ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 32,No. 1,Jan. 2017,pp. 166－174
DOI:10. 16337/j. 1004-9037. 2017. 01. 020
© 2017 by Journal of Data Acquisition and Processing
http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

基于邻域三支决策粗糙集模型的软件缺陷预测方法

李伟津[1]　郭鸿昌[2]

(1. 南京航空航天大学航天学院,南京,210016;2. 东部战区空军装备部,南京,210081)

摘　要:基于已有软件缺陷数据,建立分类模型对待测软件模块进行预测,能够提高测试效率和降低测试成本。现有基于机器学习方法对软件缺陷预测的研究大部分基于二支决策方式,存在误分率较高等问题。本文针对软件缺陷数据具有代价敏感特性且软件度量取值为连续值等特性,提出了一种基于邻域三支决策粗糙集模型的软件缺陷预测方法,该方法对易分错的待测软件模块作出延迟决策,和二支决策方法相比,降低了误分类率。在 NASA 软件数据集上的实验表明所提方法能够提高分类正确率并减小误分类代价。

关键词:软件缺陷分类;邻域三支决策粗糙集模型;三支决策

中图分类号:TP18　　**文献标志码**:A

Software Defect Prediction Method Based on Neighborhood Three-way Decision-theoretic Rough Set Model

Li Weiwei[1], Guo Hongchang[2]

(1. College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China; 2. Air Force Equipment Department of Eastern Theater Command, Nanjing, 210081, China)

Abstract: Based on existing software defect data, it is possible to improve the efficiency of software testing and reduce the test cost by establishing the classification model to predict the software modules. Most machine learning based defect prediction researches are based on two-way decision method. Since software defect prediction can be seen as a kind of cost-sensitive learning problem, and the software data has continuous values, this paper proposes a classification method based on neighborhood three-way decision-theoretic rough set model. For ambiguous testing modules, compared with two-way decision methods, this method makes a deferment decision to reduce the misclassification rate. Experimental results on NASA software datasets show that the proposed method can get a higher classification accuracy and a lower misclassification cost.

Key words: software defect classification; neighborhood three-way decision-theoretic rough set model; three-way decisions

引　　言

软件缺陷预测在减少软件开发成本和提高软件质量方面发挥着重要作用[1-3]。在软件测试过程中,

基金项目:国家重点研发计划(2016YFB1000802)资助项目;国家高技术研究发展计划("八六三"计划)(2015AA105303)资助项目。
收稿日期:2016-10-21;**修订日期**:2017-01-01

通过软件缺陷预测可以帮助软件项目管理者合理安排有限的测试时间和人力资源,从而在有限的测试资源情况下提高软件测试的有效性及软件质量。目前对软件缺陷预测的研究主要可以分为两类:(1)通过对软件模块缺陷数目的预测,将某些模块定位为高缺陷率模块或低缺陷率模块,这种方法通常将软件缺陷预测看作回归问题[4]。(2)将软件缺陷预测看作分类问题[5],将软件缺陷分类为有缺陷趋势模块和无缺陷趋势的模块,通常使用决策树、贝叶斯网络、人工神经网络和支持向量机等机器学习方法[6]。也有部分学者考虑到软件缺陷预测具有代价敏感特性,将代价敏感学习方法应用到软件缺陷分类。如 Zheng[7] 和 Arar 等[8]将基于代价敏感神经网络模型用于软件缺陷分类。Liu 等[9]将代价敏感特征选择和代价敏感 BP 神经网络应用与软件缺陷分类。不论在缺陷分类问题中是否考虑代价敏感,现有的研究都是假设缺陷分类是一个二分类问题,并应用二支决策方式,即对软件模块做出接受其为有缺陷趋势模块和拒绝其为有缺陷趋势模块的决策。二支决策方法属于立即决策方式,能够简单快速地给出分类结果,但是存在着误分类率较高的问题。简单二支决策方法会基于多数原则依据给定软件模块属于缺陷趋势模块的条件概率对其进行判定,这就导致对处于中间模糊地带不易划分的软件模块容易做出错误的决策。立即决策方式误分率较高,由此带来的误分类代价也会增加[10]。

针对此类问题提出了一种代价敏感相关的邻域三支决策粗糙集模型,并在此基础上设计了一种三支决策邻域分类方法。该分类方法考虑到软件缺陷预测的代价敏感问题,对软件模块进行分类采用三支决策方法,依据不同错误分类带来的损失代价不同,设置相应的代价函数,计算三支决策所需阈值对,将误分率高的软件模块划分到边界域中,交由专家评判或等待进一步处理,从而能够降低缺陷分类的误分类率,减少代价损失。在 NASA 软件数据集上进行了相应的对比实验,实验结果表明该方法能有效地提高分类正确率,降低误分类代价。

1 邻域三支决策粗糙集模型

考虑软件缺陷数据具有代价敏感特性,且通过对软件缺陷数据进行度量提取出的特征或属性取值为连续值的特点,结合三支决策粗糙集模型[11]和邻域粗糙集模型的优点,提出了一种邻域三支决策粗糙集模型,使之能够处理具有复杂特性的软件缺陷数据。

