

软件缺陷预测中基于排序集成的特征选择方法

姜 丽¹, 姜淑娟¹, 于 巧²

¹(中国矿业大学 计算机科学与技术学院, 江苏 徐州 221116)

²(江苏师范大学 计算机科学与技术学院, 江苏 徐州 221116)

E-mail: shjjiang@cumt.edu.cn

摘 要: 在软件缺陷预测中, 缺陷数据集中往往存在冗余或不相关特征, 需要对数据集进行特征选择. 为了避免软件缺陷预测中常见的基于排序的特征选择方法的不稳定性, 提出一种基于排序集成的特征选择方法. 首先, 分别执行相关系数、信息增益率和 ReliefF 三种特征选择方法, 得到特征排序序列, 赋予每个特征一个权重. 随后, 将三种方法得到的每个特征的权重相加求和, 作为该特征的总权重. 最后, 根据特征总权重对特征从高到低进行排序, 并按照特征百分比从前往后依次选取特征. 在实证研究中, 以 NASA 的 11 个数据集为实验对象, 采用逻辑回归算法构建预测模型, 并采用 AUC 指标度量不同预测模型的性能. 实验结果验证了基于排序集成的特征选择方法的有效性.

关键词: 软件缺陷预测; 特征选择; 特征权重; 排序集成

中图分类号: TP311

文献标识码: A

文章编号: 1000-1220(2018)07-1410-05

Feature Selection Method Based on Sorting Integration in Software Defect Prediction

JIANG Li¹, JIANG Shu-juan¹, YU Qiao²

¹(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

²(School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116, China)

Abstract: There are often redundant or irrelevant features in defect data sets in the field of software defect prediction, and feature selection is required. A feature selection method based on sorting integration is proposed which can avoid the instability of the common sorting feature selection methods. Firstly, Correlation, GainRatio and ReliefF are used respectively to obtain the feature sorting sequence based on which each feature can obtain a weight. Subsequently, the weights of each feature obtained by the three methods are added up to obtain the total weight of the feature. Finally, the features are sorted from high to low according to the total weights of the features, and the features are selected according to the percentage of the feature. In the empirical study, 11 data sets of NASA were used as experimental objects, and Logic Regression algorithm was used to construct the prediction model. Moreover, AUC metric was used to measure the classification performance of different prediction models. The experimental results show the effectiveness of the feature selection method based on sorting integration.

Key words: software defect prediction; feature selection; feature weight; sorting integration

1 引言

20 世纪 70 年代, 软件工程领域的研究人员发现软件中缺陷的分布并不是随机或平均的, 而是有规律的. 软件缺陷预测技术就是寻找这种规律的技术. 它是软件工程领域最为活跃的研究对象之一. 软件缺陷预测技术通过分析软件代码或软件开发过程, 设计出与软件缺陷相关的度量元. 随后通过挖掘软件历史仓库构建缺陷数据集. 最后基于该数据集构建缺陷预测模型. 预测被测项目中潜在的缺陷模块. Boehm^[1]指出软件缺陷在软件中的分布服从“2-8 原则”, 即 80% 的缺陷分布在 20% 的程序模块中. 软件缺陷预测技术能帮助软件测试人员定位这些可能有缺陷的模块, 使其着重针对这些模块进行测试, 修复尽可能多的软件缺陷, 提高软件测试的效率, 从而提高软件质量, 对指导和优化软件测试工作起着重要的作用.

然而, 在设计度量元的过程中, 若考虑过多的度量元可能会引入冗余特征或无关特征, 增加构建软件缺陷预测模型的时间, 甚至可能会影响软件缺陷预测模型的性能, 引起维数灾难问题. 特征选择是解决维数灾难的一种方法. 从缺陷数据集中移除冗余特征和无关特征, 选出最具代表性的特征子集, 是特征选择方法的关键. 对于提高软件缺陷预测性能有重要意义.

