一种面向对象软件缺陷的早期预测方法

张 垚,袁志海,江海燕
(中国卫星海上测控部,江苏 无锡 214400)

摘 要:软件过程早期的缺陷预测技术可以辅助软件工程决策,从而提高软件开发与测试的质量。针对面向对象软件,提出一种以分析设计模型的度量经验数据建立缺陷回归预测模型的方法,其中模型的建立使用了一种新形式的支持向量回归算法 ν-SVR。为了检验缺陷预测模型的实用价值,使用了来自真实世界的 Eclipse 项目三个版本的度量与缺陷数据集作为模型实验的训练集与测试集。结果表明,基于面向对象分析设计模型度量建立的缺陷回归预测模型可以在生命周期早期给出有效的缺陷数预测值,从而为软件工程实践提供支持。

关键词:面向对象; 软件度量; 缺陷预测; 支持向量回归
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2010)08-0037-04

An Early Defect Prediction Approach for Object Oriented Software

ZHANG Yao, YUAN Zhi-hai, JIANG Hai-yan
(Chinese Satellite Marine Tracking Center, Wuxi 214400, China)

Abstract: Predicting software defects in the early phase of software lifecycle is an effective way to improve the quality of software development and software testing. By using empirical software metric data retrieved early from object oriented analysis and design model, proposes an approach to build defect prediction model, using a new support vector regression algorithm called ν-SVR. Experiments on real world metric data from Eclipse project indicates that, the predicted number of defects could be used to assist the software engineering activities in early phase of software lifecycle.

Key words: object oriented; software metrics; defect prediction; support vector regression

0 引 言

早在20世纪80年代末,在软件质量、软件缺陷预测领域,就有系统的质量建模、度量、预测技术提出。就基于软件度量和机器学习的缺陷预测而言,一般是通过软件度量数据将问题量化,然后以此作为历史经验对未来的软件或模块做有无缺陷的分类或缺陷数量的预测。由于缺陷预测对于软件质量保证、软件测试等软件工程活动都有指导意义,除了大学等学术机构外,AT&T研究实验室、微软等企业也都有此领域的研究与应用[1,2]。

目前用于缺陷预测领域的算法众多,其中代表性的有 Khoshgoftaar 教授等人的基于树的方法、Fenton 教授等人的贝叶斯网络方法,以及人工神经网络、支持向量机等。国内这方面有李兴国教授等提出的基于支持向量回归机(ε-SVR)的可靠性早期预测方法[3]、基于支持向量机的缺陷分类[4]等工作,王青等给出了这一领域的发展综述[5]。总体而言,缺陷分类研究较多,缺陷数回归的研究较少。

文中主要研究使用生命周期早期度量数据建立缺陷回归预测模型的方法,这样的模型可以在早期即给出预测结果,从而更好地辅助软件工程决策。具体而言,预测模型建立是基于一种新形式的支持向量回归算法 ν-SVR[6]。

1 面向对象软件的早期度量

1.1 分析设计模型的静态度量与 UML

面向对象软件开发方法由于引入了封装、继承、多态等特性,与面向过程的方法有着根本性的不同。20世纪90年代中期,面向对象分析设计领域的三种主流方法:Booch 方法、Rambaugh 的 OMT 方法以及 Jocobson 的 OOSE 方法,融合发展出统一的面向对象分析(OOA)与面向对象设计(OOD)方法,诞生了统一建模语言(UML)。面向对象软件工程方法通过 OOA 建立的分析模型将现实世界的软件需求转换为类层次模

收稿日期:2009-12-15;修回日期:2010-02-05
基金项目:军队科研项目(编号略)
作者简介:张 垚(1982-),男,安徽巢湖人,助理工程师,研究方向为软件测试、软件工程。

型、对象－关系模型、对象－行为模型等，再通过 OOD 将分析模型映射到软件可以实现的设计模型，而(UML)则是这一面向对象软件过程业界通用的标准语言。

利用 UML 进行分析与设计建模，可大致划分为系统静态建模与动态建模两部分，其中前者的建模工作产品有用例图、类图等；后者有状态图、顺序图等。由于 UML 的简单性、一致性与通用性，目前已有诸多自动化度量方法[7~9]，下文所述的复杂度、耦合度度量，都可以较容易地通过对类图的自动化度量来获取。

