文章编号:1007-5321(2019)03-0127-06 **DOI**:10.13190/j.jbupt.2018-151

一种快速的特征选择框架和方法

仇利克[1], 刘 竞[2], 孙中卫[3], 赵扬帆[4]
(1. 山东外贸职业学院 信息管理系, 青岛 266100; 2. 青岛农业大学 理学与信息科学学院, 青岛 266100;
3. 青岛理工大学 信息与控制工程学院, 青岛 266100; 4. 山东青岛烟草有限公司 综合计划处, 青岛 266100)

摘要: 针对特征选择过程中准确率和计算效率不平衡问题,提出了一种快速特征选择框架(FFFS). 基于该框架,使用最小冗余最大相关方法(MRMR)选择候选特征,借助序列前向选择方法(SFS)验证性能,并通过限定迭代次数提高计算性能. 与 MRMR、SFS 和混合序列浮动前向选择算法(FDHSFFS)的对比实验结果表明,提出的快速特征选择算法 MRMR-SFS 能在预测准确率和计算效率之间取得较好的平衡.

关 键 词: 特征选择; filter; wrapper; hybrid; 性能预测; 相关系数
中图分类号: TP181 **文献标志码**: A

A Fast Feature Selection Framework and Method

QIU Li-ke[1], LIU Jing[2], SUN Zhong-wei[3], ZHAO Yang-fan[4]
(1. Information Management Department, Shandong Foreign Trade Vocational College, Qingdao 266100, China;
2. Science and Information College, Qingdao Agricultural University, Qingdao 266100, China;
3. School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266100, China;
4. Comprehensive Planning Office, Shandong Qingdao Tobacco Copany Limited, Qingdao 266100, China)

Abstract: Aiming at the imbalance between accuracy and computational efficiency in feature selection, a fast feature selection framework (FFFS) is proposed. Based on this framework, a fast feature selection algorithm, MRMR-SFS, is proposed. The minimum redundancy maximum relevance (MRMR) method is used to select the candidate features, and sequential forward selection (SFS) method is used to verify the performance of the candidate features as well. It improves the calculation efficiency by limiting the number of iterations. Comparison experiments with the MRMR, SFS and a filter-dominating hybrid sequential floating forward selection algorithms demonstrate that MRMR-SFS can balance the accuracy and computational efficiency well.

Key words: feature selection; filter; wrapper; hybrid; performance prediction; correlation coefficient

随着数据量的增加,数据中充斥的噪声和冗余也日益增加. 由于特征选择具有良好的去冗余和噪声能力,已成为数据挖掘中非常重要的预处理步骤. 通过去除无关和冗余特征,一个好的特征选择算法不仅能显著降低特征维数,而且能提高预测性能,增强对数据的理解[1-2].

特征选择算法通常可以分为 filter 方法[3]、wrapper 方法[4]和 hybrid 方法[5]三大类. filter 方法与学习算法无关,运行效率高,但难以发现特征间潜在的相关性,故选出的特征子集往往不理想. wrapper 方法将学习算法作为内嵌函数,以其准确率作为选择特征的标准,通常能发现特征间潜在的相关性,选出的特征子集往往具有良好的性能,但由于搜索空间大,且需要学习算法的介入,计算量太大,尤其

收稿日期: 2018-07-04
基金项目: 国家重点研发计划项目(2016YFC1401907); 国家自然科学基金项目(61827810)
作者简介: 仇利克(1979—), 女, 讲师, E-mail:qllike@163.com.

是随着特征维数和样本数的增加,效率太低,甚至不可接受. hybrid 方法虽然结合了 filter 方法的高效率和 wrapper 方法的高准确率特点,但也会受限于 filter 方法和 wrapper 方法的缺陷,难以真正解决计算效率和准确率之间的平衡问题.

基于以上原因,笔者提出了一种快速的特征选择框架(FFFS,a fast feature selection framework),并基于该框架提出了一种特征选择算法 MRMR-SFS(minimum redundancy maximum relevance-sequential forward selection). 该算法使用 Pearson 相关系数[6]计算特征相关性,使用 MRMR 方法选择候选特征,借助 SFS 方法验证候选特征性能,并限制了算法的迭代次数. 与当前主流算法的对比实验验证了 MRMR-SFS 算法的性能.