目前, 典型的特征选择方法有相关系数方法、信息增益率方法和 ReliefF 方法等, 不同特征选择方法的性能是有差异的, 甚至可以说差异较大. 这源于它们的方法原理不同, 得到的特征排序序列也不同. 那么是否可以选出三种常用的特征选择方法 (Correlation、GainRatio、ReliefF) 的通用特征子集, 能更好地对缺陷进行预测?

针对这个问题, 本文提出了一种基于排序集成的特征选

收稿日期: 2017-08-31 收修改稿日期: 2017-09-18 基金项目: 国家自然科学基金项目 (61673384, 61502497) 资助. 作者简介: 姜 丽, 女, 1994 年生, 硕士研究生, 研究方向为软件分析与测试、机器学习; 姜淑娟 (通信作者), 女, 1966 年生, 博士, 教授, 博士生导师, CCF 会员, 研究方向为编译技术、软件工程; 于 巧, 女, 1989 年生, 博士, 讲师, CCF 会员, 研究方向为软件分析与测试、机器学习.

择方法,它综合了三种常用的基于排序的特征选择方法,包括相关系数方法、信息增益率方法和 ReliefF 方法。该方法对于每一个缺陷数据集,在三种特征选择方法得到的特征序列的基础上,得到特征综合排名,分别选取排名靠前的占特征总数 10% 至 100% 的特征构建新的数据集,然后采用逻辑回归模型作为缺陷预测模型,并采用接受者操作特征曲线下面积 AUC 指标^[2,3]来评价不同预测模型的性能。实验结果表明:基于排序集成的特征选择方法综合了三种方法,进行特征选择后选出的特征子集更准确,避免了一种或几种特征选择方法错误选择特征子集而带来的预测模型性能低的情况。

2 研究背景与相关工作

2.1 软件缺陷预测

软件缺陷预测技术是通过分析软件代码、软件开发过程等,设计与软件缺陷相关的度量元,随后通过挖掘软件历史仓库构建缺陷数据集,对数据集进行预处理后构建缺陷预测模型,对被测项目内新程序模块有无缺陷进行预测的一种技术。

软件缺陷预测的研究内容主要包括:(1)设计与缺陷强相关的度量元,例如:McCabe 环路复杂度度量^[4],Halstead 科学度量^[5],Chidamber 和 Kemerer 提出的 CK 度量元^[6],软件演化度量元^[7]等;(2)研究数据集的预处理技术,如特征选择、去除噪声、聚类分析、抽样、维数规约、离散化和二元化等,目前研究人员重点关注缺陷预测数据集中的噪声问题、维数灾难问题和分类不平衡问题等;(3)改进预测模型。

2.2 特征选择方法

在软件缺陷预测中,特征选择是解决维数灾难的一种有效方法,目前已有的特征选择算法大致上可以分为两大类:基于子集搜索的特征选择算法和基于排序的特征选择算法。其中,基于子集搜索的特征选择算法考虑特征与类别属性之间的相关性和特征之间的关联性,但在特征较多的情况下,搜索空间太大,开销也太大。基于排序的特征选择算法根据每一个特征与类别属性之间的关联性,从高到低排序,排名越靠前,与类别属性的相关性越高,表明该特征区分不同类别的能力越强,然后从中选出排名靠前的特征,建立缺陷预测模型,这类算法效率比较高,但选出的特征子集内通常含有冗余特征。

特征选择方法还可以更细致地分为以下四类:

1) 嵌入方法:将特征选择嵌入到数据挖掘算法中,也就是数据挖掘算法本身会决定保留并使用的特征和要忽略的特征,决策树就是这类方法中的一种。

2) 过滤方法:在数据挖掘前,先使用独立于数据挖掘的算法进行特征选择,即对数据集进行预处理,产生一个特征子集。刘望舒等人^[8]基于过滤方法,提出软件缺陷预测中基于聚类分析的特征选择方法。本文基于过滤方法进行研究。

3) 包装方法:分别将每一个特征子集作为数据挖掘算法的输入,数据挖掘算法的结果反过来评价每一个特征子集,选取效果最好的特征子集。通常情况下无法枚举出全部的子集,并且特征数较多时,这类算法的开销太大。Menzies 等人^[9]和 Song 等人^[10]提出的通用缺陷预测框架中均含有基于包装方法的特征选择方法。