利用历史的 UML 模型度量经验数据建立缺陷预测模型后，当新项目完成相应的 UML 分析、设计建模工作，通过自动化工具获取新的度量数据，送至预测模型，便可获得缺陷预测结果。相对于源代码，面向对象分析设计模型是软件生命周期早期或一个迭代的早期工作产品，由此建立的模型能更早地介入软件工程实践，为质量保证与软件测试活动提供支持。

1.2 复杂度度量

面向过程软件度量很多是不适用于面向对象软件的。Chidamber 和 Kemerer 于 1994 年提出六个有着严格度量理论基础的，用于描述面向对象设计规模与复杂度的度量项，被称为 CK 度量[10]。CK 度量与 Booch 等人在面向对象分析设计方法的工作有很大的关联，正是针对 OOD 提出的，其六个度量项中除去 LOCM 外的五个经验证对类的出错机率有显著影响。CK 度量之外具有代表性的面向对象复杂度度量还有 MOOD 度量与 Lorenz 与 Kidd 提出的度量等。

1.3 耦合性度量

对于面向对象分析与设计而言，类一级的耦合性用于表述类之间依赖程度的强弱。耦合性自然也能划归复杂度范畴，这里单独提出是要体现其重要性。低耦合始终是面向对象软件开发的一个目标，原因在于高耦合会带来难以理解、难以复用等与面向对象方法初衷相背离的问题。

图 1 给出一组针对 UML 类图的耦合性度量。对于一个给定类，这些度量对由该类的方法或属性引入的关系进行计数，这些关系可以根据三个因素划分为三组：

* 给定类和与其交互的类之间关系的类型：父子关系或其它；

* 交互的类型：当前类的某个属性、方法的参数或返回类型是另一个类，或引用了另一个类的方法；

* 导入或导出：一个类通过属性、方法或方法的类型使用另一个类，即导入 import，或被另一个类使用，即导出 export。

为简化表述，将这些面向对象耦合度量根据其描述的缩写命名：缩写的第一个字母是三种类间关系类型的缩写：A 对应祖先类(Ancestor)，D 对应子孙类(Descendant)，O 对应其它(Other，即既非祖先类也非子孙类)；第二、三个字母是交互的类型：CA 对应类－属性(Class－Attribute)，CM 对应类－方法(Class－Method)，MM 对应方法－方法(Method－Method)；第四、五个字母 IC 代表导入耦合(Import Coupling)，EC 代表导出耦合(Export Coupling)。图 1 中的这些度量较全面地描述了系统的耦合性特征。