1 FFFS 特征选择框架

1.1 现有特征选择框架

目前存在 2 种主要评价相关特征的框架:对单个特征的评价和特征子集评价. 单个特征的评价框架如图 1 所示.

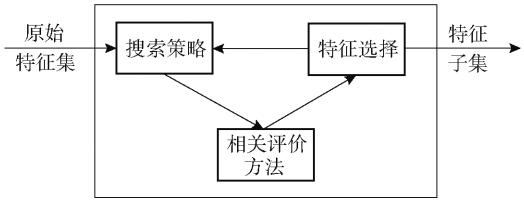

图 1 单个特征的评价框架

单个特征评价常用于 filter 方法. 通常先使用相关评价方法评价单个特征和预测值(类标)之间的相关性,并根据相关性排序,保留与预测值(类标)相关性强的特征组成特征子集,再选择合适的搜索策略搜索特征,结合相关评价方法,如互信息和相关系数,去除冗余特征,最后选择的特征子集中的特征与预测值(类标)之间的相关性强,且没有冗余. 单个特征评价的计算量通常为 $O(n)$ 或者 $O(n^2)$,其中 n 为特征维数,适合高维数据的处理,但该方法产生的结果往往与最优结果相差较远.

很多特征选择方法通过评价特征子集来处理特征相关,特征选择框架如图 2 所示. 通过搜索策略产生候选特征子集,使用评价函数评价产生候选特征子集的性能,并与之前使用此评价函数评价的性能最好的一组进行比较. 如果新的候选特征子集的性能更优,则替代之前的子集成为当前候选特征子集,产生、评价候选特征子集的过程不断循环,直到满足给定的停止条件为止. 该方法已被证明能产生较好的结果[1,7],但在候选特征子集产生和评价阶段的计算量太大,候选特征子集的数量最多可以达到 2^n 个,学习算法的评价次数最高也可达到 2^n 次. 若数据的维数很高,其计算量是巨大的.

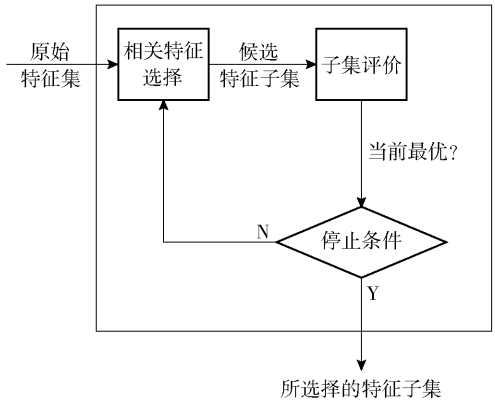

图 2 特征子集的评价框架

1.2 一种新的特征选择框架

基于 1.1 节所述 2 种特征选择框架,笔者提出了一种新的快速的 hybrid 特征选择框架 FFFS. 该框架结合了单个特征评价和特征子集评价 2 种评价标准,通过限制迭代次数,克服了单个特征评价的低准确率和特征子集评价的低效率. FFFS 特征选择框架如图 3 所示,其中,限制迭代次数为 k.

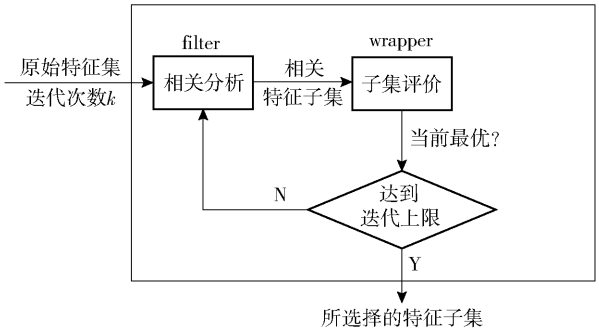

图 3 FFFS 特征选择框架

FFFS 特征选择框架分 2 部分:filter 过程和 wrapper 过程.

1) filter 过程:对单个特征进行评价,使用 filter 方法选出当前特征子集中的最优特征 F_{best},添加到当前特征子集 S,构成 S'.

2) wrapper 过程:对特征子集进行评价,使用学习算法评价 S'的性能,若其性能优于 S 的性能,则把特征子集 S'设为当前特征子集.

filter-wrapper 过程迭代 k 次,得到的特征子集即为近似最优特征子集.