4) 混合方法:Gao 等人^[11]针对一个大规模遗留电信软件系统,实现了一种混合特征选择方法,在该方法中用到了多种

特征排序评估方法和特征子集评估方法。Wang 等人^[12]尝试借助集成学习方法将 18 种特征选择方法进行组合,发现集成少数的特征选择方法的预测性能要优于集成所有特征选择方法。

不同特征选择方法得到的特征排序序列不同,执行不同特征选择方法后训练出的预测模型的性能也有差异,执行单个特征选择算法后构建的预测模型的泛化能力较差。我们提出了一种基于排序集成的特征选择方法,它属于过滤方法,集成了三种基于排序的特征选择方法,包括相关系数方法、信息增益率方法和 ReliefF 方法,综合了三种方法得到的特征排序序列,赋予每个特征一个权重,该方法能够有效提高预测模型的泛化能力,有效避免单一特征选择方法的不稳定性。

3 相关理论

3.1 特征选择方法

3.1.1 皮尔森相关系数方法(Correlation)

皮尔森相关系数方法是通过衡量各特征与类别属性之间的线性相关性,来评估一个特征对分类的贡献程度的一种方法,最终选取相关度高的特征。相关系数通过两个变量的协方差除以它们的标准差来计算,皮尔森相关系数的计算公式如公式(1)所示。

$$\begin{aligned} \text{Pearson}(X, Y) &= \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E((X - \mu_X)(Y - \mu_Y))}{\sigma_X \sigma_Y} = \\ &= \frac{E(X)E(Y) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)} \sqrt{E(Y^2) - E^2(Y)}} \\ &= \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{(\sum X^2) - \frac{(\sum X)^2}{N}} \sqrt{(\sum Y^2) - \frac{(\sum Y)^2}{N}}} \quad (1) \end{aligned}$$

相关系数值在-1 至 1 之间,其中,-1 表示该特征与类别属性完全负相关,1 表示该特征与类别完全正相关,0 表示该特征与类别属性没有关系。

3.1.2 信息增益率方法(GainRatio)

信息增益率方法根据特征给分类带来的信息的多少进行特征选择,一个特征带来的信息越多,就越重要。其计算公式如公式(2)所示。

$$\text{GainRatio}(A) = \frac{\text{Gain}(A)}{\text{SplitE}(A)} \quad (2)$$

其中 Gain(A) 为特征 A 划分数据集 S 后的信息增益,SplitE(A) 表示特征的分裂信息。可以认为信息增益率最大的那个特征是与类别属性最相关的特征。

信息增益率与信息增益相比,引入了分裂信息,特征取值点越多,分裂信息值就越大,这抵消了特征取值点的数目对特征为分类系统带来的信息量的数目带来的影响。

3.1.3 ReliefF 方法

Relief 方法在 1992 年由 Kira 和 Rendell^[13]提出,ReliefF 方法是 Relief 方法的扩展,它不仅可以处理二分类数据集,还可以处理多分类数据集。ReliefF 方法是基于实例的评估方法,它根据特征区分邻近实例能力的好坏来评估特征的价值。ReliefF 方法认为每个特征的平均权重可以通过如下计算公式求出:

$$W(A) = W(A) - \sum_{j=1}^k \text{diff}(A, R, H_j) / (mk) + \sum_{C \in \text{class}(R)} \left[\frac{p(C)}{1 - p(\text{Class}(R))} \sum_{j=1}^k \text{diff}(A, R, M_j(C)) \right] / (mk) \quad (3)$$