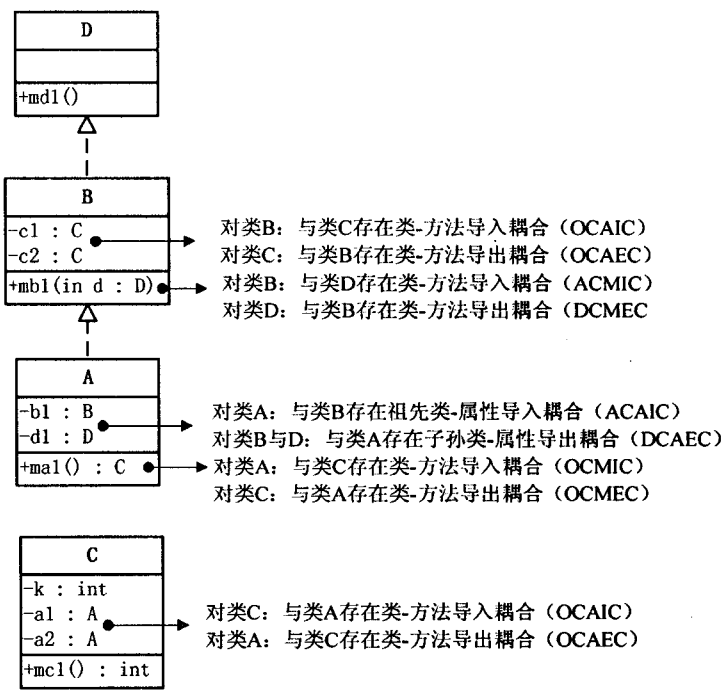

图 1 一种基于 UML 的面向对象软件耦合度量

2 建立模型的经验数据与算法

2.1 历史经验数据

建立模型所需的历史经验数据包含两方面：软件文件或模块的度量数据和相应缺陷数据。前者可由配置管理库中的历史版本中获取，常见的管理系统 Harvest、CVS、SVN 等；后者可由缺陷管理系统获取，常见的有 BugZilla 等。这些系统普遍拥有的开放特性为自动化数据收集提供了可能。对于软件能力成熟度等级高的软件组织，一般都定义有较完备的度量与分析规程，上述数据可更容易地获取到。

文中采用来自著名的开源软件 Eclipse 的真实数据进行实验，以检验在早期进行缺陷回归预测的有效

性。Eclipse 是通过 Java 实现的。具体而言,首先通过静态分析,对 Zimmermann 等发布的 Eclipse Bug Data 数据集[11]进行了扩展。原数据集在包(Package)级别主要涵盖了 2.0、2.1、3.0 版每个包的缺陷数据和源代码度量数据:前者包含由 BugZilla 与 CVS 变更记录获取的缺陷数;后者主要包含代码行、圈复杂度等度量 14 种,抽象语法树元素计数 83 种。扩展后的新数据集增加了 33 种文件、类和方法级度量。表 1 中给出数据集的基本情况。

表 1　Eclipse 这三个版本的基本情况

版本	发布日期	包数量	代码行	缺陷数标准差	缺陷数均值
2.0	2002.6	367	70 万	26	14
2.1	2003.3	423	100 万	17	8.5
3.0	2004.6	658	120 万	20.5	9.4

实验的数据集是扩展后的包级别数据集中的一个子集,其中包含描述分析设计模型关键性静态特征且编码阶段不会有太大变动的 19 种度量属性,见表 2(其中类级耦合性度量的说明可参考 Klocwork 8 的文档)。由于每个包中包含多个类,所以每种度量属性实际有均值、最大值与总和三种,因此共有 57 个属性,另外还有发布前后 6 个月期间统计的总缺陷数。

表 2　建立模型采用的面向对象分析设计模型度量

度量项	说明
LEVINHER	继承深度
NOPROTDATADECL	受保护属性数
NOPUBDATADECL	公共属性数
NOSTDATADECL	静态属性数
NOPROTMETH	受保护方法数
NOPUBMETH	公共方法数
NOMSG	消息总数
ACAIC 等 12 个类级耦合性度量	见 1.3 节

Eclipse 项目特点是大规模、全球化、分布式开发,由于 IBM 等公司的支持,其软件过程高度成熟。这些特点以及数据获取方法的客观性保证本中文实验所使用的经验数据的可靠性。值得注意的是,通过 Shapiro-Wilk 检验发现表 1 中各版本缺陷数在 0.05 置信度下均不符合正态分布,这一定程度上决定了简单的统计回归算法将难以获得较好的回归性能,因此,预测模型的建立使用了 ν-SVR 算法,下面给出简要介绍。

2.2　统计学习与 ν-SVR

支持向量机(SVM)是在统计学习理论结构风险最小准则(SRM)指导下发展出的算法工具:减小训练集上的风险的同时,还考虑降低学习算法的复杂度。支持向量回归机(SVR)用于解决回归问题,最早形式是 Vapnik 等引入的 ε-SVR,ε 是模型的一个参数,指回归超平面不敏感带的宽度,其中 ε 不敏感带边沿及之外的训练样本称为支持向量(SV),而边沿之外的称为受缚支持向量(BSV)。

ν-SVR 如同其读音(new-SVR),是一种新的与 ε-SVR 形式略有不同但本质上基本等价的支持向量回归算法,其目标为通过训练集 $(x_i, y_i) \in (\Re^N \times \Re)$, $i = 1,2,\cdots,l$ 寻找回归超平面 $f(x) = w \cdot x + b$,具体而言就是求解以下带约束最小化问题:

$$\min\ \tau(w,\xi^{(*)},\varepsilon) = \| w \|^2/2 + C(v\varepsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$\text{s.t. } (w \cdot x_i + b) - y_i \leq \varepsilon + \xi_i$$