1.1 三支决策粗糙集模型

定义 1 决策表可表示为一个四元组

$$S=\langle U,At=C\cup D,\{V_{at}\mid at\in At\},\{I_{at}\mid at\in At\}\rangle \tag{1}$$

式中:有限集合 U 为论域;At 为属性集合;C 为条件属性集;D 为决策属性集;V_{at} 表示属性 at 的值域;I_{at}:$U\rightarrow V_{at}$ 为从 U 到 V_{at} 的映射函数,通常 I_{at} 假设为单值的,任意对象 $x\in U$ 在属性 $at\in At$ 上的取值可以表示为 $I_{at}(x)$。在粗糙集领域,通常用等价类的形式来刻画或描述对象 x。对象 x 的等价类定义为

$$[x]_A=\{y\in U\mid \forall at\in A(I_{at}(x)=I_{at}(y))\} \tag{2}$$

在贝叶斯决策理论中,对于给定数据对象 x,假设 $\Omega=\{w_1,\cdots,w_2\}$ 是 x 所有可能状态的有穷集合,$\mathfrak{A}=\{a_1,\cdots,a_t\}$ 是所有可能行为的有限集合,$p(w_j\mid x)$ 表示当对象 x 的状态为 w_j 的条件概率。设 $\lambda(a_i\mid w_j)$ 为当对象实际状态为 w_j 时采取行为 a_i 的损失函数,或简记为 $\lambda_{a_iw_j}$。假设对对象 x 采取的行为为 a_i,则该行为所带来的预期风险(损失)可表示为

$$R(a_i\mid x)=\sum_{j=1}^{s}\lambda(a_i\mid w_j)\cdot p(w_j\mid x) \tag{3}$$

在决策粗糙集理论中,令 $\Omega=\{X,X^c\}$ 表示对象属于集合 X 或其补集的状态集合。$p(X\mid[x])=\frac{|X\cap[x]|}{|[x]|}$ 表示等价类 $[x]$ 中元素属于 X 的条件概率,$p(X^c\mid[x])=1-p(X\mid[x])$ 表示等价类 $[x]$ 中元素属于 X^c 的概率。$\mathfrak{A}=\{a_P,a_B,a_N\}$ 代表 3 种决策行为,其中 a_P 表示将对象划分到正域,a_B 表示将对象划分到边界域,a_N 表示将对象划分到负域。每种决策行为都伴随有相应的代价函数。在不同状态下,对对象 x 采取不同决策时会带来不同的风险代价,代价函数矩阵如表 1 所示。表 1 中 λ_{PP},λ_{BP} 和 λ_{NP}

分别表示在对象 x 实际属于 X 时分别采取的决策行为 a_P, a_B 和 a_N 的风险代价值；λ_{PN}, λ_{BN} 和 λ_{NN} 为在对象 x 实际属于 X^c 时分别采取决策行为 a_P, a_B 和 a_N 的风险代价值。依据预期风险的公式可知，对对象 x 采取相应动作时所带来的贝叶斯风险代价分别为

$$\begin{aligned}\Re_P &= \Re(a_P \mid [x]) = \lambda_{PP} \cdot p(X \mid [x]) + \lambda_{PN} \cdot p(X^c \mid [x])\\ \Re_B &= \Re(a_B \mid [x]) = \lambda_{BP} \cdot p(X \mid [x]) + \lambda_{BN} \cdot p(X^c \mid [x])\\ \Re_N &= \Re(a_N \mid [x]) = \lambda_{NP} \cdot p(X \mid [x]) + \lambda_{NN} \cdot p(X^c \mid [x])\end{aligned} \tag{4}$$

根据贝叶斯最小风险决策原则，可以得到的决策规则为

(P)若 $\Re_P \leqslant \Re_B$ 且 $\Re_P \leqslant \Re_N$，则判定 $x \in \mathrm{POS}(X)$；

(B)若 $\Re_B \leqslant \Re_P$ 且 $\Re_B \leqslant \Re_N$，则判定 $x \in \mathrm{BND}(X)$；

(N)若 $\Re_N \leqslant \Re_P$ 且 $\Re_N \leqslant \Re_B$，则判定 $x \in \mathrm{NEG}(X)$。

表 1 不同决策行为在不同状态下的风险代价

Tab. 1 Different decision costs based on different actions

决策行为	a_P	a_B	a_N
X	λ_{PP}	λ_{BP}	λ_{NP}
X^c	λ_{PN}	λ_{BN}	λ_{NN}

其中：$\mathrm{POS}(X)$ 表示 X 的正域；$\mathrm{BND}(X)$ 表示 X 的边界域；$\mathrm{NEG}(X)$ 表示 X 的负域。考虑一种特殊情况，假设损失函数满足 $\lambda_{PP} \leqslant \lambda_{BP} < \lambda_{NP}$ 和 $\lambda_{NN} \leqslant \lambda_{BN} < \lambda_{PN}$，其实际意义是：对于一实际属于 X 的对象 x，将其划分到 X 的正域所带来的风险要小于或等于将其划分到边界区域带来的风险；这两者的风险都小于将其划分到 X 的负域所带来的风险。同理，对于不属于 X 的对象 x，将其划分到 X 的负域所带来的风险要小于或等于将其划分到边界区域的风险；这两者的风险都小于将其划分到 X 的正域所带来的风险。该假设符合现实意义。又因为 $p(X \mid [x]) + p(X^c \mid [x]) = 1$，则可以推导出以下简化的三支决策规则为

