一种基于KL散度和类分离策略的特征选择算法

李晓艳[1] 张子刚[1] 张逸石[1] 张 谧[2]
(华中科技大学管理学院 武汉 430074)[1] (重庆邮电大学经济管理学院 重庆 400065)[2]

摘 要 特征选择是模式识别和机器学习中的重要环节之一,所选特征子集的质量直接影响着分类学习算法的效率及准确率。现有特征选择算法均在整个类标签集的视角下进行特征评价,并未分别考察每一类别与特征间的关系。提出了一种基于KL散度和类分离策略的特征选择算法,它采用类分离策略分别对类标签中每一类别与特征间的关系予以考察,并采用一种基于KL散度的有效距离度量类别与特征间的相关性以及特征之间的冗余性。实验结果表明,所提算法具有较高的运行效率;在所选特征质量上,所提算法显著优于经典的CFS、FCBF以及ReliefF特征选择算法。

关键词 特征选择,KL散度,类分离策略,有效距离
中图法分类号 TP181 **文献标识码** A

KL-divergence Based Feature Selection Algorithm with the Separate-class Strategy

LI Xiao-yan[1] ZHANG Zi-gang[1] ZHANG Yi-shi[1] ZHANG Mi[2]
(School of Management, Huazhong University of Science and Technology, Wuhan 430074, China)[1]
(School of Economics & Management, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)[2]

Abstract Feature selection is one of the core issues in designing pattern recognition systems and has attracted considerable attention in the literature. Most of the feature selection methods in the literature only handle relevance and redundancy analysis from the point of view of the whole class, which neglect the relation of features and the separate class labels. To this end, a novel KL-divergence based feature selection algorithm was proposed to explicitly handle the relevance and redundancy analysis for each class label with a separate-class strategy. A KL-divergence based metric of effective distance was also introduced in the algorithm to conduct the relevance and redundancy analysis. Experimental results show that the proposed algorithm is efficient and outperforms the three representative algorithms CFS, FCBF and ReliefF with respect to the quality of the selected feature subset.

Keywords Feature selection, KL-divergence, Separate-class strategy, Effective distance

1 引言

特征选择是模式识别和机器学习领域中重要的研究方向之一。从原始特征空间中选择较少的重要特征,同时排除一些不重要的干扰特征,不仅可以减小学习算法的计算复杂度,提高算法的学习能力,而且有助于寻找更精简、更易理解的学习算法模型。对特征选择的研究始于20世纪60年代。当时绝大多数学者都是从统计学的视角出发来对待与处理特征选择问题。在此之后,随着信息技术的不断发展和网络技术的不断成熟,特征选择逐渐引起了众多领域学者的广泛关注。

特征选择算法既可作为分类学习前的一个预处理操作,也可被视为学习算法中的一个组成部分。Dash和Liu等人对机器学习领域的特征选择问题做了较为深入的研究,并将特征选择问题看成一个搜索问题[1]。特征选择方法可以分为特征排序法和特征子集搜索法。其中典型的特征排序法有基于距离度量的Relief算法及其改进算法ReliefF[2],以及基于互信息的DMIFS[3]算法等。特征排序法最大的优点在于其具有高效的执行效率。其缺陷在于需要人为指定所选特征的数量,且排位靠前的m个特征所组成的特征子集并不一定是含有m个特征的特征子集中的最优特征子集[4]。造成这种现象的原因是特征之间存在冗余。特征排序法由于仅对单个特征与类标签的相关性进行评价,并未考虑特征之间的关系,因此不能甄别冗余特征。对于特征维数较高的数据集而言,由于其特征空间中存在大量的冗余特征,因此使用特征排序算法往往不能取得很好的效果[5]。为了有效地甄别冗余特征,进而搜索出最优特征子集,许多特征子集搜索算法被相继提出。其中又可根据特征相关性/冗余性分析策略的不同而将特征子集搜索算法分为两类:相关性/冗余性综合分析法和相关性/冗余性两阶段分析法。前者将特征的相关性和冗余性综合成一个指标进行考察,典型算法有基于关联性度量的CFS算法[6]、基于"最小冗余最大相关"策略的mRMR算法[7]及其改进算法[8]、基于"最小相关冗余性"和聚类技术的mRR