FFFS 特征选择框架的优势为：① 使用 filter 方法选择相关特征，降低了计算量；② 由于 filter 方法和 wrapper 方法交叉进行，避免了先使用 filter 方法移除无关或冗余特征带来的准确率下降；③ 设置了迭代次数上限，在保证预测准确率的同时进一步降低了计算效率. 基于以上框架，笔者提出了一种 hybrid 特征选择算法 MRMR-SFS.

2　MRMR-SFS 算法

MRMR-SFS 算法既适应于离散数据，也适应于连续数据，笔者仅针对连续数据给出了相关分析，离散数据的相关分析类似，只是选择的相关评价标准有所差异. 实验采用的相关评价标准为 Pearson 相关系数.

2.1　特征相关分析

在分析特征相关之前，首先区分 2 种相关性：F-相关和 Y-相关. F-相关指 2 个特征 F_i 和 F_j 之间的相关性，用 $\rho_{i,j}$ 表示；Y-相关指特征 F_i 和预测值 y 之间的相关性，用 $\rho_{i,y}$ 表示.

MRMR-SFS 算法相关特征的选择采用最大相关最小冗余思想，即在原始特征空间中寻找 Y-相关最大而 F-相关最小的子集. M 表示原始特征空间，n 表示原始特征空间的维数，S 表示当前选择的特征子集，$M_S = M - S$ 为剩余特征组成的集合. 最大相关条件通过式(1)描述：

$$\max R_V,\ R_V = \frac{1}{|S|}\sum_{F_i \in S}\rho_{i,y} \tag{1}$$

最小冗余条件通过式(2)描述：

$$\min R_D,\ R_D = \frac{1}{|S|^2}\sum_{F_i,F_j \in S}\rho_{i,j} \tag{2}$$

寻找的最优特征子集要同时满足条件(1)和条件(2)，若要同时优化条件(1)和条件(2)，需要把它们联合起来. 但由于寻求最优特征子集的计算量太大[8]，故退而寻求近似最优特征子集. 寻求近似最优特征的标准记为 PCCQ(pearson correlation coefficient quotient criterion)，有

$$\max_{F_i \in M_S}\left(\frac{\rho_{i,y}}{\frac{1}{|S|}\sum_{F_j \in S}\rho_{i,j}}\right) \tag{3}$$

MRMR-SFS 算法首先使用 PCCQ 标准选出候选特征，然后使用学习算法验证增加此候选特征能否提高预测性能. 虽然通过这种方法选择的特征子集不一定是最优的，但是子集中的特征均能提高预测性能，保证了预测准确率.

2.2　算法描述

为进一步降低计算量，MRMR-SFS 算法限定迭代次数为 k. 算法从空集开始，迭代 k 次，产生最终的近似最优特征子集 S. 具体过程见算法 1. 其中：J 为性能评价函数，F_{best} 表示使用最大相关最小冗余思想选出的近似最优候选特征.

算法 1　MRMR-SFS 算法

输入：$M(F_1, F_2, \cdots, F_n, Y)$　// 数据集 M

　　k　　// 迭代次数

输出：S

1　begin
2　for $i = 1$ to n do begin
3　　对数据集 M 中的每个特征 F_i 计算 $\rho_{i,y}$；
4　end
5　$F_j = \text{Max}(M)$；//选择第一个特征
6　$S = \{F_j\}$；
7　$M_S = M - S$；
8　计算当前特征子集 S 的性能，记为 $J(S)$；
9　for $p = 1$ to k do begin
10　　for $i = 1$ to $|M_s|$
11　　$\text{PCCQ}(i) = \dfrac{\rho_{i,y}}{\frac{1}{|S|}\sum_{F_j \in S}\rho_{i,j}}$；
12　　end
13　　获取最大 PCCQ(i)值对应的特征 F_{best}；
14　　If ($J(S \cup \{F_{\text{best}}\}) > J(S)$)
15　　$S = S \cup \{F_{\text{best}}\}$；
16　　end
17　　$M_S = M_S - \{F_{\text{best}}\}$；
18　end
19　输出 S；
20　end

第 1 个特征的选取过程(第 2 ~ 6 行)如下：获取 M 中 $\rho_{i,y}$ 值最大的特征 F_j 作为集合 S 中的第 1 个特征，把 F_j 添加到集合 S，同时从集合 M 中移除 F_j，并计算特征子集 S 的性能，记为 $J(S)$.

