基于二次学习的半监督字典学习软件缺陷预测*

张志武[1] 荆晓远[1,2] 吴 飞[3]

[1](南京邮电大学 计算机学院 南京 210023)
[2](武汉大学 软件工程国家重点实验室 武汉 430072)
[3](南京邮电大学 自动化学院 南京 210023)

摘 要 当软件历史仓库中有标记训练样本较少时,有效的预测模型难以构建.针对此问题,文中提出基于二次学习的半监督字典学习软件缺陷预测方法.在第一阶段的学习中,利用稀疏表示分类器将大量无标记样本通过概率软标记标注扩充至有标记训练样本集中.再在扩充后的训练集上进行第二阶段的鉴别字典学习,最后在学得的字典上预测缺陷倾向性.在 NASA MDP 和 PROMISE AR 数据集上的实验验证文中方法的优越性.

关键词 软件缺陷预测,二次学习,半监督学习,字典学习
中图法分类号 TP 311 **DOI** 10.16451/j.cnki.issn1003-6059.201703006
引用格式 张志武,荆晓远,吴 飞.基于二次学习的半监督字典学习软件缺陷预测.模式识别与人工智能,2017,30(3):242-250.

Twice Learning Based Semi-supervised Dictionary Learning for Software Defect Prediction

ZHANG Zhiwu[1], JING Xiaoyuan[1,2], WU Fei[3]
[1](*School of Computer*, *Nanjing University of Posts and Telecommunications*, *Nanjing* 210023)
[2](*State Key Laboratory of Software Engineering*, *Wuhan University*, *Wuhan* 430072)
[3](*School of Automation*, *Nanjing University of Posts and Telecommunications*, *Nanjing* 210023)

ABSTRACT

When the previous defect labels of modules in software history warehouse are limited, building an effective prediction model becomes a challenging problem. Aiming at this problem, a twice learning based semi-supervised learning algorithm for software defect prediction is proposed. In the first stage of learning, a large number of unlabeled samples are labeled with probability soft labels and extended to the labeled training dataset by using sparse representation classifier. Then, on this dataset discriminative dictionary learning is used for the second stage of learning. Finally, defect proneness prediction is conducted on the obtained dictionary. Experiments on the widely used NASA MDP and PROMISE AR datasets indicate the superiority of the proposed algorithm.

* 国家自然科学基金项目(No.61272273,61073113)、江苏省普通高校研究生科研创新计划项目(No.CXZZ12_0478)资助
Supported by National Natural Science Foundation of China(No.61272273,61073113), Graduate Student Innovation Research Project of Jiangsu Province(No.CXZZ12_0478)
收稿日期:2016-07-28;**录用日期**:2016-09-28
Manuscript received July 28, 2016; accepted September 28, 2016

Key Words　Software Defect Prediction, Twice Learning, Semi-supervised Learning, Dictionary Learning

Citation　ZHANG Z W, JING X Y, WU F. Twice Learning Based Semi-supervised Dictionary Learning for Software Defect Prediction. Pattern Recognition and Artificial Intelligence, 2017, 30(3): 242-250.

软件缺陷预测是软件工程领域中的一个重要研究课题,它能减轻软件代码审查或测试的负担[1-3].为了确保软件系统的稳定性,应该在软件版本发布之前尽可能多地检测出软件缺陷.然而,由于项目进度的影响和软件系统的复杂性,人们无法对每个软件模块在所有可能的条件下广泛测试.因此,准确预测软件模块的缺陷倾向性能帮助项目管理者合理分配有限的组织资源,降低软件开发成本,提高软件系统的质量.

随着机器学习和数据挖掘理论与方法不断应用于软件缺陷预测领域,软件缺陷预测已从早期的经验公式估算发展到对缺陷倾向性和分布的预测.很多机器学习技术假定有足够的软件缺陷数据用于建立预测模型,有监督的分类技术在构建预测模型时利用软件度量信息和缺陷标记信息.然而在某些情况下,由于标记软件缺陷模块需要耗费大量的人力和时间,因此,不易获得足够的有标记软件缺陷模块用于建立准确的预测模型.同时,大量无标记的软件缺陷模块却较易收集.半监督学习即是应用于这种情形下的重要研究方法.

近些年,多种不同类型的半监督学习技术已经应用于软件缺陷预测领域.Seliya 等[4-5] 提出基于期望最大化的半监督学习算法和半监督聚类方法.Catal 等[6] 利用朴素贝叶斯算法建立半监督缺陷预测模型.Jiang 等[7] 提出欠采样随机委员会方法.文献[8] 和文献[9] 提出主动半监督学习方法.Catal[10] 评估软件缺陷预测中的 4 种半监督学习方法:低密度分离、期望最大化、支持向量机和类质量归一化.Ma 等[11] 提出随机下采样的协同训练方法.Abaei 等[12] 提出半监督混合自组织映射模型.Zhang 等[13] 提出基于非负稀疏图的标签扩散方法.

在基于软件度量元的缺陷预测中,由于软件模块间的相似性,一个软件模块的度量值可由部分其它模块的度量值近似表示,而且这种表示通常是稀疏的.字典学习作为一种有效的特征学习技术,结合稀疏表示分类器能取得较好的分类效果.Jing 等[14] 已将字典学习技术成功应用于有监督情况下的软件缺陷预测,并取得较好的预测结果.如何在半监督环境下应用字典学习技术提升机器学习缺陷预测性能是一个值得研究的内容.本文将二次学习方法引入半监督缺陷预测中,提出基于二次学习的半监督字典学习软件缺陷预测方法(Twice Learning Based Semi-supervised Dictionary Learning, TLSDL),利用类别隶属度生成无标记训练样本的虚拟概率软标记.预测方法串行执行 2 个学习阶段:1)利用稀疏表示分类器计算类别隶属度,为无标记训练样本生成虚拟概率软标记,扩充有标记训练样本集.2)在扩充后的训练集上进行鉴别字典学习,在得到的类特定的鉴别字典上进行缺陷倾向性预测.在二次学习中,第一阶段学习可以充分利用无标记缺陷模块的信息,解决有标记训练样本不足的问题,实现半监督学习的扩展.第二阶段学习通过鉴别字典特征学习技术得到有利于分类的类特定字典原子,实现稀疏表示分类和预测性能的提升.