$$y_i - (w \cdot x_i + b) \leq \varepsilon + \xi_i^*$$

$$\xi^{(*)} \geq 0, \varepsilon \geq 0$$

上式中参数 C 用于调节复杂度与经验风险间的权重,$\xi^{(*)}$ 是传统的松弛变量,而 ν 是 ν-SVR 新引入的参数,其确定后 ε 将被自动计算。参数 ν 的重要性质是,以 nSV 和 nBSV 表示 SV 与 BSV 样本的数量,则 ν 分别是 nSV/l 和 nBSV/l 的上界和下界[6],通过调节参数 ν,可有效控制两者数量。Chalimourda 等还给出在不同噪声下 ν 的最优值[12]。总的来说,参数 ν 相比 ε 更直观和易用。为了数值求解和表述方便,下文中 C 实际是指 C/l。

利用拉格朗日乘子法,上述问题可转换为一个凸二次规划问题,有全局最小解。最终可获得如下形式的回归超平面:

$$f(x) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)K(x_i \cdot x) + b$$

其中 $\alpha_i^{(*)}$ 是拉格朗日乘子,其值对于非支持向量的样本都为 0,K 为满足 Mercer 条件的核函数,形式为 $K(x_i, x_j) = \Phi^{\mathrm{T}}(x_i)\Phi(x_j)$,其中通过 Φ 隐式完成低维到高维映射,可将问题空间的非线性回归问题转换为高维特征空间的线性回归问题,并且避免了维度爆炸。

3　模型实验与结果分析

3.1　实验步骤

基于 2.1 节获取的三组数据(方便起见,下文将 2.0 版的度量数据简称为 P20,2.1 版为 P21,3.0 版为 P30)以及 2.2 节所述的 ν-SVR 算法,按如下步骤建立缺陷预测模型并对预测结果进行分析:

(1)确定性能测度。理论上,采用留一法(Leave One Out)可在训练集上获得总体上性能测度的无偏估计。对于回归问题,训练时一般采用均方误差(下文简写为 MSE)作为性能测度搜索最佳参数以建立模型,

就是期望预测获得最小的绝对误差。但在噪声很大的情况下，可能无法获得小的均方误差，退求其次，能预测出较准确的顺序也是有意义的，例如对于缺陷预测，预测出模块缺陷数的排序如果和真实很接近，那么意味着可以找出风险最高的一些模块。因此，文中还引入常见的秩次相关系数 Kendall's τ（下文简写为 τ）作为性能测度，τ 越大，意味着秩次相关性越高，即预测值与真实值的顺序更为相似。

(2) 参数选择与模型建立。首先，ν - SVR 自身有参数 ν 和惩罚因子 C，其次根据采用核函数不同又会引入不同的参数。径向基函数(RBF)是最常用的通用核函数，相比较多项式核函数，RBF 只引入一个参数，记为 γ；相比较 Sigmoid 函数，RBF 可保持 Hessian 矩阵的正定而给求解带来方便；相比线性核函数，RBF 又可以隐式地完成向无限维特征空间的映射，学习能力较强。总的来说，使用 RBF 核的 ν - SVR 模型要确定三个参数，而这三个参数对模型的预测性能是有决定性作用的，必须慎重选择。

由于留一法计算量过大，文中实际是通过 5 折交叉检验和网格搜索，基于训练集搜索上述三个参数的最优值。网格搜索速度也较慢，尤其是需要在三维参数空间进行搜索时更是如此，但由于软件工程经验数据噪声形式未知，网格搜索相对其它多种参数搜索方法更可靠，且由于网格搜索任务易于并行，我们在实验中借助 MPICH 与 mpi4py 模块，使用 Python 进行 MPI 编程，将参数空间划分为众多子空间，分配到以太网的 14 个节点(节点数仍可增加)上进行分别计算，结果汇总后得出最优参数，从而一定程度上缓解了速度慢的问题。采用 MSE 和 τ 作为性能测度会搜索到不同的参数，因此每次网格搜索实验会产生两个模型。

