第 38 卷第 5 期
2018 年 9 月

杭州电子科技大学学报(自然科学版)
Journal of Hangzhou Dianzi University(Natural Sciences)

Vol. 38 No. 5
Sep. 2018

DOI:10.13954/j.cnki.hdu.2018.05.007

基于多维高斯分布概率模型的软件缺陷预测

苏　娜,方景龙

(杭州电子科技大学复杂系统建模与仿真教育部重点实验室,浙江 杭州 310018)

摘要:针对软件缺陷预测中普遍存在的样本特征冗余、缺陷数据集不平衡等问题,采用一种多维高斯分布概率模型来预测软件缺陷。通过均值向量以及协方差矩阵进行训练,根据待预测样本出现在各样本类别的概率判断得到分类结果。此外,还研究了特征个数与分类准确率之间的关系,验证了特征选择的必要性。在不同数据集上的对比实验结果表明,提出的模型具有较好的性能,弥补了普通分类算法忽视少数类样本等不足,保证预测效率的同时提高了模型整体的分类效果。

关键词:软件缺陷预测;多维高斯分布;概率模型;特征选择

中图分类号:TP183　　**文献标志码**:A　　**文章编号**:1001-9146(2018)05-0034-06

0　引　言

软件缺陷预测通过从软件过去的缺陷记录文件中提取代码属性并建立适当的模型来预测下一次发布中可能存在缺陷的组件。软件缺陷预测模型通常由预处理、特征选择、分类器训练和测试、选择指标评估等几个部分组成。S. Shivaji 等[1]研究发现,不同的预处理或特征选择算法与相同的分类算法结合具有差异性的效果,结合预处理、特征提取等预处理十分重要。特征选择的主要作用是根据一定指标来筛选出对正确分类贡献度最大的特征,以达到在提升预测效果的同时减少运算量的目的。但是,目前一般使用的特征选择存在冗余和噪声,影响模型对缺陷的判断和定位。判断软件模块是否存在缺陷的主要算法有统计学和机器学习两大类。统计学的算法包括多元回归、假设检验和参数估计等。机器学习算法包括逻辑回归(Logistic Regression, LR)[2]、支持向量机(Support Vector Machine, SVM)[3]等,在近年内应用广泛。

原始的软件缺陷数据集的数据分布不平衡,缺陷模块数量远少于无缺陷模块,事实上,对于有缺陷样本的分析更为重要[4]。传统的分类方法不能很好地应对数据集不平衡和特征冗余的情况,如 SVM 偏向于学习多数类而忽略少数类。针对软件缺陷数据普遍存在数据不平衡性严重、数据特征冗余的特点,本文提出将数据预处理、特征选择和分类算法相结合来检测软件中具有缺陷的成分,并通过实验验证了算法对于数据不平衡性和特征冗余具有鲁棒性。同时,为了更好地提升多维高斯分布概率模型的分类性能,对其特征选择进行了相应的研究,阐明了特征数量与分类效果的直接相关性,证明了特征选择的必要性。

1　多维高斯分布模型算法

1.1　一维高斯分布概率模型

高斯分布函数也称为正态分布函数,其表达式为 $f(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$,表示随机变量

收稿日期:2017-12-18

基金项目:国防基础科研计划资助项目(JCKY2016415C005);国防技术基础科研计划资助项目(JSZL2016415B002)

作者简介:苏娜(1995—),女,研究方向:软件可靠性。E-mail:nasu0601@126.com。通信作者:方景龙,研究员,研究方向:可靠性测试。E-mail:fjl@hdu.edu.cn。

$\boldsymbol{X}=(x_1,x_2,\cdots,x_n)$服从标准方差为 σ^2，数学期望为 μ 的高斯分布，n 为样本个数。$\boldsymbol{X}$ 服从高斯分布，可记作 $\boldsymbol{X}\sim N(\mu,\sigma^2)$。若 $\mu=0$ 并且 $\sigma^2=1$，则这样的高斯分布被称为标准正态分布，其表达为 $f(x)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$。

假设类 1 和类 2 的样本数据服从高斯分布，如图 1 所示。若需要判断一个样本 t 所在的类别，只需要找到该样本在不同类别中出现的概率，并依照概率大小进行判定，即只需要计算出图中的点 a 和点 b 的值。若 $a>b$ 则判断 t 属于类 1，反之属于类 2。