1　基于二次学习的半监督字典学习软件缺陷预测方法

1.1　问题定义

在静态软件缺陷预测中,假设 $X_i(X_i \in \mathbf{R}^d)$ 表示软件模块样本,为静态代码度量值构成的列向量.

$$X = [x_1, x_2, \cdots, x_n]$$

表示软件缺陷预测数据集.X 为一个 $d \times n$ 矩阵,d 为软件静态代码度量属性的个数,n 为软件缺陷预测数据集中软件模块数目,其中一定数量的软件模块带有缺陷标记(有缺陷和无缺陷),剩余部分未标记.l 为有标记数据集 X_l 中样本的数目,u 为无标记数据集 X_u 中样本的数目.

$$X = [X_l, X_u],\ l + u = n.$$

$$X_l = [x_1, x_2, \cdots, x_l],$$

$$X_u = [x_{l+1}, x_{l+2}, \cdots, x_{l+u}].$$

假设 $Y = \{Y_l, Y_u\}$ 为相应的缺陷标记,其中

$$Y_l = \{y_1, y_2, \cdots, y_l\}$$

已知,

$$Y_u = \{y_{l+1}, y_{l+2}, \cdots, y_{l+u}\}$$

未知.Y 中的标记 $y_i \in \{0,1\}$ 为二值变量,0 表示无缺陷标记,1 表示有缺陷标记.本文目标是将有监督的缺陷预测方法扩展到半监督环境中,利用大量的无标记样本 X_u 辅助少量有标记样本 X_l,建立准确的缺陷预测模型.

1.2 二次学习半监督缺陷预测模型

在静态软件缺陷预测中,字典学习技术已成功应用于有监督环境下的软件缺陷预测,并取得较好的预测结果.然而在半监督背景下不能直接使用鉴别字典学习技术,针对该问题,本文提出二次学习框架下的半监督缺陷预测模型.

二次学习是 Zhou 等[15]提出的用于分类问题的学习框架,将学习过程分成 2 个串行的学习阶段:1)训练一个泛化能力强的学习器,并利用该学习器生成一定数目的虚拟样本,这些虚拟样本能缓解训练数据不足对模型训练造成的影响;2)基于第一阶段生成的虚拟样本训练可理解性好的预测模型.Jiang 等[16]将二次学习用于软件重用问题.杨子旭等[17]将二次回归学习用于软件开发工作量预测.

受二次学习框架及其成功应用的启发,本文将二次学习框架扩充到半监督背景下的软件缺陷预测中,提出基于二次学习的半监督字典学习软件缺陷预测方法(TLSDL),具体方法如下.

第一阶段利用有标记训练样本集作为稀疏表示分类器(Sparse Representation Classifier, SRC)的预定义字典,无标记训练样本在 SRC 上进行概率软标记预测,原始无标记训练样本连同其概率软标记同时扩充至有标记训练样本集中,形成新的有标记训练样本集.

第二阶段是在新的扩充后的有标记训练样本集上,利用字典学习模型,学习得到一个有鉴别能力的类特定字典.在该字典上,利用 SRC 分类器预测新的测试模块并进行缺陷倾向性预测.本文方法利用二次学习和字典学习实现半监督场景下的软件缺陷预测,流程框架如图 1 所示.

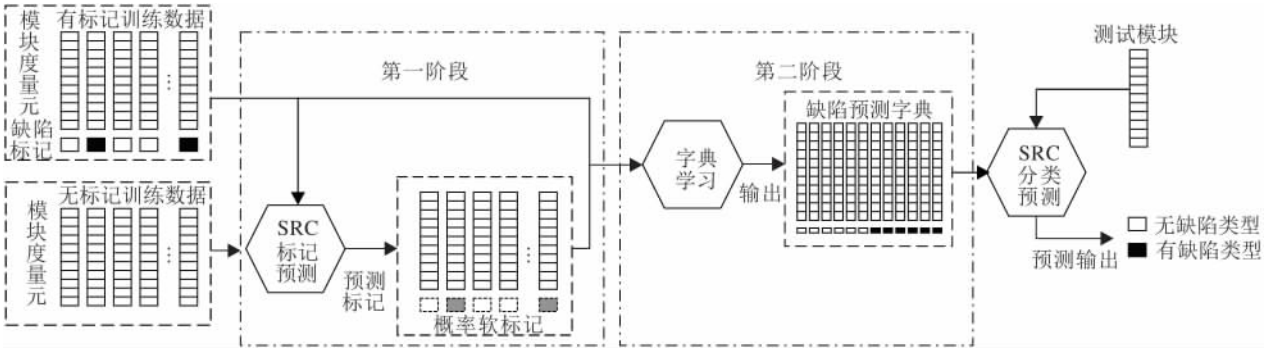


图 1 基于二次学习的半监督字典学习软件缺陷预测模型流程图

Fig.1 Flowchart of twice learning based semi-supervised dictionary learning model for software defect prediction

1.3 无标记样本的概率软标记扩充

近几年,SRC 由于其良好的性能而被广泛研究与应用[18].SRC 的目的在于寻求使用原始训练样本的线性组合对测试样本的最稀疏的表示.给定含有 c 类的训练样本集

$$\boldsymbol{A} = [\boldsymbol{A}_1, \cdots, \boldsymbol{A}_i, \cdots, \boldsymbol{A}_c] \in \mathbf{R}^{m \times n},$$

其中

$$\boldsymbol{A}_i = [\boldsymbol{A}_{i,1}, \boldsymbol{A}_{i,2}, \cdots, \boldsymbol{A}_{i,n_i}] \in \mathbf{R}^{m \times n_i}$$

为来自第 i 类的训练样本子集,z 为测试样本.稀疏表示问题表示如下:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \{ \| z - \boldsymbol{A}\boldsymbol{\alpha} \|_2^2 + \gamma \| \boldsymbol{\alpha} \|_1 \}, \quad (1)$$

其中,$\boldsymbol{\alpha}$ 为编码系数,通过 z 在 $\boldsymbol{A}$ 上的 l_1-norm 最小化求得.求得稀疏编码系数后,SRC 利用不同类的重构误差分类预测测试样本.具体分类如下:

$$identity(z) = \arg\min_i \{e_i\} = \arg\min_i \left\{ \left\| z - \boldsymbol{A}_i \hat{\boldsymbol{\alpha}}_i \right\|_2 \right\},$$

其中,$\hat{\boldsymbol{\alpha}}_i$ 为第 i 类对应的系数向量,最后测试样本 z 被指派到最小重构误差 e_i 对应的类别.