在公式(3)中 k 表示抽取的邻近实例的数目, m 表示随机抽取实例的次数, $M_j(C)$ 代表类别 C (除实例 R 所属的类别外的其他类别) 中第 j 个与实例 R 最近邻的实例, $p(C)$ 表示属于类别 C 的实例所占的比例, $p(\text{Class}(R))$ 表示与随机选取的实例 R 属于同一个类别的实例的比例, $\text{diff}(A, R_1, R_2)$ 表示实例 R_1 和 R_2 在特征 A 上的差, 它的计算公式如下:

$$\text{diff}(A, R_1, R_2) = \begin{cases} \frac{|R_1[A] - R_2[A]|}{\max(A) - \min(A)} & \text{如果 } A \text{ 是连续的} \\ 0 & \text{如果 } A \text{ 是离散的, 且 } R_1[A] = R_2[A] \\ 1 & \text{如果 } A \text{ 是离散的, 且 } R_1[A] \neq R_2[A] \end{cases} \quad (4)$$

由公式(3)可以看出, 权重的计算方法是初始权重减去相同分类的该特征值的差值, 加上不同分类的该特征值的差值. 若一个特征与分类有关, 则相同分类的实例的该特征值应该相似, 而不同分类的该特征值应该有差别, 因此, 权重高的特征分类能力强.

3.2 逻辑回归模型

逻辑回归模型(Logistic Regression, LR)^[14] 是在线性回归模型的基础上引入一个 Sigmoid 函数得到的, 是一种非线性回归方法, 适合二分类问题. 其中 Sigmoid 函数如公式(5)所示.

$$g(z) = \frac{1}{1 + e^{-z}} \quad z \in R, g(z) \in (0, 1) \quad (5)$$

3.3 预测模型评价指标

我们研究软件缺陷预测中的二分类问题时, 预测可能出现四种不同的结果, 如表 1 所示. 预测类别对应矩阵行, 实际类别对应矩阵列, 每一个单元格是对应类别的实例数目. 真阳性(TP)和真阴性(TN)都是正确分类的结果, 即得到的预测类别和实际类别相符. 假阳性(FP)和假阴性(FN)是错误分类的结果, 即得到的预测类别和实际类别不符. 好的预测结果应该是矩阵的主对角线上的数值大, 而非主对角线上的数值小.

表 1 四种预测结果

Table 1 Four prediction results

		预测类别	
		有缺陷	无缺陷
实际类别	有缺陷	真阳性(TP)	假阳性(FP)
	无缺陷	假阴性(FN)	真阴性(TN)

真阳性率(TPR)是被正确预测为有缺陷的实例数除以实际类别为有缺陷的实例总数, 即

$$TPR = \frac{TP}{TP + FN} \quad (6)$$

假阳性率(FPR)是被错误预测为有缺陷的实例数除以实际类别为无缺陷的实例总数, 即

$$FPR = \frac{FP}{FP + TN} \quad (7)$$

接受者操作特征曲线下面积就是 AUC 值^[3], 该曲线表示

的是预测模型的真阳性率和假阳性率之间的关系, 如图 1 所示. 其中 x 轴为假阳性率, y 轴为真阳性率, 曲线上的每个点对应一个阈值. 对于一种分类方法, 每个阈值下会有一个 TPR 和 FPR. 通过调整该预测模型的阈值就可以得到 ROC 曲线.

较好的预测模型的假阳性率比较低, 真阳性率比较高, 所以较好的预测模型的接受者操作特征曲线较靠近(0,1)点, 也就是靠近左上角. 主对角线代表随机预测的预测模型.

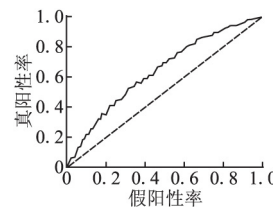


图 1 ROC 曲线

Fig. 1 ROC curve

接受者操作特征曲线下面积 AUC 是评估模型平均性能的一种方法. 如果一个预测模型的 AUC 值等于 1, 那么它是完美的; 如果一个预测模型的 AUC 值等于 0.5, 那么它和随机预测模型的效果是一样的; 如果一个预测模型的 AUC 值比另一个预测模型大, 那么它的性能比另一个模型好.

