# 基于非负稀疏图的协同训练软件缺陷预测

张志武[1],荆晓远[2,3],吴 飞[2]
(1. 南京邮电大学 计算机学院,江苏 南京 210023;
2. 南京邮电大学 自动化学院,江苏 南京 210023;
3. 武汉大学 软件工程国家重点实验室,湖北 武汉 430072)

**摘 要**:软件缺陷预测是一种可提高软件系统质量和优化测试资源分配的软件系统可靠性保证方法。当软件历史仓库中有标记训练模块较少时,应用机器学习方法构建有效的预测分类器是一个有挑战性的问题。为此,提出了一种基于非负稀疏图的协同训练软件缺陷预测方法,该方法汇集基于图的半监督学习方法和协同训练方法的优点,对无标记数据进行显示置信度估计。其利用软件模块间的相似性构建一个非负稀疏图,图中边的权重反映了样本间的相似度;利用协同训练的三个分类器对无标记样本的隐式选择和显示计算其所属类别的置信度,选取可靠的无标记样本辅助有标记样本进行训练以减少噪声数据的引入,并逐个迭代更新分类器,直至达到最大迭代次数或分类器识别率降低为止。基于 NASA MDP 数据集的验证实验结果表明,所提出的方法优于具有代表性的半监督协同训练方法。

**关键词**:非负稀疏图;协同训练;半监督学习;软件缺陷预测
**中图分类号**:TP311 **文献标识码**:A **文章编号**:1673-629X(2017)07-0038-05
doi:10.3969/j.issn.1673-629X.2017.07.009

# Defect Prediction of Co-training Software with Non-negative Sparse Graph

ZHANG Zhi-wu[1], JING Xiao-yuan[2,3], WU Fei[2]
(1. School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China;
2. School of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210023, China;
3. State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)

**Abstract**: Software defect prediction is a system reliability assurance method which can improve the quality of software system and optimize the distribution of test resources. When the previous defect labels of modules in software history warehouse are limited, building an effective prediction classifier by using machine learning methods becomes a challenging problem. Aiming at this problem, a co-training algorithm for software defect prediction based on non-negative sparse graph is proposed, which combines with the advantages of the graph-based semi-supervised learning method and the co-training method and estimates the confidence of unlabeled data. A non-negative sparse graph has been constructed by the similarity between the software modules so that the edge of the graph reflects the similarity between samples. Then three classifiers have been employed for co-training. In order to reduce the introduction of noise data, the reliable unlabeled samples have been selected for training by the implicit selection of the three classifiers and the confidence estimation of the categories. The classifiers keep to iteratively updating until the maximum number of iterations has reached or the recognition rates of classifiers have been reduced. Experimental results on NASA MDP datasets show that the proposed method is superior to the representative semi-supervised co-training method.

**Key words**: non-negative sparse graph; co-training; semi-supervised learning; software defect prediction

## 1 概 述

软件缺陷预测是软件工程领域中一个重要的研究课题,对提高软件产品的质量和优化测试资源的分配都有重要意义[1-2]。随着机器学习方法不断地应用于

**收稿日期**:2016-09-07 **修回日期**:2016-12-14 **网络出版时间**:2017-06-05
**基金项目**:国家自然科学基金资助项目(61073113,61272273);江苏省普通高校研究生科研创新计划项目(CXZZ12_0478)
**作者简介**:张志武(1981-),男,博士研究生,研究方向为模式识别、机器学习、软件工程;荆晓远,通讯作者,教授,研究方向为模式识别、机器学习、软件工程等。

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。很多机器学习方法假定有足够的缺陷数据用来建立预测模型,有监督学习方法在构建预测模型时利用了软件度量信息和缺陷标记类别信息。然而,通常情况下,标记软件缺陷模块需要耗费大量的人力物力,因此,不容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;与此同时,大量的无标记的软件缺陷模块却很容易收集。如何借助大量无标记样本来辅助少量有标记样本提高学习性能倍受学者们的关注,半监督学习是应用于这种场景下的重要研究方法。

