一种基于聚类集成的无监督特征选择方法

凌霄汉, 吉根林

(南京师范大学 数学与计算机科学学院, 江苏 南京 210097)

[摘要] 提出了一种无监督的特征选择方法, 其基本思想是利用聚类来指导特征选择, 对于无类别标签的数据样本集, 先进行聚类获得数据类标签, 再利用 ReliefF算法进行特征选择. 采用聚类集成方法解决一些聚类结果的不稳定问题, 最终特征选择结果通过多次特征选择综合得到. 实验结果表明, 该算法具有良好的特征选择性能, 在去除无关或冗余特征后可进一步提高聚类质量.

[关键词] 特征选择, 无监督学习, 集成学习

[中图分类号] TP311 [文献标识码] A [文章编号] 1672-1292(2007)03-0060-04

A Clustering Ensemble Based Unsupervised Feature Selection Approach

Ling Xiaohan, Ji Genlin

(School of Mathematics and Computer Science, Nanjing Normal University, Nanjing 210097, China)

Abstract: An unsupervised feature selection approach is proposed, which utilizes clustering to obtain the class label of data object and uses ensemble technique to resolve the instability of clustering. As clustering results generated by some algorithms are usually different from each other, feature selection performs multiply and all results are combined to produce final selected features. In addition, ReliefF is ameliorated, which is a supervised feature selection algorithm and is employed as an essential part in the approach. Experimental results show that the approach can remove redundant features and improve the quality of clustering.

Key words: feature selection, unsupervised learning, ensemble learning

0 引言

特征选择是数据挖掘过程中一个重要的预处理步骤, 它从一组 N 个特征中按一定的标准选择出由 n $(n<N)$个特征组成的特征子集, 这个子集具有比特征全集更好或与特征全集一样的分类功能. 一个典型的特征选择过程包括候选特征子集生成和评估, 按照评估方法特征选择分为两大类: 过滤方法 (filter approach)和封装方法 (wrapper approach). 过滤方法只使用数据集来评价每个特征的相关性, 它并不直接优化任何特定的分类器, 也就是说特征子集的选择和后续的分类算法无关. ReliefF[1]是这类算法的代表. 封装方法与过滤方法正好相反, 它直接优化某一特定的分类器, 使用后续分类算法来评价候选特征子集的质量. 一般说来, 过滤方法的效率比较高, 结果与采用的分类算法没有关系, 但效果稍差; 封装方法占用的运算时间较多, 结果依赖于采用的分类算法, 也因为这样其效果较好. 在分类问题中, 特征选择的研究比较成熟, 已提出多种算法[1-3]. 相比较而言, 特征选择在聚类问题中则较少受到关注. 一个重要的原因是不依赖类标签就无法评定特征子集的相关度, 当聚类的个数未知时, 问题就更加难以处理了. 无监督的特征选择因缺乏类标签信息, 其特征选择的结果在降低特征空间维度与提高聚类性能方面均有欠缺.

本文提出一种新的无监督特征选择方法, 其基本思想是利用聚类来指导特征选择, 对于无类别标签的数据样本集, 先进行聚类以获得数据类标签, 然后利用 ReliefF算法进行特征选择. 为克服一些聚类结果的不稳定性, 采用集成学习的方法, 先进行多次特征选择, 然后对各次的特征选择结果进行综合. 在每一次特

收稿日期: 2006-12-21.
基金项目: 江苏省自然科学基金 (BK2005135)资助项目.
作者简介: 凌霄汉 (1981-), 硕士研究生, 主要从事集成学习与数据挖掘方面的学习与研究. E-mail: nolen0@163.com
通讯联系人: 吉根林 (1964-), 教授, 博士生导师, 主要从事数据库与数据挖掘、机器学习等方面的教学与研究. E-mail: jigenlin@njnu.edu.cn

征选择时, 采用了 boosting[4]中的方法对重要样本加强学习. 实验结果表明, 本文算法具有良好的特征选择性能, 在去除无关或冗余特征后可进一步提高聚类质量.

1 ReliefF算法

ReliefF算法是有监督特征选择的代表性算法, 具有优良的性能, 其基本思想是随机抽取样本对该样本的若干近邻进行学习, 计算每一个特征的权重, 然后该过程迭代若干次不断对特征的权值进行更新, 最后选择权值较大的若干个特征.