(3)预测模型推广性能测试。为充分考察预测模型的推广性能，实验中使用老版本数据建立的模型，使用新版的数据做测试数据集，用实际缺陷数与模型预测的缺陷数做比较。

3.2 结果分析

根据上述步骤进行表 3 中所列实验：分别利用 P20、P21 以及合并 P20 与 P21 得到的混合数据集作为训练集，在三维参数空间的 4320 个点中搜索最佳参数建立模型，再利用 P30 做推广性能测试。另外，表 3 给出了各实验运行时间，容易发现，训练集规模增大导致 ν - SVR 中二次规划问题的数值求解时间极大地增加，这也是支持向量机算法目前仍难以完全回避的一个根本性问题。

表 4 结果显示，三组实验建立的六个模型在 P30 上预测的缺陷数与实际值相比 MSE 始终较大，但两者线性相关系数基本都大于 0.6，最高达到 0.67，秩次相关性也达到 0.35 以上，这说明缺陷数的预测值与实际值间有较为显著的共变特征，意味着预测值在实际中将有应用意义：将预测值作为风险指标，用于辅助确定 SQA 工作重点、制定软件测试优先级；在测试完成时，又可通过预测值来检验测试充分性等。

表 3 模型试验列表

实验	内容	训练集规模	运行时间(秒)
1	P20 = > P30	367	982
2	P21 = > P30	423	1919
2	P20 + P21 = > P30	790	8241

表 4 实验结果及对比

实验	性能测度	最优参数			P30 上性能测度		
		C	γ	ν	MSE	相关系数*	τ*
1	MSE	2048	0.016	0.38	236.6	0.67	0.37
	τ	128	0.25	0.76	241.8	0.67	0.39
2	MSE	8192	0.008	0.34	292.1	0.58	0.23
	τ	128	1	0.84	278.9	0.62	0.37
3	MSE	256	0.5	0.31	255.0	0.65	0.37
	τ	128	1	0.72	266.0	0.66	0.37

(* 相关系数在 0.01 水平下是显著的)

另外，对实验结果有如下发现与结论：

●P20 建立的模型结果较好，P21 的相对较差，P20 + P21 介于二者之间。这是真实世界度量数据中的不确定性决定的。理论上，经验的积累应该意味着预测性能的提升，但由于 P21 中缺陷数据存在若干异常值，导致了其基础上建立模型的性能的下降。当然，性能下降程度并不明显。

●以 MSE 最小化建立的模型在 P30 上推广的 MSE 也稍小；以 τ 最大化建立的模型 P30 上推广的 τ 也稍大。因此，要获得更精确的缺陷数预测值，可用 MSE 为性能测度；要获得更准确的缺陷数排序序列，可用 τ 为性能测度。

●使用 τ 建立的模型参数始终较为稳定，推广性能也较稳定，对训练集中噪声较不敏感。而以 MSE 为性能测度总倾向于寻找更大的 C，结合 2.2 节论述可知，这不利于模型推广性能。例如实验 2 中以 MSE 为性能测度建立的模型，其推广性能就有易观察到下降。

4 结束语

文中主要针对面向对象软件，通过真实世界数据集上的实验，发现使用分析设计模型度量经验数据可建立有效的缺陷回归预测模型。模型可以在软件生命周期早期给出有效的缺陷数预测值，从而可以给软件

(下转第 44 页)

完成整个语义检索。

4 结束语

语义网技术改变了信息资源的描述方式，它对信息资源进行统一的语义描述，将语义网技术引入信息检索领域，为信息检索开创了一个崭新的研究方向—语义信息检索，随着语义网技术的研究越来越纯熟，语义信息检索的各项技术也更加成熟。文中提出的语义信息检索框架是基于语义网方法与技术的，利用语义网中的本体技术对网络资源进行描述，然后将用户端给定的关键词与本体中信息资源进行匹配，得到检索的查询条件，针对该查询条件本体模型进行遍历，最终找到与关键词相匹配的本体实例，这些实例就是最终所要查询的结果。文中还提出了基于语义信息检索框架的信息检索算法，在未来的工作中将针对该算法设计出语义查询系统，实现与用户的交互。

参考文献：

[1] Rinaldi A M. An Ontology - Driven Approach for Semantic Information Retrieval on the Web[J]. ACM Transaction on Internet Technology, 2009, 9(3): 1 - 10.

[2] Antoniou G, Van Harmelen F. 语义网基础教程[M]. 陈小平，译. 北京：机械工业出版社，2008: 1 - 3.