于巧等人^[15]指出由于 AUC 指标具有一个很好的性能: 当训练集中不同类别的实例分布变化时, 即分类不平衡情况下, AUC 值能够保持相对稳定, 所以使用 AUC 作为评价指标会更加准确可靠. Jiang 等人^[16]也通过实验证明了 AUC 指标的误差更小且更稳定, 且在实证研究中 AUC 指标被广泛应用于评价预测模型的性能^[8,15]. 基于以上原因, 本文实验都在以 AUC 为性能评价指标的基础上进行.

4 本文方法

本文提出一种基于排序集成的特征选择方法, 它结合了 Correlation、GainRatio、ReliefF 三种特征选择方法, 结合的具体方法如下: (1) 分别执行三种特征选择方法, 得到特征排序序列; (2) 对于每一个特征, 将特征总数减去该特征的序列号作为权重赋给该特征; (3) 对于每一个特征, 将三种方法得到的权重相加求和, 得到该特征的总权重; (4) 根据特征总权重对特征从高到低进行排序, 得到特征排序序列; (5) 从特征序列中从前往后依次选取排名前百分之 x 的特征. 具体算法如算法 1 所示.

算法 1. 基于排序集成的特征选择方法

Input: dataset, the percentage of features x

Output: the order of indexes

```

1  begin
2  a[] = correlation( dataset );
3  b[] = gainRatio( dataset );
4  c[] = reliefF( dataset );
5  n = numAttributes( dataset );
6  for each feature i in a do
7      weight[a[i]] = n-i;
8      for each feature j in b do
9          if a[i] = b[j] then
10             weight[a[i]] = n-j + weight[a[i]];
11      end_for
12      for each feature k in c do
13          if a[i] = c[k] then
14             weight[a[i]] = n-k + weight[a[i]];
15      end_for
16  end_for
17  sort weight[] in descending order;
18  get the order of indexes which are the top x
   percentage of features;
19  end

```

该方法在本质上是一种投票法. 第 2 行至第 4 行, 分别执

行 Correlation 方法、GainRatio 方法、ReliefF 方法,得到三个特征排序序列,与类别属性最相关的特征排在前面,越靠后表示与类别属性相关性越弱。第 5 行至第 16 行,将特征总数减去该特征的序列号作为权重赋给该特征,并将每种方法得到的每个特征的权重分别相加求和,得到该特征的总权重。第 17 行根据特征总权重对特征从高到低进行排序,排序越靠前的特征权重越大,即该特征越重要,得到基于排序集成的特征选择方法获得的特征排序序列。第 18 行选取排名前百分之 x 的特征,作为最终选出的特征子集。

5 实验评估

本文为了研究基于排序集成的特征选择方法对软件缺陷预测性能的影响,提出了如下问题:在软件缺陷预测问题中,与三种常用的基于排序的特征选择方法(相关系数方法、信息增益率方法、ReliefF 方法)相比,本文提出的基于排序集成的特征选择方法能否提高缺陷预测模型的性能?

5.1 实验对象

本文实验以美国国家航空航天局 NASA 发布的 11 个缺陷数据集为实验对象,包括 CM1、JM1、KC3、MC1、MC2、MW1、PC1、PC2、PC3、PC4 和 PC5,这些数据集均为 PROMISE 库¹中的缺陷数据集。其中的特征均为软件代码度量元,包括 McCabe 环路复杂度度量和 Halstead 科学度量。McCabe 环路复杂度度量是对程序内部结构的复杂度进行分析,认为循环和选择所构成的环路越多,程序就越复杂,就越容易出现缺陷;Halstead 科学度量根据操作符与操作数度量模块的缺陷倾向性。数据集的基本信息如表 2 所示。

表 2 缺陷数据集的基本信息

Table 2 Basic information of the defect data set

数据集	特征数	实例数	有缺陷数	无缺陷数
CM1	37	327	42	285
JM1	21	7782	1672	6110
KC3	39	194	36	158
MC1	38	1988	46	1942
MC2	39	125	44	81
MW1	37	253	27	226
PC1	37	705	61	644
PC2	36	745	16	729
PC3	37	1077	134	943
PC4	37	1287	177	1110
PC5	38	1711	471	1240

