Software defect prediction using Bayesian networks笔记

文章中文名是基于贝叶斯网络的软件缺陷预测，在早期的软件缺陷预测研究中，静态代码特性被更多地使用。但后来人们认识到，除了静态代码度量对缺陷预测的影响之外，其他度量（如过程度量）也是有效的，应该加以研究。例如，Fenton和Neil（1999）认为，单靠静态代码度量无法准确地预测软件缺陷。为了支持这一观点，我们认为，如果一个软件有缺陷，这可能与以下因素之一有关：项目的规范可能由于相互矛盾的需求或缺少的功能而出错。它可能太复杂，无法实现或没有很好的记录。设计可能很差，可能没有考虑到所有的需求，或者可能错误地反映了一些需求。开发人员没有足够的资格参与该项目。可能存在项目管理问题，软件生命周期方法可能没有得到很好的遵循。软件的测试可能不够，因此在测试期间可能无法修复某些缺陷。以上因素都与静态代码度量无关，并且所有这些因素都可能很好地影响缺陷倾向性。所以，问题是哪些因素或指标对缺陷有效，我们如何衡量它们的影响？这个问题是本篇文章的主要研究内容。

本文的研究方法是使用贝叶斯网络研究度量之间的关系和度量指标对缺陷倾向的影响。贝叶斯网络是一种图形表示，它显示了我们感兴趣的一组变量之间的概率因果关系或影响关系。使用贝叶斯网络有几个实际的因素。首先，贝叶斯网络能够模拟一组变量对网络中另一个变量的概率影响。考虑到节点父母的概率，可以计算出他们孩子的概率。其次，贝叶斯网络可以解决数据丢失问题。贝叶斯网络的这一方面对于缺陷预测非常重要，因为缺陷预测数据集中的某些模块可能缺少一些度量。从软件或过程度量的全部或一个有效子集除了静态代码度量之外还必须一起考虑的角度来看，贝叶斯网络模型是同时考虑多个过程或产品度量并度量其效果的一个很好的候选模型。本文建立了度量与缺陷之间的贝叶斯网络，以衡量哪些度量对缺陷的影响更为重要，并探讨它们之间的影响关系。通过对该网络的学习，我们发现了整个软件系统的缺陷概率、度量对准确预测缺陷的贡献程度以及度量与缺陷之间的概率影响关系。

在贝叶斯网络结构学习中，搜索空间由基于给定变量（节点）的有向无环图的所有可能结构组成。通常，如果没有启发式方法，很难枚举所有这些可能的有向无环图。因为，当节点数增加时，搜索空间呈指数增长，几乎不可能搜索整个空间。给定一个数据集，由Cooper和Herskovits提出的K2算法，启发式地搜索最可能的贝叶斯网络结构（Cooper和Herskovits 1992）。该算法根据节点的排序，为每个节点寻找父节点，增加了贝叶斯网络的得分。如果将某个节点添加到节点的父节点集不会增加贝叶斯网络的得分，K2将停止进一步查找节点的父节点。由于贝叶斯网络中节点的排序是已知的，因此搜索空间比不使用启发式方法需要搜索的整个空间要小得多。此外，已知的排序可以确保贝叶斯网络中没有循环，因此也不需要检查循环。

作者引入了一个新的度量标准，称之为缺乏编码质量（LOCQ），用来度量源代码的质量。在Netbeans中运行PMD源代码分析器插件，为选定的每个开源Apache项目类生成LOCQ值。PMD检查给定的源代码并查找潜在的问题，如可能的错误、死代码、次优代码、过度复杂的表达式和重复代码。它计算软件系统中每个类和包检测到的问题数。作者认为，这种度量给出了源代码质量的概念，并且与缺陷有关系。正因为如此，作者在实验中加入了LOCQ度量，并试图了解它与缺陷倾向性和其他文献中众所周知的静态代码度量之间的关系。

为了用K2算法学习贝叶斯网络，需要指定节点的顺序。这就是为什么，作者决定在生成贝叶斯网络之前，考虑到软件度量对缺陷的影响，对软件度量进行排序，根据作者分析生成三组度量，其中LOC、CBO和LOCQ位于Group1中，WMC和RFC位于group2中，LCOM、DIT和NOC位于Group3中。1组计量比2组计量更重要，2组计量比3组计量对缺陷的影响更重要。

作者没有详细介绍实验实施过程，详细步骤在WEKA中实施。本文主要介绍了实验结果，实验结果的比较篇幅占了一大半。本文的最终实验结论是：作者使用Promise数据仓库中的9个数据集，证明RFC、LOC和LOCQ对缺陷倾向性的影响更大另一方面，NOC和DIT对缺陷的影响是有限的，不可信的。

读完文章最直接的感受是实验数据的分析以及与同类型的实验相比较很重要，另一方面本文也提供了一个实验思路：将其他软件和过程度量包含在贝叶斯网络的模型中，以揭示它们之间的关系，并确定在缺陷预测中最有用的度量。不必处理大量的软件度量，只关注最有效的度量将提高缺陷预测研究的成功率。