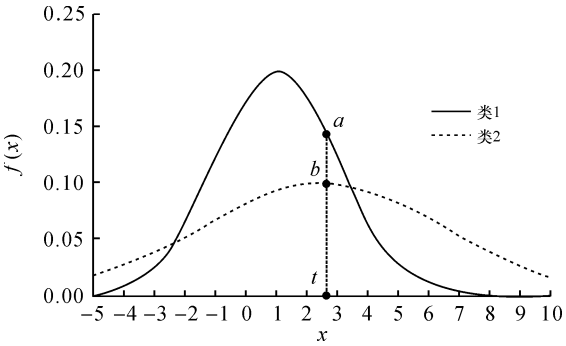


图 1　一维高斯分布判定方法

1.2　多维高斯分布概率模型

本文提出的多维高斯分布概率模型(Multivariate Normal Distribution，MND)的思想起源于单元的、一维的高斯分布判定方法。就软件缺陷预测问题而言，数据集中的每一个样本都是多维的，假如将每一个特征对应到单元的高斯分布，那么对于任意样本 t 可以求得该样本的每一个特征在任一类的概率，再将每一个特征对应的概率分别累积，便可以求得样本 t 在所有类中出现的各个概率值。但是，概率累积需要以特征独立为前提，由于缺陷预测数据集中许多特征都是相关特征，如代码行数(LOC_TOTAL)和注释行数(LOC_COMMENTS)，显然无法保证特征之间独立性，而多维高斯分布的判定模型可以有效地规避这个问题。

假设 $x_{ij}^k(i=1,2,\cdots,m_k;j=1,2,\cdots,n;k=1,2,\cdots,s)$为一个数据集中属于第 k 类的第 i 个样本的第 j 个属性。其中 m_k 为第 k 类中样本的总个数，n 为所有样本的特征数量，s 为样本类别的总数。首先，计算第 k 类中样本在所有特征上的均值向量 $\boldsymbol{\mu}_k$：

$$\boldsymbol{\mu}_k=\frac{1}{m_k}\sum_{i=1}^{m_k}\boldsymbol{X}_i^k \tag{1}$$

根据式(1) 求解第 k 类样本的协方差矩阵

$$\boldsymbol{\Sigma}_k=\frac{1}{m_k}\sum_{i=1}^{m_k}(\boldsymbol{x}_i^k-\boldsymbol{\mu}_k)(\boldsymbol{x}_i^k-\boldsymbol{\mu}_k)^{\mathrm{T}} \tag{2}$$

根据多维高斯分布模型公式，计算样本 t 出现在各类的综合分布概率 $P_k(t)$：

$$P_k(t)=\frac{1}{(2\pi)^{\frac{m_k}{2}}\left|\boldsymbol{\Sigma}\right|^{\frac{1}{2}}}\exp\left(\frac{1}{2}(t-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(t-\boldsymbol{\mu}_k)\right) \tag{3}$$

最后，判断样本 t 所属的类别

$$\mathrm{C}_t=\underset{k}{\operatorname{argmax}}P_k(t) \tag{4}$$

2　实验及结果分析

2.1　数据集

实验使用的 NASA 软件缺陷公共数据库包含 12 个软件缺陷数据集，基本信息如表 1 所示。数据库由若干个专门用于进行软件属性研究的数据集组成，每个数据集代表 NASA 软件系统或者子系统，其中包含一些软件模块及相应的故障数据，这些模块以静态代码度量作为特征，被转换为数值化的样例[5-6]。当某软件模块存在一个或多个缺陷时，其样本标记为正类，反之则为负类。NASA 软件缺陷公共数据库被许多研究者作为比较其算法优劣的通用数据，如文献[7-8]针对该数据库，分别使用了关联规则算法和朴素贝叶斯(Naive Bayesian，NB)来预测缺陷。为了体现本文算法在平衡性不同的数据集上的表现，在数据库中的 CM1，JM1 等 12 个数据集进行实验。数据集的属性包括代码行数(LOC_TOTAL)、空行数(LOC_BLANK)等基本的代码特征。为了方便实验，本文不考虑缺失值。

表 1 NASA MDP 数据集描述

数据集参数	CM1	JM1	KC1	KC3	MW1	MC1	MC2	PC1	PC2	PC3	PC4	PC5
属性数	37	21	21	39	37	38	39	37	36	37	37	38
样本总数	344	9 593	2 096	200	364	9 277	127	759	1 585	1 125	1 399	17 001
缺陷样本数	42	1 759	325	36	27	68	44	60	16	140	178	503
缺陷率/%	12.21	18.34	15.51	18.00	7.42	0.73	34.65	7.91	1.01	12.44	12.72	2.96