到稿日期:2012-02-02 返修日期:2012-05-19 本文受国家自然科学基金项目(60973085)资助。
李晓艳(1975-),女,博士生,主要研究方向为模式识别与电子商务;张子刚(1947-),男,教授,博士生导师,主要研究方向为信息管理与知识管理;张逸石(1986-),男,博士生,主要研究方向为模式识别与电子商务,E-mail:easezh@126.com(通信作者);张 谧(1987-),男,硕士生,主要研究方向为电子商务。

算法[9]等；后者则分别对特征的相关性和冗余性进行考察，典型算法有基于对称不确定性度量的 FCBF 算法[10]和基于马尔科夫毯(Markov blanket)的 IAMB 算法[11]及其改进算法[12]等。一般而言，相关性/冗余性两阶段分析法在效率和所选特征质量上要优于相关性/冗余性综合分析法[13]。

然而，不论是上述的特征排序算法还是特征子集搜索算法，它们均在整个类标签集的视角下进行特征的选择，并未分别对每一类别与特征的关系予以考察，因此往往导致一些总体相关性较强但与部分类标签相关性较弱的特征被选中，最终造成分类准确率的下降。针对这一问题提出一种类分离策略，以实现在单个类标签视角下的特征选择。此外，就以往的特征选择算法所使用的度量工具无法度量单个类标签和特征之间关系的困难，提出了一种基于 KL 散度的有效距离度量方法，以配合类分离策略进行特征的相关性/冗余性分析。实验结果表明，所提特征选择算法具有较好的运行效率，且在所选特征子集质量上要显著优于现有有代表性的特征选择算法。

2 KL 散度与类分离策略

2.1 KL 散度

KL 散度(Kullback-Leibler divergence)，又称 KL 距离，是一种用于描述两个分布之间差异性的工具。KL 散度具有如下形式：

$$D_{KL}(p\|q)=\sum_{x\in\Omega}p(x)\log\frac{p(x)}{q(x)} \tag{1}$$

式中，p 和 q 是概率空间 Ω 下的两个概率分布。$D_{KL}(p\|q)$ 称为分布 p 关于分布 q 的 KL 散度，$D_{KL}(q\|p)$ 则为分布 q 关于分布 p 的 KL 散度。由式(1)易知 $D_{KL}(p\|q)\neq D_{KL}(q\|p)$，即 KL 散度不满足对称性。对于形如 $D_{KL}(p\|q)$ 的 KL 散度而言，称分布 p 为真实分布，分布 q 为分布 p 的近似分布。$D_{KL}(p\|q)$ 的取值越大，表明真实分布 p 与近似分布 q 越相异，反之则越相近。根据 Jensen 不等式，易得 $D_{KL}(p\|q)\geqslant 0$，等号成立当且仅当 $p=q$。

2.2 类分离策略

特征选择的最终目的是减小表达数据的特征空间的维度以提高分类器性能和效率，同时分类性能的优劣最终体现在每一类别中被正确分类的实例数量百分比上，因此在这个意义上，分别对每一个类别进行相关性和冗余性分析可以很自然地作为一种特征选择策略。可以称这种相关性和冗余性分析策略为类分离策略。图 1 给出了一个该策略的示意图。