(P)若 $p(X \mid [x]) \geqslant \alpha$ 且 $p(X \mid [x]) \geqslant \gamma$，则判定 $x \in \mathrm{POS}(X)$；

(B)若 $p(X \mid [x]) \leqslant \alpha$ 且 $p(X \mid [x]) \geqslant \beta$，则判定 $x \in \mathrm{BND}(X)$；

(N)若 $p(X \mid [x]) \leqslant \beta$ 且 $p(X \mid [x]) \leqslant \gamma$，则判定 $x \in \mathrm{NEG}(X)$。

其中 α, β 和 γ 值分别为

$$\begin{aligned}\alpha &= \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}\\ \beta &= \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}\\ \gamma &= \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}\end{aligned} \tag{5}$$

考虑条件

$$\frac{(\lambda_{NP} - \lambda_{BP})}{(\lambda_{BN} - \lambda_{NN})} > \frac{(\lambda_{BP} - \lambda_{PP})}{(\lambda_{PN} - \lambda_{BN})} \tag{6}$$

可得 $0 \leqslant \beta < \gamma < \alpha \leqslant 1$，则可以进一步简化上述三支决策规则为

(P1)若 $p(X \mid [x]) \geqslant \alpha$，则判定 $x \in \mathrm{POS}(X)$；

(B1)若 $\beta < p(X \mid [x]) < \alpha$，则判定 $x \in \mathrm{BND}(X)$；

(N1)若 $p(X \mid [x]) \leqslant \beta$，则判定 $x \in \mathrm{NEG}(X)$。

1.2 邻域粗糙集模型

经典粗糙集方法只能处理离散值，而为了能够处理数值型数据，很多学者将其扩展到邻域系统中。一种最具代表性的模型是文献[12]提出的基于距离的邻域粗糙集模型。

定义 2 给定决策表 S，对象 $x_i \in U$ 且 $A \subseteq At$，在子空间 A 中，x_i 的邻域粒子 $\delta_A(x_i)$ 定义为

$$\delta_A(x_i) = \{x_j \mid x_j \in U, \Delta_A(x_i, x_j) \leqslant \delta\} \tag{7}$$

式中 Δ 为度量函数，两个对象之间的 Minkowski 距离定义为

$$\Delta_M(x_i,x_j)=(\sum_{k=1}^{N}|I_{a_k}(x_i)-I_{a_k}(x_j)|^M)^{1/M} \tag{8}$$

式中:x_i 和 x_j 为 N 维空间 $At=\{a_1,a_2,\ldots,a_N\}$ 中的两个对象。Minkowski 距离也被称为:(1)如果 $M=1$,Δ_1 为 Manhattan 距离;(2)如果 $M=2$,Δ_2 为 Euclidean 距离;(3)如果 $M=\infty$,Δ_∞ 为 Chebyshev 距离。给定度量空间$\langle U,\Delta\rangle$,粒系统由邻域粒子族$\{\delta(x_i)\mid x_i\in U\}$组成,该系统基于覆盖,而不基于划分。显而易见,基于划分的经典粗糙集可以看作是基于覆盖的邻域粗糙集的一个特例,只要令 $\delta=0$ 即可。$\langle U,N_A\rangle$是定义在属性集 A 上的邻域近似空间。C 是关于邻域关系 N 对 U 的一个覆盖,那么由覆盖 C 推导出的邻域粒子族可用来刻画邻域粗糙近似空间。对于任意子集 $X\subseteq U$,基于覆盖 C 的邻域粗糙集的下近似$\underline{N_C}(X)$及上近似$\overline{N_C}(X)$定义为

$$\underline{N_C}(X)=\{x_i\mid\delta(x_i)\subseteq U,x_i\in U\}$$

$$\overline{N_C}(X)=\{x_i\mid\delta(x_i)\cap X\neq\varnothing,x_i\in U\} \tag{9}$$

根据粗糙集的上下近似定义,邻域粗糙集的正域、边界域和负域可表示为

$$\mathrm{POS}_C(X)=\underline{N_C}(X)$$

$$\mathrm{BND}_C(X)=\overline{N_C}(X)-\underline{N_C}(X)$$

$$\mathrm{NEG}_C(X)=U-\mathrm{POS}_C(X)\cup\mathrm{BND}_C(X)=U-\overline{N_C}(X) \tag{10}$$