目前半监督学习主要有四种范式:基于生成式模型的方法、基于低密度划分的方法、基于图的方法和基于不一致的方法。近年来,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 和 Khoshgoftaar 提出了基于期望最大化的半监督学习算法[3] 和半监督聚类方法[4]。Catal 和 Diri[5] 利用朴素贝叶斯算法建立半监督缺陷预测模型。Jiang 等[6] 提出了欠采样随机委员会方法。Li 等[7] 和 Thung 等[8] 分别提出了一种主动半监督学习方法。Catal[9] 评估了软件缺陷预测中的四种半监督学习方法:低密度分离、期望最大化、支持向量机和类质量归一化。Ma 等[10] 提出了一种随机下采样的协同训练方法。Abaei 等[11] 提出了半监督混合自组织映射模型。Zhang 等[12] 提出了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以由部分其他模块的度量值近似表示,而且这种表示通常是稀疏的。利用这种稀疏性可以为缺陷数据构建一个能反映数据关系的稀疏图,Zhang 等[12] 的研究验证了在这种稀疏图上基于图的半监督学习方法的有效性。为此,结合基于图的半监督学习方法和协同训练方法的优点,提出了一种基于非负稀疏图的协同训练软件缺陷预测方法(Non-negative Sparse Graph Based Co-training,NSGCT)。利用非负稀疏图上的训练数据间的稀疏表示关系,对无标记数据进行显示置信度估计,通过对用于辅助分类器训练的无标记样本的有效选取,减少了噪声数据的引入,提高了协同训练算法的性能。

## 2　基于非负稀疏图的协同训练缺陷预测方法

### 2.1　方法框架

在静态软件缺陷预测中,假设 $\boldsymbol{X}_i(\boldsymbol{X}_i \in \mathbb{R}^d)$ 表示软件模块样本,它是由静态代码度量值构成的列向量。$\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$ 表示软件缺陷预测数据集。$\boldsymbol{X}$ 是一个 $d\times n$ 矩阵,$d$ 是软件静态代码度量属性的个数,$n$ 是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),剩余部分没有标记。$l$ 是有标记数据集 $\boldsymbol{X}_l$ 中样本的数目,$u$ 是无标记数据集 $\boldsymbol{X}_u$ 中样本的数目。$\boldsymbol{X}=\{\boldsymbol{X}_l,\boldsymbol{X}_u\}$ $(l+u=n)$,其中 $\boldsymbol{X}_l=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_l\}$,$\boldsymbol{X}_u=\{\boldsymbol{x}_{l+1},\boldsymbol{x}_{l+2},\cdots,\boldsymbol{x}_{l+u}\}$。假设 $Y=\{Y_l,Y_u\}$ 是相应的缺陷标记,这里 $Y_l=\{y_1,y_2,\cdots,y_l\}$ 已知,$Y_u=\{y_{l+1},y_{l+2},\cdots,y_{l+u}\}$ 未知。$Y$ 中的标记 $y_i\in\{-1,1\}$ 是二值变量,这里 $-1$ 表示无缺陷标记,1 表示有缺陷标记。其目标是利用大量的无标记样本 $\boldsymbol{X}_u$ 去辅助少量有标记样本 $\boldsymbol{X}_l$ 进行协同训练,得到分类能力强的预测分类器。

