工程应用

doi: 10.3969/j.issn.1673-5692.2018.04.017

基于改进 PSO 与 NB 算法的软件缺陷预测模型

马振宇[1]，张　威[1,*]，王晓岭[2]，高　飞[3]，刘福胜[1]

(1. 装甲兵工程学院，北京　100072; 2. 中国人民解放军 91431 部队，湛江　524002;
3. 中国人民解放军 9596 部队，武汉　430300)

摘　要: 伴随着软件在当今社会不断地更新换代，自身的缺陷也就随之出现，并且暴露出的问题越来越多，所以能够很好地使用软件缺陷预测技术是十分重要的，进而大幅度减少使用软件过程中的成本。本文通过将含有遗传思想的粒子群算法(M－PSO)和朴素贝叶斯(NB)相融合，对样本属性进行分离，把朴素贝叶斯分类的错误率看做是粒子群算法(PSO)里的适应函数，建立软件缺陷预测模型。借鉴 NASA 数据进行模拟实验，并通过与其他方法进行对比实验，证明在相似的方法中，M－PSO 与朴素贝叶斯相结合的算法预测能力最强。

关键词: 粒子群优化算法; 朴素贝叶斯算法; 软件缺陷预测
中图分类号: TP311　　**文献标识码:** A　　**文章编号:** 1673-5692(2018)04-460-05

Software Defect Prediction Model Based on Improved PSO and NB Algorithm

MA Zhen-Yu[1], ZHANG Wei[1], Wang Xiao-Ling[2], GAO Fei[3], LIU Fu-Sheng[1]
(1. Academy of Armored Force Engineering, Beijing 100072, China; 2. unit 91431 of PLA; 3. unit 95942 of PLA)

Abstract: With the development and evolution of software in the temporary era, its defects and problems have been increasingly exposed. Therefore, it is very important that the defect prediction technology is taken good advantage. In addition, the cost can be decreased during the using procedure. This essay separates sample attributes, regards error rate of Naïve Bayesian Model as fitness function of Particle Swarm Optimization and builds software defect prediction model via combining M-PSO and NB. It also demonstrates that the combination of M-PSO and NB is the best way among similar methods by referencing NASA data and comparing other methods.

Key words: Particle Swarm Optimization; Naive Bayesian; soft defect prediction

0　引　言

近几年，随着科技的不断发展，软件在各个领域都得到广泛的应用，软件已然对国家经济走向以及国民生活品质的高低起到至关重要的影响作用。为了能最大限度的利用好软件，我们就需要在软件开发的第一步把好关，尽可能多的预测出缺陷，降低日后软件在使用过程中所需花费更大的代价。诸如软件的再次开发、维护成本甚至直接造成软件的失效，因此软件缺陷预测技术[1-3]越来越受到人们的瞩目。

为了寻找到一种高效地预测方法，可以更早地、更准地、更快地发现软件潜在的缺陷，那么就可以通过高效地预测技术节省更多的资源。本文借鉴遗传思想，将遗传算法中变异的思想融入到粒子群算法中，通过变异算子使得种群里的粒子更新得到更好

收稿日期: 2018-06-01　　**修订日期:** 2018-07-02
基金项目: 军队科研计划项目(2015H04)

地速度和位置;然后将该方法与朴素贝叶斯算法相结合,得到最优的分离点以及最低的错误率;最后建立软件缺陷预测模型,提高软件预测能力[4]。

1　相关工作

1.1　朴素贝叶斯

贝叶斯定理在 250 多年前就被发明,在当今领域内有着无法撼动的位置。贝叶斯分类是一系列该分类算法的统称,这些算法都是根据贝叶斯定理演变而来,所以都叫做贝叶斯分类。在该分类算法的领域里,朴素贝叶斯算法(Naive Bayesian)[5]是其中应用最为广泛的算法之一,本文将其做以重点介绍。