定理 1 给定两个信息系统$\langle U,N_1\rangle$和$\langle U,N_2\rangle$,分别对应两个非负 δ_1 和 δ_2,如果 $\delta_1\leqslant\delta_2$,则有(1)$\forall x_i\in U:\delta_1(x_i)\leqslant\delta_2(x_i)$;(2)$\forall X\subseteq U:\underline{N_2}(X)\subseteq\underline{N_1}(X)$,$\overline{N_1}(X)\subseteq\overline{N_2}(X)$。具体证明和邻域粗糙集的其他概念详细可见文献[12],在此不予赘述。

1.3 领域三支决策粗糙集模型

三支决策粗糙集模型是一种代价敏感学习模型,而且其通过引入三支决策所需的阈值,使得该模型对于有噪音的数据具有较高的容忍度,但是经典的三支决策粗糙集模型只能处理离散值的数据,无法直接处理连续值,现有三支决策相关研究也大部分着重于离散数据的处理[13,14]。而邻域粗糙集模型则提供了一种直接处理连续值的方法,基于此,结合这两种模型的优点,提出了一种邻域三支决策粗糙集模型,使之能够直接处理具有代价敏感特性且取值为连续值的软件缺陷数据。在决策粗糙集和其他粗糙集模型里,对象 x 是用等价类$[x]$形式来刻画。而在邻域三支决策粗糙集模型中,采用邻域粒子 $\delta(x)$来表示,则对象 x 属于某一类 X 的概率表示为 $p(X\mid\delta(x))=\dfrac{|X\cap\delta(x)|}{|\delta(x)|}$,则 X 的上近似和下近似值可定义如下。

定义 3 给定信息系统$\langle U,N\rangle$及代价函数矩阵,基于代价函数矩阵和决策粗糙集模型,可以计算得出阈值函数(α,β),其中 $0\leqslant\beta\leqslant\alpha\leqslant1$,对于任何子集 $X\subseteq U$,在子空间 $B\subseteq At$,X 的(α,β)下近似$\underline{N_B^{(\alpha,\beta)}}(X)$及上近似$\overline{N_B^{(\alpha,\beta)}}(X)$定义为

$$\underline{N_B^{(\alpha,\beta)}}(X)=\{x_i\mid p(X\mid\delta_B(x_i))>\alpha,x_i\in U\}$$

$$\overline{N_B^{(\alpha,\beta)}}(X)=\{x_i\mid p(X\mid\delta_B(x_i))\geq\beta,x_i\in U\} \tag{11}$$

对于任意子集 $X\subseteq U$,在子空间 $B\subseteq At$,X 的正域、边界域和负域定义为

$$\mathrm{POS}_B^{(\alpha,\beta)}(X)=\underline{N_B^{(\alpha,\beta)}}(X)$$

$$\mathrm{BND}_B^{(\alpha,\beta)}(X)=\overline{N_B^{(\alpha,\beta)}}(X)-\underline{N_B^{(\alpha,\beta)}}(X)$$

$$\mathrm{NEG}_B^{(\alpha,\beta)}(X)=U-(\mathrm{POS}_B^{(\alpha,\beta)}(X)\cup\mathrm{BND}_B^{(\alpha,\beta)}(X))=U-\overline{N_B^{(\alpha,\beta)}}(X) \tag{12}$$

在决策表中，$\pi_D=\{D_1,D_2,\cdots,D_m\}$ 是对论域 U 的划分，表示 m 个决策类。邻域三支决策粗糙集模型下 π_D 的正域、边界域和负域可定义为

$$\begin{aligned}\mathrm{POS}_{\mathrm{B}}^{(\alpha,\beta)}(\pi_D)&=\bigcup_{1\leqslant i\leqslant m}\mathrm{POS}_{\mathrm{B}}^{(\alpha,\beta)}(D_i)\\ \mathrm{BND}_{\mathrm{B}}^{(\alpha,\beta)}(\pi_D)&=\bigcup_{1\leqslant i\leqslant m}\mathrm{BND}_{\mathrm{B}}^{(\alpha,\beta)}(D_i)\\ \mathrm{NEG}_{\mathrm{B}}^{(\alpha,\beta)}(\pi_D)&=U-\mathrm{POS}_{\mathrm{B}}^{(\alpha,\beta)}(\pi_D)\bigcup\mathrm{BND}_{\mathrm{B}}^{(\alpha,\beta)}(\pi_D)\end{aligned}\tag{13}$$

决策粗糙集可以看作是邻域三支决策粗糙集的一种特例，即 $\delta=0$。当 $\delta=0$，$\delta(x)$ 表示等价关系。另外，若邻域决策粗糙集中的 $\alpha=1$ 且 $\beta=0$ 时，则邻域三支决策粗糙集模型转化为经典邻域粗糙集模型。

定理 2 给定信息系统 $<U,N>$，两个非负值 α_1 和 α_2，如果 $\alpha_1\leqslant\alpha_2$ 且 δ 值相同，则有 $\forall X\subseteq U$：$\mathrm{POS}_{\mathrm{B}}^{(\alpha_2,\beta)}(X)\subseteq\mathrm{POS}_{\mathrm{B}}^{(\alpha_1,\beta)}(X)$。