设 $X = \{x_1, x_2, \cdots, x_N\}$ 是样本全集, 样本 $x_i = [x_{i1}, x_{i2}, \cdots, x_M]$, 其中 x_{ij} $(j = 1, 2, \cdots, M)$ 表示第 i 个样本的第 j 个特征值, $\boldsymbol{w} = [w_1, w_2, \cdots, w_M]$ 表示 M 个特征的权值向量. ReliefF算法首先令 $w_i = 0 (1 \leqslant i \leqslant M)$, 初始化时各特征的权值相等即重要性相同. 算法执行 m 次迭代, 每次迭代随机抽取一个样本 x_i, 找出 r 个与 x_i 同类的最近邻样本 h_j $(j = 1, 2, \cdots, r)$, 然后在每个与 x_i 不同类的样本集中找出 r 个与 x_i 最近邻的样本 k_{lj} $(j = 1, 2, \cdots, r,\ l \neq \mathrm{class}(x_i),\ \mathrm{class}(x_i)$ 表示 x_i 的类别 $)$. 最后对权值向量进行更新, 如式 (1) 所示:

$$w_i = w_i - \sum_{j=1}^{r} \mathrm{diff}(F_i, x_i, h_j)/(m \times r) + \sum_{l \neq \mathrm{class}(x_i)} \left(\frac{p(l)}{1 - p(\mathrm{class}(x_i))} \sum_{j=1}^{r} \mathrm{diff}(F_i, x_i, k_{lj}) \right) /(m \times r). \tag{1}$$

式中, $\mathrm{diff}(F_i, x_1, x_2) = \dfrac{|\mathrm{value}(F_i, x_1) - \mathrm{value}(F_i, x_2)|}{\max(F_i) - \min(F_i)}$, F_i 为第 i 个特征, $\mathrm{value}(F_i, x_i)$ 为样本 x_i 的第 i 个特征的值, $\max(F_i)$ 为所有样本中第 i 个特征的最大值, $\min(F_i)$ 为所有样本中第 i 个特征的最小值; $P(l)$ 为第 l 类出现的概率, 可以用第 l 类的样本数除以数据集中样本的总数. 算法执行 m 轮后, 就可以得到各特征的权重. ReliefF算法中 $\mathrm{diff}(\cdot)$ 函数定义了两个样本关于某一特征的差异, 当两个样本属于同一类时, 这种差异表明该特征对分类不利, 所以式 (1) 第二项前为 "-"; 当两个样本不属于同一类时, 差异对分类有利, 所以第三项前为 "+".

2 算法分析与描述

ReliefF算法只能用于有监督的特征选择, 因为算法执行过程中对数据的类别标签有很强的依赖性. 为了将该算法用于无监督的特征选择, 本文提出: 对无类别标签的数据集先进行聚类以获得数据类标签, 然后利用 ReliefF算法进行特征选择. 这种方法在实际使用时必须考虑一个重要的因素, 那就是不少聚类算法的结果是不稳定的. 以 k-means算法为例, 在设置了相同的聚类簇数后, 多次运行算法其聚类结果一般是不同的. 于是, 聚类后再进行特征选择就有可能使得这次特征选择的结果较好而另一次的结果较差, 即聚类结果的不稳定性会影响特征选择的结果. 集成学习方法在提高分类器泛化能力方面显示出了极为显著的效果, 该技术已被成功引入聚类问题中[5]. 为此, 本文将这种聚类集成的思想应用到无监督特征选择方法中, 具体做法是: 对数据集多次进行无监督特征选择, 然后综合各次的特征选择结果来获得最终的特征.

ReliefF算法在计算特征的权重时, 对数据集采用了随机取样的方法. 当抽取的样本数足够多时, 从概率的角度来看, 这些样本来自整个数据集分布空间的各个类群中, 因此保证了算法对数据集的充分学习. 但是, 该算法若不加修改直接使用在基于聚类集成的无监督特征选择方法中, 将会失去集成的意义. 本文在仔细研究像 k-means这些聚类不稳定的算法后, 发现多次聚类结果的不同主要表现在各类别边界的样本上面.