由于 SRC 泛化能力较好,本文利用 SRC 标记预测训练数据集中的无标记样本.针对训练集中的无标记样本 x_i,利用每个子类的训练样本的表示能力计算 x_i 对应于该类的隶属度,把这个隶属度作为 x_i 指派到该类的概率软标记.对于无标记样本 x_i 和类别 j,概率软标记 $p(x_i, j)$ 定义为

$$p(x_i, j) = \frac{1}{e_{ij}} \left(\sum_{j=1}^{c} \frac{1}{e_{ij}} \right)^{-1}, \quad (2)$$

其中

$$e_{ij} = \| x_i - \boldsymbol{A}_j \alpha_{ij} \|_2^2,$$

$$\sum_{j=1}^{c} p(x_i, j) = 1,\ 0 \leqslant p(x_i, j) \leqslant 1.$$

这种概率软标记定义扩充至整个训练样本集中的任

意 x_i,可以定义如下:

$$p(x_i,j)=\begin{cases}1, & x_i\text{有标记},\ j=y_i\\ 0, & x_i\text{有标记},\ j\neq y_i\\ \dfrac{1}{e_{ij}}\left(\sum\limits_{j=1}^{c}\dfrac{1}{e_{ij}}\right)^{-1}, & x_i\text{无标记}\end{cases}\tag{3}$$

于是整个训练样本的隶属度矩阵为

$$\boldsymbol{P}=\begin{bmatrix}P_{1,1} & P_{1,2} & \cdots & P_{1,c}\\ P_{2,1} & P_{2,2} & \cdots & P_{2,c}\\ \vdots & \vdots & & \vdots\\ P_{c,1} & P_{c,2} & \cdots & P_{c,c}\\ P_{c+1,1} & P_{c+1,2} & \cdots & P_{c+1,c}\end{bmatrix},$$

其中 $P_{c+1,j}$ 为无标记样本在第 j 类上的隶属度.无标记样本连同其概率软标记一起加入到有标记训练样本集中进行下一阶段的学习,解决半监督情况下有标记训练样本数量不足的问题.

1.4　类特定鉴别字典学习

在第一学习阶段,利用泛化能力强的 SRC 得到无标记样本的概率软标记,这个标记是本文生成的虚拟标记,为了训练一个可理解性好的预测模型,在第二学习阶段,利用类特定的字典学习方法求得有鉴别能力的字典,使用这种鉴别字典取代由原始样本构成的预定义字典,提升 SRC 的分类预测能力.

假设

$$\boldsymbol{A}=[\boldsymbol{A}_1\quad \boldsymbol{A}_2]$$

表示有监督环境下软件缺陷预测的训练模块集合,$\boldsymbol{A}_i(i=1,2)$ 表示来自第 i 类的训练模块子集.类特定的鉴别字典学习方法[19-20]从有缺陷和无缺陷软件模块中学得一个结构化的字典

$$\boldsymbol{D}=[\boldsymbol{D}_1\quad \boldsymbol{D}_2],$$

求解问题如下:

$$J_{(\boldsymbol{D},\boldsymbol{\alpha})}=\arg\min_{(\boldsymbol{D},\boldsymbol{\alpha})}\{r(\boldsymbol{A},\boldsymbol{D},\boldsymbol{\alpha})+\lambda\|\boldsymbol{\alpha}\|_1\},$$

其中,$r(\boldsymbol{A},\boldsymbol{D},\boldsymbol{\alpha})$ 为鉴别项,$\|\boldsymbol{\alpha}\|_1$ 为稀疏限制,λ 为平衡因子.针对缺陷预测这种两类分类问题,鉴别项可表示为

$$r(\boldsymbol{A},\boldsymbol{D},\boldsymbol{\alpha})=\sum_{i=1}^{2}\left[\|\boldsymbol{A}_i-\boldsymbol{D}\boldsymbol{\alpha}_i\|_{\mathrm{F}}^2+\|\boldsymbol{A}_i-\boldsymbol{D}_i\boldsymbol{\alpha}_i^i\|_{\mathrm{F}}^2+\sum_{\substack{j=1\\ j\neq i}}^{2}\|\boldsymbol{D}_j\alpha_i^j\|_{\mathrm{F}}^2\right].$$

在半监督学习环境下,由于第一学习阶段已将无标记训练样本根据式(3)打上概率软标记,扩充后的训练样本集可表示为

$$\boldsymbol{A}=[\boldsymbol{A}_1\quad \boldsymbol{A}_2\quad \boldsymbol{A}_3],$$

把概率软标记引入到字典学习中,整个 TLSDL 的目标函数表示为

$$J_{(\boldsymbol{D},\boldsymbol{\alpha})}=\arg\min_{(\boldsymbol{D},\boldsymbol{\alpha})}\left\{\sum_{i=1}^{3}\sum_{j=1}^{2}P_{ij}\left(\|\boldsymbol{A}_i-\boldsymbol{D}\boldsymbol{\alpha}_j\|_{\mathrm{F}}^2+\|\boldsymbol{A}_i-\boldsymbol{D}_j\boldsymbol{\alpha}_i^j\|_{\mathrm{F}}^2+\sum_{\substack{k=1\\ k\neq j}}^{2}\|\boldsymbol{D}_k\boldsymbol{\alpha}_j^k\|_{\mathrm{F}}^2\right)+\lambda\|\boldsymbol{\alpha}\|_1\right\}.\tag{4}$$

上式的优化求解可分解成 2 个子问题:固定 $\boldsymbol{D}$ 更新 $\boldsymbol{\alpha}$ 和固定 $\boldsymbol{\alpha}$ 更新 $\boldsymbol{D}$.优化程序是迭代实现鉴别字典 $\boldsymbol{D}$ 和相应系数矩阵 $\boldsymbol{\alpha}$ 的求解.

