(芥川奖)名著导演的第一个轰动西方的日本电影，讲述了一件凶杀案。故事有四个人物：武士(多襄丸)，通奸女人(真纱)，被杀者(武弘，真纱的丈夫)，以及目击者樵夫。在法庭上，所有人对杀死一个人的事实，完全确认，但是，对犯罪事实的陈述则完全不同。

武士说：他在被杀者同意的条件下，公平决斗时杀了对方。
真纱说：由于丈夫对她的轻蔑，使得她误杀了她的丈夫。
被杀者通过巫婆之口说：因强盗要他处置他的妻子，他决定自杀。
证人樵夫说：武士要女子跟他，她要丈夫与武士决斗，她丈夫拒绝。女子激将之，两个男人决斗。但是，他私藏了宝刀，证词可信吗？

问题情况难以预料，或“公说公有理，婆说婆有理”，称为**Rashomon**问题，例如，这次美国中期选举难以预料，可以称为“美国中期选举**Rashomon**”。

Rashomon问题的解释

“杀人案件”是数据集合，当事人的陈述是模型，当事人需要使得自己目标最优，而法庭以法律目标找出最优的模型。

(1)法律目标：一个案件，多个陈述，法庭辨别哪个陈述合理。

相同目标有多个模型(**Breiman**)：对数据集合，不同算法导致多个模型，对目标(法律)，它们相同(判处一个人有罪)。但是，对问题的解释不尽相同(哪个人有罪)。那个模型解释最合理？

(2)律师目标：一个案件，为某个嫌疑人辩护，有所侧重。

多个有意义的模型(我们的扩展)：数据集合中有多个有意义的模型(多个当事人彼此目标不同)，是数据集合的不同侧面，不同用户需要不同的模型(偏好，对自己当事人有利)。

模型多样性的说明

Rashomon不是新问题，认知科学的二义图，特征选择、特征抽取、核函数选择等等都属于这类问题。只是没有作为普遍存在的问题。

解答有多个模型 → 哪个模型最优? → 对问题的可解释性

多个有意义模型 → 需要哪个模型? → 用户需求(偏好)

困难：为了获得一个好的解答，机器学习要求在样本空间上搜索，而模型多样性则必须在模型空间上搜索，以获得一个满足用户需求或对问题世界可解释性的解答。

次属性原理是模型空间搜索的规则。

模型多样性之一---特征选择

特征选择：模型多样性问题的最简单的任务。

经典方法：根据统计的方法，计算特征与数据标号之间的相关程度，由此，对特征排列为一个序。在矛盾样本个数不变(误差率不变) 与特征数量限制下，决定被选择的特征集合。

我们的思考：最终是对特征排序，假设用户了解特征的重要性，仅仅不能排成全序或者只能对部分特征排序，在矛盾样本个数不变的条件下，计算对给定特征序最小的特征集合。

特征选择以reduct作为解答。

内　容

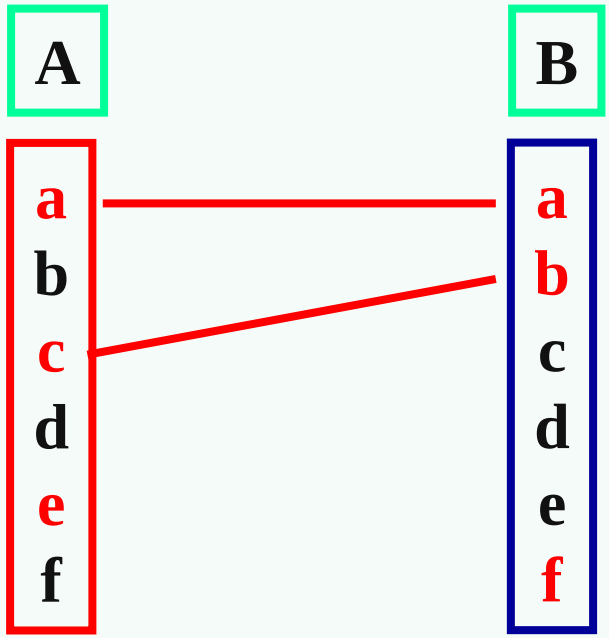
根据需求，从Reduct集合(模型)中搜索最优模型是NP-Hard。需要解决搜索规则问题，次属性定理(韩素青，2004)是其基础。

限制：特征值域为离散或符号。

1. 属性序：我们将特征称为属性。
2. Reduct：特征选择就是计算一个reduct。
3. 计算reduct的算法：语义问题。
4. 次属性定理：在reduct空间上搜索对给定属性序最优解的规则。
5. 搜索最优模型的算法。

Yao的设想(2005)

给定属性序，S：a→b→c→d→e→f。
如果存在两个reduct，A={a, c, e}与B={a, b, f}，



对属性序S，A中的c比B中的b靠后，且在A中，除了a，没有比c更靠前的reduct属性。

B比A更好的解答。

Yao设想是NP-Hard

击中集问题(Hitting Set)是NP-hard。

从击中集问题可在多项式时间规约到计算特定*reduct*问题，而**Yao**的设想可描述为计算特定属性序的*reduct*问题。

基于Yao设想的最优reduct问题是NP-hard!