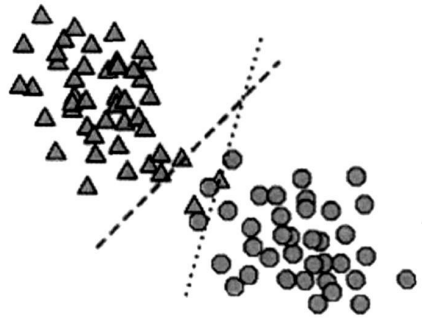

图 1 聚类边界样本的变化

Fig.1 Change of cluster boundary

如图 1所示, 两条斜线分别是两次聚类所确定的聚类边界分隔线, 从中可以看到这两次聚类是有变化的, 但显然变化不大, 主要是类边界的样本容易导致错分. 这些类边界上发生

变化的样本一般只占总的数据集的极小一部分,ReliefF随机抽取样本进行学习不能保证每次都抽到类边界样本.这样,ReliefF在各次聚类结果上的抽取一般只集中在那些相对稳定类属变化不大的样本上面,进而导致各次特征选择的结果趋于相同,集成的效果得不到有效发挥.在 Boosting[4]中采用了一种对重要样本加强学习的方法,即每轮迭代训练新学习器的时候,上一轮错分的样本将以较大的概率出现在本次学习器的训练集中.本文算法采用了一种与之类似的样本学习策略对 ReliefF加以改进以满足集成对多个学习器间差异性的要求.在 ReliefF抽取样本的时候,先让其进行正常的随机抽取,然后再让它对每一个与上轮聚类的类别不一致的样本进行学习.可以看出,这种改进就是保证 ReliefF算法一定抽取到那些在聚类边界上发生类别从属变化的样本,这样在一定程度上使得各次特征选择的结果反映出各次聚类的变化与不同点.

多次聚类之间的类别对应也是使用集成学习方法需要解决的一个重要问题.本文采用了一种启发式方法,即当两个聚类的簇之间包含的相同样本数最多时就认为这两个簇有类别对应关系,剩下的簇依次类推.

综上所述,基于聚类集成的无监督特征选择算法描述如下:

输入: k为聚类簇数, T为集成的规模, m 为随机抽取样本个数, r为近邻的个数, X 为样本集.

输出: 各个特征的权值 $\{w_1, w_2, \cdots, w_M\}$.

算法:

(1) $t=0$; $\{C_1^t, C_2^t, \cdots, C_k^t\}$ = k-means(X); //初始化聚类

(2) for($t=1$; $t<=T$; $t++$)

{ $\{C_1^t, C_2^t, \cdots, C_k^t\}$ = k-means(X); // C_i^t为第 t次聚类的第 i个簇

Γ = find($\{C_1^{t-1}, C_2^{t-1}, \cdots, C_k^{t-1}\}$, $\{C_1^t, C_2^t, \cdots, C_k^t\}$); //求解类别有变化的样本集合

unique(Γ); //去除 Γ 中重复样本

for($i=1$; $i<=M$; $i++$) $w_i^t=0$ // w_i^t为第 t次特征选择的第 i个特征的权值

for($j=1$; $j<=m$; $j++$) //随机选择 m 个样本

{ x_i = random select(X);

$\Gamma = \Gamma + \{x_i\}$;

}

for each x_i in Γ do

{ $\{h_1, h_2, \cdots, h_r\}$ = findhits(X); //找出 r个同类最近邻

$\{k_1, k_2, \cdots, k_{r\times(k-1)}\}$ = findmisses(X); //在每个与 x_i 不同类中分别找出 r个最近邻

update($\{w_1^t, w_2^t, \cdots, w_M^t\}$); //根据式(1)更新每个特征的权值

}

}

(3) for($i=1$; $i<=M$; $i++$)

$$w_i = \frac{1}{T}\sum_{t=1}^{T} w_i^t;$$

上述算法使用 k-means聚类,如果使用其他聚类算法进行特征选择,则可以做相应修改.

3 实验结果与分析

为测试算法性能,本文使用了 UCI机器学习数据库[6]中的 3个数据集,这 3个数据集在特征数、实际类别数、样本数方面逐渐增加,如表 1所示.因 k-means算法需设定聚类的簇数,为简单起见,实验中直接设置其聚类数为数据集的实际类别数.为进行比较,同时测试了文献[7]的一种无监督特征选择方法,该方法把特征选择转换为估计问题,而这种估计是由 EM (Expectation Maximization)算法完成的.无监督特征选择的目的是去除冗余特征以减小聚类的时间开销并提高聚类性能.在 Modha和 Spangler[8]的工作中利用了数据的分类信息来评价聚类结果的好坏,即当数据有分类信息时,可认为该分类信息在一定程度上表达了

表 1　实验所用数据集

Table 1　Data set used in the experiment

数据集	特征个数	类别个数	样本个数
Iris	4	3	150
Wine	13	3	178
Image Segmentation	19	7	2310

数据的一些内部分布特性. 如果该分类信息没有被聚类过程所利用, 则可以用它来评价聚类性能, 其度量标准 Micro-precision定义如下:

$$\text{Micro-precision} = \frac{1}{n}\sum_{i=1}^{k}\alpha_i. \tag{2}$$

式中, n 为数据集样本总数; k 为聚类的簇数; α_i 为聚类的簇与已知数据集类别对应后, 簇当中被正确归为相应类别的样本个数. Micro-precision的值越大, 表示在该数据集上聚类效果越好.