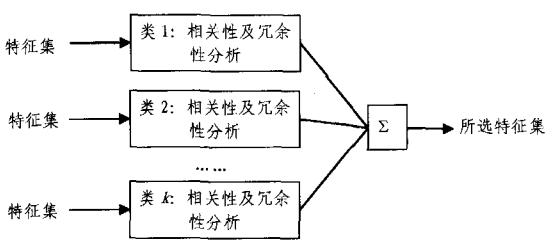

图 1 类分离策略示意图

图 1 简单而直观地给出了类分离策略的主要意图：对每个类标签而言，分别对特征空间中的特征进行相关性与冗余性分析，然后综合每一类中所选特征作为最终的特征子集并输出。然而现有主流的相关性和冗余性度量工具(例如 χ^2 检验、信息熵以及互信息等)仅能够在类标签集合 C 的视角下对特征 F 进行相关性和冗余性的度量，即能够在统计或信息论意义上测量 C 和 F 之间的“有效距离”；而在单个类标签 $c_i\in C$ 的视角下，这些度量工具则无法度量特征 F 与当前类标签 c_i 的“有效距离”，因此需要设计一种新的度量方法，以实现在单个类标签的视角下度量特征的相关性和冗余性。在第 3 节中，首先给出在单个类标签视角下基于 KL 散度的“有效距离”的正式定义，进而基于该度量和类分离策略，提出一种有效的特征选择算法。

3 基于 KL 散度和类分离策略的特征选择算法 DSCS

3.1 基于 KL 散度的有效距离及特征搜索策略

给定数据集 $\boldsymbol{U}=\boldsymbol{D}(\boldsymbol{F},C)$，$\boldsymbol{G}\subset\boldsymbol{F}$，根据文献[10]，若对于特征 $F\in\boldsymbol{F}$，$\exists c\in C, P(F|\boldsymbol{G},C=c)\neq P(F|\boldsymbol{G})$(以下将 $P(F|\boldsymbol{G},C=c)$ 简记为 $P_c(F|\boldsymbol{G})$)，则称 F 为给定 $\boldsymbol{G}$ 时类标签 C 的相关特征，否则称 F 为给定 $\boldsymbol{G}$ 时类标签 C 的不相关特征或 $\boldsymbol{G}$ 的冗余特征。因此，可以采取一种相关/冗余分析策略同时进行相关特征和冗余特征的判别，这种判别方式被称为特征的有效性判别。采用一种基于 KL 散度的有效距离来进行特征的有效性判别可以确保判别的可靠性。

定义 1(有效距离) 给定一个特征空间为 $\boldsymbol{F}=\{F_1,\cdots,F_n\}$、类标签集为 $C=\{c_1,\cdots,c_k\}$ 的数据集 $\boldsymbol{D}$，特征 $F_i(1\leqslant i\leqslant n)$ 到类标签 $c_j(1\leqslant j\leqslant k)$ 的有效距离为

$$ValDis_{\boldsymbol{G}}(F_i,c_j)=\sum_{f_{\boldsymbol{G}}\in\boldsymbol{G}}P_{c_j}(f_{\boldsymbol{G}})D_{KL}(P_{c_j}(F_i|f_{\boldsymbol{G}})\|P(F_i|f_{\boldsymbol{G}})) \tag{2}$$

式中，$\boldsymbol{G}\subset\boldsymbol{F}$ 为 $\boldsymbol{F}$ 中的一个特征子集。特别地，若 $\boldsymbol{G}=\varnothing$，则 F_i 到 c_j 的有效距离为

$$ValDis(F_i,c_j)=D_{KL}(P_{c_j}(F_i)\|P(F_i)) \tag{3}$$

式(2)中，$D_{KL}(P_{c_j}(F_i|f_{\boldsymbol{G}})\|P(F_i|f_{\boldsymbol{G}}))$ 即分布 $P_{c_j}(F_i|\boldsymbol{G})$ 和 $P(F_i|\boldsymbol{G})$ 之间的 KL 散度，用以描述两分布之间的“差距”。由 KL 散度的定义知，$P_{c_j}(F_i|\boldsymbol{G})$ 和 $P(F_i|\boldsymbol{G})$ 之间的“差距”越大，则表明特征 F_i 对类标签 c_j 的区分性越强，即 F_i 为 c_j 的相关特征，反之则为不相关特征或冗余特征。$P_{c_j}(f_{\boldsymbol{G}})$ 为在 $C=c_j$ 下特征取值 $f_{\boldsymbol{G}}$ 的分布。该分布可以看作在数据集中含有 $f_{\boldsymbol{G}}$ 的样本上估计 $P_{c_j}(F_i|\boldsymbol{G})$ 的可靠性系数，即若在类标签为 c_j 的样本集上含有 $f_{\boldsymbol{G}}$ 的样本越多，则对 $P_{c_j}(F_i|\boldsymbol{G})$ 和以及 $D_{KL}(P_{c_j}(F_i|f_{\boldsymbol{G}})\|P(F_i|f_{\boldsymbol{G}}))$ 的估计越“可靠”，其对应的可靠性系数 $P_{c_j}(f_{\boldsymbol{G}})$ 越大。该系数能够在一定程度上降低因含有 $f_{\boldsymbol{G}}$ 的样本有限而存在的估计偏差所带来的影响。为进一步避免该偏差所带来的影响，采用“最大有效特征”的搜索方式进行特征的搜索。