第 1 列是数据集名称,第 2 列表示特征数,第 3、4、5 列分别表示全部的实例数,有缺陷的实例数,无缺陷的实例数。

5.2 实验设计

为了验证本文提出的基于排序集成的特征选择方法的有效性,本文通过编程实现了该方法。实验环境为 Windows 7, 32 位 2G RAM, Open JDK 1.7, WEKA 3.7。

利用 4 种特征选择方法,10 个百分比和表 2 中的 11 个数据集进行组合实验。

首先选取一个数据集,分别采用相关系数方法、信息增益

率方法、ReliefF 方法和基于排序集成的特征选择方法对该数据集进行特征选择;然后利用逻辑回归方法构建预测模型,利用 10 折交叉验证评估该预测模型的性能,得到该预测模型的性能评价指标 AUC 的值;最后对 11 个数据集做重复实验,通过对这些 AUC 值画折线图,比较本文提出的方法与三种方法的性能差异,验证软件缺陷预测中基于排序集成的特征选择方法的有效性。在实验评估中,我们使用 WEKA 软件包实现特征选择算法、逻辑回归分类算法等,各特征选择方法除选取的特征数量参数按特征百分比设置外,其他的参数均采用 WEKA 默认的参数设置。

10 折交叉验证是评估分类方法性能的一种常用的方法,即将数据集分成 10 等份,依次选取其中的 9 份作为训练集,剩下的 1 份作为测试集,进行 10 次训练和测试,每次都会得到预测模型评价指标的值,将 10 次的平均值作为最终的评估值。在执行交叉验证之前,先使用随机数发生器将数据随机化,使用的随机种子不同会使数据的随机化也不同,对同一数据集的交叉验证就会产生不同的结果。为了使实验多次运行都能得到完全相同的结果,也就是为了使实验具有可重复性,本文实验指定“1234”为固定的随机种子。

5.3 实验结果与分析

与三种常用的基于排序的特征选择方法(相关系数、信息增益率、ReliefF 方法)相比,本文提出的基于排序集成的特征选择方法是否能够提高缺陷预测模型的性能?

图 2 表示在每一个数据集上执行四种特征选择方法后的预测性能的对比。横坐标表示采用特征选择方法选取的特征百分比(10% 至 100%, 100% 表示不进行特征选择),纵坐标表示评价指标 AUC 的值,其中 Assembly 表示本文提出的基于排序集成的特征选择方法。

从图 2 可以看出:基于排序集成的特征选择方法能在一定程度上消除某一种或某几种特征选择方法的不稳定性。以子图(h)为例,ReliefF 方法在从数据集 PC2 中选取 10% 的特征时,由于选取的最佳特征子集不准确,构建的预测模型与随机预测模型相似,性能较差。执行基于排序集成的特征选择方法后得到的预测模型性能较好,较 ReliefF 特征选择方法性能提升了 40.21%,原因在于基于排序集成的特征选择方法会综合考虑三种特征选择方法选取的特征子集的优劣,选出更优质的特征子集。

从图 2 的子图(c)(d)(e)(g)(j)可以看出:在只选取 10% 的特征时,基于排序集成的特征选择方法会选出较三种方法更好的特征子集,构建的预测模型的预测性能更好。

除了在上述 10% 的情况下执行基于排序集成的特征选择方法后会提高缺陷预测性能外,在某些数据集选取某些百分比的特征后,也会获得比执行其他三种方法好的性能。以子图(h)为例,从 PC2 数据集中选取 40% 的特征后的预测模型的预测性能与相关系数方法、信息增益率方法、ReliefF 方法相比,分别提高了 7.78%、6.08% 和 4.98%。

基于上述分析,基于排序集成的特征选择方法能有效避免单一特征选择方法错误地选择特征子集,而带来的预测模型性能低的情况,也避免了选择特征选择方法的盲目性。

¹ The Promise Repository of Empirical Software Engineering Data <http://openscience.us/repo/>