对每个数据集进行 5次实验, 每次实验除了用两种特征选择算法再聚类外, 还使用全部特征直接聚类. 各数据集聚类的 Micro-precision对比如表 2所示.

表 2 实验性能 Micro-precision比较

Table 2 Comparison of Micro-precision

数据集	Iris			Wine			Image Segmentation		
实验次数	全部特征聚类	EM 算法	本文算法	全部特征聚类	EM 算法	本文算法	全部特征聚类	EM 算法	本文算法
1	0.912	0.944	0.939	0.667	0.768	0.867	0.672	0.833	0.858
2	0.899	0.932	0.946	0.68	0.822	0.78	0.601	0.85	0.891
3	0.902	0.951	0.949	0.643	0.83	0.884	0.64	0.793	0.82
4	0.932	0.937	0.958	0.677	0.799	0.858	0.611	0.868	0.901
5	0.922	0.933	0.95	0.691	0.852	0.875	0.627	0.851	0.885
平均值	0.913	0.939	0.948	0.672	0.814	0.852	0.63	0.839	0.871
标准差	0.014	0.008	0.007	0.018	0.032	0.042	0.028	0.028	0.033

观察表 2的 Micro-precision对比, 在 3个数据集上本文的无监督特征选择算法与文献 [7]的 EM 算法均好于没有进行特征选择的聚类结果. 直观上, 对每个数据样本知道的信息越多, 聚类的效果越好. 但实践中并非如此, 有些特征可能是冗余信息对聚类毫无帮助, 有些特征甚至是噪音反而会降低聚类的效果. 另外, 在 Iris数据集上本文算法与 EM 算法的 Micro-precision值相近, 在 Wine和 Image Segmentation数据集上则比 EM 算法略好. 表 3显示了第 5次实验本文算法与 EM 算法所选择的特征, 这些特征按重要性排序, 以后一半排序中相邻特征权值差距最大的作为选择的断点. 总的来说, 本文算法在进行特征选择之后提高了聚类性能, 实验对比结果显示其是有效可行的.

表 3 特征选择结果

Table 3 Results of feature selection

数据集	EM 算法	本文算法
Iris	4, 3	4, 3
Wine	6, 12, 7, 2, 1, 10, 4, 11	6, 12, 9, 11, 7, 5, 13, 1, 4
Image Segmentation	16, 9, 4, 15, 13, 11, 14, 5, 8, 17, 19, 10	12, 9, 10, 16, 14, 5, 7, 6, 11, 17, 19

4 结束语

本文对于无类标签的数据样本集, 提出了一种新的特征选择方法, 利用聚类来指导特征选择, 先进行聚类获得数据类标签, 再用 ReliefF算法进行特征选择. 采用聚类集成的方法解决聚类结果的不稳定性问题. 实验结果表明, 本文算法具有良好的特征选择性能, 在去除无关或冗余特征后可进一步提高聚类质量.

[参考文献] (References)

[1] Kononenko I. Estimating attributes: analysis and extensions of relief[C] // Proceedings of the 7th European Conference on Machine Learning. Berlin: Springer, 1994: 171-182.

[2] Liu H, Setiono R. Feature selection and classification: a probabilistic wrapper approach[C] // Proceedings of the 9th International Conference on Industrial and Engineering Applications of AI and ES. Fukuoka: Springer, 1996: 419-424.

[3] Dash M, Liu H. Feature selection for classification[J]. Intelligent Data Analysis, 1997, 1(3): 131-156.

[4] Schapire R E. The strength of weak learnability[J]. Machine Learning, 1990, 5(2): 197-227.

[5] Fred A L N, Jain A K. Data clustering using evidence accumulation[C] // Proceedings of the 16th International Conference on Pattern Recognition. Quebec: IEEE Press, 2002: 276-280.

[6] Newman D J, Hettich S, Blake C L, et al. UCI repository of machine learning databases [EB/OL]. [2006-12-21] http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[7] Martin H C Law, Mário A T Figueiredo, Anil K Jain. Simultaneous feature selection and clustering using mixture models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(9): 1154-1166.

[8] Modha D S, Spangler W S. Feature weighting in k-means clustering[J]. Machine Learning, 2003, 52(3): 217-237.

[责任编辑: 严海琳]