基于非负稀疏图的协同训练缺陷预测方法框架如图 1 所示。

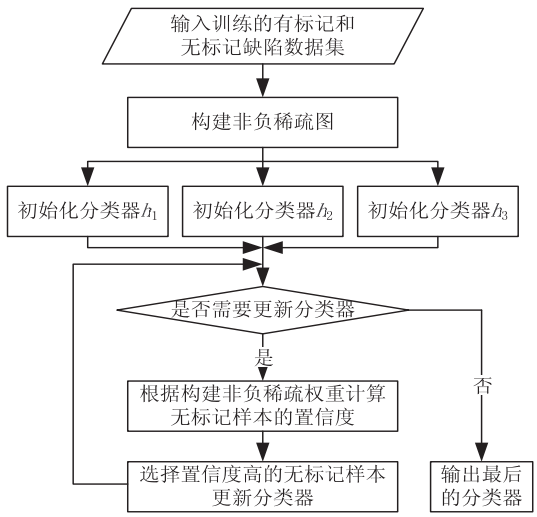


图 1　基于非负稀疏图的协同训练缺陷预测模型

首先,在软件缺陷训练集 $\boldsymbol{X}$ 上构建非负稀疏图,图的顶点是软件缺陷模块,边是由非负稀疏编码算法求得的能反映顶点间连接关系的非负稀疏相似权重;然后,对有标记样本集 $\boldsymbol{X}_l$ 进行 Bootstrap 重采样,构造协同训练中的三个分类器的初始训练集,并利用它们对分类器进行初始化;接着,在每一轮迭代中,每次轮流选定一个分类器作为主分类器,其他两个分类器作为辅助分类器分别对 $\boldsymbol{X}_u$ 上的无标记软件模块进行分类,并将分类一致的软件模块及其分类标记放入缓冲池中,利用建立的非负稀疏图计算无标记样本的置信度,将缓冲池中置信度高的无标记软件模块加入到主分类器中对主分类器进行更新;最后,直到达到最大的迭代次数或者三个分类器的分类误差都没有减小时,算法终止。

### 2.2　非负稀疏图的构建

基于图的半监督学习用图 $G=(V,E)$ 来表示数据之间的关系,图中节点 $V$ 表示数据,非负权重边 $E$ 表示数据点之间的关系,图的构造是基于图的方法的核心。传统的基于图的机器学习方法采用 RBF 函数($w_{ij}=$

$\exp(-\|x_i - x_j\|^2/(2\sigma^2))$ 来计算边的权重。但是，$\sigma$ 是靠经验设定的自由参数，它将显著影响学习结果而且很难得到一个最优的值。更糟糕的是，当只有少数有标记样本可用时，没有一种可靠的方法对其进行选择[13]。

为了避免传统基于图的方法中的参数选择问题，提出利用每个数据点 $\boldsymbol{x}_i$ 的稀疏表示信息构建图 $G$。基于稀疏编码构建 $l^1$ 图如下：

$$\min_{\boldsymbol{w}} \|\boldsymbol{x}_i - \boldsymbol{X}\boldsymbol{w}_i\|_2^2 + \lambda \|\boldsymbol{w}_i\|_1 \quad \text{s.t.} \quad w_{ij} \geqslant 0 \tag{1}$$

其中，参数 $\lambda$ 是一个用来平衡重构误差和稀疏程度的平衡因子；$\boldsymbol{x}_i$ 对应的非负稀疏向量 $\boldsymbol{w}_i$ 表示训练样本集中除 $\boldsymbol{x}_i$ 外其他样本对 $\boldsymbol{x}_i$ 稀疏表示的贡献程度。因此，它能本质上反映样本间的聚类关系。此外，非负稀疏表示自然地表示了图上节点的连接关系。这样能同时得到连接关系和权重。当所有样本的非负稀疏权重都求得后，稀疏矩阵 $\boldsymbol{W}$ 构建如下：

$$W(i,j) = \begin{cases} w_{ij} \Big/ \sum_{t=1}^{n-1} w_{it} & j < i \\ w_{i(j-i)} \Big/ \sum_{t=1}^{n-1} w_{it} & j > i \end{cases} \tag{2}$$

其中，$W(i,j)$ 反映了样本 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 间的相似关系和属于同一聚类的概率，因此，$\boldsymbol{W}$ 可看成是图 $G$ 的权重矩阵。