借鉴概率和数理统计的思想将 NB 分类。NB 是贝叶斯算法的一种特殊情况,它简化了属性变量之间的关系,即简化为相互独立。虽然这个条件在一定范围里制约了朴素贝叶斯算法,但是与此同时,朴素贝叶斯算法使得对参数的估计大量简化,并通常可以忽略数据丢失带来的影响,算法本身也就简单化了,因而从较广的范围上降低了朴素贝叶斯算法的复杂度。

朴素贝叶斯算法模型如下:每一个数据样本都有很多的属性,把它的所有属性构建成一个 n 维的特征向量 $X=\{x_1,x_2,\cdots x_n\}$,分别对这 n 个属性进行 $A_1,A_2,\cdots A_n$ 度量。假设有 m 个类 $Y=\{y_1,y_2,\cdots y_m\}$,对于给出的一个未知样本 X,预估 X 隶属于具有最大后验概率的类里,那么 NB 算法将未知样本分到 y_i,也就是满足下式:

$$P(y_i|X) > P(y_j|X),\ 1 \leqslant j \leqslant m,\ j \neq i \quad (1)$$

其中 $P(y_i|X)$ 最高对应类的 y_i 称作最大后验假定,由贝叶斯定理,得到:

$$P(y_i|X) = \frac{P(X|y_i)P(y_i)}{p(X)} \quad (2)$$

因为 $P(X)$ 一般情况下都是一个常数,所以只要让 $P(X|y_i)P(y_i)$ 最大即可。类的先验概率可以用公式 $P(y_i)=\frac{s_i}{s}$ 得到,其中 s_i 为类 y_i 中的训练样本个数,而 s 为训练样本总数。有一种特殊情况,假如在类的先验概率不知的情况下,一般假设该类概率相等,也就是满足 $P(y_1)=P(y_2)=\cdots=P(y_m)$。

由于数据样本集具有很多的属性,它们之间的关系错综复杂,计算 $P(X|y_i)$ 的开销可能会异常的大,为了减少计算 $P(X|y_i)$ 的开销。我们假设各个属性相互独立,即在属性之间,没有任何联系,既有:

$$P(X|y_i) = \prod_{k=1}^{n} P(x_k|y_i) \quad (3)$$

其中每个样本的属性条件概率 $P(x_1|y_i)$,$P(x_2|y_i)$,…,$P(x_n|y_i)$ 可由训练样本估算。其中 A_k 分为两种情况讨论,即:

(1)若 A_k 是连续的,我们一般就假设这个属性满足高斯分布,得到:

$$P(X_k|y_i) = g(x_k,\mu_{y_i},\sigma_{y_i}) = \frac{1}{\sqrt{2\pi}\sigma_{y_i}} e^{\frac{(x_k-\mu_{y_i})^2}{2\sigma_{y_i}^2}} \quad (4)$$

其中,给出 y_i 样本属性值 A_k,$g(x_k,\mu_{y_i},\sigma_{y_i})$ 是属性 A_k 的高斯密度函数,而 μ_{y_i},σ_{y_i} 分别为平均值和标准差。

(2)若 A_k 是离散的,则 $P(X_k|y_i)=\frac{s_{ik}}{s_i}$,其中 S_{ik} 是在属性 A_k 上含有值 x_k 的类 y_i 的样本个数。

1.2　粒子群算法

粒子群算法[6-7]是人类根据鸟类群体的行为表象提出的一个算法,其核心思想就是模拟鸟类群体一起飞行寻食的过程,通过鸟类种群里各个小鸟的相互协作,反馈出自身最好的位置,然后再在鸟类种群里寻找到整个种群最佳的位置。现在我们把种群里的每一只鸟看成一个粒子,演化成现在的粒子群算法。就是在维数为 D 的搜寻空间里,有一粒规模数为 N 的种群粒子。然后规范粒子群相关参数,其中第 i 个粒子位置为:$x_i=(x_{i1},x_{i2},\cdots,x_{iD})$,第 i 个粒子速度为:$v_i=(v_{i1},v_{i2},\cdots,v_{iD})$,第 i 个粒子历史最优位置为:$p_i=(p_{i1},p_{i2},\cdots,p_{iD})$,所有粒子最优位置为:$p_g=(p_{g1},p_{g2},\cdots,p_{gD})$,位置更新公式为:$v_{iD}(t+1)=\omega v_{iD}(t)+c_1r_1\times(p_{id}(t)-x_{iD}(t))+c_2r_2(p_{gD}(t)-x_{iD}(t))$ 速度更新公式为:$x_{iD}(t+1)=x_{iD}(t)+v_{iD}(t+1)$。

