基于聚类的特征选择方法

蒋盛益,郑 琪,张倩生
(广东外语外贸大学信息学院,广东广州 510006)

摘 要: 本文提出了一种度量特征区分度的定义,进而提出一种基于聚类的特征选择方法 CBFS. 该方法时间复杂度与数据集的大小和特征个数成近似线性关系,适合于大规模数据集中的特征选择;该方法对数据类型没有限制,适用于混合类型数据. 在 UCI 数据集上的实验结果表明,与文献中的方法相比,本文方法具有较好的性能,说明提出的特征选择方法是有效和实用的.

关键词: 聚类; 特征区分度; 特征选择
中图分类号: TP391 **文献标识码**: A **文章编号**: 0372-2112 (2008) 12A-157-04

Clustering-Based Feature Selection

JIANG Sheng-yi, ZHENG Qi, ZHANG Qian-sheng
(School of Informatics, Guangdong University of Foreign Studies, Guangzhou, Guangdong 510006, China)

Abstract: The authors come up with a definition of measuring differentiations between features, and then put forward a method of clustering-based feature selection (Below referred to as CBFS). The time complexity of the method is nearly linear with both the size of dataset and the number of features. Besides, the method is applicable to the selection of features in large dataset. It can particularly handle data with both Nominal and Continuous Features. The results of the experiment on UCI datasets show that the method is effective and practicable.

Key words: clustering; differentiation of feature; feature selection

1 引言

特征选择是指从一组原始特征集合中选择具有代表性的特征子集,使其保留原有数据的大部分信息,即所选择的特征子集可以像原来的全部特征一样用来正确区分数据集中的每个数据对象. 特征选择作为数据预处理的一类方法,是数据挖掘、机器学习和模式识别中的一个重要而棘手的问题. 特征选择的研究开始于上个世纪的六十年代[1],已取得许多研究成果,有基于信息熵、粗糙集、神经网络、支持向量机的各类特征选择算法[2~8]. 已经证明最优特征子集的搜索是一个 NP 问题,除了穷举搜索,不能保证得到最优解,因此人们致力于用启发式搜索算法寻找近似最优解.

特征选择算法主要有两种框架,即过滤式(Filter)特征选择算法和封装式(Wrapper)特征选择算法. 过滤式特征选择算法是将特征选择作为一个预处理过程,利用数据的内在特性对选取的特征子集进行评价,独立于学习算法,通常是选择与目标函数相关度大的特征或者特征子集,该类算法通常运行效率较高而适用于大规模数据集. 而封装式特征选择算法则将后续学习算法的结果作为特征子集评价准则的一部分,根据算法生成规则的分类精度选择特征子集,该类算法具有使得生成规则分类精度高的优点,但特征选择效率低.

本文研究 Filter 式特征选择方法,以聚类为基本工具,利用不同簇间在各个特征上的差异定义的区分度作为特征可分性判据,然后按可分性判据的大小进行特征重要性排序,最后根据重要性的变化规律选取特征子集. 实验结果表明,相比于经典的 IFN[2]、Relief[3]、ABB[4]等特征选择算法,本文的算法在性能上有明显的提高.

2 基于聚类的特征选择方法

2.1 方法描述

本文采用文献[9]给出的一趟聚类算法作为对数据进行划分的基本方法. 假设数据集 D 有 t 个不同类别的数据,每个数据有 m 个特征,其中有 m_C 个分类特征和 m_N 个数值特征, $m = m_C + m_N$. 不妨设分类特征位于数值

收稿日期:2008-05-09;修回日期:2008-10-01
基金项目:国家自然科学基金(No. 60673191);广东省高等学校自然科学研究重点项目(No. 06Z012);广东外语外贸大学科研创新团队项目(No. GW2006-TA-005)

特征之前,用 $D_i(1 \le i \le m)$ 表示第 i 个特征取值的集合.

定义 1 给定簇 C, $a \in D_i$, a 在 C 中关于 D_i 的频度定义为 C 在 D_i 上的投影中包含 a 的次数,并用 $Freq_{C|D_i}(a)$ 表示, $Freq_{C|D_i}(a) = |\{object | object \in C, object.D_i = a\}|$.

定义 2 给定簇 C, C 的摘要信息 CSI(Cluster Summary Information)定义为: $CSI = \{kind, n, Summary\}$,其中 $kind$ 为簇的类别, $n = |C|$ 为 C 的大小, $Summary$ 由分类型特征中不同取值的频度信息和数值型特征的质心两部分构成,即: $Summary = \{\langle Stat_i, Cen\rangle | Stat_i = \{(a, Freq_{C|D_i}(a)) | a \in D_i\}, 1 \le i \le m_C, Cen = (c_{m_C+1}, c_{m_C+2}, \cdots, c_{m_C+m_N})\}$.