定义 2(最大有效特征) 若特征 F^* 满足

$$\max_{F\in\boldsymbol{F}-\boldsymbol{G}}ValDis_{\boldsymbol{G}}(F,c_j)\geqslant\delta_j \tag{4}$$

则称 F^* 为 c_j 的一个最大有效特征。其中，δ_j 为类标签 c_j 所对应的有效性判别阈值。

由于存在样本有限而导致的估计偏差，因此一个估计的较短有效距离往往可能是实际上的“无效距离”(即实际有效距离为 0)。定义 2 中引入的有效性判别阈值 δ_j 可在一定程度上避免这种错误的发生。引入有效性阈值虽然解决了上述误差，却带来了新的“估计为无效而实际有效”的误差。定义 2 在引入有效性判别阈值的基础上，仅针对当前具有最大有效距离的特征进行有效性判别，以在最大程度上避免由单纯引入有效性判别阈值而造成的“估计为无效而实际有效”的误差。所提算法将根据定义 2 对每一类标签的最大有效特征进

行搜索。

3.2 DSCS算法及实现

综合上述分析，基于KL散度和类分离策略的DSCS算法(KL-Divergence and Separate-Classes Strategy based feature selection algorithm)伪代码描述如下。

算法1 DSCS特征选择算法

Input: A training dataset $\mathbf{U}=\mathbf{D}(\mathbf{F},C)$

Output: Selected feature subset $\mathbf{G}$

1. Initialize: $\mathbf{G}=\emptyset$, TempSet$=\emptyset$
2. Repeat
3. For i=1 to $|C|$, do
4. If ($F^* \in \mathbf{F}-\mathbf{G}$ satisfying $\max_{F\in F-G} ValDis_G(F,c_i)$ and $ValDis_G(F^*,c_i)\geqslant\delta_i$
5. TempSet=TempSet$\cup F^*$
6. End If
7. End For
8. $\mathbf{G}=\mathbf{G}\cup$TempSet
9. TempSet$=\emptyset$
10. Until $\mathbf{G}$ has not changed

DSCS算法在当前已选特征子集$\boldsymbol{G}$的基础上依据类分离策略对类标签进行遍历，搜索出当前每一类标签中的最大有效特征，并在遍历完后将其统一选入特征子集$\boldsymbol{G}$中，如此迭代直到$\boldsymbol{G}$不再变化为止。因此，最坏情况下算法需要遍历类标签$(|\boldsymbol{F}|+(|\boldsymbol{F}|-1)+(|\boldsymbol{F}|-2)+\cdots+(|\boldsymbol{F}|-|\boldsymbol{G}|))\sim O(|\boldsymbol{F}|^2)$次。受文献[13]的启发，给出一种有效距离的计算方法，基于该方法，DSCS算法的时间复杂度仅为$O(|C||\boldsymbol{F}|^2(N+r))$，其中$r$为特征取值数的上限。

定理1 给定一个含有N个样本的数据集$\boldsymbol{U}=\boldsymbol{D}(\boldsymbol{F},C)$，其中$C=\{c_1,\cdots,c_k\}$，则有

$$\sum_{j=1}^{k}\frac{N_j}{N}\cdot ValDis_{\boldsymbol{G}}(F,c_j)=\hat{I}(F;C|\boldsymbol{G}) \tag{5}$$

式中，N_j为整个数据集中类标签为c_j的样本数，$\hat{I}(F;C|\boldsymbol{G})$为采用文献[13]中条件互信息估计方法所估计的条件互信息。