首先,假设 $\boldsymbol{D}$ 固定,式(4)中的目标函数简化成一个稀疏编码问题求解

$$\boldsymbol{\alpha}=[\boldsymbol{\alpha}_1\quad \boldsymbol{\alpha}_2],$$

逐个求解 $\boldsymbol{\alpha}_1$ 和 $\boldsymbol{\alpha}_2$,当求解其中一个时,固定另外一个.因此,式(4)可重写为

$$J_{(\boldsymbol{\alpha}_j)}=\arg\min_{(\boldsymbol{\alpha}_j)}\left\{\sum_{i=1}^{3}P_{ij}\left(\|\boldsymbol{A}_i-\boldsymbol{D}\boldsymbol{\alpha}_j\|_{\mathrm{F}}^2+\|\boldsymbol{A}_i-\boldsymbol{D}_j\boldsymbol{\alpha}_i^j\|_{\mathrm{F}}^2+\sum_{\substack{k=1\\ k\neq j}}^{2}\|\boldsymbol{D}_k\boldsymbol{\alpha}_j^k\|_{\mathrm{F}}^2\right)+\lambda\|\boldsymbol{\alpha}_j\|_1\right\}.\tag{5}$$

上式可根据文献[21]提供的迭代投影方法(Iterative Projection Method, IPM)求解.

当 $\boldsymbol{\alpha}$ 固定时,逐个求解 $\boldsymbol{D}_1$ 和 $\boldsymbol{D}_2$,当求解其中一个时,固定另外一个.因此,式(4)可以重写为

$$J_{(\boldsymbol{D}_j)}=\arg\min_{(\boldsymbol{D}_j)}\left\{\sum_{i=1}^{3}P_{ij}\left(\left\|\boldsymbol{A}-\boldsymbol{D}_j\boldsymbol{\alpha}^j-\sum_{\substack{k=1\\ k\neq j}}^{2}\boldsymbol{D}_k\boldsymbol{\alpha}^k\right\|_{\mathrm{F}}^2+\|\boldsymbol{A}_i-\boldsymbol{D}_j\boldsymbol{\alpha}_i^j\|_{\mathrm{F}}^2+\sum_{\substack{k=1\\ k\neq j}}^{2}\|\boldsymbol{D}_k\boldsymbol{\alpha}_j^k\|_{\mathrm{F}}^2\right)\right\}.\tag{6}$$

上式为一个二次规划问题,可使用文献[20]中的算法求解.

最后,测试样本 z 在求得的字典 $\boldsymbol{D}$ 上利用 SRC 进行稀疏编码和分类预测,SRC 利用

$$identity(z)=\arg\min_j\{\boldsymbol{e}_j\mid j=1,2\}=\arg\min_j\left\{\left\|z-\boldsymbol{A}_j\hat{\boldsymbol{\alpha}}_j\right\|_2\,\middle|\,j=1,2\right\}\tag{7}$$

将测试样本 z 指派到最小重构误差 $\boldsymbol{e}_j$ 对应的类别中.

TLSDL 的实现步骤如下.

算法　TLSDL

输入　软件缺陷数据集 $X=\{x_1,\cdots,x_l,x_{l+1},\cdots,x_n\}$,测试样本 z,

有标记训练样本集 $\boldsymbol{X}_l = \{\boldsymbol{x}_i\}_{i=1}^{l}$,

无标记训练样本集 $\boldsymbol{X}_u = \{\boldsymbol{x}_i\}_{i=l+1}^{n}$

输出　字典 $\boldsymbol{D}$ 和测试样本 z 的预测标记 y_z

step 1　利用 SRC 分类器对无标记样本进行第一阶段学习,将有标记样本集作为预定义字典,根据式(2)计算无标记样本对应于每个类的隶属度 $p(\boldsymbol{x}_n, j)$.

step 2　将无标记训练样本数据扩充至有标记训练集中,根据式(3)计算所有训练样本的概率软标记.

step 3　在扩充后的有标记训练样本集上进行第二阶段学习,使用有标记数据初始化每个类的子字典 $\boldsymbol{D}_j$.

step 4　固定字典 $\boldsymbol{D}$,使用式(5)更新编码系数 $\boldsymbol{\alpha}_j$.

step 5　固定编码系数 $\boldsymbol{\alpha}$,使用式(6)更新字典 $\boldsymbol{D}_j$.

step 6　返回 step 4,直到 $J_{(\boldsymbol{D},\boldsymbol{\alpha})}$ 在相邻迭代中足够接近或达到最大迭代次数.

step 7　输出字典 $\boldsymbol{D}$,对于测试样本 z,利用式(7)预测标记 y_z.

1.5　算法复杂度分析

TLSDL 的时间复杂度主要包含 3 部分:第一阶段学习中的概率软标记计算时,利用式(1)求解稀疏编码问题;第二阶段学习中式(5)的编码系数优化问题;式(6)中的字典优化问题.具体地,第一阶段学习中的稀疏编码问题近似时间复杂度为 $mO(q^2l^{\varepsilon})$ [23],$\varepsilon \geqslant 1.2$ 为一个常量,q 为软件缺陷样本度量元个数,l 为整个有标记训练样本个数,m 为训练数据集中无标记模块的数量.第二阶段学习中编码系数优化问题也是一个稀疏编码问题,它的近似时间复杂度为 $nO(q^2r^{\varepsilon})$,r 为整个字典 D 中的字典原子数目,n 为整个训练数据集中模块的数量.第二阶段学习中字典原子更新的时间复杂度为 $\sum_j r_j O(2nq)$,r_j 为子字典 D_j 中字典原子的数目.因此,整个 TLSDL 的时间复杂度为

$$mO(q^2l^{\varepsilon}) + k\left[nO(q^2r^{\varepsilon}) + \sum_j r_j O(2nq)\right],$$

k 为第二阶段学习中的迭代总次数.

2　实验及结果分析

本节通过实验验证 TLSDL 的性能,并与其它半监督缺陷预测方法进行对比分析.

2.1　实验数据库

实验采用软件缺陷预测研究中广泛使用的开放数据集,包括 NASA Metrics Data Program (MDP)和 PROMISE (AR).

MDP 数据库是美国航空航天局提供的开放软件缺陷数据库,本文从该仓库中选取 5 个项目库作为实验数据.每个项目代表一个采用 C 语言或 Java 语言编程的 NASA MDP 软件系统或子系统,包含相应的缺陷标记数据和多种静态代码度量.仓库使用一个缺陷跟踪系统记录每个软件模块的缺陷数目.NASA MDP 仓库的静态代码度量都是依照与软件质量密切相关的 Halsted、LOC、McCabe 等软件规模及复杂度度量方法生成.

表 1 给出 5 个所选数据集的具体描述信息.根据文献 [23] 和文献 [24],清洗实验数据集,移除其中重复的数据和矛盾数据.