注:为简单计,本文对于簇的类别采用投票机制确定,即将簇中对象最多的类别作为簇的类别.

比较一个特征在两个带有类别的簇上的区分程度需要考虑两个因素,其一是两个簇在这个特征上取值的差异,其二是两个簇的大小,为此提出定义 3.

定义 3 簇 C_1 与 C_2 在特征 D_i 上的区分程度定义为 $d_i(kind_{C_1}, kind_{C_2}) = \frac{|C_1| + |C_2|}{|C_1| + |C_2|} dif(C_i^{(1)}, C_i^{(2)})$,这里 $kind_{C_1}, kind_{C_2}$ 分别是簇 C_1 与 C_2 的类别, $dif(C_i^{(1)}, C_i^{(2)})$ 体现了两个簇在特征 D_i 上的差异,对于数值特征 D_i 有 $dif(C_i^{(1)}, C_i^{(2)}) = |c_i^{(1)} - c_i^{(2)}|$,对于分类特征 D_i 有 $dif(C_i^{(1)}, C_i^{(2)}) = 1 - \frac{1}{|C_1| + |C_2|} \sum_{p_i \in (C_1|D_i) \cap (C_2|D_i)} Freq_{C_1|D_i}(p_i) \cdot Freq_{C_2|D_i}(p_i)$.

定义 4 特征 D_i 的区分度定义为 $f_i = \frac{Max_i - Min_i}{Mean_i}$,这里 Max_i, Min_i, $Mean_i$ 分别是定义 3 中得到的 $d_i(kind_{C_1}, kind_{C_2})$ 当 C_1、C_2 遍历一趟聚类得到的不同簇时的最大值、最小值和平均值.

定义 3、定义 4 体现了不同特征在不同类之间的区分能力,一个好的特征应该使同类样本之间的特征值相同或相近,而不同类样本之间的特征值不同或者差别很大,或者说一个好的特征应该具有较大的区分度.因为我们的目标是找到能使不同类尽可能被分开的特征,对于两个特征 X 和 Y,如果特征 X 在簇间的区分度更大,则说明特征 X 能更好地区分不同的簇,该特征重要程度较高,应优先选择特征 X.由于特征之间可能存在关联关系,当去除部分特征后,余下特征之间的重要性可能发生变化,因此可能需要多步选择,直至结果稳定.基于前面的分析,本文提出一种基于聚类的特征选择算法,聚类时不使用标志位,但利用标志位对聚类结果进行标识,采用投票机制确定每个簇的类别,利用带类别的簇间差异性来度量特征重要性进而选择特征子集.算法描述如下:

Step 1 重复以下操作 s 次

Step 1.1 聚类:随机选择聚类阈值,使用一趟聚类算法对数据集 D 进行聚类,并采用投票机制确定每个簇的类别,得到带类别信息的聚类结果 $C = \{C_1, C_2, \cdots, C_k\}$;

Step 1.2 按照定义 3 计算每个特征 D_i 上任何一个簇到其它簇之间的区分度(同时标明两个类别信息);

Step 1.3 分类汇总 Step1.2 得到的每个特征 D_i 在不同类别之间的区分度的平均值;

Step 2 计算每个特征 D_i 在不同类别之间的区分度的总的平均值 $Mean_i$,进一步计算每个特征上不同类别之间平均区分度的最大值 Max_i、最小值 Min_i;

Step 3 计算每个特征 D_i 在不同类别上的区分度 $f_i = \frac{Max_i - Min_i}{Mean_i}$;

Step 4 对特征 $D_1 \sim D_m$ 按 f_i 降序排列得到 f_i^* $(i = 1, \cdots, m)$;

Step 5 在 f_i^* $(i = 1, \cdots, m)$ 的折线图中找到急剧变化的点或拐点 i_0, $f_i^* \sim f_{i0}^*$ 即为选择的特征子集.

2.2 时间复杂度分析

为简化分析,假定在 s 次循环中,每次最终产生的 CSI 个数最大为 k,每个分类特征 i 有 n_i 个不同的取值.