证明：令Φ为$\boldsymbol{G}$中特征取值组合空间，$\phi_i\in\Phi$为$\boldsymbol{G}$中第i种存在于$\boldsymbol{U}$中的取值组合，Φ中实际存在于$\boldsymbol{U}$中的取值组合个数为q，$\boldsymbol{U}_i\subset\boldsymbol{U}$为含有$\phi_i$的子数据集，$N_{U_i}$为$\boldsymbol{U}_i$中的样本数，$N_{U_i\wedge c_j}$为$\boldsymbol{U}_i$中类标签为$c_j$的样本数，$\pi_{U_i}(\cdot)$和$\pi_{U_i\wedge c_j}(\cdot)$分别表示特征在$\boldsymbol{U}_i$和$\boldsymbol{U}_i$中类标签为$c_j$的样本上某一取值的频数。根据文献[13]，易得

$$\hat{I}(F;C|\boldsymbol{G})=\sum_{j=1}^{k}\frac{N_j}{N}\sum_{i=1}^{q}\frac{N_{U_i\wedge c_j}}{N_j}\left(\sum_{f\in F}\frac{\pi_{U_i\wedge c_j}(f)}{N_{U_i\wedge c_j}}\log\frac{\frac{\pi_{U_i\wedge c_j}(f)}{N_{U_i\wedge c_j}}}{\frac{\pi_{U_i}(f)}{N_{U_i}}}\right)$$

$$=\sum_{j=1}^{k}\frac{N_j}{N}\underbrace{\sum_{i=1}^{q}\hat{P}_{c_j}(\phi_i)\left(\sum_{f\in F}\hat{P}_{c_j}(f|\phi_i)\log\frac{\hat{P}_{c_j}(f|\phi_i)}{\hat{P}(f|\phi_i)}\right)}_{\Theta_{ij}} \tag{6}$$

在$ValDis_G(F,c_j)$中，若以$\pi(\cdot)/N$来估计$P(\cdot)$，根据定义1，有$ValDis_G(F,c_j)=\sum_{i=1}^{q}\Theta_{ij}$。代入(7)式即得定理1。

结合定理1，采用文献[13]中复杂度为$O(|\boldsymbol{G}|(N+r))$的估计方法(其中r为特征的取值上限)来估计$ValDis_G(F,c_j)$，从而确保算法在当前已选特征子集为$\boldsymbol{G}$时遍历一次类标签的时间复杂度为$O(|C||\boldsymbol{G}|(N+r))$(分析过程请参考文献[13])。结合前面的分析，可知DSCS算法的最坏时间复杂度为$O(|C||\boldsymbol{F}|^2(N+r))$。

4 实验及结果分析

实验中选取著名的特征子集搜索算法CFS[6]和FCBF[10]以及经典的特征排序算法ReliefF[2]与DSCS算法进行比较。采用2个经典的分类器kNN[14]和C4.5[15]对特征选择后的数据集进行分类，并对分类结果进行评价。对于CFS算法，采用Best-First(BF)法作为其特征子集搜索策略；对于FCBF算法，根据文献[10]将其相关性判定阈值γ设为0；对于ReliefF算法，根据文献[2]将近邻数设为5，迭代参数设为30；对于本文的DSCS算法，将有效性判别阈值ξ_i均设为0.01。实验中DSCS算法采用Java实现，并在Weka平台下运行。CFS、FCBF和ReliefF算法以及kNN和C4.5分类器均可在Weka中直接调用。对于含有连续型特征的数据集，在实验前采用MDL方法进行离散化处理。离散化处理仅针对特征选择过程而使用。所有实验均在2.4GHz CPU、4G Memory PC机上的64位Linux操作系统下完成。

选用8个著名的基准数据集进行实验。其中Mushroom、Kr-vs-kp、Musk2、DNA、Madelon和Gisette为UCI库[1)]中的常用数据集，Ibn-sina和Sylva则为著名的挑战数据集(challenge dataset)[2)]。表1给出了这些数据集的详细信息。