内　容

1. 属性序：我们将特征称为属性。
2. **Reduct**：特征选择就是计算一个**reduct**。
3. 计算**reduct**的算法：语义问题。
4. 次属性定理：在**reduct**空间上搜索对给定属性序最优解的规则。
5. 搜索最优模型的算法。

Reduct定义

条件属性子集$R \subseteq C$是信息表$<U, C \cup D>$的**reduct**，如果R满足下述条件，

$$POS_R(D)=POS_C(D)$$

$$\forall r \in R, POS_R(D) \neq POS_{R-\{r\}}(D)$$

本质：删除R中的任一属性，矛盾对象增加(改变错误率)。

已有特征选择方法计算的属性子集与**reduct**的区别是：既不考虑属性冗余，也不考虑误差率是否改变。

差别矩阵(Discernibility Matrix)

给定<U, C∪{d}>。差别元素定义为：

$$m_{ij}=\{a : a(x)\neq a(y), \forall a\in C\cup\{d\}, x, y\in U\}$$

信息表

U	a	b	c	d	D
No.1	1	0	0	1	1
No.2	1	0	0	0	1
No.3	0	0	0	0	0
No.4	1	1	0	3	0
No.5	1	1	1	2	2
No.6	2	1	0	2	2
No.7	2	2	2	2	2

计算差别矩阵

差别矩阵

	1	2	3	4	5	6	7
1	d						
2	adD	aD					
3	bdD	bdD	abdD				
4	bcdD	bcdD	abcdD	cdD			
5	abdD	abdD	abdD	adD	ac		
6	abcdD	abcdD	abcdD	abcdD	abc	bc	
7							

定义在差别矩阵上的等价关系L(S)

给定属性序S: $a_1 \rightarrow a_2 \rightarrow \ldots \rightarrow a_k$。M是一个信息表的差别矩阵，$\alpha \in M$是差别元素，且从左到右满足S。

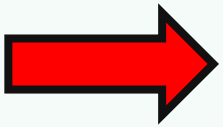
令a_j是α从左到右的第一个属性，称为α的标记属性。

根据每个差别元素的标记属性，可以将差别矩阵划分为不同的等价类。由此，定义等价关系L(S)：

$$L(S)=\{\alpha \mid \alpha=a_j\{b\}, \alpha\text{从左到右满足}S, \alpha \in M\}$$

对差别矩阵M的划分记为：M/L(S)。

内　容



1. 属性序：我们将特征称为属性。
2. Reduct：特征选择就是计算一个reduct。
3. 计算reduct的算法：语义问题。
4. 次属性定理：在reduct空间上搜索对给定属性序最优解的规则。
5. 搜索最优模型的算法。

基于属性序的reduct算法(王珏与王驹)

属性序S: a→b→c→d

差别矩阵： M/L(S)

[a]: ~~a, ad, abd, abcd~~

[b]: ~~bd, bcd~~

[c]: ~~cd~~

[d]:

M=∅

最大非空等价类标号为c

计算reduct

Reduct={ a b c }

[c]≠∅, 删除所有带c的元素。

[b]≠∅, 删除所有带b的元素。

[a]≠∅, 删除所有带a的元素。

由于这个算法对reduct完备且对给定属性序唯一，这个算法可以记为映射形式： R=DM(S)