证明 对于任意 $y\in\mathrm{POS}_{\mathrm{B}}^{(\alpha_1,\beta)}(X)$，则有 $p(X\mid\delta_B(y))>\alpha_2$。因为 $\alpha_1\leqslant\alpha_2$，故有 $p(X\mid\delta_B(y))>\alpha_1$，推导出 $y\in\mathrm{POS}_{\mathrm{B}}^{(\alpha_2,\beta)}(X)$，因此，$\mathrm{POS}_{\mathrm{B}}^{(\alpha_2,\beta)}(X)\subseteq\mathrm{POS}_{\mathrm{B}}^{(\alpha_1,\beta)}(X)$。

定理 3 给定信息系统 $\langle U,N\rangle$，两个非负值 β_1 和 β_2，若 $\beta_1\leqslant\beta_2$ 且 δ 值均相等，则有 $\forall X\subseteq U$：$\mathrm{BND}_{\mathrm{B}}^{(\alpha,\beta_2)}(X)\subseteq\mathrm{BND}_{\mathrm{B}}^{(\alpha,\beta_1)}(X)$。

证明 对于任意 $y\in\mathrm{BND}_{\mathrm{B}}^{(\alpha,\beta_2)}(X)$，则有 $\alpha>p(X\mid\delta_{\mathrm{B}}(y))\geqslant\beta_2$，因为 $\beta_1\leqslant\beta_2$，故有 $\alpha>p(X\mid\delta_{\mathrm{B}}(y))\geqslant\beta_1$，推导出 $y\in\mathrm{BND}_{\mathrm{B}}^{(\alpha,\beta_1)}(X)$，因此，$\mathrm{BND}_{\mathrm{B}}^{(\alpha,\beta_2)}(X)\subseteq\mathrm{BND}_{\mathrm{B}}^{(\alpha,\beta_1)}(X)$。

定理 2 表示随着 α 值的增大，正域单调性减小，定理 3 表示随着 β 值的增大，正域单调性减小。这两个定理表明通过修改 (α,β) 的值能调整 X 的正域和边界域的大小。

2 基于三支决策邻域分类器的软件缺陷分类方法

Hu 等基于邻域粗糙集模型提出了一种二支决策邻域分类器，该分类器可以直接对连续型数据进行处理。该分类器在对对象 x 进行分类时，通常分配 $\delta(x)$ 中样本占多数的决策类标 D_+。二支决策分类器的优势是能够快速地对测试对象做出立即决策。然而，这种立即决策方式常常伴随更多的预测错误和更大的决策代价。假定存在一个模棱两可的对象，它被分到类别 D_+ 的概率为 51%，分到类别 D_- 的概率为 49%。按照多数准则将其划分到 D_+ 中时，则意味着其有 49%概率被分错。针对二支决策分类的劣势，三支决策方法则将模糊不清的对象划分到边界域中，作出延迟决策，等待专家的进一步处理。基于三支决策理论和邻域二支决策分类器，在邻域三支决策粗糙集模型下设计了一种三支决策邻域分类器用于软件缺陷分类，该分类器的目的是减少分类错误率。该分类算法具体思想如下：对于待分类的软件模块 x，计算其属于有缺陷趋势类别 D_+ 的概率 $p(D_+\mid\delta_B(x))$，通过判断该概率值与基于代价函数矩阵计算出的阈值对 (α,β) 之间的大小关系作出相应的三支决策，具体如下：

(P)若 $p(D_+\mid\delta_B(x))>\alpha$，则 x 属于 D_+；

(B)若 $\beta\leqslant p(D_+\mid\delta_B(x))\leqslant\alpha$，则 x 需进一步检查；

(N)若 $p(D_+\mid\delta_B(x))<\beta$，则 x 属于 D_-。

x 若为接受决策则将其划分到 D_+ 中，即为有缺陷趋势模块，x 若为拒绝决策则将其划分到 D_- 中，即为无缺陷趋势模块，x 若为延迟决策则需要等待做进一步检查。详细算法如下所示。

算法 1 Three-way decisions based neighborhood classifier (TDNEC)输入：Training set：$\langle U,C,D\rangle$；

Test object：x；

Parameter ω；

Cost matrix$\{\lambda_{ij}\}$.

输出：Class of x.