第 $j(j \geqslant 2)$ 个特征的选取过程如下：计算 M_S 中每个特征 F_i 的 PCCQ(i)值，从计算结果中选择最大的 PCCQ(i)值，最大 PCCQ(i)值对应的特征记为 F_{best}(第 10 ~ 13 行). 用学习算法计算添加 F_{best} 到当前特征子集 S 后的性能，若 $J(S \cup \{F_{\text{best}}\}) > J(S)$，说明增加 F_{best} 到当前特征子集后性能提高，则更新 $S = S \cup \{F_{\text{best}}\}$. 以上过程迭代 k 次，最后输出近似最优特征子集 S.

MRMR-SFS 算法的计算量主要由 2 部分组成：基于最大相关最小冗余思想筛选候选特征产生的计算量；学习算法验证特征子集性能产生的计算量. 前者的计算量主要集中在 F-相关值的计算上，算法的计算量与 n^2 成正比，为 $O(kn^2)$. 因为每筛选出一个候选特征都要使用学习算法验证添加此候选特征到当前特征子集后的性能，共筛选 k 次，所以后者产生的计算量为 $O(kA)$，其中，A 为单次学习算法性能验证的计算量.

3 实验和分析

实验使用 3 个基准数据集和 1 个真实数据集来分析 MRMR-SFS 算法相比于其他特征选择算法的优点和不足.

3.1 数据集

实验使用数据集见表 1.

表 1 实验数据集信息描述

数据集	维数	样本数
EC	23	93
SP	32	1 044
CC	101	1 985
UJIL	528	19 937

基准数据集来自 UCI 机器学习库. 其中，Student Performance 简写为 SP，Communities and Crime 简写为 CC，UJIndoorLoc 简写为 UJIL. 特征集 SP 和 CC 中的文本数据都已处理为数值数据. 耗电量(energy consumption)数据集来自某公司一年的耗电量测试数据，简写为 EC.

3.2 评价方法和标准

实验使用的评价标准如下：

1) 所选择的特征维数；

2) 不同特征选择算法运行的时间；

3) 不同特征选择算法的预测误差.

实验使用相对误差来评价所选择的特征子集的性能，相对误差能很好地反映预测的可信程度，计算公式为

$$R = \frac{1}{m}\sum_{i=1}^{m}\frac{|y_i - \hat{y}_i|}{|y_i|} \tag{4}$$

其中：y_i 为实际值，$\hat{y}_i$ 为预测值，m 为测试集中的样本数.

MRMR-SFS 算法分别与 filter 方法、wrapper 方法和 hybrid 方法(对应 3 种不同的特征选择框架)做了比较. MRMR 方法是 filter 方法中最具代表性的算法之一，SFS 方法是 wrapper 方法中的经典算法，FDHSFFS 是 hybrid 方法中的典型算法. 因此，将本文算法分别与以上 3 种算法进行了对比实验. 经证明，MRMR 算法使用除法结合方式得到的特征子集性能更优[8]. 因此，filter 方法选择的是 MRMR 算法的除法结合方式，记为 MRMRQ. 所有算法都使用 Pearson 相关系数判定相关特征.

所有算法的运行环境为 Intel(R) Core(TM) i5-4200U，8 GB 内存，Matlab 2016.

3.3 实验结果与分析

实验中，MRMR-SFS 算法使用多项式和局部加权线性回归算法(LWLR，locally weighted linear regression)作为特征子集的评价函数，采用 10 折交叉验证训练参数，留一交叉验证(LOOCV，leave one out cross validation)计算测试误差，取 5 次运行结果的平均值.

1) 低维数据集实验结果和分析

针对低维数据集 EC 和 SP，由于运行时间短，计算效率可以忽略，最重要的是误差，所以本文算法主要与 MRMRQ、FDHSFFS、SFS 比较预测误差.

EC 数据集上的实验结果见表 2. 实验中，MRMR-SFS 的迭代上限 k 设置为 6，FDHSFFS 所选择的特征维数上限为 4. 第 1 列的算法模型中，MRMR-SFS-P 表示 MRMR-SFS 中的性能评价算法使用的是多项式算法，MRMR-SFS-L 表示 MRMR-SFS 中的性能评价算法使用的是 LWLR 算法，以此类推. 第 2 列显示了不同算法所选择的特征子集的平均维数和标准差，第 3 列显示了不同特征选择算法的运行时间和标准差，第 4 列显示了 LOOCV 预测误差和标准差.