6 结束语

本文提出一种基于排序集成的特征选择方法,集成了相关系数方法、信息增益率方法、ReliefF 方法,能避免单一特征选择方法的不稳定性。由于综合了三种方法,执行基于排序集成的特征选择方法后选出的特征子集更准确,避免了单一特征选择方法错误选择特征子集,而带来的预测模型性能低的情况,能消除某一或几种特征选择方法的不稳定性。

实验中观察到有些情况下采用基于排序集成的特征选择方法的预测性能不会得到较大的提升,因为相关系数方法、信息增益率方法、ReliefF 方法都只考虑特征与类别的相关性,而没有考虑冗余特征。基于排序集成的特征选择方法是三种方法的结合,也不会考虑特征冗余。假如一个数据集的特征之

情况,能消除某一或几种特征选择方法的不稳定性。实验中观察到有些情况下采用基于排序集成的特征选择方法的预测性能不会得到较大的提升,因为相关系数方法、信息增益率方法、ReliefF 方法都只考虑特征与类别的相关性,而没有考虑冗余特征。基于排序集成的特征选择方法是三种方法的结合,也不会考虑特征冗余。假如一个数据集的特征之

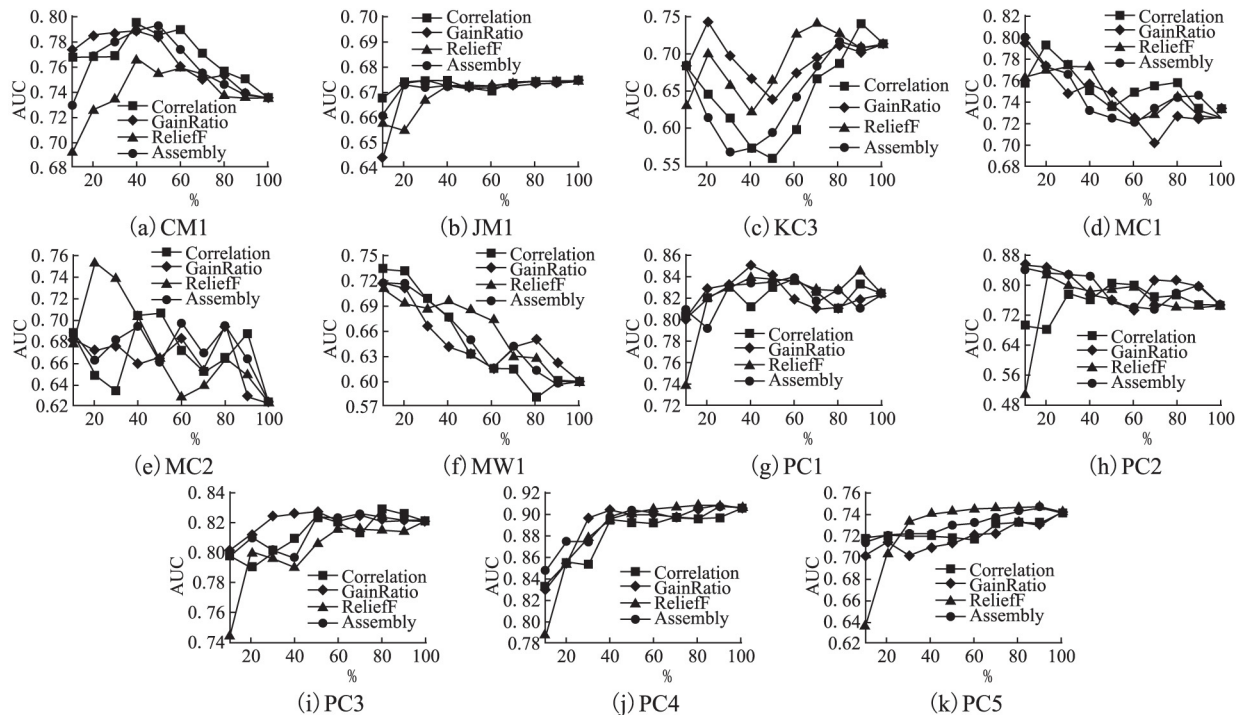


图2 基于排序集成的特征选择方法与其他方法的比较

Fig. 2 Comparison of feature selection algorithm based on sort integration with other methods