BEGIN

(1)Compute the decision threshold α,β based on $\{\lambda_{ij}\}$；//基于代价函数矩阵计算阈值对

(2)FOR each s in U
(3)Compute the distance $\Delta(x,s)$ between x and s with the used norm;//计算距离
(4)MIN=min($\Delta(x,s)$);
(5)MAX=max($\Delta(x,s)$);
(6)END FOR
(7) $\delta(x)$ =MIN+ ω ·(MAX−MIN); //计算其邻域
(8) $p = p(D_+ \mid \delta(x))$;
(9)IF $p > \alpha$
(10)Assign D_+ to test object and x is a defect-prone module;//将其判定为有缺陷模块
(11)ELSE IF $\beta \leqslant p \leqslant \alpha$
(12) x is in the boundary region of D_+ and x needs to further-examined;//等待进一步处理
(13)ELSE
(14) x is a non-defect-prone module; //将其判定为无缺陷模块
(15)END IF
END BEGIN

在该算法中,邻域粒子的大小由阈值 δ 决定,采用文献[12]中建议的方式,由待测对象 x 的局部和全局信息动态决定,则

$$\delta = \min(\Delta(s_i,x)) + \omega \cdot (\max(\Delta(s_i,x)) - \min(\Delta(s_i,x))) \qquad \omega \leqslant 1 \tag{14}$$

式中:$s_i(i=1,\ldots,n)$为训练对象集;$\min(\Delta(s_i,x))$和 $\max(\Delta(s_i,x)$分别表示 s_i 和测试对象 x 最小和最大距离值。

3 实验验证

本节通过实验来考察所提三支决策邻域分类器在软件缺陷分类任务上的性能,实验对比三支决策邻域分类器(TDNEC)、二支决策邻域分类器(NEC)[12],C4.5,k-NN 和 SVM 在分类准确率、F 值和误分类代价上的性能。

3.1 实验数据集和参数设置

表 2 的 11 个数据集均是 NASA 的实际项目,来自于公共的 PROMISE 库[15],包括了卫星飞行软件、卫星模拟器软件和地面站数据管理软件等。数据集覆盖了 3 种编程语言。在这些数据集中,缺陷模块的百分率分布为 3.0%～32.29%。,每个数据采样描述一个模块的属性是否是缺陷。模块的属性包括 McCabe 度量值、Halstead 度量值、操作符数和代码行数等。

表 2 NASA 数据集

Tab.2 NASA data sets

数据集名称	来源	描述	语言	样本	属性	缺陷/%
CM1	NASA	航天器装置	C	498	22	9.830
JM1	NASA	卫星模拟器	C	7 782	22	21.400
KC2	NASA	地面数据存储管理	C++	522	21	20.490
KC3	NASA	地面数据存储管理	JAVA	458	39	9.380
MC2	NASA	电视制导系统	C++	61	39	32.290
MW1	NASA	零重力下的燃烧实验	C++	403	37	7.690
PC1	NASA	绕地球轨道的卫星飞行软件	C++	1 109	21	6.940
PC2	NASA	绕地球轨道卫星飞行软件	C++	745	37	2.147
PC3	NASA	绕地球轨道卫星飞行软件	C++	1 077	38	12.400
PC4	NASA	绕地球轨道卫星飞行软件	C++	1 458	38	12.200
PC5	NASA	绕地球轨道卫星飞行软件	C++	17 186	39	3.000

实验的相关参数设置如表 3 所示。由于采用 10 倍交叉验证,在试验结果中仅给出平均结果。对于每个数据集,随机产生 10 组不同的代价函数,即对每个分类任务运行 10 次 10 倍交叉验证。ω 值参考文献[12]的设置,ω 值介于 0 和 0.1。

表 3 实验各参数设置

Tab. 3 Parameter setting

参数	值
Platform	Eclipse with WEKA(version 3.5)
$\{\lambda_{ij}\}$	10 groups, generated randomly
Distance	Euclidean distance Δ_2
ω	(0,0.1]
C4.5,k-NN, SVM	default values in WEKA
cross validation	10-folds

3.2 评价标准

令 N_{PP} 表示分类器将实际为有缺陷的模块判定正确的个数,N_{NN} 表示分类器将实际为无缺陷的模块判定正确的个数,N_{PN} 表示将无缺陷的模块判定为有缺陷模块的个数,N_{NP} 表示将有缺陷的模块判定为无缺陷模块的个数,则分类正确率定义为

$$\text{accuracy}=\frac{N_{PP}+N_{NN}}{N_{PP}+N_{NP}+N_{NN}+N_{PN}} \tag{15}$$

覆盖率定义为

$$\text{coverage}=\frac{N_{PP}+N_{NP}+N_{NN}+N_{PN}}{N_{PP}+N_{NP}+N_{BP}+N_{BN}+N_{NN}+N_{PN}} \tag{16}$$

TDNEC 基于三支决策,一些对象会被分类到边界域中,因此,TDNEC 覆盖率的值常常小于 1。在大多数情况下,分类正确率和覆盖率是一种 tradeoff 的关系,常用 F 值来表示分类器的折衷性能。基于分类正确率和覆盖率的 F 值定义为

$$F=2\cdot\frac{\text{accuracy}\cdot\text{coverage}}{\text{accuracy}+\text{coverage}} \tag{17}$$

基于表 1 给定的代价函数,误分类代价定义为

$$\text{cost}=\lambda_{PN}\cdot N_{PN}+\lambda_{NP}\cdot N_{NP}+\lambda_{BN}\cdot N_{BN}+\lambda_{BP}\cdot N_{BP} \tag{18}$$