Step 1.1 在最坏情况下聚类算法时间复杂度为 $O(n \cdot k(\sum_{i=1}^{m_C} n_i + m_N))$[9],期望的时间复杂度为 $O(n \cdot k \cdot m)$;

Step 1.2 需要计算任意两个簇在每个特征上的差异,其时间复杂度为 $O(k^2(\sum_{i=1}^{m_C} n_i + m_N))$;

Step 1.3 在 Step 1.2 计算时即分类统计不同类别之间区分度之和,只需求 $\frac{t(t+1)}{2}$ 个数之和,时间复杂度为 $O(m \cdot t^2)$;

由于 $k < N$, Step 1 的时间复杂度为 $O(N \cdot k \cdot (\sum_{i=1}^{m_C} n_i + m_N)) + O(m \cdot t^2)$,期望的时间复杂度为 $O(N \cdot k \cdot m)$;

Step 2 对 s 次循环得到的值求均值、最大值、最小值,时间复杂度为 $O(m \cdot m \cdot t^2)$;

Step 3 计算每个特征的区分度,时间复杂度为 $O(m)$;

Step 4 对 m 个特征上的区分度数据进行排序,使用快速排序法,时间复杂度为 $O(m \log m)$;

Step 5 采用交互式方法或差分法确定急剧变化之点,时间复杂度为 $O(m)$.

由此可见,由于数据类别数 t 通常较小,整个算法的时间开销主要体现在 Step 1,复杂度为 $O(s \cdot N \cdot k(\sum_{i=1}^{m_c} n_i + m_N))$,其时间复杂度与数据集大小成线性关系,与特征个数以及最终的簇个数成近似线性关系,这使得本文方法具有好的扩展性,可用于大规模数据集中的特征选择. 相对于信息熵、粗糙集等传统的特征选择方法,本文方法计算简单,它对离散特征数据和连续特征数据均可处理,大大简化了特征选择过程,提高了特征选择效率,可分性效果好.

3　实验结果分析

本文主要从特征子集的大小、特征选择前后分类准确率的变化两个大的方面来进行比较,以评价特征选择算法.

3.1　实验数据与方法

我们从 UCI 数据集[10]中选取了 11 个数据集进行了测试,并与文献中的经典方法进行了性能对比. 表 1 列出了这些数据集的特征,选取的这些数据集有较广泛的代表性:这些数据集的特征维数从几个到数十个不等、数据类型有单一的离散型或连续型特征数据、也有同时包含离散型和连续型特征的混合类型数据、有些数据集包含有较多的缺失数据,可以较好地验证特征选择方法在实际数据集上的性能. 为验证我们的特征选择算法的效果,对每个数据集在全部特征集合和选取的特征集合上使用 Weka 软件[11,12]中的 C4.5 分类算法进行对比测试,以比较对应性能的变化.

表 1　实验数据集汇总

Dataset	Nominal attributes	Continuous attributes	Total attributes	Instance size	Number of Class
Breast	0	9	9	699	2
Chess	36	0	36	3196	2
Credit	9	6	14	690	2
pima-diabetes	0	8	8	768	2
Glass	0	9	9	214	6
Heart	7	6	13	270	2
Iris	0	4	4	150	3
KDDcup99	7	34	41	494020	23
Liver	0	6	6	345	2
Lung-cancer	56	0	56	32	3
Wine	0	13	13	178	3

3.2　实验结果比较

在每个数据集上,首先运行我们的特征选择算法,记录下特征选择后的特征子集,然后在各特征选择算法处理后的训练数据集上训练得到 C4.5 决策树分类器,接着得到 C4.5 分类器在测试集上的分类错误率并记录下来. 表 2 给出了在这些选取的数据集上 IFN、Relief、ABB、CBFS 等特征选择算法得到的特征集的大小.

表 3 列出了 C4.5 在 IFN、Relief、ABB、CBFS 等特征选择算法处理后的分类错误率(Error rate),为便于比较,表的第一列列出了分类器在原始数据集上的分类错误率. 对数据集采用随机选取的 2/3 的数据作为训练集,余下的作为测试集的策略划分数据集,同时在每个数据集上对数据随机打乱顺序并测试 10 次,以 10 次的平均指标作为评估的结果.

