基于相对决策熵的决策树算法及其在入侵检测中的应用

江 峰[1] 王春平[2] 曾惠芬[3]
(青岛科技大学信息科学与技术学院 青岛 266061)[1]
(浙江工业大学计算机科学与技术学院 杭州 310023)[2] (九江职业技术学院 九江 332007)[3]

摘 要 为了弥补传统决策树算法的不足,提出一种基于相对决策熵的决策树算法DTRDE。首先,将Shannon提出的信息熵引入到粗糙集理论中,定义一个相对决策熵的概念,并利用相对决策熵来度量属性的重要性;其次,在算法DTRDE中,采用基于相对决策熵的属性重要性以及粗糙集中的属性依赖性来选择分离属性,并且利用粗糙集中的属性约简技术来删除冗余的属性,旨在降低算法的计算复杂性;最后,将该算法应用于网络入侵检测。在KDD Cup99数据集上的实验表明,DTRDE算法比传统的基于信息熵的算法具有更高的检测率,而其计算开销则与传统方法接近。

关键词 决策树,粗糙集,信息熵,相对决策熵,属性重要性,入侵检测
中图法分类号 TP181 **文献标识码** A

Relative Decision Entropy Based Decision Tree Algorithm and its Application in Intrusion Detection

JIANG Feng[1] WANG Chun-ping[2] ZENG Hui-fen[3]
(College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China)[1]
(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)[2]
(Jiujiang Vocational and Technical College, Jiujiang 332007, China)[3]

Abstract To overcome the disadvantages of traditional decision tree algorithms, this paper proposed a relative decision entropy based decision tree algorithm DTRDE. First, we introduced the information entropy proposed by Shannon into rough set theory, defined a concept of relative decision entropy, and utilized the relative decision entropy to measure the significance of attributes. Second, in algorithm DTRDE, we adopted the relative decision entropy based significance of attributes and the dependency of attributes in rough sets to select splitting attributes. And we used the attribute reduction technology in rough sets to delete the redundant attributes, aiming to reduce the computation complexity of our algorithm. Finally, we applied the proposed algorithm to network intrusion detection. The experiments on KDD Cup99 dataset demonstrate that DTRDE algorithm has higher detection rate than the traditional information entropy based algorithms, and its computational expense is simliar to those of the traditional methods.

Keywords Decision tree, Rough sets, Information entropy, Relative decision entropy, Significance of attributes, Intrusion detection

目前,基于信息熵的决策树算法还存在很多问题[1,2],例如决策树中子树有重复;某些属性在树中的某一路径上被多次检验,从而降低了分类的效果[3]。尤其是当数据集中存在大量冗余属性时,这类方法所生成的决策树过于庞大,且效率低下。

如何将决策树与其他数据挖掘技术相结合以取长补短,成为近年来的研究热点,其中决策树与粗糙集的结合引起了广泛关注[3-6]。自Pawlak提出粗糙集理论以来[7,8],粗糙集受到了广泛关注。属性约简、属性重要性和依赖性是其核心内容[7-13]。

近年来,已经有不少基于粗糙集的决策树算法被提出[3-6]。为了进一步解决传统算法所存在的问题,提出一种新的决策树算法DTRDE,并将该算法应用于入侵检测[14-17]。DTRDE采用一种新的分离属性选择标准——相对决策熵。与ID3及C4.5所采用的信息增益和信息增益率相比[1,2],相对决策熵的计算更为简单有效。与现有的基于粗糙集的决策树算法所采用的基于正区域的分离属性选择标准相比[3-6],相对决策熵更为全面地反映了每个属性对决策分类的贡献。另外,在DTRDE中考虑了属性之间的依赖性和相关性,从而弥补了传统方法对于属性的依赖性强调不够这一缺点。

为了将DTRDE更好地应用于入侵检测,利用粗糙集中的属性约简技术来预先删除与决策无关的属性[9-13],以降低算法的复杂性。

1 相关概念介绍

在粗糙集中,信息表是一个四元组 $IS=(U,A,V,f)$,其

到稿日期:2011-06-11 返修日期:2011-09-29 本文受国家自然科学基金项目(60802042,61103246),山东省自然科学基金项目(ZR2011FQ005,ZR2011FQ026,ZR2010FQ027)资助。