2　基于遗传思想的 PSO 算法

为了保持粒子的高效性,根据遗传算法[8-10]里的变异算法,避免求解过程中陷于局部最优解。我们给出一个将变异思想融入 PSO 里的改进算法,即 M-PSO(Mutation Swarm Optimization))。在历代更新进程中使用变异手段,使之最终得到最优解。

把变异思想融入到 PSO 当中,通过其算法里的迭代公式进行变异操作。我们用 x_t^i 代替 x_{iD},在种群里第 i 个粒子的历史最优解对应的 $x_{\max}^i$ 代替 p_{iD},用

该种群的历史最优解对应的 $X^i_{\max}$ 代替 p_{gD}

$$\Delta x^i_{\max,t} = \Delta x^i_{\max,t-1} + (x^i_{\max,t} - x^i_{\max,t-1})/t \quad (5)$$

而 *PSO* 的具体变异迭代公式为：

$$\begin{cases} \Delta x^i_{\max,t+1} = \Delta x^i_{\max,t} + c_1 r_1 (x^i_{\max} - x^i_t) + \\ \qquad c_2 r_2 (X^i_{\max} - x^i_t) \\ x^i_{t+1} = x^i_t + \Delta x^i_{\max,t+1} \end{cases} \quad (6)$$

在以上公式中，我们可以得出一些结论。第一块通过参数变量可以有效地预计变异的幅度和运动轨迹。第二块则是通过变异操作来达到目的的。

曾经有种极端的改变方法。将一部分粒子的原有运动轨迹直接变异为逆方向，大大增强了该群的多样性，提升了 PSO 的寻优能力。就是把遗传思想里的变异率应用到粒子群算法中，用种群的规模大小去乘以变异率，算出需要变异的粒子个数，选定等同数目的粒子个数，不再按原来的迭代方程式进行速度的更新，而是选取种群现状中最优解的逆方向定为该粒子的飞行轨迹。这样以来就扩大了粒子自身的搜寻范围，消弱了粒子聚集的可能性，避免了“早熟”现象的出现。

而本文使用一种通俗易懂的方法，将遗传的思想融入到 PSO 里，去增加粒子群的多样性。假设当第 t 代时，种群里历史最高适应值 G_{best} 和该代粒子平均最高适应值 $\bar{G}_{best} = \frac{1}{n}\sum_{i=1}^{n} P_{best}$。假如 $G_{best}(t+1)$ 适应值大于 $G_{best}(t)$ 的适应值，或者 $\bar{G}_{best}(t+1)$ 的平均适应值大于 $\bar{G}_{best}(t)$ 的平均适应值，就证明了该代粒子是朝着最优解的目标靠近的。先假设 $\beta = |G_{best} - \bar{G}_{best}|$，当计算开始时，因为粒子间存在着很大的差异，$\beta$ 应该远远大于 0。而当在迭代了很多次以后，该种群粒子的历史最优适应值应该减去该代粒子平均最高适应值的绝对值就会越接近于 0，当 β 一直处于 0 的附近时，却一直不能终止迭代，就说明该种群已经陷入了局部最优的问题，则需要对该种群的粒子采取变异方案，以使粒子跳离局部最优问题，重新获得多样性，以便更好地寻找最优解。

其基本实现步骤如下，即图 1 所示。

第一步：初始化该种群中所有粒子的初始位置和速度。假定粒子的种群数量为 N。

第二步：通过适应值函数，算出每个粒子最初的适应值。将每个粒子在取适应值时所处位置设为 P_{best}，然后在根据该群体里的初始粒子，选择最优的适应值所对应的位置设为 G_{best}。