对属性序的语义

算法语义：对给定属性序，在属性序的中间某个位置开始，计算一个**reduct**，换句话说，算法认为属性序中间这位置的属性最重要。向右的所有属性最不重要。

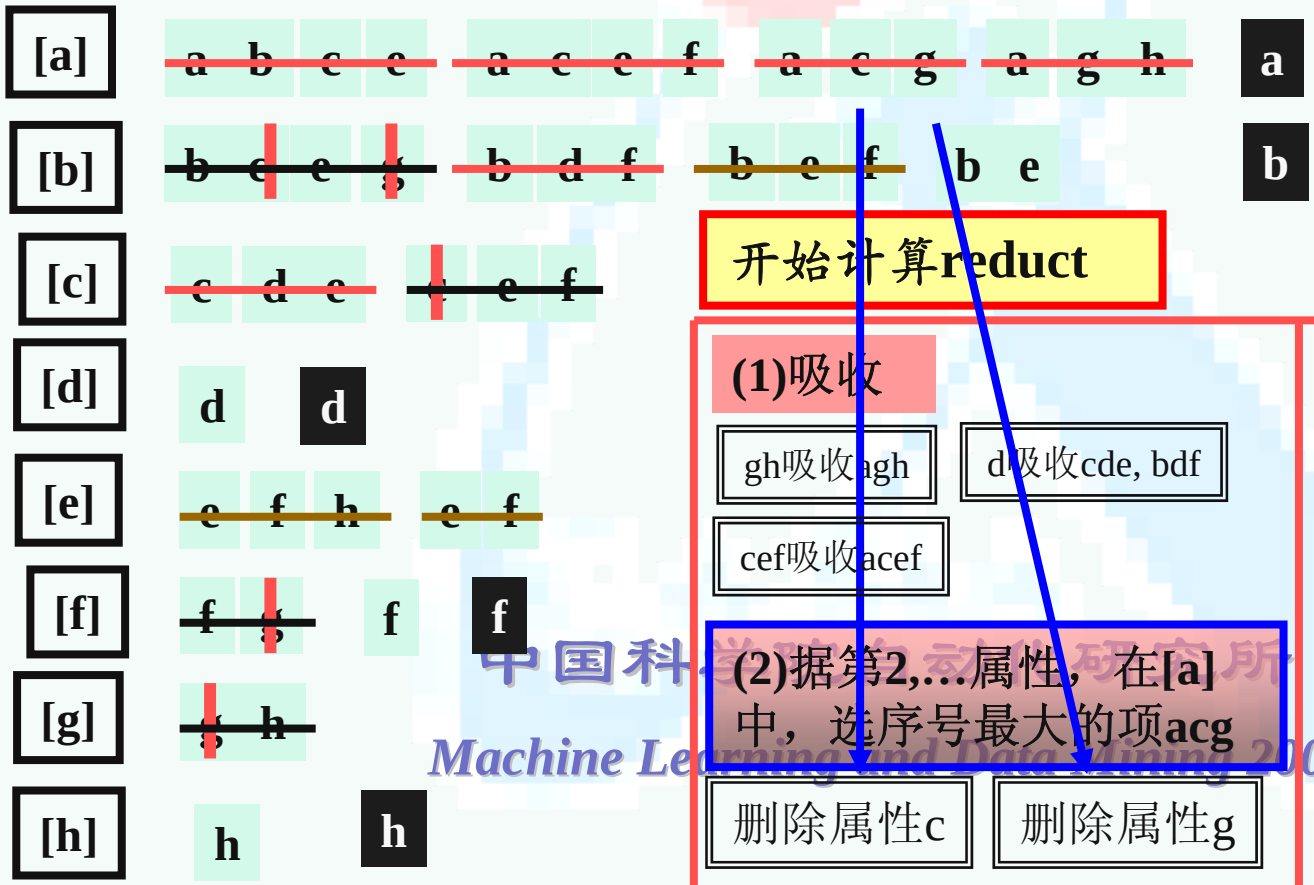
用户语义：给定属性序，从左到右(从右到左)，依次表述属性重要性。

算法对属性序的语义解释与用户解释不一致。

赵凯的尝试(2002)

差别矩阵：M/L(S)

属性序S: a→b→c→d→e→f→g→h



赵凯算法也不满足Yao的设想

差别矩阵：M/L(S)

属性序S: a→b→c→d→e

[a] ~~ab~~ ~~ac~~

[b]

[c] ~~cd~~ ~~ce~~

[d] ~~d~~

[e] ~~e~~

M=∅

计算reduct

Reduct={ a d e }

(1) 考虑属性a，并选择ac。

(2)删除属性c。删除[a]，a是reduct属性。

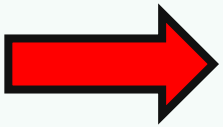
(3) 根据属性序整理。

(4) d与e是reduct属性。

对属性序S，最优Reduct={ a c }

内　容

1. 属性序：我们将特征称为属性。
2. Reduct：特征选择就是计算一个reduct。
3. 计算reduct的算法：语义问题。
4. 次属性定理：在reduct空间上搜索给定属性序最优解的规则。
5. 搜索最优模型的算法。