[3] Zhou Qi, Wang Chong, Xiong Miao, et al. SPARK: Adapting Keyword Query to Semantic Search[C]//In: Proceedings of ISWC'2007. [s. l.]: [s. n.], 2007.

[4] 丁晟春，顾德访. Jena 在实现基于 Ontology 的语义检索中的应用研究[J]. 现代图书情报技术，2005(10): 5 - 9.

[5] 黄 敏，赖茂生. 语义检索研究综述[J]. 图书情报工作，2008(6): 63 - 66.

[6] 王继东，张 瑜，李 娜. 基于本体的语义检索技术研究与实现[J]. 计算机技术与发展，2009, 19(10): 134 - 137.

[7] 董海凤. 一个完整的基于语义网的信息搜索模型[J]. 计算机技术与发展，2009, 19(8): 1 - 3.

[8] 黄日茂. 语义 Web 知识表示方法的研究[D]. 贵阳：贵州大学，2006.

[9] Anyanwu K, Sheth A. ρ - Queries: Enabling querying for semantic associations on the semantic web[C]//Proc. of the 12th int'l Conf. on WWW. New York: ACM, 2003: 690 - 699.

[10] Aleman - Meza B, Halaschek C, Arpinar I B. Context - aware semantic association ranking[C]//Proc. of the 1st int'l Workshop on Semantic Web and Databases. Berlin, Germany: Humboldt - University, 2003: 33 - 50.

[11] 袁 杰，赵建民，朱信忠，等. 基于本体的领域 Web 搜索模型与架构[J]. 计算机时代，2008(5): 22 - 25.

[12] 王晓东，张 合，王红涛. 基于 Ontology 的语义信息检索模型研究[J]. 计算机工程与设计，2008(6): 39 - 41.

(上接第 40 页)

测试、软件质量保证等工程实践提供定量的数据支持。文中最后给出了通过不同性能测度建立的模型推广性能的简单对比，发现除传统的均方误差外，秩次相关性也是一种有价值的性能测度。如何根据软件缺陷回归预测的特点和应用需求确定合适的性能测度是值得进一步研究的问题。

参考文献：

[1] Zimmermann T, Nagappan N, Gall H, et al. Cross - project defect prediction: a large scale experiment on data vs. domain vs. process[C]//In: ESEC/FSE 2009. Amsterdam: ACM, 2009: 91 - 100.

[2] Ostrand T J, Weyuker E J, Bell R M. Automating algorithms for the identification of fault - prone files[C]//In: Proceedings of the 2007 international symposium on Software testing and analysis. London, United Kingdom: ACM, 2007: 219 - 227.

[3] 李兴国，舒艳华，李 嘉. 基于支持向量机的软件可靠性早期预测[J]. 合肥工业大学学报：自然科学版，2007(7): 859 - 863.

[4] Xing F, Guo P, Lyu M R. A novel method for early software quality prediction based on support vector machine[C]//In: Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering. Citeseer: [s. n.], 2005: 213 - 222.

[5] 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.

[6] Schölkopf B, Smola A J, Williamson R C, et al. New Support Vector Algorithms[J]. Neural Computation, 2000, 12(5): 1207 - 1245.

[7] 董 琳. 基于 UML 的软件度量[J]. 计算机工程，2008(22): 55 - 56.

[8] 王 悠，罗燕京，易福华，等. 基于用例的软件复杂度估算及应用[J]. 计算机技术与发展，2007, 17(7): 196 - 199.

[9] 姚 琨. 基于 UML 的需求分析模型和设计模型的度量研究[D]. 合肥：合肥工业大学，2006.

[10] Chidamber S R, Kemerer C F. A metrics suite for object oriented design[J]. IEEE Transactions on Software Engineering, 1994, 20(6): 476 - 493.

[11] Zimmermann T, Premraj R, Zeller A. Predicting defects for eclipse[C]//In: ICSE 2007 Workshops, PROMISE'07. [s. l.]: [s. n.], 2007.

[12] Chalimourda A, Schölkopf B, Smola A J. Experimentally optimal ν in support vector regression for different noise models and parameter settings[J]. Neural Networks, 2004, 17(1): 127 - 141.