第三步：根据粒子位置和速度公式更新每个新粒子的最优位置和最佳速度。用其更新的位置和速度带入适应函数，计算出新的适应值。

第四步：判断 β 是否趋于 0，如果是就对该代粒子采取变异手段，通过变异迭代公式去产生新的粒子，然后求解与其对应的新的适应值。否则直接跳过第五步，进行第六步。

第五步：比较所有适应值。把该代的每一个粒子的适应值分别于对应的上一代粒子乃至历代粒子的适应值进行对比，假如该代粒子的适应值比上一代粒子乃至历代粒子的适应值都要好，就把该粒子的位置更新为现在最好的位置，否则不进行替换。再拿下一代的每一个粒子的适应值去和历史全局的适应值进行比较，假如该带的适应值高于历史种群的适应值，就用该粒子的位置更换历史种群最优位置，否则不进行更换。

第六步：判别该方法的终止条件，如果算法大于最多迭代次数或者找到最优解，立刻终止。否则回到第三步，接着进行运算。

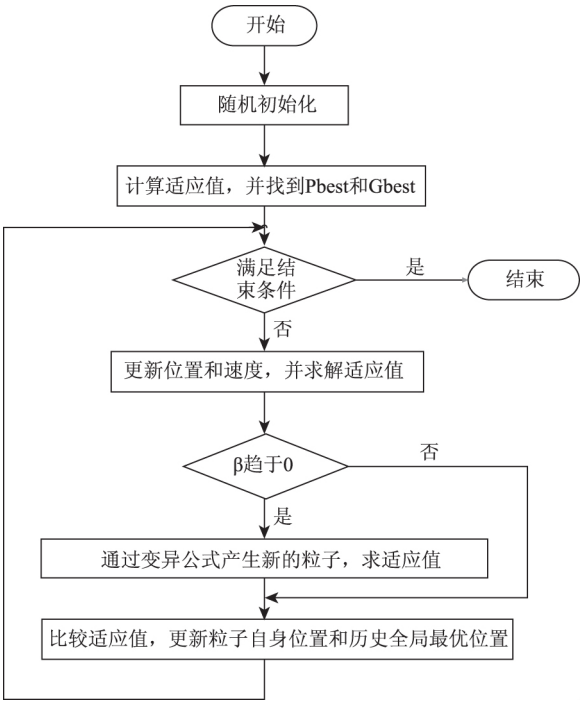

图 1 M-PSO 算法实现步骤

3 基于 M – PSO 与 NB 的软件缺陷预测模型

3.1 基于 M – PSO 与 NB 的属性离散化

把各个样本的属性值离散化，不但能够避免由

于属性值连续带来的大量计算量,而且能使预测结果简单明了和便于应用。我们把 M – PSO 与朴素贝叶斯相结合来完成属性值连续问题的离散化。为了更好地叙述这个方法,我们不妨假定样本的属性数量为 N。粒子群探索范围的上下界分别是:

$$P_{\max} = (p_{\max 1}, p_{\max 2}, \cdots, p_{\max N})$$
$$P_{\min} = (p_{\min 1}, p_{\min 2}, \cdots, p_{\min N}) \quad (7)$$

其中 P_{maxi}和 P_{mini}分别代表该样本第 i 个属性的最大值(上界)和最小值(下界)。各个样本的属性值的离散化其实就是在每个属性所属的范围内寻找几个分离点,假设将属性划分出 a 个等级,那么就需要找到 $a-1$ 个分离点。

3.2　M-PSO 与 NB 实现介绍

各个样本本身的属性离散化本质就是找寻该样本的分离点,由于没有统一的方法能够使分离结果做到最好,所以这也是实现该问题的一个难点。我们将 NB 的分类错误率当做粒子群算法里的适应函数,使用本文提出的 M-PSO 算法搜寻能够使得适应函数值最小的来找到最佳分离点。

第一步:对种群的粒子进行初始化,粒子群维数为 $N=20$,$w=0.3$,$c_1=c_2=1.5$,$r_1=r_2$ 为 0 到 1 的随机数。最大迭代次数为 $T_{\max}=50$,初始迭代次数为 $t=0$。然后把属性划分为 3 个级别,即有 2 个分离点。