2.2 评估指标

软件缺陷预测属于偏斜类问题,表现为训练集中有非常多的同一种类的样例,只有很少或没有其他类的样例。研究发现,单着眼于算法预测的正确率是存在偏颇的,因为对于少数类的评价会被忽略,而软件缺陷预测中对于作为少数类的缺陷样本的正确判断意义更为重要,所以需要使用更合适的度量来评价算法的性能。

混淆矩阵(confusion matrix)又称为可能性表格或错误矩阵,采用矩阵形式来呈现算法性能的可视化效果,其列代表预测值,行代表实际类别。矩阵包含了关于样本真实类别和被预测类别的信息,通常一个缺陷预测算法的性能可以通过矩阵内的数据进行分析。二分类问题混淆矩阵如表 2 所示,N_{TP},N_{FP},N_{FN},N_{TN}分别表示相应类别的样本数量。

表 2 二分类问题的混淆矩阵

预测类别	实际类别	
	正类	负类
正类	真正类(TP)	假正类(FP)
负类	假负类(FN)	真负类(TN)

本文采用评估算法性能的指标如下:

(1)正确率(Accuracy)表示被准确预测的样本在所有样本中的占比,$P_{\text{Accuracy}}=\dfrac{N_{FP}+N_{TP}}{N_{TP}+N_{FP}+N_{FN}+N_{TN}}$。

(2)查准率(Precision)表示被判定为有缺陷样本当中真正存在缺陷的样本占比,$P_{\text{Precision}}=\dfrac{N_{TP}}{N_{TP}+N_{FP}}$。

(3)召回率(Recall)表示实际存在缺陷样本中被预测出缺陷的样本占比,$P_{\text{Recall}}=\dfrac{N_{TP}}{N_{TP}+N_{FN}}$。

(4)F-Measure 为一个综合指标,可以更全面衡量预测算法的整体性能,$F_\beta=\dfrac{(1+\beta^2)\times P_{\text{Recall}}\times P_{\text{Precision}}}{\beta^2\times P_{\text{Recall}}+P_{\text{Precision}}}$,本文取参数 $\beta=1$。

2.3 实验与结果分析

因为软件缺陷数据集存在诸多冗余的特征,所以,首先采用信息增益(Information Gain, IG)[9]算法对数据集中的特征进行筛选。然后,在 2.1 节的 12 个 NASA MDP 数据集上采用十折交叉验证法将 MND 模型与在缺陷预测领域内的经典算法——朴素贝叶斯(NB)、决策树(OneR,J48)以及支持向量机 SVM 进行比较。实验结果如表 3 和表 4 所示。

表 3 不同缺陷预测算法在 NASA 数据集上的正确率和查准率

数据集	正确率%					查准率				
	NB	OneR	J48	MND	SVM	NB	OneR	J48	MND	SVM
CM1	84.91	86.05	87.51	85.59	83.61	0.31	0.13	0.40	0.33	0.33
JM1	84.47	87.95	91.30	81.87	81.75	0.36	0.42	0.33	0.55	0.25
KC1	81.35	81.40	81.89	84.64	84.64	0.49	0.48	0.43	0.61	0.36
KC3	83.21	83.11	85.12	88.50	82.60	0.42	0.39	0.48	0.49	0.25
MC1	80.00	80.50	83.50	96.33	99.44	0.31	0.38	0.33	0.63	0.49
MC2	96.64	99.34	99.42	75.00	84.56	0.08	0.67	0.69	0.47	0.13
MW1	73.27	69.29	69.17	89.62	89.90	0.62	0.44	0.47	0.57	0.00
PC1	88.53	92.22	92.22	91.07	91.79	0.34	0.53	0.41	0.47	0.45
PC2	95.77	98.86	98.93	93.23	98.96	0.07	0.00	0.00	0.00	0.00
PC3	77.70	86.66	87.64	83.04	87.89	0.16	0.43	0.43	0.97	0.32
PC4	86.28	88.99	88.28	88.71	87.09	0.50	0.66	0.60	0.48	0.28
PC5	97.12	96.85	97.03	75.79	83.41	0.39	0.45	0.57	0.41	0.49
平均值	85.77	87.60	88.50	86.11	87.97	0.34	0.42	0.43	0.50	0.28