属性序：事实与性质

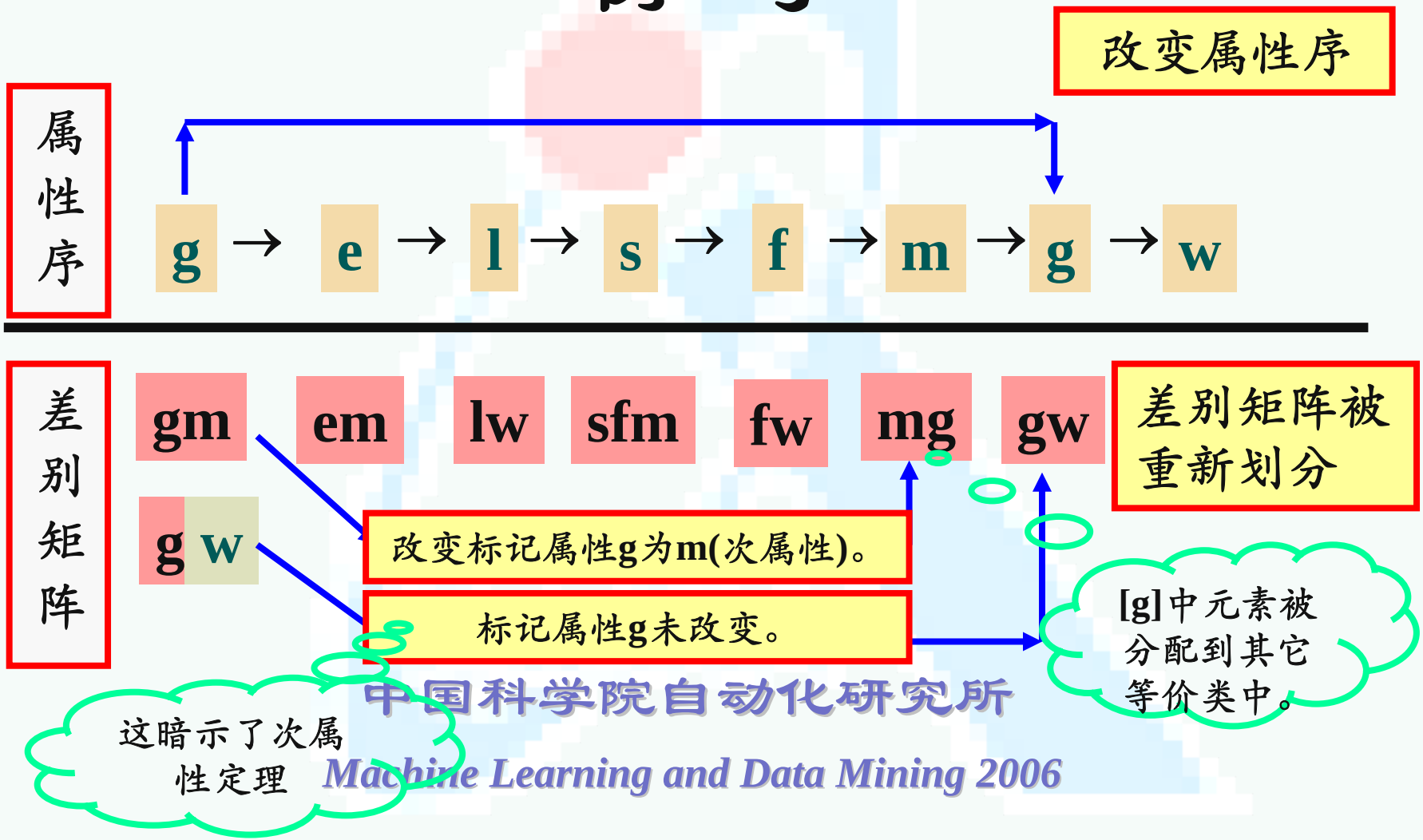
事实：对给定属性序S，移动其中任一个属性到其它位置，产生一个新的属性序S’。

如果DM(S)=DM(S’)，需要满足什么条件？

性质：由于属性序变化，必然导致差别元素的属性排序变化，根据在差别矩阵上定义的等价关系L(S)，必然导致M/L(S)≠M/L(S’)。

右移一个属性，某些差别元素的标记属性将右移，这样，原差别元素的次属性将成为新属性序M/L(S’)的标记属性。

例　子


改变属性序
属性序
g → e → l → s → f → m → g → w
差别矩阵
gm em lw sfm fw mg gw
g w
改变标记属性g为m(次属性)。
标记属性g未改变。
差别矩阵被重新划分
[g]中元素被分配到其它等价类中。
这暗示了次属性定理
中国科学院自动化研究所
Machine Learning and Data Mining 2006

次属性的定义

M/L(S)={[1], [2], … , [k]}，其中[j]是L(S)划分M的一个等价类。令R=DM(S)，是关于S的一个reduct。

令P⊆R。对S，在P中的所有属性序号大于(R-P)中的属性序号，P中最小属性序号记为j。

相对等价类：假设[j-1]≠∅，所有α∈[j-1]，满足α∩P=∅的差别元素集合，称为M/L(S)的相对等价类，记为[j-1]'。

次属性：对S，令[j-1]'≠∅，Σ是[j-1]'中所有差别元素排列第二的属性序号集合，其中最大的记为m=Max(Σ)，属性为a_m，a_m称为等价类[j-1]的次属性，m称为[j-1]次属性属性序标号。

例　子

属性序S: b(1)→c(2)→f(3)→a(4)→d(5)→e(6)→g(7)

M是差别矩阵， M/L(S):

1. [b]: { b, ~~bc~~ }
2. [c]: { ~~cfg~~, cde, ~~cf~~ }
3. [f]: { fa, fg }

[a]，[d]，[e]，[g]为空。

REDUCT={b, c, f}

f是reduct属性，删带f的元素。

b是核属性，删除bc。

计算相对等价类

1. [b]: {b}
2. [c]: {cde}
3. [f]: { fa, fg }

[b]：核属性，次属性无穷大。

[c]：次属性为d(m=5)。

[f]：由于g的属性序号为7，大于a的4，它的次属性为g(m=7)。

次属性定理(韩素青，2004)

给定信息表<U, C∪{d}>,其中C={a_1,a_2,…a_k}，对定义在C上的属性序P，右移属性a_i(1≤i≤k)到y (i<y≤k)位置获得属性序Q。对P，设[j]的次属性的属性序标号为m。则P与Q具有相同reduct，当且仅当y<m。