PROMISE code repository 数据库是美国西弗吉尼亚大学 PROMISE 网站提供的软件工程领域常用开放数据库.其中 AR 系列数据集来自于土耳其的一家白色家电制造商使用 C 语言开发的嵌入式软件系统.

表 1 同时也给出 5 个 AR 数据集的描述信息,这些描述属性也是由 Halsted、LOC、McCabe 等软件度量方法产生.

表 1　实验数据集信息

Table 1　Information of experimental datasets

项目	度量个数	缺陷模块数	模块总数	缺陷率/%
JM1	22	1672	7755	21.56
KC1	22	314	1192	26.34
PC3	38	134	1073	12.49
PC4	38	177	1287	13.75
PC5	39	471	1691	27.85
AR1	29	9	121	7.44
AR3	29	8	63	12.70
AR4	29	20	107	18.69
AR5	29	8	36	22.22
AR6	29	15	101	14.85

2.2　评估度量

实验采用 4 个广泛使用的评估度量:召回率(Probability of Detection, PD)、误报率(Probability of False Alarm, PF)、F 值和受试者工作特征曲线下的面积(Area Under Receiver Operating Characteristic

Curve，AUC)值.这 4 个度量可使用 4 个二元分类结果正确性统计方式定义,包括正确分类的肯定实例数(True Positive，TP)、正确分类的否定实例数(True Negative，TN)、错误分类为肯定实例的否定实例数(False Positive，FP)、错误分类为否定实例的肯定实例数(False Negative，FN).

召回率表示正确分类的有缺陷模块数占总的有缺陷模块数的比率,定义为

$$PD=\frac{TP}{TP+FN}.$$

它是软件缺陷预测中一个重要指标,期望预测模型能预测尽可能多的有缺陷模块.

误报率表示无缺陷模块错分为有缺陷模块的数量占总的无缺陷模块数的比率,定义为

$$PF=\frac{FP}{FP+TN}.$$

对于软件缺陷预测,精确度预测模型的准确程度,定义为

$$precision=\frac{TP}{TP+FP}.$$

一个好的预测模型应该有好的召回率和精确度,然而,这 2 个度量指标之间存在一定的权衡,F 值是召回率和精确度二者的调和均值度量,定义为

$$F\text{-}measure=\frac{2\cdot PD\cdot precision}{PD+precision}.$$

AUC 用于估计受试者工作特征曲线下的面积.受试者工作特征(Receiver Operating Characteristic，ROC)曲线由二维坐标中的一系列(PF,PD)点对组成,X 轴表示 PF 值,Y 轴表示 PD 值.预测模型的性能可用单个标量值 AUC 表示.

上述所有估计度量的取值范围为 [0,1],一个理想的缺陷预测模型应有高的 PD、F 值、AUC 和低的 PF.实验使用 PD、PF、F 值和 AUC 评估半监督缺陷预测方法的性能.

2.3　实验方法与设置

为了评估 TLSDL 的性能,对比 5 个代表性的半监督缺陷预测方法:迭代的半监督方法(Iterative Semi-supervised Approach-Fitting the Fits，FTF)[25]、欠采样随机委员会方法(Random Committee with Under-Sampling，ROCUS)[7]、低密度分离方法(Low-Density Separation，LDS)[10]、主动半监督缺陷分类方法(Active Semi-supervised Defect Categorization，ASDC)[9]和非负稀疏图标记扩散方法(Nonnegative Sparse Graph Based Label Propagation，NSGLP)[13].

对于实验选用的2个缺陷数据库中的所有项目,对所有方法采用 10 折交叉验证方式重复 10 次实验.在 10 折交叉实验中,9 份数据用于训练,1 份数据用于测试,在用于训练的 9 份数据中,按照给定的标记率 μ 对训练集划分为有标记数据和无标记数据.实验使用 3 种不同的标记率:10%,20%和 30%.TLSDL 中的参数 λ 是通过大范围搜索后根据最好的 F 值选择,

$$\lambda=0.001,0.01,0.1,1,10,100,1000.$$

2.4　实验结果分析

表 2 给出对比方法在 MDP 和 AR 中的 10 个缺陷预测项目上的预测结果,标记率 μ = 10%,20%,30%.从表 2 可以看出,TLSDL 的 PD、F 值和 AUC 值在绝大多数情况下都高于其它半监督缺陷预测方法.TLSDL 在 PD 上的性能提升验证了类特定的鉴别字典学习技术具备较强的特征学习能力,可以提升软件缺陷预测的召回率.同时,TLSDL 在综合度量 F 值和 AUC 值上的性能提升验证了二次学习方法能充分利用无标记样本辅助有标记样本进行半监督学习.从平均值上看,相比已有的对比方法,TLSDL 在 PD 值上至少提高 0.05;在综合度量指标 F 值和 AUC 值上,相比已有对比方法,TLSDL 分别至少提高 0.04 和 0.03.从度量指标 PF 值上看,TLSDL 虽然不能一直取得最好的值(最小值),但是它能取得与竞争方法可比的结果.

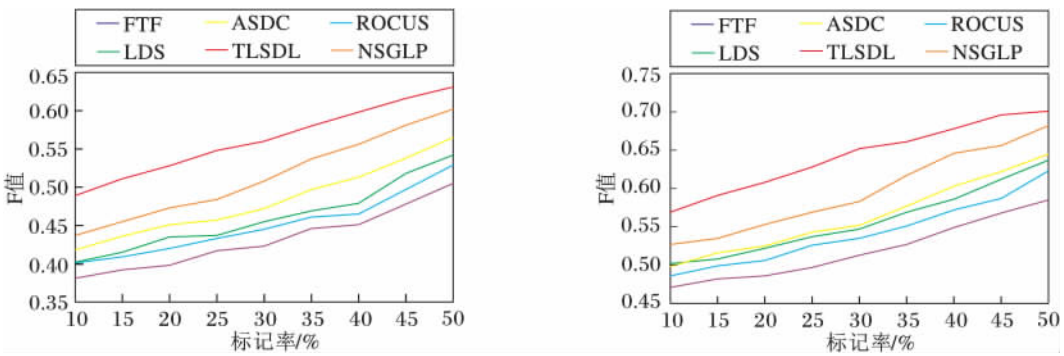
为了验证 TLSDL 在半监督情形下的有效性和研究有标记模块的数量对方法性能的影响,本文通过不断改变标记率 μ 对比所有方法的预测性能.分别从 MDP 和 AR 数据集中抽取一个项目库作为研究实例,观察预测结果随标记率的变化情况,这些项目库为 JM1 和 AR4,分别表示大规模项目集和小规模项目集.所有对比方法在这 2 个项目库上的 F 值随标记率的变化情况见图 2.