江 峰 博士,副教授,主要研究方向为人工智能、粗糙集理论等。

中 U 和 A 分别表示对象集和属性集；V 是所有属性论域的并，即 $V=\bigcup_{a\in A}V_a$，其中 V_a 为属性 a 的值域；$f:U\times A\to V$ 是一个信息函数，使得对任意 $a\in A$ 以及 $x\in U$，$f(x,a)\in V_a$[7,8]。

进一步，A 又可以划分为两个不相交的子集——条件属性集 C 和决策属性集 D。这种特殊的信息表被称为决策表，简记 $DT=(U,C,D,V,f)$。

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 $IND(B)$ 为 $IND(B)=\{(x,y)\in U\times U:\forall a\in B(f(x,a)=f(y,a))\}$。

$IND(B)$ 是 U 上的一个等价关系，将 U 划分成多个等价类，所有这些等价类的集合就构成 U 的一个划分，记为 $U/IND(B)$。

定义 1(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-上近似和 B-下近似分别定义为

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

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

定义 2(粗糙度) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $B\subseteq C\cup D$ 和 $X\subseteq U(X\neq\emptyset)$，集合 X 的 B-粗糙度定义为

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

定义 3(正区域) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $B\subseteq C$，定义 D 的 B-正区域为

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

定义 4(属性依赖度) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $c\in C$，D 对 c 的依赖度定义为

$$\gamma_c(D)=|Pos_{\{c\}}(D)|/|U|$$

2 相对决策熵

为了度量不确定性，人们将信息熵引入到粗糙集[18]，提出了知识熵、粗糙熵和条件熵等模型[9,11,19,20]。

传统的决策树算法采用基于信息熵的分离属性选择标准，这些方法还存在诸多问题。因此，本文将定义一种新的信息熵模型——相对决策熵，并基于该模型来选择分离属性。

定义 5(相对决策熵) 给定决策表 $DT=(U,C,D,V,f)$，令 $U/IND(D)=\{D_1,\cdots,D_m\}$ 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

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

式中，$\rho_B(D_i)$ 为 D_i 的 B-粗糙度，$1\leqslant i\leqslant m$。

与粗糙集中现有的信息熵模型不同，相对决策熵采用 Pawlak 所提出的粗糙度进行定义。粗糙度已被证明是一种刻画集合不确定性的有效方式，被广泛应用于粗糙集的各个领域，本文进一步利用它来刻画知识的不确定性，并且基于该不确定性度量模型设计出一种新的分离属性选择标准。

定理 1 给定决策表 $DT=(U,C,D,V,f)$，其中 $U=\{x_1,x_2,\cdots,x_n\}$。令 $U/IND(D)=\{D_1,\cdots,D_m\}$ 为 $IND(D)$ 对 U 的划分，对任意 $B\subseteq C$，有

(1) $0\leqslant RDE(D,B)\leqslant n$；

(2) 当 $U/IND(B)=\{\{x_1\},\{x_2\},\cdots,\{x_n\}\}$ 时，$RDE(D,B)$ 取最小值 0；

(3) 当 $U/IND(D)=\{\{x_1\},\{x_2\},\cdots,\{x_n\}\}(n>1)$ 且 $U/IND(B)=\{U\}$ 时，$RDE(D,B)$ 取最大值 n。

证明：

(1) 由定义 1 可知，对任意 $B\subseteq C$ 和 $X\subseteq U$，$\overline{X}_B\geqslant\underline{X}_B$。再由定义 2 可以得出，对任意 $B\subseteq C$ 和 $D_i\in U/IND(D)$，$0\leqslant\rho_B(D_i)\leqslant 1$，其中 $1\leqslant i\leqslant m$[10]。

由定义 5 可知，$RDE(D,B)=\sum_{i=1}^{m}\rho_B(D_i)\log_2(\rho_B(D_i)+1)$，因此，$0\leqslant RDE(D,B)\leqslant m$，其中 m 是一个变量，并且 $1\leqslant m\leqslant n$。由此可以得出 $0\leqslant RDE(D,B)\leqslant n$。

(2) 当 $U/IND(B)=\{\{x_1\},\{x_2\},\cdots,\{x_n\}\}$ 时，由定义 1 可知，对任意 $D_i\in U/IND(D)$，$D_{i\,B}=D_{i\,B}=D_i$，$1\leqslant i\leqslant m$。再由定义 2 可知，对任意 $1\leqslant i\leqslant m$，$\rho_B(D_i)=0$。因此，$RDE(D,B)=0$。