只考虑右移，左移将是它的对称系统。

证明思路(1)---核属性

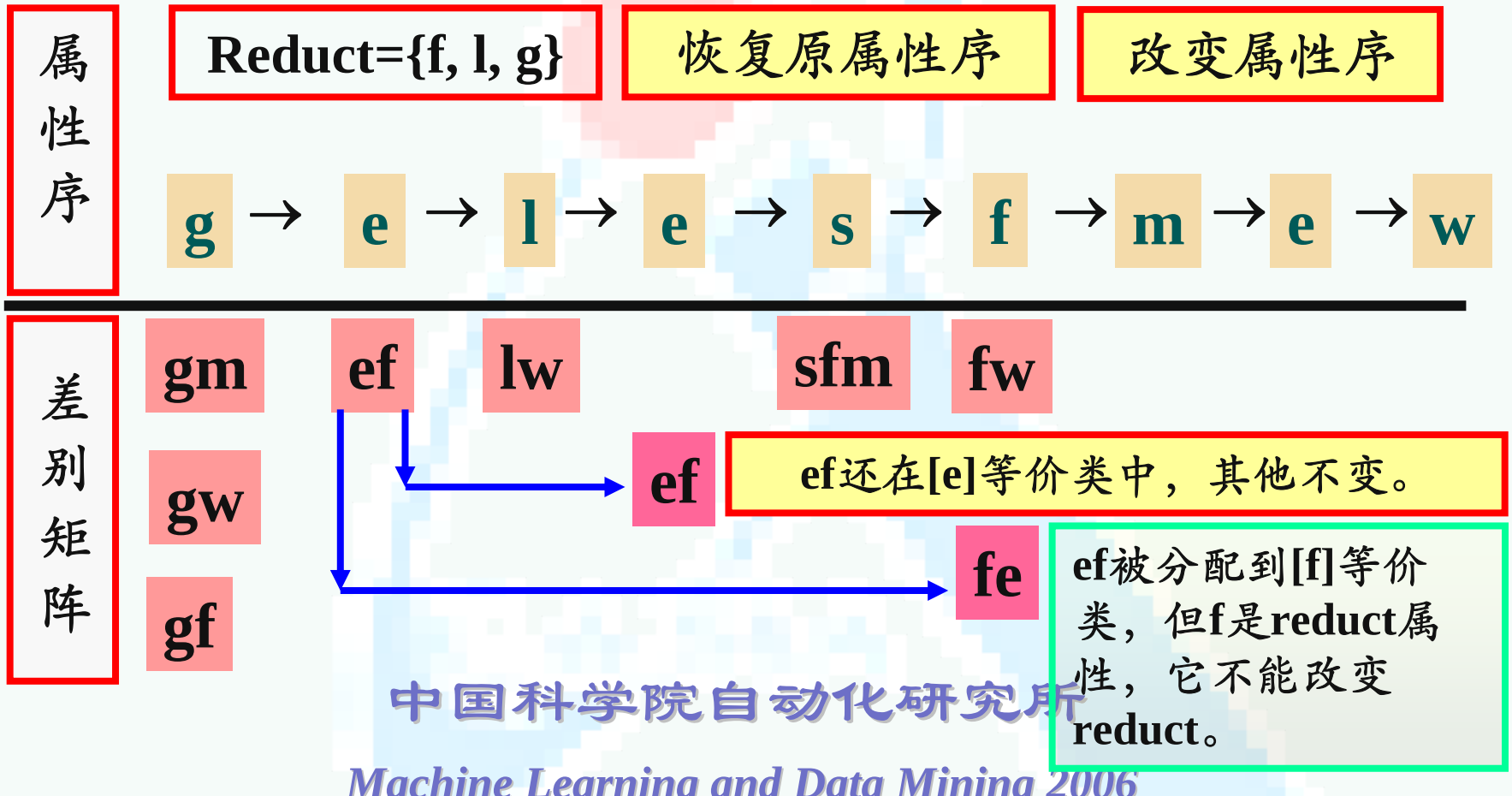
核属性右移，reduct不变。令c是一个核属性.

(1)核属性是给定信息表所有reduct的reduct属性。

(2)核属性对应的相对等价类中只有一个差别元素，就是核属性本身，{b}。没有次属性！

核属性可右移到任何位置，它的达到界m，定义为无穷大

证明思路(2)---非reduct属性



证明思路(2)---非reduct属性

非reduct属性右移，reduct不变。令r是一个非reduct属性.

(1)[r]中差别元素全部被分配到其他等价类中。它们不可能变为reduct属性，这样，这些元素将被其他reduct属性删除。

(2)[r]中差别元素全部被保留，显然，reduct不能改变。

(3)[r]部分被分配到其他等价类中，部分被保留在[r]。综合上述两种情况，改变reduct不可能。

注意：非reduct属性的相对等价类为空集，因此，它与核属性一样，没有次属性。m定义为无穷大，非reduct属性右移的界任意.

证明思路(3)---reduct属性

属性序

Reduct={f, l, g}　恢复原属性序　改变属性序

g → e → l → s → f → m → g → w → g

差别矩阵

gm　ef　lw　sfm　fw　mg　gw　wg

gw　移g到[g]次属性w前，[g]非空，g是reduct属性。解不变.