表 2　各特征选择算法得到的特征子集大小

Dataset	Total attributes	IFN	Relief	ABB	CBFS	Selected Features by CBFS
Breast	9	6	7	6	6	{1,2,3,4,6,7}
Chess	36	27	33	31	22	{3,7,8,10,13,14,15,16,18,19,21,22,23,25,27,28,29,30,31,32,33,35}
Credit	14	10	12	9	5	{3,8,9,11,15}
pima_diabetes	8	4	7	6	4	{3,4,6,8}
Glass	9	6	8	8	6	{1,3,6,7,8,9}
Heart	13	10	11	9	9	{2,3,4,8,9,10,11,12,13}
Iris	4	3	2	3	2	{3,4}
KDDcup99	41	/	/	/	12	{1,3,23,24,25,26,29,33,34,36,38,39}
Liver	6	1	4	2	3	{3,4,6}
Lung-cancer	56	54	53	52	11	{1,9,17,19,33,40,43,46,47,48,50}
Wine	13	10	10	11	5	{6,7,11,12,13}
Average	16.8	13.1	14.7	13.7	7.3	
Average dimensionality reduction		22%	12.5%	18.5%	56.5%	

表 3　C4.5 在全部特征和选择的特征子集上的分类错误率(%)对比

Dataset	Before FS	After IFN	After Relief	After ABB	After CBFS
Breast	7.3	6.0	6.4	6.4	5.67
Chess	0.8	3.0	10.2	6.0	4.56
Credit	13.8	15.1	13.4	15.1	15.19
pima_diabetes	28.6	27.7	22.3	22.7	32.96
Glass	36.6	39.4	56.3	49.3	30.32
Heart	20.0	22.0	26.0	34.0	20.88
Iris	5.46	6.0	4.0	10.0	5.27
Liver	32.4	39.2	45.1	45.1	38.2
Lung-cancer	15.62	44.4	88.9	66.7	15.31
Wine	3.3	8.3	5.0	21.7	8.25
KDDcup99_20000	0.2794	/	/	/	1
Average	16.39	21.11	27.76	27.7	17.66

表 2～表 3 表明特性选择算法 CBFS 较 Relief、IFN、ABB 算法可以选择更有效的特征子集,展现了 CBFS 的有效性. 经特征选择算法 CBFS 选择特征子集后,特征维数平均下降了 50%多,而其他特征选择算法平均维数下降在 20%左右;经特征选择算法 CBFS 选择子集后,尽管分类错误率比特征选择前有所上升,但较其他

方法低 3.5～10 %.

4 结论

本文提出了一种基于聚类的过滤型特征选择算法,该算法适用于混合类型数据,具有近似线性时间复杂度,适用于大规模数据集.实验结果表明本文的特征选择算法独立于特定的分类算法,具有较好的性能(效率高、特征子集较小、对分类质量影响小),与文献中的经典方法比较性能具有一定的优势.

参考文献:

[1] Lewis P M. The characteristic selection problem in recognition system[J]. IRE Transaction on Information Theory, 1962, 8(2):171 - 178.

[2] Mark Last, Abraham Kandel, Oded Maimon. Information-theoretic algorithm for feature selection[J]. Pattern Recognition Letters, 2001, 22(6):799 - 811.

[3] Kononenko I. Estimating attributes: analysis and extensions of RELIEF[A]. Proc of ECML[C]. Catania, Italy, Springer-Verlag New York, 1994. 171 - 182.

[4] Liu H, Motoda H. Feature Selection for Knowledge Discovery and Data Mining[M]. Klumwer, Boston. 1998.

[5] Hu Q H, Xie Z X, Yu D R. Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation[J]. Pattern Recognition, 2007, 40(12):3509 - 3521.

[6] Swiniarski R W, Skowron A. Rough set methods in feature selection and recognition[J]. Pattern Recognition Letters, 2003, 24(6):833 - 849.

[7] Neumann J, Schnorr C, Steidl G. Combined SVM-based feature selection and classification[J]. Machine Learning, 2005, 61(1):129 - 150.

[8] Huang J J, Cai Y Z, Xu X M. A hybrid genetic algorithm for feature selection wrapper based on mutual information[J]. Pattern Recognition Letters, 2007, 28(13):1825 - 1844.

[9] Jiang S Y, Song X Y, et al. A clustering-based method for unsupervised intrusion detections[J]. Pattern Recognition Letters, 2006, 27(7):802 - 810.

[10] Merz C J, Merphy P. UCI repository of machine learning databases [OB/OL]. URL: http://www.ics.uci.edu/～mlearn/MLRRepository.html, 1996.

[11] Witten I H, Frank E. Data Mining: Practical Machine Learning Tools and Techniques (2nd Edition)[M]. Morgan Kaufmann, San Francisco, 2005.

[12] http://www.cs.waikato.ac.nz/ml/weha/

作者简介:

蒋盛益 男,1963 年生,湖南省隆回县人,博士,教授,主要研究领域为数据挖掘、网络安全.

E-mail:jiangshengyi@163.com