表1 实验数据集描述

	Datasets	Features	Instances	Classes
1	Mushroom	22	8124	2
2	Kr-vs-kp	36	3196	2
3	Ibn-sina	92	20722	2
4	Musk2	166	6598	2
5	DNA	180	3186	3
6	Sylva	216	72626	2
7	Madelon	500	2000	2
8	Gisette	5000	6000	2

4.1 所选特征规模及运行时间

算法在所选数据集上的平均所选特征数以及算法的平均运行时间分别如图2和图3所示。

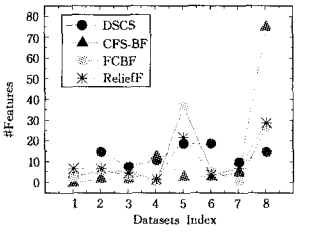

图2 算法所选特征数

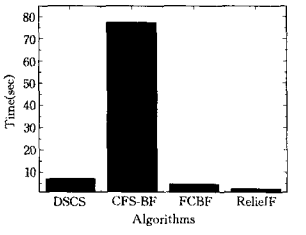

图3 算法平均运行时间

图2的横坐标为数据集索引(见表1)。从图2可以看出，CFS-BF算法在8个数据集上的所选特征数波动性最强，而DSCS算法仅在4到20个特征的范围内波动，较另外3种特征选择算法更加稳定。从图3可以看出，CFS-BF算法在8

1) http://archive.ics.uci.edu/ml/

2) http://www.causality.inf.ethz.ch/activelearning.php?page=datasets#cont

个数据集上的平均运行时间在70s～80s之间，远大于其它3种特征选择算法。较CFS-BF，DSCS、FCBF和ReliefF算法的平均执行时间均在10s以下，具有较高的运行效率。

4.2 分类性能评价

表2和表3出了kNN和C4.5分类器在所选数据集上采用10次10折交叉验证法所获得的平均分类准确率。表中"Unselect"一列给出了2种分类器在每一个数据集上直接分类的结果；"DSCS"、"CFS-BF"和"FCBF"列则分别给出了2种分类器在经过相应特征选择算法降维后的数据集上的分类结果。记号"+"/"−"表示当前分类结果显著优/显著差于原始数据集(Unselect列)上的分类结果。实验采用成对双样本t检验法进行显著性检验，并取显著性水平$\alpha=0.01$。Avg.一行给出了每种情况下分类结果的平均值。尾行的"W/T/L"(Wins/Ties/Losses)表示在经特征选择算法降维后的数据集上的分类结果显著优于/不显著/显著差于原始数据集上分类结果的次数。4种特征选择算法中的最优结果在表中用粗体标识。

表2 kNN上的分类结果(%)

	Unselect	DSCS	CFS-BF	FCBF	ReliefF
Mushroom	100	**100**	98.52−	98.57−	**100**
Kr-vs-kp	96.12	**98.15+**	90.43−	94.18−	93.39−
Ibn-sina	96.22	**96.36+**	92.55−	93.06−	95.59−
Musk2	94.68	**94.74+**	93.01−	91.42−	93.19−
DNA	74.55	**90.59+**	87.16+	83.13+	85.99+
Sylva	98.17	**99.78+**	98.76+	98.68+	98.52+
Madelon	55.10	**86.50+**	77.20+	56.10+	84.47+
Gisette	79.91	**91.61+**	87.59+	83.73+	83.43+
Avg.	86.84	**94.72**	90.65	87.36	91.82
W/T/L		7/1/0	4/0/4	4/0/4	4/1/3

表3 C4.5上的分类结果(%)

	Unselect	DSCS	CFS-BF	FCBF	ReliefF
Mushroom	100	**100**	98.52−	98.62−	**100**
Kr-vs-kp	99.44	**98.47−**	90.43−	94.03−	93.34−
Ibn-sina	97.32	**96.96−**	93.87−	94.22−	96.65−
Musk2	96.84	**95.76−**	95.72−	91.22−	93.36−
DNA	92.54	**94.60+**	87.16−	91.37−	89.49−
Sylva	99.39	**99.71+**	99.11−	99.17−	99.17−
Madelon	69.08	78.26+	73.31+	63.80−	**78.36+**
Gisette	87.20	**92.95+**	87.15	83.67−	87.40+
Avg.	92.73	**94.59**	90.66	89.51	92.22
W/T/L		4/1/3	1/1/6	0/0/8	2/1/5