从图 2 可以看出,当标记率增加时,所有的对比方法的预测性能都得以提升,TLSDL 一直优于其它对比方法.特别是,在标记率很低时,相比其它方法,TLSDL 预测性能更高.这是因为标记率很低时,训练样本中只有少量有标记样本,存在大量无标记样本,TLSDL 能充分利用这些无标记样本信息辅助有标记样本构建预测模型,实验表明,TLSDL 是一种有效的半监督学习方法.

表 2 2 个数据集上不同标记率 μ 下 6 种方法的平均预测性能

Table 2 Average prediction performance of 6 methods with different labeled rate μ on 2 datasets

项目	评估度量	FTF			ROCUS			LDS			ASDC			NSGLP			TLSDL		
		μ=10%	μ=20%	μ=30%	μ=10%	μ=20%	μ=30%	μ=10%	μ=20%	μ=30%	μ=10%	μ=20%	μ=30%	μ=10%	μ=20%	μ=30%	μ=10%	μ=20%	μ=30%
JM1	PD	0.44	0.45	0.52	0.60	0.62	0.67	0.58	0.66	0.70	0.59	0.62	0.65	0.63	0.67	0.73	0.67	0.74	0.78
	PF	0.23	0.24	0.29	0.33	0.35	0.39	0.36	0.36	0.38	0.36	0.38	0.39	0.35	0.36	0.37	0.27	0.28	0.28
	F值	0.38	0.39	0.40	0.42	0.42	0.43	0.39	0.45	0.45	0.40	0.41	0.42	0.46	0.47	0.48	0.49	0.54	0.56
	AUC	0.65	0.67	0.68	0.74	0.75	0.76	0.71	0.76	0.78	0.71	0.73	0.74	0.72	0.74	0.80	0.76	0.79	0.83
KC1	PD	0.42	0.46	0.49	0.51	0.54	0.58	0.61	0.65	0.67	0.56	0.60	0.64	0.68	0.71	0.75	0.72	0.75	0.78
	PF	0.13	0.14	0.16	0.20	0.22	0.25	0.27	0.30	0.31	0.23	0.24	0.26	0.26	0.29	0.31	0.27	0.29	0.31
	F值	0.47	0.49	0.50	0.49	0.50	0.51	0.51	0.52	0.53	0.51	0.53	0.54	0.56	0.56	0.57	0.57	0.58	0.59
	AUC	0.62	0.62	0.65	0.66	0.69	0.70	0.68	0.69	0.72	0.68	0.69	0.71	0.72	0.75	0.77	0.73	0.76	0.78
PC3	PD	0.33	0.35	0.39	0.37	0.42	0.45	0.42	0.47	0.51	0.48	0.52	0.55	0.67	0.72	0.74	0.68	0.74	0.79
	PF	0.15	0.16	0.17	0.18	0.21	0.22	0.21	0.22	0.24	0.24	0.24	0.25	0.26	0.28	0.29	0.26	0.28	0.30
	F值	0.27	0.28	0.30	0.28	0.29	0.30	0.29	0.31	0.32	0.30	0.32	0.33	0.38	0.39	0.39	0.38	0.40	0.40
	AUC	0.62	0.70	0.71	0.63	0.67	0.69	0.64	0.68	0.72	0.62	0.66	0.72	0.66	0.73	0.75	0.67	0.75	0.77
PC4	PD	0.53	0.56	0.62	0.58	0.64	0.67	0.61	0.66	0.69	0.63	0.70	0.75	0.70	0.74	0.78	0.73	0.76	0.81
	PF	0.17	0.18	0.20	0.20	0.23	0.24	0.21	0.23	0.23	0.21	0.23	0.24	0.24	0.25	0.26	0.21	0.22	0.22
	F值	0.40	0.41	0.42	0.40	0.41	0.41	0.41	0.42	0.43	0.42	0.44	0.45	0.43	0.44	0.45	0.47	0.48	0.50
	AUC	0.70	0.75	0.78	0.71	0.75	0.77	0.73	0.76	0.77	0.72	0.75	0.79	0.76	0.80	0.81	0.77	0.82	0.83
PC5	PD	0.53	0.55	0.57	0.55	0.56	0.60	0.64	0.66	0.70	0.63	0.67	0.72	0.75	0.78	0.82	0.77	0.81	0.84
	PF	0.19	0.22	0.23	0.21	0.22	0.23	0.25	0.25	0.27	0.22	0.23	0.25	0.23	0.25	0.26	0.21	0.23	0.25
	F值	0.52	0.52	0.53	0.53	0.53	0.55	0.56	0.57	0.58	0.57	0.59	0.61	0.64	0.64	0.66	0.67	0.67	0.68
	AUC	0.76	0.78	0.79	0.80	0.82	0.83	0.79	0.81	0.82	0.80	0.81	0.82	0.81	0.83	0.84	0.82	0.83	0.85
AR1	PD	0.45	0.48	0.52	0.47	0.53	0.55	0.50	0.55	0.59	0.51	0.55	0.57	0.53	0.58	0.62	0.57	0.61	0.67
	PF	0.19	0.21	0.22	0.21	0.22	0.23	0.21	0.23	0.24	0.23	0.24	0.25	0.20	0.22	0.25	0.19	0.22	0.24
	F值	0.33	0.34	0.35	0.34	0.37	0.38	0.38	0.39	0.41	0.39	0.41	0.43	0.41	0.43	0.44	0.45	0.48	0.50
	AUC	0.62	0.65	0.66	0.64	0.67	0.69	0.66	0.68	0.71	0.66	0.69	0.71	0.68	0.70	0.73	0.71	0.73	0.77
AR3	PD	0.53	0.56	0.58	0.53	0.57	0.59	0.51	0.55	0.58	0.58	0.60	0.63	0.62	0.65	0.68	0.66	0.68	0.71
	PF	0.17	0.18	0.20	0.19	0.21	0.22	0.15	0.16	0.18	0.14	0.16	0.17	0.19	0.22	0.23	0.13	0.15	0.16
	F值	0.37	0.40	0.42	0.39	0.41	0.43	0.39	0.41	0.44	0.40	0.43	0.45	0.45	0.48	0.50	0.47	0.50	0.52
	AUC	0.65	0.68	0.70	0.67	0.70	0.71	0.65	0.69	0.72	0.69	0.72	0.74	0.70	0.72	0.75	0.72	0.76	0.79
AR4	PD	0.58	0.59	0.62	0.59	0.61	0.63	0.52	0.55	0.57	0.57	0.60	0.62	0.61	0.64	0.66	0.65	0.68	0.70
	PF	0.19	0.20	0.21	0.21	0.22	0.24	0.11	0.12	0.13	0.13	0.16	0.17	0.13	0.14	0.16	0.11	0.12	0.12
	F值	0.47	0.49	0.51	0.49	0.51	0.53	0.50	0.52	0.55	0.50	0.53	0.55	0.53	0.55	0.58	0.57	0.61	0.65
	AUC	0.63	0.65	0.67	0.65	0.66	0.68	0.63	0.66	0.68	0.67	0.69	0.71	0.70	0.72	0.74	0.71	0.73	0.76
AR5	PD	0.62	0.65	0.67	0.65	0.68	0.70	0.67	0.71	0.73	0.69	0.72	0.75	0.71	0.73	0.75	0.80	0.83	0.85
	PF	0.18	0.20	0.21	0.18	0.19	0.22	0.21	0.21	0.22	0.18	0.20	0.21	0.16	0.17	0.19	0.16	0.17	0.18
	F值	0.46	0.47	0.49	0.47	0.49	0.50	0.47	0.50	0.52	0.50	0.52	0.54	0.52	0.54	0.57	0.60	0.62	0.64
	AUC	0.62	0.65	0.69	0.64	0.68	0.70	0.66	0.69	0.72	0.68	0.71	0.73	0.69	0.72	0.74	0.73	0.75	0.79
AR6	PD	0.56	0.58	0.61	0.55	0.59	0.62	0.57	0.60	0.62	0.59	0.62	0.64	0.61	0.65	0.66	0.64	0.68	0.70
	PF	0.27	0.30	0.33	0.25	0.27	0.28	0.21	0.23	0.24	0.22	0.23	0.25	0.15	0.17	0.18	0.12	0.14	0.15
	F值	0.52	0.55	0.58	0.54	0.58	0.60	0.57	0.60	0.62	0.58	0.61	0.64	0.62	0.66	0.68	0.68	0.71	0.73
	AUC	0.60	0.63	0.65	0.63	0.64	0.68	0.64	0.67	0.70	0.65	0.68	0.71	0.70	0.73	0.75	0.72	0.76	0.79
平均值	PD		0.53			0.57			0.60			0.62			0.68			0.73	
	PF		0.20			0.23			0.23			0.23			0.23			0.21	
	F值		0.43			0.45			0.47			0.48			0.52			0.56	
	AUC		0.67			0.70			0.71			0.71			0.74			0.77	