表 4　不同缺陷预测算法在 NASA 数据集上的召回率和 F-Measure

数据集	召回率					F-Measure				
	NB	OneR	J48	MND	SVM	NB	OneR	J48	MND	SVM
CM1	0.34	0.08	0.12	0.32	0.13	0.32	0.10	0.18	0.32	0.19
JM1	0.52	0.22	0.47	0.48	0.48	0.43	0.29	0.39	0.51	0.33
KC1	0.20	0.11	0.15	0.49	0.64	0.28	0.18	0.22	0.54	0.46
KC3	0.35	0.17	0.21	0.52	0.33	0.38	0.24	0.29	0.50	0.28
MC1	0.28	0.18	0.45	0.11	0.83	0.29	0.24	0.38	0.19	0.62
MC2	0.28	0.25	0.25	0.32	0.50	0.12	0.36	0.37	0.38	0.21
MW1	0.36	0.32	0.27	0.35	0.00	0.46	0.37	0.34	0.43	0.00
PC1	0.36	0.18	0.16	0.44	0.25	0.35	0.27	0.23	0.45	0.32
PC2	0.30	0.00	0.00	0.06	0.00	0.11	0.00	0.00	0.00	0.00
PC3	0.56	0.10	0.00	0.17	0.62	0.25	0.16	0.00	0.29	0.42
PC4	0.41	0.28	0.32	0.57	0.53	0.45	0.39	0.42	0.52	0.37
PC5	0.19	0.28	0.27	0.66	0.65	0.26	0.35	0.37	0.51	0.56
平均值	0.35	0.18	0.22	0.37	0.41	0.31	0.25	0.27	0.39	0.31

由表 3 可以看出,MND 模型的正确率并不是最优,但其查准率在多达 6 个数据集上表现突出。说明 MND 模型进行缺陷预测时,不会倾向于多数类而忽视少数类。表 4 中,SVM 的召回率更高,值得一提的是其召回率在 MC1 数据集上达到了 0.83,结合表 3 不难看出,SVM 的查准率低而召回率高。但从综合指标 F-Measure 的评估上看,MND 更胜一筹,其整体分类效果更优。