(3) 当 $U/IND(D)=\{\{x_1\},\{x_2\},\cdots,\{x_n\}\}(n>1)$ 且 $U/IND(B)=\{U\}$ 时，由定义 1 可知，对任意 $D_i\in U/IND(D)$，$D_{i\,B}=\emptyset$，$1\leqslant i\leqslant n$。再由定义 2 可知，对任意 $1\leqslant i\leqslant n$，$\rho_B(D_i)=1$。因此，$RDE(D,B)=\sum_{i=1}^{n}1\times 1=n$。

定理 2 给定决策表 $DT=(U,C,D,V,f)$，其中 $U/IND(D)=\{D_1,\cdots,D_m\}$。对任意 $P,Q\subseteq C$，如果 $P\supseteq Q$，则 $RDE(D,P)\leqslant RDE(D,Q)$。

证明：

在粗糙集理论中，根据定义 2 可以得出这样的结论，即如果 $P\supseteq Q$，则对任意 $X\subseteq U$，$\rho_P(X)\leqslant\rho_Q(X)$[10]。

因此，如果 $P\supseteq Q$，则对任意 $D_i\in U/IND(D)$，$\rho_P(D_i)\leqslant\rho_Q(D_i)$，$1\leqslant i\leqslant m$。再由定义 5 可知，$RDE(D,P)=\sum_{i=1}^{m}\rho_P(D_i)\log_2(\rho_P(D_i)+1)\leqslant\sum_{i=1}^{m}\rho_Q(D_i)\log_2(\rho_Q(D_i)+1)=RDE(D,Q)$。这样，定理 2 得证。

定义 6(属性重要性) 给定决策表 $DT=(U,C,D,V,f)$，对任意 $B\subset C$ 和 $c\in C-B$，我们将 c 相对于 B 和 D 的重要性定义为

$$Sig(c,B,D)=RDE(D,B)-RDE(D,B\cup\{c\})$$

3 算法 DTRDE

算法 1 DTRDE

输入：决策表 $T_1=(U,C,D,V,f)$

输出：规则集 R

Function Main(T_1)

(1) 对 C 中的连续型属性分别采用等频区间和等宽区间方法进行离散化[21]，令 $T_2=(U_2,C,D,V_2,f_2)$ 为离散化后的决策表。

(2) 对 T_2 中的属性集 C 采用刘少辉等的约简方法计算其约简[12]，令 Red 为 C 的约简。

(3) 根据约简 Red，去掉 T_2 中的冗余属性，并删除重复记录，令 $T_3=(U_3,Red,D,V_3,f_3)$ 为约简后的决策表。

(4) 调用函数 Decision_Tree(T_3)，在 T_3 上生成决策树 Tr。

(5) 遍历 Tr 中每条由根至叶节点的路径，生成规则集 R。

(6) 返回 R。

Function Decision_Tree(T_{temp})

(注：$T_{temp}=(U',C',D,V',f')$ 为当前正在使用的临时决策表)

(1) 令 B=Red−C'。

(2) 如果 B=∅，则令 RDE(D,B)=0；否则通过对 U'中的对象进行基数排序，来分别计算 U'/IND(D)=$\{D_1,\cdots,D_m\}$，U'/IND(B)和 U'/IND(B∪D)。并对任意 $D_i\in U'/IND(D)$，计算 $\rho_B(D_i)$，$1\leqslant i\leqslant m$。最后，计算 RDE(D,B)。

(3) 对任意 $c\in C'$，反复执行：

(3.1) 通过对 U'中的对象进行基数排序，来计算 U'/IND(B∪

{c})和 U′/IND(B∪{c}∪D)。

(3.2) 对任意 $D_i \in U'/IND(D)$，计算 $\rho_{B\cup\{c\}}(D_i)$，$1 \leqslant i \leqslant m$。

(3.3) 计算 RDE(D,B∪{c})。

(3.4) 计算 Sig(c,B,D)=DE(D,B)−RDE(D,B∪{c})。

(4) 从 C′中选择重要性最大的属性作为分离属性。

(5) 如果存在多个属性的重要性同为最大(假设 A 包含了所有这些属性)，则对任意 $a \in A$，计算 $\gamma_a(D)$，并选择值最大的作为分离属性。