表 2 EC 数据集上的实验结果

模型	特征维数	运行时间/s	平均误差/%
Full set-P	23.00 ± 0.00	0.03 ± 0.00	8.78 ± 0.00
MRMR-SFS-P	4.60 ± 0.89	0.09 ± 0.01	**7.89 ± 0.46**
FDHSFFS-P	4.00 ± 0.00	0.19 ± 0.11	8.25 ± 0.95
MRMRQ-P	**3.00 ± 0.00**	**0.07 ± 0.00**	11.59 ± 0.00
SFS-P	4.60 ± 1.67	0.85 ± 0.35	8.23 ± 1.27
Full set -L	23.00 ± 0.00	0.03 ± 0.01	10.55 ± 0.00
MRMR-SFS-L	**2.00 ± 0.00**	0.36 ± 0.37	10.30 ± 0.00
FDHSFFS-L	3.60 ± 0.89	0.22 ± 0.08	10.71 ± 0.63
MRMRQ-L	3.00 ± 0.00	**0.07 ± 0.00**	13.88 ± 0.00
SFS-L	4.00 ± 0.71	0.88 ± 0.16	**8.29 ± 0.86**

可以看出，所有的特征选择算法都有很好的降维效果，且选出的特征子集的维数近似. 由于 EC 数

据集的维数和样本数都较少，所以，运行时间差距不是很大. 针对 EC 数据集，无论使用多项式算法还是 LWLR 算法，本文算法的误差和 SFS 算法相近，均低于 FDHSFFS 算法，明显低于 MRMRQ 算法，相差分别为 3.7%（多项式算法）和 3.6%（LWLR）.

数据集 SP 上的实验结果见表 3. 实验中，MRMR-SFS 的迭代上限 k 设置为 6，FDHSFFS 所选择的特征维数上限为 4.

表 3　数据集 SP 上的实验结果

模型	特征维数	运行时间/s	平均误差/%
Full set -P	32.00 ± 0.00	1.14 ± 0.04	5.83 ± 0.00
MRMR-SFS-P	**1.80 ± 0.84**	1.32 ± 0.22	4.61 ± 0.21
FDHSFFS-P	**1.80 ± 0.84**	3.77 ± 1.27	4.57 ± 0.06
MRMRQ-P	2.00 ± 0.00	**0.10 ± 0.00**	4.97 ± 0.00
SFS-P	2.00 ± 0.00	6.82 ± 0.07	**4.54 ± 0.04**
Full set -L	32.00 ± 0.00	4.28 ± 0.04	5.64 ± 0.00
MRMR-SFS -L	1.60 ± 0.55	16.63 ± 0.09	4.60 ± 0.08
FDHSFFS-L	**1.40 ± 0.55**	92.25 ± 15.73	**4.56 ± 0.00**
MRMRQ-L	2.00 ± 0.00	**0.60 ± 0.00**	4.74 ± 0.00
SFS-L	2.20 ± 1.30	199.5 ± 118.56	4.70 ± 0.07

可以看出，不同算法选出的特征维数近似. 由于 MRMRQ 没有验证算法的介入，所以运行时间最短，其次是 MRMR-SFS、FDHSFFS 和 SFS 算法. MRMR-SFS 算法虽然限制了迭代次数，预测误差却与 FDHSFFS 和 SFS 近似，均略低于 MRMRQ 算法.

2）高维数据集实验结果和分析

针对高维数据集 CC 和 UJIL，在保证低预测误差的同时要保证高计算效率. 因此，本文算法分别与 MRMRQ、FDHSFFS、SFS 在预测误差和计算效率 2 方面进行对比实验.

数据集 CC 上的实验结果见表 4. 实验中，MRMR-SFS 的迭代上限 k 设置为 10，FDHSFFS 所选择的特征维数上限为 10. 可以看出，所有特征选择算法都取得了很好的降维效果，MRMRQ 的降维效果最好. 随着特征维数的增加，MRMR-SFS 的计算效率凸显，针对多项式算法（LWLR 算法），FDHSFFS 和 SFS 的运行时间分别为 MRMR-SFS 的 15 倍（4 倍）和 76 倍（71 倍）. 由于 MRMRQ 算法中没有学习算法的介入，所以其运行效率最高，也正因为此，其误差最大，比 MRMR-SFS 高 16.39%（多项式算法）和 16.41%（LWLR 算法），其他算法的误差近似.