(a) JM1 (b) R4

图 2 JM1 和 AR4 上 6 种方法的 F 值随标记率的变化情况

Fig.2 F-measure values of 6 methods changing with labeled rate on JM1 and AR4

3　结 束 语

本文提出基于二次学习的半监督字典学习软件缺陷预测方法.通过二次学习,将大量的无标记软件模块连同学习到的概率软标记扩充至有标记训练集中,在扩充后的训练集上,利用类特定字典学习方法进行特征学习,最后在学得的有鉴别能力的字典上进行稀疏表示分类,预测待测样本的缺陷倾向性.在 MDP 和 AR 数据集上的实验验证 TLSDL 的有效性.

如何设计一个有效的半监督学习方法解决有缺陷训练样本极少(甚至只有单个样本)时软件缺陷预测模型的构建问题是一个值得研究的方向.另一个研究工作是当现有软件工程项目中缺少充足的有标记数据时,利用跨公司的项目数据提高软件缺陷预测模型的性能.

参 考 文 献

[1] CATAL C, DIRI B. A Systematic Review of Software Fault Prediction Studies. Expert Systems with Applications, 2009, 36(4): 7346-7354.

[2] HALL T, BEECHAM S, BOWES D, *et al*. A Systematic Literature Review on Fault Prediction Performance in Software Engineering. IEEE Transactions on Software Engineering, 2012, 38(6): 1276-1304.

[3] 何 亮,宋擒豹,沈钧毅.基于 Boosting 的集成 k-NN 软件缺陷预测方法.模式识别与人工智能, 2012, 25(5): 792-802.
(HE L, SONG Q B, SHEN J Y. Boosting-Based k-NN Learning for Software Defect Prediction. Pattern Recognition and Artificial Intelligence, 2012, 25(5): 792-802.)

[4] SELIYA N, KHOSHGOFTAAR T M. Software Quality Estimation with Limited Fault Data: A Semi-supervised Learning Perspective. Software Quality Journal, 2007, 15(3): 327-344.

[5] SELIYA N, KHOSHGOFTAAR T M. Software Quality Analysis of Unlabeled Program Modules with Semisupervised Clustering. IEEE Transactions on Systems, Man, and Cybernetics (Systems and Humans), 2007, 37(2): 201-211.

[6] CATAL C, DIRI B. Unlabelled Extra Data Do Not Always Mean Extra Performance for Semi-supervised Fault Prediction. Expert Systems, 2009, 26(5): 458-471.

[7] JIANG Y, LI M, ZHOU Z H. Software Defect Detection with ROCUS. Journal of Computer Science and Technology, 2011, 26(2): 328-342.

[8] LI M, ZHANG H Y, WU R X, *et al*. Sample-Based Software Defect Prediction with Active and Semi-supervised Learning. Automated Software Engineering, 2012, 19(2): 201-230.

[9] THUNG F, LE X B D, LO D. Active Semi-supervised Defect Categorization // Proc of the 23rd IEEE International Conference on Program Comprehension. Piscataway, USA: IEEE, 2015: 60-70.

[10] CATAL C. A Comparison of Semi-supervised Classification Approaches for Software Defect Prediction. Journal of Intelligent Systems, 2014, 23(1): 75-82.

[11] MA Y, PAN W W, ZHU S Z, *et al*. An Improved Semi-supervised Learning Method for Software Defect Prediction. Journal of Intelligent & Fuzzy Systems, 2014, 27(5): 2473-2480.