间具有很强的关联性,那么从中选择最优的特征子集时就很难考虑到特征冗余问题。如何加入去除冗余特征的方法是未来值得研究的内容。

References:

- [1] Boehm B W. Value-based software engineering: overview and agenda[M]. Berlin: Springer, 2006: 3-14.
- [2] Bradley A P. The use of the area under the ROC curve in the evaluation of machine learning algorithms[J]. Pattern Recognition, 1997, 30(7): 1145-1159.
- [3] Huang J, Ling C X. Using AUC and accuracy in evaluating learning algorithms[J]. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2005, 17(3): 299-310.
- [4] McCabe T J. A complexity measure[J]. IEEE Transactions on Software Engineering (TSE), 1976, SE-2(4): 308-320.
- [5] Halstead M H. Elements of software science[M]. New York, USA: Elsevier, 1977.
- [6] Chidamber S R, Kemerer C F. A metrics suite for object oriented design[J]. IEEE Transactions on Software Engineering (TSE), 1994, 20(6): 476-493.
- [7] Wang Dan-dan, Wang Qing. Improving the performance of defect prediction based on evolution data[J]. Journal of Software, 2016, 27(12): 3014-3029.
- [8] Liu Wang-shu, Chen Xiang, Gu Qing, et al. A cluster-analysis-based feature-selection method for software defect prediction[J]. Science China: Information Science, 2016, 46(9): 1298-1320.
- [9] Menzies T, Greenwald J, Frank A. Data mining static code attributes to learn defect predictors[J]. IEEE Transactions on Software Engineering (TSE), 2007, 33(1): 637-640.
- [10] Song Q, Jia Z, Shepperd M, et al. A general software defect-prone-ness prediction framework[J]. IEEE Transactions on Software Engineering (TSE), 2011, 37(3): 356-370.
- [11] Gao K, Khoshgoftar T M, Wang H, et al. Choosing software metrics for defect prediction: an investigation on feature selection techniques[J]. Software: Practice and Experience (SPE), 2011, 41(5): 579-606.
- [12] Wang H, Khoshgoftar T M, Napolitano A. A comparative study of ensemble feature selection techniques for software defect prediction[C]. Proceedings of the 9th International Conference on Machine Learning and Applications (ICMLA 2010). Piscataway, USA: IEEE, 2010: 135-140.
- [13] Kira K, Rendell L A. A practical approach to feature selection[C]. Proceedings of the 9th International Workshop on Machine Learning (ML 1992). Burlington, Massachusetts: Morgan Kaufmann, 1992: 249-256.
- [14] Cessie S L, Houwelingen J C V. Ridge estimators in logistic regression[J]. Journal of the Royal Statistical Society, 1992, 41(1): 191-201.
- [15] Yu Qiao, Jiang Shu-juan, Zhang Yan-mei, et al. The impact study of class imbalance on the performance of software defect prediction models[J]. Chinese Journal of Computers, 2018, 41(4): 809-824.
- [16] Jiang Y, Lin J, Cukic B, et al. Variance analysis in software fault prediction models[C]. Proceedings of the 20th International Symposium on Software Reliability Engineering (ISSRE 2009). Los Alamitos, CA: IEEE Computer Society, 2009: 99-108.

附中文参考文献:

- [7] 王丹丹, 王青. 基于演化数据的软件缺陷预测性能改进[J]. 软件学报, 2016, 27(12): 3014-3029.
- [8] 刘望舒, 陈翔, 顾庆, 等. 软件缺陷预测中基于聚类分析的特征选择方法[J]. 中国科学: 信息科学, 2016, 46(9): 1298-1320.
- [15] 于巧, 姜淑娟, 张艳梅, 等. 分类不平衡对软件缺陷预测模型性能的影响研究[J]. 计算机学报, 2018, 41(4): 809-824.