由于向量 $\boldsymbol{w}_i$ 是非负稀疏的，可称基于式(2)的图为非负稀疏图(Non-negative Sparse Graph, NSG)。NSG 中关键的步骤是计算式(1)中的非负稀疏编码，如算法 1 所述。

算法 1：非负稀疏编码算法。

输入：数据集 $\boldsymbol{X} \in \mathbb{R}^{d\times n}$，编码样本 $\boldsymbol{g} \in \mathbb{R}^{d\times 1}$ 和最近邻数 $n_{knn}$($n_{knn} \leqslant \min(n,d)$)

输出：稀疏编码 $\boldsymbol{w}$

(1)标准化 $\boldsymbol{X}$ 中所有列和编码样本 $g$，使它们有统一的 $l^2$ 模；

(2)根据最近邻准则，计算 $\boldsymbol{X}$ 中 $g$ 的最近邻子集 $I$，记 $\boldsymbol{X}^1 = \{x_i \mid i \in I\}$；

(3)求解如下正则非负最小二乘问题：

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \|\boldsymbol{X}^1\boldsymbol{w} - \boldsymbol{g}\|_2^2 + \lambda \|\boldsymbol{w}\|_1 \quad \text{s.t.} \quad \boldsymbol{w} \geqslant 0$$

(4)令 $w = 0 \in \mathbb{R}^{n\times 1}$ 并且置 $\boldsymbol{w}_I = \boldsymbol{w}^*$。

### 2.3 无标记样本置信度估计

在基于图的半监督学习中，通过最小化数据标记和图的不一致性来寻找无标记数据的标记。基于稀疏相似矩阵 $\boldsymbol{W}$，可构造如下不一致性代价函数(Cost Function)：

$$F(\boldsymbol{W},y) = F_L(\boldsymbol{W},y) + CF_U(\boldsymbol{W},y_U) \tag{3}$$

其中，$F_L(\boldsymbol{W},y)$ 为有标记模块和无标记模块之间的不一致性；$F_U(\boldsymbol{W},y_U)$ 为无标记模块之间的不一致性；$C$ 为用来衡量 $F_U(\boldsymbol{W},y_U)$ 重要性的常量。

为无标记模块分配标记时遵循下面两个准则：

(1)两个相似度高的无标记模块应具有相同的标记；

(2)与有标记模块有高相似度的无标记模块应具有与有标记模块一致的标记。

根据基于图的半监督学习理论，有标记模块和无标记模块之间的不一致性 $F_L(\boldsymbol{W},y)$ 定义为：

$$F_L(\boldsymbol{W},y) = \sum_{i=1}^{l}\sum_{j=1}^{u} W_{i,j}(y_{Li} - y_{Uj})^2 \tag{4}$$

无标记数据之间的不一致性 $F_U(\boldsymbol{W},y_U)$ 定义为：

$$F_U(\boldsymbol{W},y_U) = \sum_{i,j=1}^{u} W_{i,j}(y_{Ui} - y_{Uj})^2 \tag{5}$$

在每一轮的协同训练迭代过程中，要保留有标记模块的真实标记信息，用 $h(x_i)$ 表示 $x_i$ 的预测标记，则目标代价函数为：

$$\min F(\boldsymbol{W},y) \quad \text{s.t.} \quad h(x_i) = y_{Li}, i = 1,2,\cdots,l \tag{6}$$

将式(3)~(5)代入式(6)，可得：

$$\min F(\boldsymbol{W},y) = \min \sum_{i=1}^{l}\sum_{j=1}^{u} W_{i,j}(y_{Li} - y_{Uj})^2 + C\sum_{i,j=1}^{u} W_{i,j}(y_{Ui} - y_{Uj})^2 \quad \text{s.t.} \quad h(x_i) = y_{Li}, i = 1,2,\cdots,l \tag{7}$$