(6) 创建节点 N，用分离属性 t 来标记 N，并令 C′=C′−{t}。

(7) 通过对 U′中对象进行基数排序，来计算 $U'/IND(\{t\})=\{S_1,\cdots,S_k\}$。

(8) 对任意 $S_j \in U'/IND(\{t\})$，反复执行：

(8.1) 建立 N 的一个分支，并标记相应的属性值。

(8.2) 如果 S_j 中的样本都来自于同一个决策类别，则新建一个标有该类别名的叶子节点。

(8.3) 如果 S_j 中的样本来自于多个不同的决策类别，并且 C′=∅，则新建一个叶子节点，并以支持度最大的决策类别来标记它。

(8.4) 如果 S_j 中的样本来自于多个不同的决策类别，并且 C′=∅，则创建 T_{temp} 的一个子表 $T_{sub}=(S_j,C',D,V_{sub},f_{sub})$，并且调用函数 Decision_Tree($T_{sub}$)来生成子树。

通常，计算划分 $U/IND(B)$ 的复杂度为 $O(|U|^2)$。在算法 1 中，采用预先对 U 中对象进行基数排序，然后求 $U/IND(B)$ 的方法[13]，从而使得计算 $U/IND(B)$ 的复杂度降为 $O(|B|\times|U|)$。

在最坏的情况下，算法 1 的 Main 函数中第(4)步的时间复杂度为 $O(|Red|^3\times|U_3|)$。此外，算法 1 的 Main 中第(1)步的复杂度为 $O(|C|\times|U|)$、第(2)步的复杂度为 $O(|C|^2\times|U_2|\times\log|U_2|)$。因此，算法 1 的复杂度为 $O(|Red|^3\times|U_3|+|C|^2\times|U_2|\times\log|U_2|)$。

为将 DTRDE 更好地用于入侵检测，进行如下预处理：

(1) 属性离散化。入侵检测中存在着大量的连续型属性，例如 KDD Cup99 中有 41 个属性，其中 34 个是连续型属性，只有 7 个是离散型属性[22]。然而，决策树更适合于处理离散型属性。为了将决策树应用于入侵检测，需要事先对其进行离散化。本文将主要采用两种无监督的离散化算法来处理连续型属性。

(2) 属性约简。面对海量的数据，为了实时地检测入侵，必须进行属性约简。在入侵检测之前将那些冗余的属性过滤掉，可以有效降低这些属性对入侵检测的干扰，加快检测时间，并提高检测的准确率。

4 实验

在 KDD Cup99 数据集上进行实验[22,23]所用的硬件环境是 Intel 处理器 2.0GHz，2GB 内存。我们将比较 DTRDE、ID3 和 C4.5 的性能，其中 DTRDE 采用 Pascal 语言实现；ID3 和 C4.5 使用 Weka 中所提供的分类器[21]。

4.1 实验数据

由于 KDD Cup99 太大，因此选取它的一个子集“10%-KDD”来进行实验。10%-KDD 中共有 22 种攻击类型[22]。然而，10%-KDD 数据集仍然太大。因此，仿照文献[24]的做法，对该数据集中的各种攻击类型和正常连接，依据不同比例采用随机不放回的方式抽取一定数量的记录，最终产生一个包含 15095 条记录的数据集“Final-Dataset”。随机抽取的策略如表 1 所列。

表 1 各种攻击类型和正常连接所包含的记录数

攻击类型与正常连接	初始记录数	随机抽取的记录数
back	2203	220
buffer_overflow	30	30
ftp_write	8	8
guess_passwd	53	53
imap	12	12
ipsweep	1247	125
land	21	21
loadmodule	9	9
multihop	7	7
neptune	107201	1072
nmap	231	231
normal	97278	9730
perl	3	3
phf	4	4
pod	264	264
portsweep	1040	104
rootkit	10	10
satan	1589	160
smurf	280790	2808
spy	2	2
teardrop	979	100
warezclient	1020	102
warezmaster	20	20
Total	494021	15095

4.2 实验步骤与设置

(I)数据准备与离散化。我们采用等宽区间(Equal Width Binning，EW)和等频区间(Equal Frequency Binning，EF)算法来离散化 Final-Dataset 中的属性。对于这两种算法，区间数都设置为 3[21]。