3.3 实验结果及分析

表 4~6 为三支决策邻域分类器,NEC,C4.5,k-NN 和 SVM 五种分类器在分类正确率、F 值和误分类代价上的性能对比结果。对于分类正确率,TDNEC 在 5 个数据集上表现优于其他算法,SVM 表现其次,4 个最优,C4.5 有 2 个最优。TDNEC 由于采用三支决策方法,对于模棱两可的对象都将被延迟做进一步的检验,这在理论上保证了三支决策方法具有较高的分类精度。对于 F 值,SVM 最好,有 7 个最优,C4.5 其次,有 2 个最优,NEC 有 1 个最优。对于 TDNEC 的表现,这是可预期的,因为 F 值是由分类正确率和覆盖度共同决定的,虽然 TDNEC 能够取得较好的分类正确率,TDNEC 的覆盖度小于 1,而其他算法的覆盖度都等于 1,由此决定了 TNDEC 在一定程度上无法取得较高的 F 值。对于误分类代价,TDNEC 有 5 个数据集上最优,SVM 表现其次,4 个最优,C4.5 有 2 个最优。从理论上分析,TDNEC 基于最小化贝叶斯决策代价理论,这保证了 TDNEC 能够得到较小的误分类代价。

表 4 各分类器在航天软件缺陷数据上分类正确率的对比结果(加粗表示最好的值)

Tab. 4 Comparison results of classification accuracy on NASA datasets (The best values in bold)

Data set	TDNEC	NEC	C4.5	k-NN	SVM
cm1	**0.8839±0.0095**	0.8680±0.0020	0.8308±0.0106	0.7846±0.0078	0.8779±0.0000
jm1	**0.8071±0.0121**	0.7726±0.0019	0.7842±0.0024	0.7179±0.0013	0.7863±0.0002
kc2	0.8159±0.0024	0.8126±0.0018	**0.8303±0.0066**	0.8080±0.0064	0.8255±0.0035
kc3	0.7598±0.0142	0.7428±0.0134	0.7959±0.0106	0.7500±0.0095	**0.8196±0.0024**
mc2	**0.7145±0.0229**	0.6808±0.0218	0.6264±0.0173	0.6976±0.0155	0.7024±0.0216
mw1	0.8259±0.0097	0.8241±0.0062	0.8727±0.0105	0.8178±0.0101	**0.8929±0.0012**
pc1	**0.9257±0.0082**	0.9129±0.0014	0.9018±0.0063	0.8970±0.0037	0.9122±0.0004
pc2	0.9774±0.0005	0.9769±0.0011	0.9785±0.0000	0.9568±0.0038	**0.9785±0.0000**
pc3	0.8547±0.0049	0.8282±0.0052	0.8546±0.0098	0.8494±0.0023	**0.8756±0.0000**
pc4	**0.9044±0.0038**	0.8909±0.0036	0.8962±0.0044	0.8740±0.0035	0.8909±0.0007

表 5 各分类器在航天软件缺陷数据上 F 值的对比结果(加粗表示最好的值)

Tab. 5 Comparison results of F-measure on NASA datasets (The best values in bold)

Data set	TDNEC	NEC	C4.5	k-NN	SVM
cm1	0.9114±0.0221	0.9293±0.0012	0.9076±0.0063	0.8793±0.0049	**0.9350±0.0000**
jm1	0.7725±0.1067	0.8717±0.0012	0.8791±0.0015	0.8358±0.0009	**0.8804±0.0004**
kc2	0.8917±0.0046	0.8966±0.0011	0.9073±0.0040	0.8938±0.0039	0.9044±0.0021
kc3	0.8578±0.0090	0.8523±0.0088	0.8863±0.0066	0.8571±0.0062	**0.9008±0.0015**
mc2	0.8025±0.0147	0.8099±0.0155	0.7702±0.0131	0.8218±0.0107	**0.8250±0.0150**
mw1	0.9014±0.0061	0.9036±0.0038	0.9320±0.0060	0.8997±0.0061	**0.9434±0.0007**
pc1	0.9445±0.0128	**0.9545±0.0007**	0.9484±0.0035	0.9457±0.0021	0.9541±0.0002
pc2	0.9870±0.0024	0.9883±0.0005	0.9891±0.0000	0.9779±0.0020	**0.9891±0.0000**
pc3	0.8831±0.0098	0.9060±0.0031	0.9216±0.0058	0.9186±0.0013	**0.9337±0.000**
pc4	0.9248±0.0051	0.9423±0.0020	**0.9453±0.0025**	0.9328±0.0020	0.9423±0.0004

表 6 各分类器在航天软件缺陷数据上误分类代价的对比结果(加粗表示最好的值)

Tab. 6 Comparison results of misclassification cost on NASA datasets (The best values in bold)