由表2可知，DSCS算法使kNN获得94.72%的平均分类准确率，在4种特征选择算法中最优。不仅如此，DSCS算法在所有数据集上均使kNN的分类准确率达到最优。此外，除了在Mushroom数据集上与原始数据集上的分类准确率相等外，在其他7个数据集上DSCS算法对应的分类准确率均显著优于原始数据集上的分类准确率。而表现次之的ReliefF算法也仅仅获得了4次显著优、1次差别不显著和3次显著差的结果。表3中DSCS算法仍然使C4.5获得了94.59%的最优平均分类准确率，仅在Madelon上获得了劣于ReliefF的结果。此外，DSCS获得了4次显著优和1次差别不显著，是4个特征选择算法中的最好情况。上述结果充分显示了本文所提DSCS算法的性能显著优于CFS-BF、FCBF和ReliefF算法。

结束语 本文提出了一种基于KL散度和类分离策略的DSCS特征选择算法。该算法基于类分离策略，采用一种基于KL散度的有效距离对每一类别与特征的相关性及特征间的冗余性予以考察，进而搜索出一个具有较好类区分能力的特征子集。实验结果表明，DSCS算法具有较高的执行效率，同时在不同数据集上DSCS算法所选特征规模具有很好的稳定性。在kNN以及C4.5上的分类实验结果表明，DSCS算法明显优于经典的CFS-BF、FCBF和ReliefF特征选择方法，证实了DSCS算法的有效性及优越性。

参 考 文 献

[1] Dash M, Liu H. Feature selection for classification [J]. Intelligent Data Analysis, 1997, 1(1-4): 131-156

[2] Robnik-Sikonja M, Kononenko I. Theoretical and empirical analysis of ReliefF and RReliefF [J]. Machine Learning, 2003, 53: 23-69

[3] Liu H, Sun J, Liu L, et al. Feature selection with dynamic mutual information [J]. Pattern Recognition, 2009, 42(7): 1330-1339

[4] Jain A K, Duin R P W, Mao J. Statistical pattern recognition: A review [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(1): 4-37

[5] 任永功，林楠. DPFS：一种基于动态规划的文本特征选择算法 [J]. 计算机科学，2009，36(6)：188-191

[6] Hall M A. Correlation-based feature selection for discrete and numeric class machine learning [C]// Proceedings of the 7th International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 2000: 359-366

[7] Ding C, Peng H. Minimum Redundancy Feature Selection from Microarray Gene Expression Data [C] // Proceedings of the IEEE Computer Society Conference on Bioinformatics. Washington DC: IEEE Computer Society Press, 2003: 523-528

[8] Peng H, Long F, Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(8): 1226-1238

[9] Sotoca J M, Pla F. Supervised feature selection by clustering usingconditional mutual information-based distances [J]. Pattern Recognition, 2010, 43(6): 2068-2081

[10] Yu L, Liu H. Efficient feature selection via analysis of relevance and redundancy [J]. Journal of Machine Learning Research, 2004, 5: 1205-1224

[11] Tsamardinos I, Aliferis C, Statnikov A. Local causal and Markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation [J]. Journal of Machine Learning Research, 2010, 11: 171-234

[12] Zhang Y, Zhang Z, Liu K, et al. An improved IAMB algorithm for Markov blanket discovery [J]. Journal of Computers, 2010, 5(11): 1755-1761

[13] 张逸石，陈传波. 基于最小联合互信息亏损的最优特征选择算法 [J]. 计算机科学，2011，38(12)：200-205

[14] Hall P, Park B U, Samworth R J. Choice of neighbor order in nearest-neighbor classification [J]. Annals of Statistics, 2008, 36(5): 2135-2152

[15] Kotsiantis S B. Supervised machine learning: A review of classification techniques [J]. Informatica, 2007, 31: 249-268