对每个经过 EW 和 EF 离散化之后的数据集，将其随机分成两个部分：训练集(占 60%)和测试集(占 40%)。

(II)数据约简。采用刘少辉等的约简算法[12]，在每个训练集上计算约简集，并删除冗余属性和重复记录。

(III)生成决策树与决策规则。分别采用 DTRDE、ID3 和 C4.5 算法，在每个约简后的训练集上创建决策树，并生成决策规则。

(IV)分类测试。对前面所产生的决策规则在测试集上进行测试，以验证算法的分类性能。对 ID3 和 C4.5，测试过程同样借助 Weka 来完成[21]。对 DTRDE，采用 Rosetta 中所提供的 Standard Voting 方法来实现[25]，相关设置如下：CLASSIFIER = StandardVoter; FALLBACK = True; FALLBACK. CLASS=DoS。

4.3 实验结果

表 2 列出了在由 EW 所离散化的数据集上的分类结果。

表 2 EW 所离散化的数据集上的分类结果

决策树算法	建模所使用的属性个数	检测率(%)	建模时间(s)
DTRDE	24	97.748	1.187
ID3	41	97.665	1.06
C4.5	41	97.582	1.63

注：对于 DTRDE 算法而言，建模时间包括之前的约简时间。

从表 2 可以看出，在由 EW 所离散化的数据集上，DTRDE 的性能要好于 ID3 和 C4.5。这是因为 DTRDE 的检测率要高于 ID3 和 C4.5，而它的建模时间与这两个算法很接近。

另外，表 3 给出了在由 EF 所离散化的数据集上的分类结果。

表 3 EF 所离散化的数据集上的分类结果

决策树算法	建模所使用的属性个数	检测率(%)	建模时间(s)
DTRDE	15	99.304	0.792
ID3	41	99.023	0.83
C4.5	41	98.924	0.74

从表 3 可以看出，在由 EF 所离散化的数据集上，DTRDE 的性能也要好于 ID3 和 C4.5。这也证明了所提算法的有效性。

结束语 本文将粗糙集与决策树有机地结合在一起，基于相对决策熵提出一种新的分离属性选择标准，并给出相应的决策树算法。在建树之前，该算法利用数据约简技术进行预剪枝，以有效降低决策树的规模。在从 KDD Cup99 中所随机抽取的数据子集上的实验表明，该算法的入侵检测效果要好于其他算法。但由于实验数据集的规模有限，该算法的真实检测性能还有待进一步的考证。

在后续工作中，计划将该算法扩展到邻域粗糙集模型中[26]，设计一种不需要离散化就能够直接处理连续型和离散型属性的算法。

参 考 文 献

[1] Quinlan R. Induction of decision trees[J]. Machine Learning, 1986,1(1):81-106

[2] Quinlan R. C4.5:Programs for Machine Learning[M]. Morgan Kaufmann,1993

[3] Huang L J,Huang M H,Guo B. A new method for constructing decision tree based on rough set theory[C]// 2007 IEEE Int. Conf. on Granular Computing. 2007:241-244

[4] Li X P,Dong M. An algorithm for constructing decision tree based on variable precision rough set model[C]// The 4th Int. Conf. on Natural Computation. 2008:280-283

[5] Wei J M,Huang D,Wang S Q. Rough set based decision tree[C]// Proc. of the 4th World Congress on Intelligent Control and Automation. 2002:426-431

[6] Bai J S,Fan B,Xue J Y. Knowledge representation and acquisition approach based on decision tree[C]// Int. Conf. on Natural Language Processing and Knowledge Engineering. 2003:533-538

[7] Pawlak Z. Rough Sets[J]. Int. J. Comput. Informat. Sci. ,1982, 11(5):341-356

[8] Pawlak Z. Rough Sets:Theoretical Aspects of Reasoning about Data[M]. Kluwer Academic Publishing,1991

[9] 苗夺谦，胡桂荣. 知识约简的一种启发式算法[J]. 计算机研究与发展，1999,36(6):681-684

[10] 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

[11] 王国胤，于洪，等. 基于条件信息熵的决策表约简[J]. 计算机学报，2002,25(7):759-766

[12] 刘少辉，盛秋戬，等. Rough 集高效算法的研究[J]. 计算机学报，2003,26(5):525-529

[13] 徐章艳，刘作鹏，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

[14] Anderson J P. Computer Security Threat Monitoring and Surveillance[M]. James P. Anderson Co. ,USA,1980