Data set	TDNEC	NEC	C4.5	k-NN	SVM
cm1	18.011±4.948	19.402±5.562	25.030±7.819	31.790±9.590	**17.942±5.112**
jm1	**713.577±198.062**	992.259±307.421	941.308±291.721	1231.147±381.403	932.759±289.645
kc2	38.111±18.650	38.876±20.222	**34.992±17.930**	39.772±20.149	35.981±17.878
kc3	12.423±8.594	13.293±9.371	10.485±7.427	12.800±8.934	**9.277±6.478**
mc2	**17.079±8.833**	20.350±11.075	23.335±11.519	19.185±10.264	18.945±10.120
mw1	14.694±8.692	14.561±8.305	10.715±6.610	15.422±8.865	**8.966±5.274**
pc1	**24.339±12.295**	27.780±13.477	31.453±16.306	33.032±16.178	28.078±13.730
pc2	**7.142±4.787**	7.966±5.381	7.308±4.803	14.871±9.970	7.308±4.803
pc3	67.272±38.506	78.488±49.170	66.769±41.460	69.331±44.307	**57.301±36.803**
pc4	**57.888±30.353**	64.680±34.600	62.529±35.238	75.015±40.042	64.928±34.610

4 结束语

本文针对软件缺陷数据具有代价敏感特性且属性值为连续型数据等特点，提出了一种基于邻域三支决策粗糙集模型的软件缺陷预测方法。在邻域三支决策粗糙集模型中，既可以通过代价函数矩阵求出三支决策所需的阈值，又能通过邻域系统来表示和计算连续型数据。基于该模型提出的三支决策分类器对于具有较高确信度和较低确信度的待测软件模块作出明确的接受和拒绝决策，而对于模糊不清的待测软件模块则做出延迟决策，交由专家进一步处理。在 NASA 数据集上的实验结果表明，三支决策邻域分类器在大部分数据集上能够取得较高的分类正确率和较低的误分类代价。

参考文献：

[1] Huai J P. Views about future networked software technologies[J]. Communications of the CCF, 2008, 4(1):19-26.

[2] President's information technology advisory committee. Computational Science: Ensuring America's competitiveness[R]. Washington: PITAC, 2005.

[3] Ramler R, Wolfmaier K. Economic perspectives in test automation: Balancing automated and manual testing with opportunity cost[C]//Proceedings of the 2006 International Workshop on Automation of Software Test. New York, USA: ACM, 2006: 85-91.

[4] 王青，伍书剑，李明树. 软件缺陷预测技术[J]. 软件学报，2008，19(7)：1565-1580.
Wang Qing, Wu Shujian, Li Mingshu. Software defect prediction[J]. Journal of Software, 2008, 19(7): 1565-1580.

[5] Lessmann S, Baesens B, Mues C, et al. Benchmarking classification models for software defect prediction: A proposed framework and novel findings[J]. Software Engineering, IEEE Transactions on, 2008, 34(4):485-496.

[6] 黎铭，霍轩. 半监督软件缺陷挖掘研究综述[J]. 数据采集与处理，2016，31(1)：56-64.
Li Ming, Huo Xuan. Software defect mining based on semi-supervised learning [J]. Journal of Data Acquisition and Processing, 2016, 31(1): 56-64.

[7] Zheng J. Cost-sensitive boosting neural networks for software defect prediction [J]. Expert Systems with Applications, 2010, 37(6): 4537-4543.

[8] Arar O F, Ayan K. Software defect prediction using cost-sensitive neural network[J]. Applied Soft Computing, 2015, 33: 263-277.

[9] Liu M X, Miao L S, Zhang D Q. Two-stage cost-sensitive learning for software defect prediction[J]. IEEE Transactions on Reliability, 2014, 63(2): 676-686.

[10] Li W W, Huang Z Q, Li Q. Three-way decisions based software defect prediction[J]. Knowledge-Based Systems, 2016, 91: 263-274.

[11] Yao Y Y. The superiority of three-way decisions in probabilistic rough set models[J]. Information Sciences, 2011, 181(6): 1080-1096.

[12] Hu Q H, Yu D R, Xie Z X. Neighborhood classifiers[J]. Expert Systems with Applications: An International Journal, 2008, 34: 866-876.

[13] 杨霁琳，张贤勇，唐孝. 基于三支决策的模糊信息系统 OWA 算子参数选择[J]. 数据采集与处理，2016，31(6)：1156-1163.
Yang Jilin, Zhang Xianyong, Tang Xiao. Three-way decisions based parameter selection of OWA operations in fuzzy information system[J]. Journal of Data Acquisition and Processing, 2016, 31(6): 1156-1163.

[14] Jia X Y, Liao W H, Tang Z M, et al. Minimum cost attribute reduction in decision-theoretic rough set models[J]. Information Sciences, 2013, 219: 151-167.

[15] Sayyad S J, Menzies T J. The PROMISE repository of software engineering databases[EB/OL]. University of Ottawa, Canada, http://promise.site.uottawa.ca/SERepository, 2006-01-01/2017-01-10.

作者简介：

李伟漳(1981-)，女，助理研究员，研究方向：软件挖掘、机器学习，E-mail：liweiwei@nuaa.edu.cn。

郭鸿昌(1979-)，男，工程师，研究方向：软件工程、机器学习。