表 4　数据集 CC 上的实验结果

模型	特征维数	运行时间/s	平均误差/%
Full set -P	101.00 ± 0.00	24.45 ± 0.85	73.42 ± 0.00
MRMR-SFS-P	7.20 ± 1.10	4.40 ± 0.16	**63.33 ± 0.58**
FDHSFFS-P	9.80 ± 0.45	65.43 ± 26.85	63.83 ± 1.25
MRMRQ-P	**2.00 ± 0.00**	**0.30 ± 0.00**	79.72 ± 0.00
SFS-P	8.20 ± 0.84	336.58 ± 52.73	64.91 ± 0.63
Full set -L	101.00 ± 0.00	39.12 ± 1.01	80.58 ± 0.00
MRMR-SFS-L	8.00 ± 0.71	154.75 ± 2.41	**64.15 ± 0.47**
FDHSFFS-L	10.00 ± 0.00	671.15 ± 255.19	64.44 ± 1.30
MRMRQ-L	**2.00 ± 0.00**	**4.50 ± 0.00**	80.56 ± 0.00
SFS-L	8.40 ± 2.19	11 008.99 ± 2 903.51	67.06 ± 0.80

数据集 UJIL 上的实验结果见表 5. 实验中，MRMR-SFS 的迭代上限 k 设置为 10（多项式算法）和 4（LWLR 算法），FDHSFFS 所选择的特征维数上限为 10（多项式算法）和 4（LWLR 算法）. 由于数据集 UJIL 维数和样本数都较高，SFS 算法使用多项式作为评价函数时的运行时间已经超过 288 000 s，故未进行 SFS 实验. 在使用 LWLR 作为评价函数的特征选择过程中，由于算法的运行时间过长，所有算法只取了 10 折交叉验证运行一次的误差进行性能比较.

表 5　数据集 UJIL 上的实验结果

模型	特征维数	运行时间/s	平均误差/%
Full set -P	528.00 ± 0.00	15 298.90 ± 132.91	20.89 ± 0.00
MRMR-SFS-P	2.00 ± 0.00	187.57 ± 1.30	**20.40 ± 0.00**
FDHSFFS-P	10.00 ± 0.00	4 201.62 ± 185.89	20.16 ± 0.54
MRMRQ-P	**1.00 ± 0.00**	**35.84 ± 0.00**	20.86 ± 0.00
Full set -L	528.00	18 916.28	98.83
MRMR-SFS-L	**1.00**	6 424.10	**20.86**
FDHSFFS-L	3.00	82 451.61	21.36
MRMRQ-L	**1.00**	**34.15**	**20.86**

由表 5 可以看出，MRMR-SFS 的降维效果与 MRMRQ 近似，其次是 FDHSFFS 算法. 本文算法的误差稍低于其他算法，相差不大，但计算效率远远高于 SFS 和 FDHSFFS 算法.

综上分析，不同特征选择算法在 4 个数据集上选择的特征子集的维数近似. 特征维数较低时（$n < 30$），运行时间差别不大（EC 和 SP 数据集），但维数越高，MRMR-SFS 相对于 FDHSFFS 和 SFS 算法的计算优势越明显. 例如，UJIL 数据集上，FDHSFFS 和 SFS 多项式算法的运行时间分别为 MRMR-SFS 的 23 倍和 1 548 倍（SFS 的运行时间按照 288 000 s 估算，实

际运行时间还要长). 尽管如此,MRMR-SFS 得到的误差却与 FDHSFFS、SFS 算法近似,甚至更低.

由于 MRMRQ 没有学习算法的介入,其计算效率始终是最优的,正因为如此,其在 EC 和 CC 数据集上的误差很大,与最低值的差值最高为 16.41%,预测效果很不稳定. 而 MRMR-SFS 算法的误差或者是最低的,或者和最低值相近,预测性能与 MRMRQ 相比更稳定.

3.4 迭代次数对性能的影响

MRMR-SFS 算法的迭代上限设置为 k,即 MRMR-SFS 所选择特征子集的维数不会超过 k. 不同的数据集,k 值不同,MRMR-SFS 选出的特征子集就可能不同,运行时间和误差就会有差异. 在数据集 SP、CC 和 UJIL 上分别验证了 MRMR-SFS-P 的 k 值变化对性能的影响,如图 4 所示. 其中,SP 数据集 k 取值上限为 30,CC 和 UJIL 数据集 k 取值上限为 60.