[15] Li X Y,Ye N. Decision tree classifiers for computer intrusion detection[J]. Journal of Parallel and Distributed Computing Practices,2001,4(2):179-190

[16] Kruegel C,Toth T. Using decision trees to improve signature-based intrusion detection[C]// Symp. on Recent Advances in Intrusion Detection. 2003:173-191

[17] Amor N B,Benferhat S,Elouedi Z. Naive Bayes vs decision trees in intrusion detection systems [C]// ACM Symp. on Applied Computing. 2004:420-424

[18] Shannon C E. The mathematical theory of communication[J]. Bell System Technical Journal,1948,27(3/4):373-423

[19] Liang J Y,Shi Z Z. The information entropy,rough entropy and knowledge granulation in rough set theory[J]. Int. Journal of Uncertainty,Fuzziness and Knowledge Based Systems,2004,12(1):37-46

[20] Düntsch I,Gediga G. Uncertainty measures of rough set prediction[J]. Artificial Intelligence,1998,106:109-137

[21] Witten I H,Frank E. Data Mining:Practical Machine Learning Tools and Techniques with Java Implementations[M]. Morgan Kaufmann,2000

[22] KDD Cup 99 Dataset[OL]. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

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

[24] 陈仕涛，陈国龙，等. 基于粒子群优化和邻域约简的入侵检测日志数据特征选择[J]. 计算机研究与发展，2010,47(7):1261-1267

[25] Øhrn A. Rosetta Technical Reference Manual. 2000

[26] Hu Q H,Yu D R,Liu J F,et al. Neighborhood rough set based heterogeneous feature subset selection [J]. Information Scien-ces,2008,178(18):3577-3594

(上接第 219 页)

参 考 文 献

[1] Tenenbaum J B,De Silva V, Langford J C. A global geometric framework for nonlinear dimensionality reduction[J]. Science, 2000,290:2319-2323

[2] Roweis S T,Saul L K. Nonlinear Dimensionality Reduction by Locally Linear Embedding[J]. Science,2000,290:2323-2326

[3] Belkin M,Niyogi P. Laplacian eigenmaps for dimensionality reduction and data rep resentation[J]. Neural Computing,2003,15(6):1373-1396

[4] Donoho D L,Grimes C. Hessian Eigenmaps:Locally Linear Embedding Techniques for High Dimensional Data[J]. Proceedings of the National Academy of Sciences of the United States of America,2003,100(10):5591-5596

[5] Zhang Z,Zha H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment[J]. Journal of Shanghai University(English Edition),2004,8(4):406-424

[6] de Ridder D, Kouropteva O, Okun O, et al. Supervised locally linear embedding[J]. Artificial Neural Networks and Neural Information Processing,2003,2714:333-341

[7] Kouropteva O, Okun O, Pietikainen M. Supervised Locally Linear Embedding Algorithm for Pattern Recognition[J]. Pattern Recognition and Image Analysis,2003,2652:386-394

[8] Martin H C L,Anil K J. Incremental Nonlinear Dimensionality Reduction by Manifold Learning[J]. IEEE Trans. Pattern Analysis and Machine Intelligence,2006,28(3):377-391

[9] Kouropteva O,Okun O. Incremental locally linear embedding algorithm[J]. Pattern Recognition,2005,38(10):1764-1767

[10] Liu X, Yin J, Feng Z, et al. Incremental manifold learning via tangent space alignment[J]. ANNPR,2006,4087:107-121

[11] 曾宪华，罗四维. 动态增量流形学习算法[J]. 计算机研究与发展，2007,44(9):1462-1468

[12] Jia P,et al. Incremental Laplacian eigenmaps by preserving adjacent information between data points[J]. Pattern Recognition, 2009,30:1457-1463

[13] Abdel-Mannan O,Hamza A B,Youssef A. Incremental Hessian Locally Linear Embedding Algorithm[C]// Proc. of the 9th International Symposium on Signal Processing and Its Applications. 2007

[14] 李厚森，成礼智. 增量 Hessian LLE 算法研究[J]. 计算机工程，2010,37(6):159-161