第二步:依据每个粒子的坐标分离各个样本的属性,在应用朴素贝叶斯算法得到分类错误率,也是在比较该粒子坐标的好坏。然后更替粒子群里各个粒子最优位置以及种群最优位置。

第三步:按照文章里指出的 M – PSO 算法更替该粒子种群两个指标。

第四步:$t=t+1$,判别 t 是不是小于 T,如果是,立刻回到第二步,否则继续。

第五步:输出最佳分离点和最小错误率,得到软件缺陷预测模型。

3.3　软件缺陷预测模型

建立软件缺陷预测模型,即图 2 所示。

第一步:收集软件数据,输入总样本数以及各样本属性。

第二步:求解各个样本属性的的最大值(上界)与最小值(下界)。

第三步:根据文献 [11] 中,葛贺贺给出的训练样本与测试样本的分配比例(3∶2),本文为了方便进行实验结果的对比分析,所以选择一样的分配比例。然后随机进行样本的分组。

第四步:将 M – PSO 与 NB 算法相结合,计算每个样本的分离点,将样本属性离散化。

第五步:基于第四步的结果,建立软件缺陷预测模型。

第六步:输出预测结果。

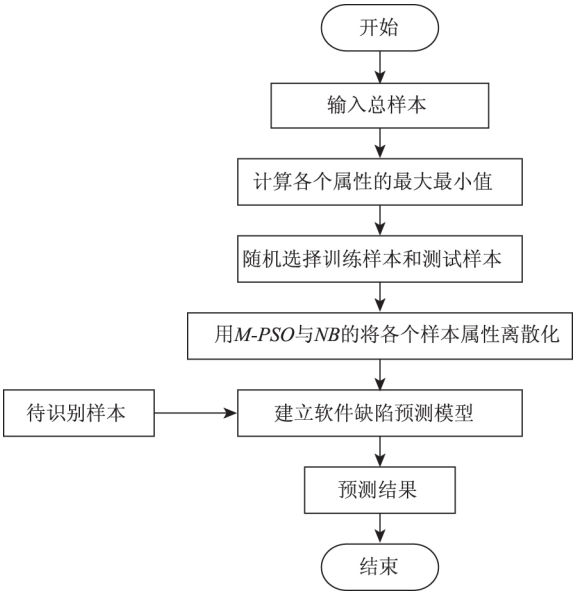

图 2　软件缺陷预测模型

4　实验分析

为了更好地对比实验效果,本文采用 NASA 提供的数据包,里面总样本个数有 10 855 个,并包括 21 个属性。我们首先从总共的样本数里任意抽选 6 000个当做训练样本,然后从剩余的里面任意抽选 4 000 个当做测试样本;然后根据数据离散化属性,从每个属性的最大值和最小值之间找出 2 个分离点,即把属性分为了 3 个等级;最后的实验结果如表 1所示。

进一步我们可以得到分类错误率,在使用 M-PSO与朴素贝叶斯方法相融合的情况下,使得错误率有了良好的改观,控制到 15.12%。比其他算法计算出来的错误率更好。比如在文章《Knowledge Discovery in Databases: An Attribute-oriented Approach》[12] 中的错误率为 20.45%,在文章《Extracting Classification Rule of Software Diagnosis Using Modified MEPA》[13] 中的错误率为 18.87,而在极为相似算法的文章《基于 PSO 和朴素贝叶斯的软件缺

表 1 数据离散化结果

属性	属性区间	离散化结果	
		分离点 1	分离点 2
1	[1,3782]	498.583	3011.375
2	[1,463]	156.739	366.194
3	[1,173]	89.492	149.736
4	[1,467]	301.492	431.379
5	[0,8109]	3260.839	6491.493
6	[0,83018]	28269.491	59278.391
7	[0,1]	0.618	0.951
8	[0,442]	197.395	481.782
9	[0,549]	197.386	482.381
10	[0,3.00483e+007]	1.37469e+007	2.98371e+007
11	[0,28]	10.694	21.158
12	[0,2.00018e+006]	0.09823e+007	0.16893e+007
13	[0,3018]	1973.387	2668.379
14	[0,321]	117.593	298.491
15	[0,506]	114.396	398.158
16	[0,99]	35.672	93.597
17	[0,397]	63.496	329.376
18	[0,1397]	683.369	1034.285
19	[0,5820]	128.385	4928.479
20	[0,4017]	892.358	2172.259
21	[1,716]	102.629	610.374