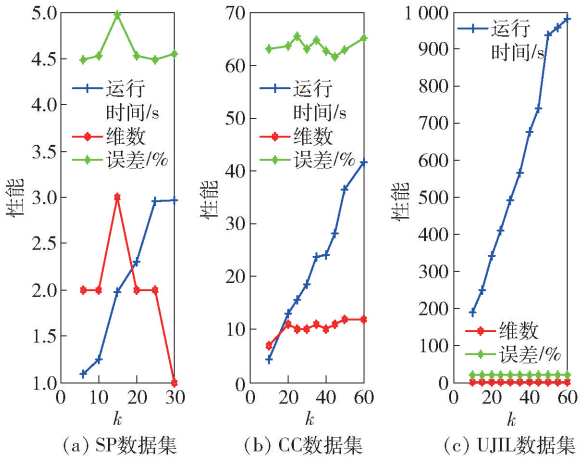

图 4 迭代次数对 MRMR-SFS 算法性能的影响

由图 4 可以看出,随着迭代次数 k 的增加,算法运行时间呈缓慢上升趋势,但所选特征子集的维数并没有随着 k 值的增加而呈上升趋势,平均误差也没有随着 k 值的增加而呈下降趋势. 实验刚开始时,MRMR-SFS-P 所选择特征子集的维数呈上升趋势,随后趋于稳定,这是由于近似最优特征子集的维数是比较稳定的. 实验刚开始时,平均误差有所上升,随着维数的增加,可能会选择出更优的特征组合,因此误差趋于稳定,仅有较小的波动.

综上分析,迭代次数 k 并不是越大越好,MRMR-SFS 算法能在较少的迭代次数(以上 4 个数据集实验过程中选择的 k 值为 $k = n \times 0.2$,其中,n 为原始特征集的维数)内选择出近似最优特征子集. k 值的选择可根据经验,或者经过多次尝试取误差最小时对应的 k 值.

4 结束语

分析了目前常用的 3 类特征选择算法的优点和不足,提出了一种快速特征选择框架 FFFS,基于该框架,提出了一种 MRMR-SFS 特征选择算法,该算法使用 MRMR 方法选择候选特征,借助 SFS 方法验证特征的性能,并限制了算法的迭代次数. 4 个数据集上的实验结果表明,与 SFS、FDHSFFS 算法相比,MRMR-SFS 算法的性能更优,而与 MRMRQ 算法相比,由于 MRMR-SFS 在特征选择过程中有学习算法的介入,其计算效率低于 MRMRQ,也正因为如此,其预测准确率在某些数据集上远高于 MRMRQ,最大相差 16.41%,具有更稳定的预测性能. 简单快速地确定 k 值是下一步的研究方向.

参考文献:

[1] Koller D, Sahami M. Toward optimal feature selection [C]// Proceedings of the 13th International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1996: 284-292.

[2] Ahmad A, Dey L. A feature selection technique for classificatory analysis [J]. Pattern Recognition Letters, 2005, 26(1): 43-56.

[3] Yu Lei, Liu Huan. Efficient feature selection via analysis of relevance and redundancy [J]. Journal of Machine Learning Research, 2004(5): 1205-1224.

[4] Marill T, Green D. On the effectiveness of receptors in recognition systems [J]. IEEE Transactions on Information Theory, 1963, 9(1): 11-17.

[5] El Akadi A, Amine A, El Ouardighi A, et al. A two-stage gene selection scheme utilizing MRMR filter and GA wrapper [J]. Knowledge and Information Systems, 2011, 26(3): 487-500.

[6] Kohavi R, John G H. Wrappers for feature subset selection [J]. Artificial Intelligence, 1997, 97(1-2): 273-324.

[7] 崔鸿雁, 徐帅, 张利锋, 等. 机器学习中的特征选择方法研究及展望[J]. 北京邮电大学学报, 2018, 41(1): 1-12.
Cui Hongyan, Xu Shuai, Zhang Lifeng, et al. The key techniques and future vision of feature selection in machine learning [J]. Journal of Beijing University of Posts and Telecommunications, 2018, 41(1): 1-12.

[8] Ding C, Peng Hanchuan. Minimum redundancy feature selection from microarray gene expression data [J]. Journal of Bioinformatics and Computational Biology, 2005, 3(2): 185-205.