移g到[g]次属性w后，[g]空，[w]变，g不是reduct属性。解变.

[g]的次属性为w

证明思路(3)---reduct属性

Reduct属性r右移，未超过m，reduct不变。

Reduct属性r右移到[r]的次属性之前，在新属性序下，至少存在一个差别元素，被保留在新属性序的[r]中，r还是reduct属性。其他被分配到其他等价类的差别元素，不可能改变reduct。

Reduct属性r右移，超过m，reduct一定改变。

Reduct属性r右移到[r]的次属性之后，在新属性序下，原属性序下的[r]的全部差别元素被分配到其他等价类中，在新属性序下的[r]成为空集，r一定不是新序的reduct属性，两个属性序的reducts一定不同。

多个属性移动

从一个属性序变为另一个属性序，需要移动多个属性。

可以连续移动属性的条件是，各个等价类的次属性不能改变。不幸的是，只要移动非**reduct**属性，次属性必然改变！

属性范序：给定属性序**P**，根据**DM**算法计算它的**reduct R**，将**R**中的所有属性移至属性序的最左边，由此构成一个新的属性序**Q**，称为**P**对**R**的属性范序。

根据次属性定理，**DM(P)=DM(Q)**。

多个属性移动需要基于属性范序。

次属性的计算方法

由于次属性定理的证明是构造性的，因此，可以计算。

(1)计算属性范序。
(2)对非reduct属性排序，使其与给定属性序一致。
(3)计算reduct属性的次属性。
(4)对reduct属性排序，使其与给定属性序一致。

在树表示基础上，已被发展的计算次属性算法的复杂性与对象个数呈近似线性关系。(韩素青，2006)

我们不可能详细介绍这个证明，上述是证明思路的说明。

为什么仅关注次属性

一个重要的事实:

可以容易地证明，从一个属性序变为另一个属性序，需要移动的每个属性只须移动一次。

3属性，4属性，…，k属性等等永远不可能变为新属性序下M/L(S')的标记属性。因此，在给定属性序下移动属性，它们不可能成为判定reduct改变的特征。

只有次属性是判定reduct改变的特征!

计算Reduct的树表示算法(赵岷，2004)