为了得到无标记模块的置信度，式(7)的最小化问题等同于最小化式(8)[14-15]：

$$\bar{F} = \sum_{i=1}^{u}(p_i - q_i) \tag{8}$$

其中：

$$p_i = \sum_{j=1}^{l} W_{i,j}(h_i - y_j)^2\delta(y_j,1) + \frac{C}{2}\sum_{j=1}^{u} W_{i,j}(h_i - h_j)^2 \tag{9}$$

$$q_i = \sum_{j=1}^{l} W_{i,j}(h_i - y_j)^2\delta(y_j,-1) + \frac{C}{2}\sum_{j=1}^{u} W_{i,j}(h_i - h_j)^2 \tag{10}$$

当 $x = y$ 时，$\delta(x,y) = 1$，否则 $\delta(x,y) = 0$。$p_i$ 和 $q_i$ 分别表示无标记模块 $\boldsymbol{x}_i$ 属于有缺陷类和无缺陷类的置信度，其标记 $z_i$ 可用函数 $\text{sign}(p_i - q_i)$ 求得，无标记模块 $\boldsymbol{x}_i$ 分配这个标记的置信度为 $|p_i - q_i|$。

### 2.4 算法描述

算法 2 给出了基于非负稀疏图的协同训练缺陷预测算法的实现步骤。

算法 2：基于非负稀疏图的协同训练缺陷预测

算法。

输入:有标记训练缺陷数据集 $\boldsymbol{X}_l$ 和无标记训练缺陷数据集 $\boldsymbol{X}_u$,迭代次数 $T$

输出:最终分类器 $h_1$, $h_2$, $h_3$

(1)利用算法 1 和式(2)构建非负稀疏矩阵 $\boldsymbol{W}$;

(2)在有标记训练缺陷数据集 $\boldsymbol{X}_l$ 进行随机 Bootstrap 重采样形成三个集合,用其初始化分类器 $h_i$;

(3)对每一个无标记模块计算 $p_i$, $q_i$,样本的标记 $z_i$ 和样本的置信度 $|p_i - q_i|$;

(4)将一个分类器作为主分类器,其余两个作为辅助分类器对无标记模块进行投票,投票相同的无标记模块放入到缓冲池 buffer($i$) 中;

(5)在 buffer($i$) 中选取置信度 $|p_i - q_i|$ 较高的无标记模块更新分类器;

(6)三个分类器的分类误差都没有减小或者达到最大迭代次数 $T$ 时退出算法,否则转到步骤 3。

## 3　实　验

为证明提出方法的有效性,将 NSGCT 方法与其他协同训练方法在 NASA 数据库上进行实验对比验证。

### 3.1　实验数据库

实验中采用了来自软件缺陷预测研究中广泛使用的开放数据集 NASA Metrics Data Program (MDP)。NASA MDP repository 数据库是美国航空航天局提供的开放软件缺陷数据库,从该仓库中选取五个项目库作为实验数据。每一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,其中包含了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一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的。表 1 给出了五个所选数据集的度量元个数、模块总数、有缺陷模块数及其所占比率等具体描述信息。根据文献[16]的建议,对实验数据集进行了清洗,移除了重复数据和矛盾数据。

表 1　NASA MDP 数据集

| 项目 | 度量个数 | 缺陷模块数 | 模块总数 | 缺陷率/% |
|---|---|---|---|---|
| JM1 | 22 | 1672 | 7 755 | 21.56 |
| KC1 | 22 | 314 | 1 192 | 26.34 |
| PC3 | 38 | 134 | 1 073 | 12.49 |
| PC4 | 38 | 177 | 1 287 | 13.75 |
| PC5 | 39 | 471 | 1 691 | 27.85 |

### 3.2　评估度量

在缺陷预测数据集上,一般采用信息检索评测指标对缺陷预测二类分类模型进行评估度量。包括召回率(PD)、误报率(PF)、$F$ -measure 和 AUC。由于软件缺陷预测数据的类不平衡性,采用综合评价指标 AUC 评估预测模型的性能。AUC 表示接收者操作特征(Receiver Operating Characteristic, ROC)曲线下的面积,这种评价指标可以有效避免类不平衡问题带来的影响。