陷预测模型》[11] 中的错误率仍为 16.8%，证明了该方法的有效性。

5 结 语

本文通过将遗传算法里的变异思想融入到 PSO 中，提出了两个相融合的方法 M－PSO，以此为基础，和朴素贝叶斯算法相融合，构建出软件缺陷预测模型。实验结果证明，由于 M－PSO 算法改良了粒子群算法本身收敛速度较慢的缺点，所以可以更加高效地预测缺陷，使得软件有了更高的可靠性。

但对于多样本多属性的数据，会出现收敛速度较慢的现象，所以在今后的研究工作中，需要采用更高效的分类算法，提高预测效果。

参考文献:

[1] Muhammad Dhiauddin Mohamed Suffian, Suhaimi Ibrahim. A Prediction Model for System Testing Defects using Regression Analysis [J]. International Journal of Soft Computing And Software Engineering, 2012, 2(7): 55-68.

[2] Richard Baker, Ibrahim Habli. An Empirical Evaluation of Mutation Testing for Improving the Test Quality of Safety-Critical Software [J]. Transactions on Software Engineering, 2013, 39(6): 787-805.

[3] Mrinal Singh Rawat, Sanjay Kumar Dubey. Software Defect Prediction Models for Quality Improvement: A Literature Study [J]. International Journal of Computer Science Issues, 2012, 9(2): 288-296.

[4] 刘世军. 装备软件保障技术研究 [J]. 中国电子科学研究院学报, 2008, 3(6): 639-643.

[5] 余民杰，王元亮. 朴素贝叶斯分类算法研究 [J]. 行业科技.

[6] 廉师友. 人工智能技术导论 [M]. 西安: 西安电子科技大学出版社, 2007.

[7] 马振宇，张威，毕学军，等. 基于优化 PSO-BP 算法的软件缺陷预测模型 [J]. 计算机工程与设计, 2016, 37(2): 413-417.

[8] 刘晶晶，吴传生. 一种带交叉算子的改进的粒子群优化算法 [J]. 青岛科技大学学报, 2008, 29(1): 77-79.

[9] 王文义，秦广军，王若雨. 基于粒子群算法的遗传算法研究 [J]. 计算机科学, 2007, 34(8): 145-147.

[10] 巩永光. 粒子群算法与遗传算法的结合研究 [J]. 济宁学院学报, 2008, 29(6): 20-22.

[11] 葛贺贺，金 聪，叶俊民. 基于 PSO 和朴素贝叶斯的软件缺陷预测模型 [J]. 计算机工程, 2011, 37(12): 36-37.

[12] Han Jiawei, Cai Yandong, Cercone N. Knowledge Discovery in Databases: An Attribute-oriented Approach [C] //Proc. of the 18th International Conference on Very Large Data Bases. Vancouver, Canada: [s. n.], 1992: 547-559.

[13] Chen Jr-Shian, Cheng Ching-Hsue. Extracting Classification Rule of Software Diagnosis Using Modified MEPA [J]. Expert Systems with Applications, 2008, 34(1): 411-418.

作者简介

马振宇(1991—)，新疆人，博士研究生，主要研究方向为软件测试、软件缺陷预测、软件可靠性;
E-mail: 625181316@qq.com

张 威(1968—)，新疆人，教授，主要研究方向为软件工程、软件测试、军用软件保障;

王晓岭(1990—)，山东人，硕士研究生，主要研究方向为战场侦察;

高 飞(1990—)，山东人，硕士研究生，主要研究方向为军事信息系统、软件测试;

刘福胜(1990—)，河北人，副教授，主要研究方向为装备保障。