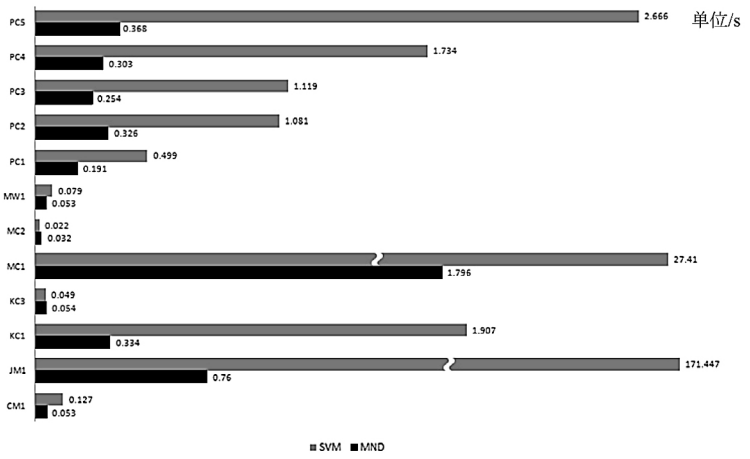


图 2　MND 和 SVM 的平均缺陷预测时间比较

在运行效率方面,实验还比较了整体性能较好的 MND 和 SVM 算法在不同数据集上的平均预测时间,结果如图 2 所示。

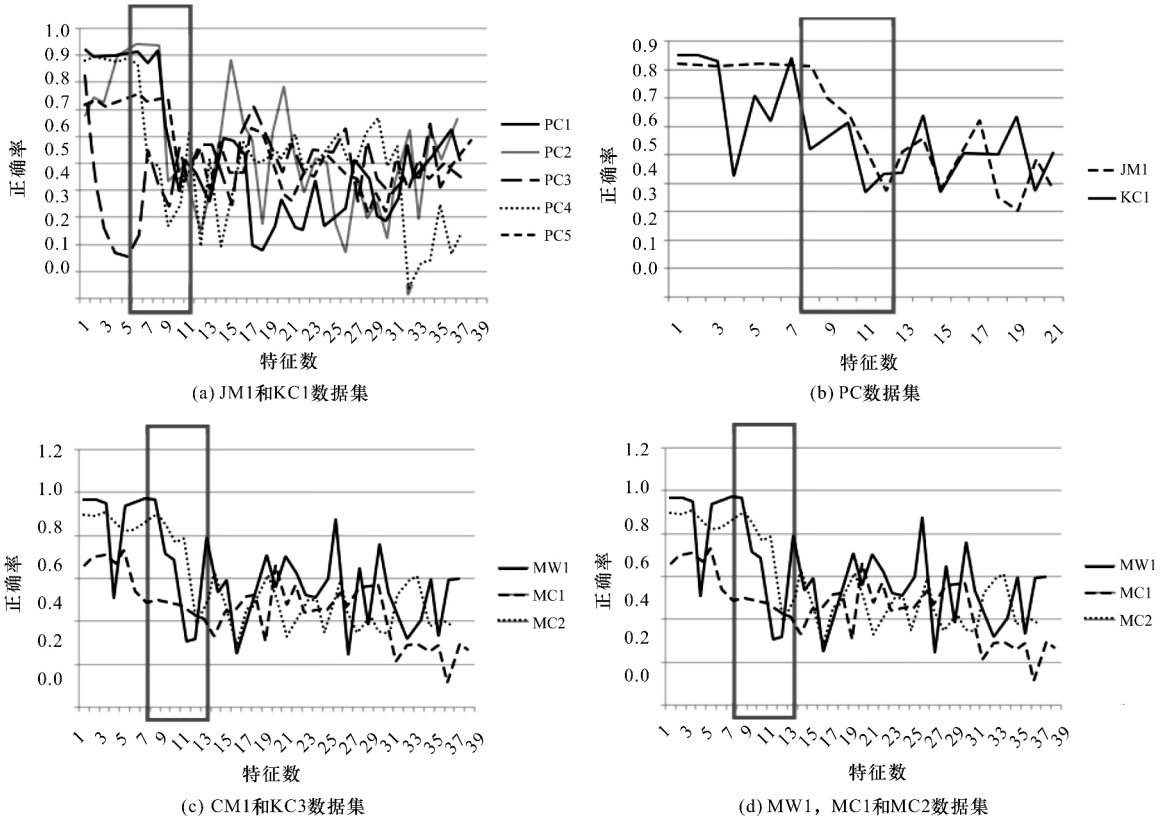
对比结果表明,除了在 MC2 和 KC3 数据集上,SVM 的平均缺陷预测时间比 MND 分别快 0.010 s 和 0.005 s,其余几乎都是 MND 的数倍。综合预测效果和预测时间两方面,MND 在保证整体分类效果的基础上,还能维持较高的预测效率,说明其整体预测性能最佳。

综合实验结果,本文提出的 MND 模型通过改进传统的一维高斯分布模型,使用多维度均值向量以及协方差矩阵进行训练,有效提高了分类性能,能够准确迅速地识别软件缺陷。具体表现在:(1)相比经典算法,有较高的召回率和 F-measure,克服了数据的有偏性;(2)与准确率一样优秀的 SVM 相比,平均预测时间更短,运算速度更快;(3)简洁有效地给出了判定过程和依据,仅需要极少数特征就可以获得高正确率;(4)避免了在单元高斯分布中特征的独立性假设问题,加强了鲁棒性。

2.4　预测正确率与特征个数的关系

特征选择是依据给定的评价标准选择一个最能保持数据原始特性的最优特征子集的过程[10]。为了定量描述特征个数和正确率之间的关系,首先按特征对实际缺陷类别的信息增益值从大到小排序,实验时按序列逐步添加特征并进行预测,结果如图 3 所示。

图 3 中,当特征个数增加时,正确率整体呈下降趋势。这是由于特征个数增多时,特征之间存在的冗余和噪声相应增多。此外,图中方框标记了不同数据集上的正确率在某个特征数量区间中明显下降的趋势,表明在缺陷预测过程中不宜采用过多的特征。从另一个角度来看,在度量软件时,使用过多的具有类似功能的特征是没有意义的。



(a) JM1和KC1数据集 (b) PC数据集

(c) CM1和KC3数据集 (d) MW1，MC1和MC2数据集

图 3 特征个数与缺陷预测正确率的关系图

3 结束语

数据集的有偏性和特征冗余导致了软件缺陷较难被精准预测的问题。本文基于多维高斯分布提出了一种高效且稳定的缺陷预测算法。通过计算每个类别中样本的特征均值和协方差矩阵得到其在不同类别中多维高斯分布模型，并通过分布概率预测样本所属类别。一系列评估指标的对比实验验证了本文提出的 MND 模型在预测缺陷方面所具备的高效性能，在分类效率和查准率上有着一定的综合性能。然而，本文算法在正确率上的表现并不十分突出，下一步将通过对动态代码特征度量的研究来提高预测正确率。

参考文献

[1] SHIVAJI S, WHITEHEAD E J, AKELLA R, et al. Reducing features to improve code change-based bug prediction[J]. IEEE Transactions on Software Engineering, 2013,39(4):552-569.