### 3.3　实验结果与分析

为了对比 NSGCT 算法与现有主流协同训练算法的分类性能,采用 Co-training 算法[17]、Tri-training 算法[18]和 RusTri[10]作为对比方法,采用支持向量机(Support Vector Machine,SVM)作为基础分类器。在实验中,数据集被随机划分为三部分,10% 作为测试集,30% 作为有标记训练集,60% 作为无标记训练集。重复实验 20 次,最后给出平均的 AUC 值。在实验过程中,每轮迭代中取置信度较高的前 10% 的无标记样本辅助分类器训练能取得较好的效果,同时,算法的迭代次数取 20 时,分类器的分类误差下降速度趋于平稳。

表 2 给出了对比方法在 NASA MDP 数据库中的 5 个缺陷预测项目上的预测结果。

表 2　对比方法在 NASA MDP 数据库上的平均 AUC

| 训练集 | Co-training | Tri-training | RusTri | NSGCT |
|---|---|---|---|---|
| JM1 | 0.58 | 0.63 | 0.64 | 0.69 |
| KC1 | 0.60 | 0.58 | 0.70 | 0.73 |
| PC3 | 0.57 | 0.59 | 0.69 | 0.73 |
| PC4 | 0.68 | 0.70 | 0.81 | 0.84 |
| PC5 | 0.67 | 0.68 | 0.88 | 0.92 |
| Avg. | 0.62 | 0.64 | 0.74 | 0.78 |

从表 2 可以看出,NSGCT 方法的 AUC 值在绝大多数情况下都高于其他方法。从平均值上看,NSGCT 比对比方法至少提高 0.04(0.78-0.74)。NSGCT 在综合度量 AUC 值上的性能提升验证了基于非负稀疏图的协同训练半监督方法,能够在训练过程中通过对无标记数据的有效挑选在一定程度上防止噪声数据的引入,它能够充分利用无标记样本辅助有标记样本进行半监督学习。

## 4　结束语

为提高软件缺陷的预测能力和水平,提出了基于非负稀疏图的协同训练半监督缺陷预测方法。该方法结合基于图的半监督学习方法和协同训练方法的优点,利用非负稀疏图表示软件模块间的相似关系,并通过计算无标记模块的类别置信度,有效选取可靠样本进行协同训练。基于 NASA MDP 数据库的实验结果表明,该方法性能最优,能显著提高缺陷预测指标 AUC 的值,在软件缺陷预测中具有较好的有效性。

**参考文献：**


[1] Catal C, Diri B. A systematic review of software fault prediction studies[J]. 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[J]. IEEE Transactions on Software Engineering, 2012, 38(6): 1276-1304.

[3] Seliya N, Khoshgoftaar T M. Software quality estimation with limited fault data: a semi-supervised learning perspective[J]. Software Quality Journal, 2007, 15(3): 327-344.

[4] Seliya N, Khoshgoftaar T M. Software quality analysis of unlabeled program modules with semisupervised clustering[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2007, 37(2): 201-211.

[5] Catal C, Diri B. Unlabelled extra data do not always mean extra performance for semi-supervised fault prediction[J]. Expert Systems, 2009, 26(5): 458-471.

[6] Jiang Y, Li M, Zhou Z H. Software defect detection with ROCUS[J]. Journal of Computer Science and Technology, 2011, 26(2): 328-342.

[7] Li M, Zhang H, Wu R, et al. Sample-based software defect prediction with active and semi-supervised learning[J]. Automated Software Engineering, 2012, 19(2): 201-230.

[8] Thung F, Le X B D, Lo D. Active semi-supervised defect categorization[C]//Proceedings of the 2015 IEEE 23rd international conference on program comprehension. [s. l.]: IEEE, 2015: 60-70.

[9] Catal C. A comparison of semi-supervised classification approaches for software defect prediction[J]. Journal of Intelligent Systems, 2014, 23(1): 75-82.