可以剪枝的子树

A单支，D单支　A单支，D多支

A多支，D每个分支多支

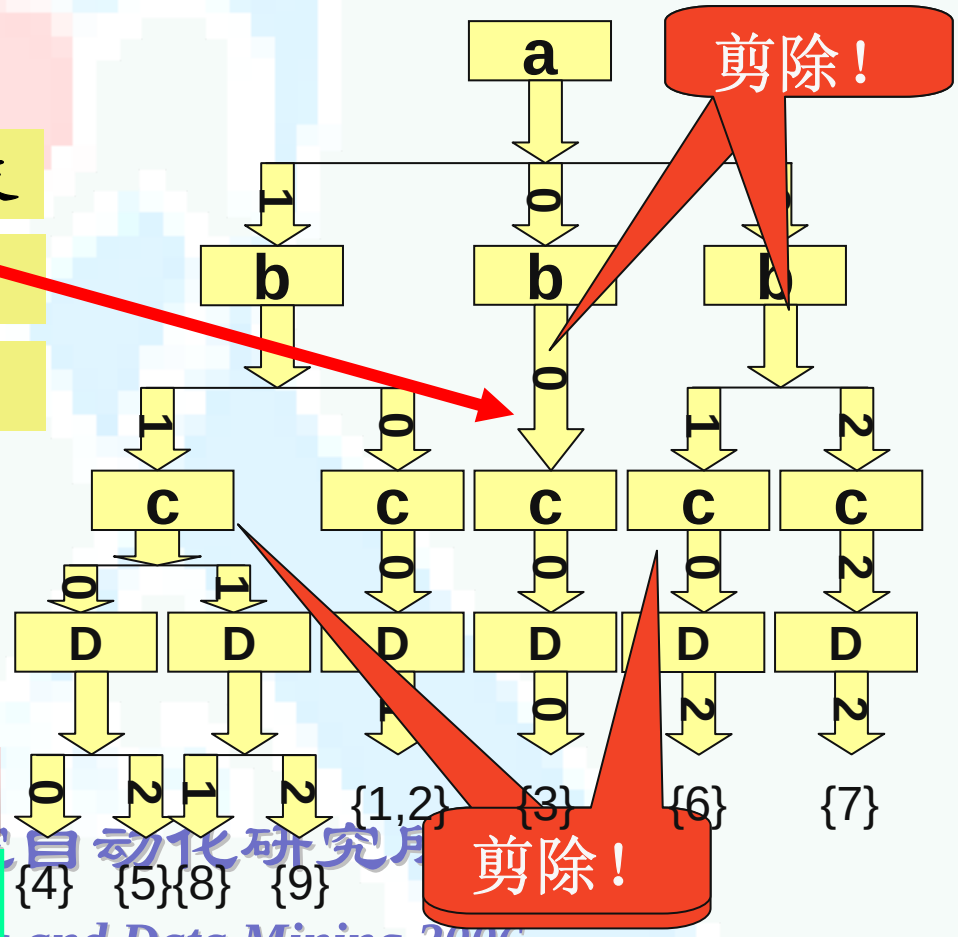
A多支，D每个分支有相同的值

不可以剪枝的子树

A多支，D每支单支，且决策属性值不同

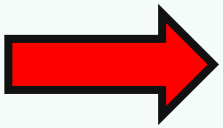
A多支，D每支，单支多支混合

这个算法记为：TRX(S)。



内　容

1. 属性序：我们将特征称为属性。
2. Reduct：特征选择就是计算一个reduct。
3. 计算reduct的算法：语义问题。
4. 次属性定理：在reduct空间上搜索给定属性序最优解的规则。
5. 搜索最优模型的算法。



标准属性范序(梁洪力)

标准属性范序：令P是属性序，R=DM(P)，Q是P对R的属性范序，如果Q中R的reduct属性和非reduct属性均按照属性序P从左到右排列顺序相同，则Q称为对R的标准属性范序。DM(P)=DM(Q)。

例子：令P：a→b→c→d→e是属性序，R={a, c, e}是reduct，a→c→e→b→d与c→a→e→b→d是两个属性范序，但是，前者是标准属性范序。

标准属性范序将保证次属性不再改变。

基本概念

假设Q是一个属性标准范序。

将属性a_i右移至Q末尾得到属性序Q'，称Q'为Q的**邻居**，由Q的所有邻居组成的集合称为Q的**邻域**，记作N(Q)。

令R_Q=DM(Q)与$R_{Q'}$=DM(Q'). (或TRX).

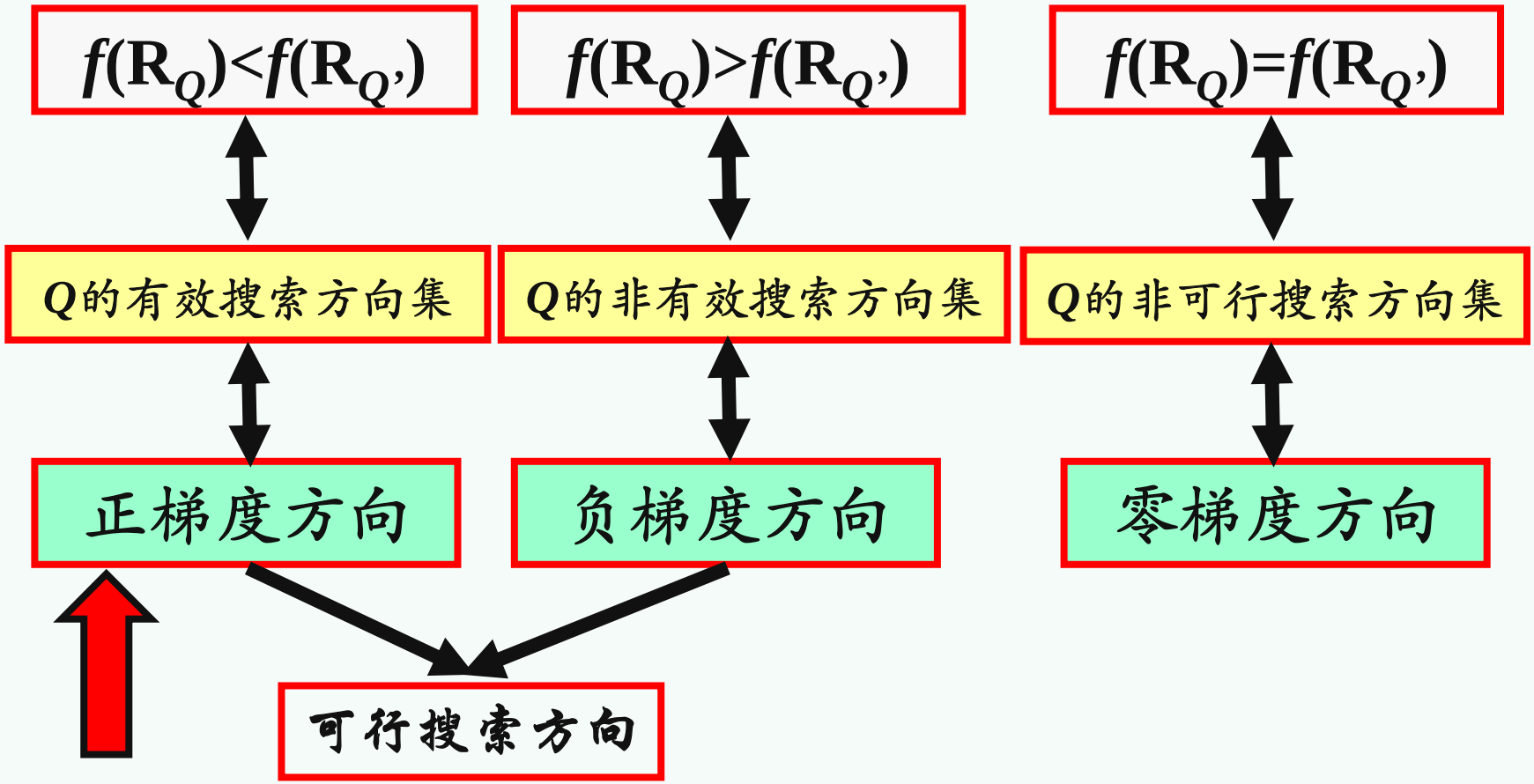
如果Q的邻居Q'满足$R_Q \neq R_{Q'}$，则称Q'为Q的**可行搜索方向**。Q的所有可行搜索方向称为Q的可行搜索方向集，记作 N'(Q)。