[2] SUFFIAN M D M, ABDULLAH M R. Establishing a defect prediction model using a combination of product metrics as predictors via six sigma methodology[C]// Information Technology. IEEE, 2010:1087-1092.

[3] JIN C, LIU J A. Applications of support vector mathine and unsupervised learning for predicting maintainability using object-oriented metrics[C]// Second International Conference on Multimedia and Information Technology. IEEE Computer Society, 2010:24-27.

[4] KRAWCZYK B. Learning from imbalanced data: open challenges and future directions[J]. Progress in Artificial Intelligence, 2016,5(4):1-12.

[5] YAMANAKA A, AOKI T, OGAWA S, et al. GPU-accelerated phase-field simulation of dendritic solidification in a binary alloy[J]. Journal of Crystal Growth, 2011,318(1):40-45.

[6] BENNIN K E, TODA K, KAMEI Y, et al. Empirical evaluation of cross-release effort-aware defect prediction models[C] // IEEE International Conference on Software Quality, Reliability and Security. IEEE, 2016:214-221.

[7] KAMEI Y, SHIHAB E, ADAMS B, et al. A large-scale empirical study of just-in-time quality assurance[J]. IEEE Transactions on Software Engineering, 2013,39(6):757-773.

[8] TURHAN B, BENER A. Analysis of naive bayes' assumptions on software fault data: an empirical study[J]. Data & Knowledge Engineering, 2009,68(2):278-290.

[9] CZIBULA G, MARIAN Z, CZIBULA I G. Software defect prediction using relational association rule mining[J]. Information Sciences, 2014,264(183):260-278.

[10] LEI S. A feature selection method based on information gain and genetic algorithm[C]// International Conference on Computer Science and Electronics Engineering. IEEE Computer Society, 2012:355-358.

Software Defects Detection Based on Multivariate Gaussian Distribution Probability Model

SU Na, FANG Jinglong

(*Ministry of Education Key Laboratory of Complex Systems Modeling and Simulation, Hangzhou Dianzi University, Hangzhou Zhejiang* 310018, *China*)

Abstract: In view of the common characteristics of data, such as imbalance and feature redundancy, in software defect prediction, a new classification model is proposed based on multivariate Gaussian distribution probability model. The model is built by mean vector of feature and the corresponding covariance matrix, and then makes judgment by the probabilities of the test samples belonging to each class. In addition, the relationship between the number of features and the classification accuracy is studied, and the necessity of feature selection is verified. The experimental results show that the proposed model has a good performance, which makes up the general classification algorithm tend to ignore a small number of samples and other deficiencies, to ensure that the prediction of the data efficiency at the same time to ensure a good overall classification effect.

Key words: software defect prediction; multivariate Gaussian distribution; probability model; feature selection

(**上接第** 23 **页**)

Improved Algorithm for Infrared Image Edge Detection Based on Prewitt Operator

AN Jianyao[1], LI Jinxin[1], SUN Shuangping[2]

(1. *School of Electronic Information Engineering, Hangzhou Dianzi University, Hangzhou Zhejiang* 310018, *China*;
2. *School of Automation, Hangzhou Dianzi University, Hangzhou Zhejiang* 310018, *China*)

Abstract: The traditional Prewitt algorithm has some disadvantages such as the missing of edge structure, the appearance of rough edge, false edges and inaccurate positioning caused by selecting threshold artificially. Thus, an improved algorithm for dynamic threshold of the human visual properties based on Prewitt operator was proposed. Firstly, the improved algorithm increases the number of directional templates to make edge detection more accurate. Secondly, this paper splits the flat areas in the image based on the human visual properties. Lastly, detects edge used dynamic threshold, thins edge based on the rough edge features at the same time. Results show that, compared with the classical algorithm, the proposed method can solve the shortcoming of the Prewitt operator rough edge effectively, the precision improved by 6% and detect edge structure more completely, and can position more accurately.

Key words: edge detection; Prewitt operator; human visual properties; dynamic threshold