[10] Ma Y, Pan W, Zhu S, et al. An improved semi-supervised learning method for software defect prediction[J]. Journal of Intelligent & Fuzzy Systems, 2014, 27(5): 2473-2480.

[11] Abaei G, Selamat A, Fujita H. An empirical study based on semi-supervised hybrid self-organizing map for software fault prediction[J]. Knowledge-Based Systems, 2015, 74: 28-39.

[12] Zhang Z W, Jing X Y, Wang T J. Label propagation based semi-supervised learning for software defect prediction[J]. Automated Software Engineering, 2017, 24(1): 47-69.

[13] Zhou D, Bousquet O, Lal T N, et al. Learning with local and global consistency[J]. Advances in Neural Information Processing Systems, 2004, 16(4): 321-328.

[14] Mallapragada P K, Jin R, Jain A K, et al. Semiboost: boosting for semi-supervised learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(11): 2000-2014.

[15] 郭 涛, 李贵洋, 兰 霞. 基于图的半监督协同训练算法[J]. 计算机工程, 2012, 38(13): 163-165.

[16] Shepperd M, Song Q, Sun Z, et al. Data quality: some comments on the NASA software defect datasets[J]. IEEE Transactions on Software Engineering, 2013, 39(9): 1208-1215.

[17] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training[C]//Proceedings of the eleventh annual conference on computational learning theory. [s. l.]: ACM, 1998: 92-100.

[18] Zhou Z H, Li M. Tri-training: exploiting unlabeled data using three classifiers[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(11): 1529-1541.


(上接第 37 页)


[3] 朝乐门, 张 勇, 邢春晓. 面向开放关联数据的知识地图研究[J]. 图书情报工作, 2012, 56(10): 17-24.

[4] 李 孟, 曹 晟, 秦志光, 等. 一种网络导航学习路径生成方法[J]. 电子科技大学学报, 2014, 43(4): 596-600.

[5] Lin F R, Hsueh C M. Knowledge map creation and maintenance for virtual communities of practice[J]. Information Processing & Management, 2006, 42(2): 551-568.

[6] 顾亦然, 王 兵, 孟繁荣. 一种基于 K-Shell 的复杂网络重要节点发现算法[J]. 计算机技术与发展, 2015, 25(9): 70-74.

[7] Hoser B, Hotho A, Jäschke R, et al. Semantic network analysis of ontologies[C]//European semantic web conference. Berlin: Springer, 2006: 514-529.

[8] Ortiz-Arroyo D. Analysis of semantic networks using complex networks concepts[C]//International conference on flexible query answering systems. [s. l.]: Springer, 2013: 134-142.

[9] 张 晗, 刘双梅. 中心度指标对语义述谓网络概念抽取的比较分析—以疾病治疗学研究为例[J]. 现代图书情报技术, 2013, 29(6): 30-35.

[10] 林德明, 陈超美, 刘则渊. 共被引网络中介中心性的 Zipf—Pareto 分布研究[J]. 情报学报, 2011, 30(1): 76-82.

[11] Diallo S Y, Lynch C J, Gore R, et al. Identifying key papers within a journal via network centrality measures[J]. Scientometrics, 2016, 107(3): 1005-1020.

[12] 许海云, 方 曙, 付鑫金. 基于特征向量中心度加权的期刊影响因子研究[J]. 情报理论与实践, 2011, 34(11): 108-112.

[13] 朱 骥, 杨 华, 牛北方, 等. Motif 识别算法简介及软件性能研究[J]. 计算机应用研究, 2006, 23(10): 66-69.

[14] Ribeiro P, Silva F. G-Tries: a data structure for storing and finding subgraphs[J]. Data Mining and Knowledge Discovery, 2014, 28(2): 337-377.

[15] Landis J R, Koch G G. The measurement of observer agreement for categorical data[J]. Biometrics, 1977, 33(1): 159-174.