[12] ABAEI G, SELAMAT A, FUJITA H. An Empirical Study Based on Semi-supervised Hybrid Self-organizing Map for Software Fault Prediction. Knowledge-Based Systems, 2015, 74: 28-39.

[13] ZHANG Z W, JING X Y, WANG T J. Label Propagation Based Semi-supervised Learning for Software Defect Prediction. Automated Software Engineering, 2017, 24(1): 47-69.

[14] JING X Y, YING S, ZHANG Z W, *et al*. Dictionary Learning Based Software Defect Prediction // Proc of the 36th International Conference on Software Engineering. New York, USA: ACM, 2014: 414-423.

[15] ZHOU Z H, JIANG Y. Medical Diagnosis with C4.5 Rule Preceded by Artificial Neural Network Ensemble. IEEE Transactions on Information Technology in Biomedicine, 2003, 7(1): 37-42.

[16] JIANG Y, LI M, ZHOU Z H. Mining Extremely Small Data Sets with Application to Software Reuse. Software: Practice & Experience, 2009, 39(4): 423-440.

[17] 杨子旭,黎 铭.二次回归学习及其在软件开发工作量预测上的应用.模式识别与人工智能, 2015, 28(1): 59-64.
(YANG Z X, LI M. Twice Regression Learning and Its Application on Software Effort Estimation. Pattern Recognition and Artificial Intelligence, 2015, 28(1): 59-64.)

[18] WRIGHT J, YANG A Y, GANESH A, *et al*. Robust Face Recognition via Sparse Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(2): 210-227.

[19] RAMIREZ I, SPRECHMANN P, SAPIRO G. Classification and Clustering via Dictionary Learning with Structured Incoherence and Shared Features // Proc of the IEEE Conference on Computer Vision and Pattern Recognition. New York, USA: IEEE, 2010: 3501-3508.

[20] YANG M, ZHANG L, YANG J, *et al*. Metaface Learning for Sparse Representation Based Face Recognition // Proc of the 17th IEEE International Conference on Image Processing. New York, USA: IEEE, 2010: 1601-1604.

[21] ROSASCO L, MOSCI S F, SANTORO M, *et al*. Iterative Projection Methods for Structured Sparsity Regularization. Technical Reports, MIT-CSAIL-TR-2009-050, CBCL-282. Cambridge, USA: Massachusetts Institute of Technology, 2009.

[22] YANG M, ZHANG L, FENG X C, *et al*. Sparse Representation Based Fisher Discrimination Dictionary Learning for Image Classification. International Journal of Computer Vision, 2014, 109(3): 209-232.

[23] GRAY D, BOWES D, DAVEY N, *et al*. The Misuse of the NASA Metrics Data Program Data Sets for Automated Software Defect Prediction // Proc of the 15th Annual Conference on Evaluation & Assessment in Software Engineering. London, UK: IET, 2011: 96-103.

[24] SHEPPERD M, SONG Q B, SUN Z B, *et al*. Data Quality: Some

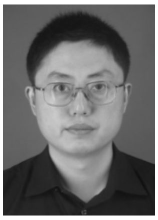
Comments on the NASA Software Defect Datasets. IEEE Transactions on Software Engineering, 2013, 39(9): 1208-1215.

[25] LU H H, CUKIC B, CULP M. An Iterative Semi-supervised Approach to Software Fault Prediction // Proc of the 7th International Conference on Predictive Models in Software Engineering. New York, USA: ACM, 2011. DOI: 10.1145/2020390.2020405.

作者简介

张志武,男,1981 年生,博士研究生,主要研究方向为模式识别、机器学习、软件工程.E-mail:zhangzw@ njupt.edu.cn.

(**ZHANG Zhiwu**, born in 1981, Ph.D. candidate. His research interests include pattern recognition, machine learning and software engineering.)



荆晓远(通讯作者),男,1971 年生,博士,教授,主要研究方向为模式识别、机器学习、软件工程.E-mail:jingxy_2000@ 126.com.

(**JING Xiaoyuan** (Corresponding author), born in 1971, Ph.D., professor. His research interests include pattern recognition, machine learning and software engineering.)

吴 飞,男,1989 年生,博士,讲师,主要研究方向为模式识别、机器学习、软件工程.E-mail:wufei_8888@ 126.com.

(**WU Fei**, born in 1989, Ph.D., lecturer. His research interests include pattern recognition, machine learning and software engineering.)

* *

第 5 届中国计算机学会大数据学术会议(CCF BigData 2017)征文通知

继 2013 年~2016 年成功召开四届 CCF 大数据学术会议后,第 5 届中国计算机学会大数据学术会议(CCF BigData 2017)将于 2017 年 10 月 14 日至 16 日在广东省深圳市举行。本届会议由中国计算机学会(CCF)主办,CCF 大数据专家委员会与深圳大学联合承办。本届会议拟突出与国际和多学科的交叉融合。会议将邀请多位院士和国际大数据领域的顶级专家学者作大会特邀报告,邀请 Science 编辑部的高级编辑组织以 Scientific Data in Science 为主题的高峰论坛,还将组织专题论坛、青年论坛和分会场口头报告等多种形式的学术交流。会议特别设立最佳学术论文奖、最佳应用论文奖和最佳学生论文奖。

一、征文范围(但不限于)

- 数据科学基础理论与方法
- 数据科学与大数据趋势与未来
- 大数据系统构架与基础设施
- 大数据采集与预处理技术
- 大数据存储管理模型、技术与系统
- 大数据并行计算模型、框架与系统
- 主流开源大数据系统优化与应用实践
- 大数据分析挖掘与智能计算方法与系统
- 高性能大数据学习架构、算法与系统
- 大数据可视化分析与计算
- 大数据共享开放技术方法与标准
- 大数据隐私与安全保护
- 大数据系统解决方案与工具平台
- 大数据行业与政府应用

二、征文要求

1、中文论文版面标准 A4 幅面,Word 或 PDF 格式,稿件按照《计算机学报》、《计算机研究与发展》等一级学报期刊论文篇幅要求撰写。英文论文版面标准 A4 幅面,Word 或 PDF 格式,稿件按照 IEEE Transactions on Big Data 等期刊论文篇幅和格式要求撰写。

2、会议官网:http://csse.szu.edu.cn/ccfbigdata2017/

3、投稿网站:http://easychair.org/conferences/?conf=ccfbigdata2017

(下转第 268 页)