如果Q'满足$f(R_Q) < f(R_{Q'})$，则称Q'为Q的**有效搜索方向**。

$$f(R_Q) = \sum_{a \in R_Q} 2^{m-i(a)}$$

其中，i(a)是属性a在属性序Q中的序号。

搜索规则的定性描述


$f(R_Q)<f(R_{Q'})$
$f(R_Q)>f(R_{Q'})$
$f(R_Q)=f(R_{Q'})$
Q的有效搜索方向集
Q的非有效搜索方向集
Q的非可行搜索方向集
正梯度方向
负梯度方向
零梯度方向
可行搜索方向

搜索规则1：可行搜索方向

只有核属性与非reduct属性的m为无穷大。

搜索规则1

移动核属性与非reduct属性是非可行搜索方向。

搜索规则2：有效搜索方向

规则1排除零梯度方向，还需排除负梯度方向。

根据次属性定理，可以证明下属命题：

用户给定属性序P: $a_1 \rightarrow a_2 \rightarrow \ldots \rightarrow a_m$，$R=\{a_{j1}, a_{j2}, \ldots, a_{jp}\}$是它的*reduct*，并根据属性序*P*从左到右排列。如果*reduct R*的前*k*个属性$a_{j1}, \ldots, a_{jk}$与属性序*P*的前*k*个属性相同，即，$a_{ji}=a_i$，$1 \le i \le p$，在属性序P上，将a_i移动到属性序的末尾，并获得新的属性序Q。假设$R_P=DM(P)$与$R_Q=DM(Q)$，则$f(R_Q)<f(R_P)$。

搜索规则2

命题是属性序*Q*属于*P*非有效搜索方向集(负梯度)的充分不必要条件。

邻域贪婪reduct算法(梁洪力)

基于两个搜索规则的reduct优化算法称邻域贪婪reduct算法。

邻域贪婪reduct算法

(1)根据DM(或TRX)算法计算用户给定的属性序的reduct R, 并构造其标准属性范序*P*。

(2)在*P*的可行搜索方向集内选择一个使$f(R_Q)$最大的有效搜索方向Q。

(3)直到*P*的有效搜索方向集为空，算法停止。

总结：特征选择

Rashomon问题的要点是考虑偏好的模型(或模型可解释性)，这里，我们仅讨论了其中的一个比较容易解决的特征选择问题。

我们希望建立特征选择一般性理论框架---包括模型选择，并将这个框架建立在序语言与**reduct**理论上，这项研究仅仅是起点。

总结：属性序与特征选择

无特定目标的一般性讨论属性序没有意义！

例如，根据属性序选择前n个属性的信息表，没有什么意义。

Filter类：使得大于阈值的属性被选择。不能保证没有属性冗余，且不能保证误差率不变。

我们方法：**Reduct**保证没有属性冗余且不改变误差率，属性序保证解答是用户需要的，并其作为特征选择算法的一部分。

如果我们增加一个要求---reduct，则Filter类算法是我们算法的一个特例。

总结：模型空间搜索

机器学习的理念：在统计目标驱动下，在样本空间上搜索一个逼近最优模型的路径(例如，分界面)。其搜索规则最常用的就是梯度下降法，或类似方法。

我们的考虑：在偏好目标驱动下，在模型空间(reduct集)上搜索最符合偏好的模型。

次属性原理是在模型空间上的搜索规则，我们希望它能够成为解决模型多样性(用户偏好解释)的优化搜索的基本原理之一。

总结：问题

由于这个框架的所有部分，只使用了等价关系，而没有使用**reduct**理论中规定的特殊等价关系，因此，将其扩展到连续特征情况应该是可以考虑的问题，尽管在理论上还有一些困难。

差别矩阵可以考虑为在特定空间上样本之间的“距离”，因此，是否可以与统计机器学习与集群机器学习建立联系。

次属性定理可以推广到其他机器学习方法中吗?

由于在机器学习中，统计量的作用一般不是描述问题世界的性质，而是为了排序(Fuzzy方法更是如此)，因此，序结构是否可以成为机器学习算法的一部分?

选择一个明确的目标，一直朝它奔去。你可能永远达不到你的目的，但是，在你前进途中，一定会找到一些有兴趣的东西。

---费利克斯•克莱因---

谢 谢

中国科学院自动化研究所

Machine Learning and Data Mining 2006