DOI:10.19850/j.cnki.2096-4706.2020.21.004

数据挖掘在软件工程领域中的应用探讨

张海霞

（广州华南商贸职业学院，广东 广州 510650）

摘　要：数据挖掘技术是当前发展速度最为迅猛的技术之一，为“互联网 +”相关大数据技术的发展提供了助力，其应用对于数据整理以及分析都有非常重要的影响。广州华南商贸职业学院针对数据挖掘技术在软件工程中的应用进行了分析研究，对数据挖掘技术进行了理论分析，并以软件知识库为例分析数据挖掘技术在软件知识库中的应用，深入分析当前数据挖掘技术具体应用，为数据挖掘技术在软件工程领域发展提供参考。

关键词：数据挖掘；软件工程；挖掘类型；软件知识库

中图分类号：TP311　　**文献标识码：**A　　**文章编号：**2096-4706（2020）21-0013-04

Discussion on the Application of Data Mining in the Field of Software Engineering

ZHANG Haixia

（Guangzhou South China Business Trade College，Guangzhou 510650，China）

Abstract：Data mining technology is one of the most rapidly developing technologies at present，which provides a boost for the development of “internet plus” related big data technology，and its application has a very important impact on data collation and analysis. Guangzhou South China Business Trade College conducted analysis and research on the application of data mining technology in software engineering，and conducted theoretical analysis on data mining technology. The application of data mining technology in software knowledge base is analyzed by taking software knowledge base as an example. The specific application of current data mining technology is deeply analyzed to provide reference for the development of data mining technology in software engineering field.

Keywords：data mining；software engineering；mining type；software knowledge base

0　引　言

随着用户对软件系统要求的不断提升，越来越多大型软件系统开发人员需要不断地优化软件代码设计，以满足用户日益增长的需求，但是在代码优化过程中需要面临各种各样的问题。例如，大型软件开发人员必须使用一些有效的方法、工具去理解软件系统的具体框架，之后对框架传播源代码进行优化与变更，在这过程中良好的系统开发文档，可以帮助开发人员更好地理解大型软件系统，但遗憾的是大多数软件系统开发人员并没有记录自己工作的习惯，能够留存下来的文件大多残缺不全，因此一旦需要对大型系统进行修改或者完善时，便需要对系统源代码进行查看，但是此种方式所需要耗费的时间往往比较长，很大程度上会出现难以遵守软件交付周期的风险。因此为了应对在大型系统开发中相关文档残缺不全等问题，通过对 Subversion（SVN）软件中系统修改或者更新的历史信息深度挖掘，加快对大型系统软件框架理解的进程，便于开发人员在软件更新前期快速实现新功能模块添加与源代码更新。针对此种情况通过将数据挖掘技术运用在知识库中，以实现最终提升源代码修改与新模块更新效率的目的。广州华南商贸职业学院针对上述情况，提出一种数据挖掘在软件工程领域中的应用方法，为教学研究打下良好基础。

收稿日期：2020-10-09

1　数据挖掘技术的基本概述

当前，社会正处于信息化时代，信息数据是社会发展过程中的重要产物，对数据信息的有效处理，在一定程度上关系着社会发展的生产力，对于信息技术的应用也有非常重要的作用 [1]。在信息技术应用的过程中，软件开发应用是其重要组成部分，而在软件开发应用的过程中，数据的分析和处理是其核心环节，数据挖掘技术的应用可以提升软件开发的效果，保证软件开发应用更加精准 [2]。

数据挖掘技术在具体应用的过程中也是一项复杂的技术应用，其主要包括信息收集、数据集成、数据规约、数据清理、数据变换、数据挖掘过程、功能模式测试及评估等不同部分组成，对数据能够进行有效的处理 [3]。

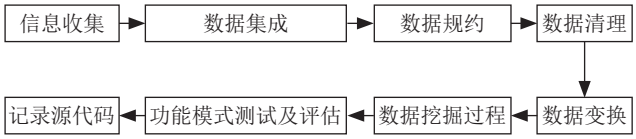

图 1　数据挖掘具体流程

图 1 为数据挖掘的具体流程，图中反映不同流程分担着不同的数据处理任务，信息收集任务可以将软件系统中相关信息整合收集出来；数据集成在相应空间内，利用预先设定好的数据规约对数据进行筛选，将其中对软件系统运行、更新等完全无作用的多余数据信息清除掉；根据软件系统开发功能删减、优化等需要改变软件系统部分代码，重新

构建软件框架，改变原有软件各子系统交互关系；软件进行功能模式测试及评估，记录优化后系统源代码。

2 分析数据挖掘技术在软件知识库中的应用

2.1 软件解构

在软件系统优化过程中需要先对软件系统本身框架进行分析与理解，便于掌握软件总系统与各个子系统之间关系，及系统之间存在的交互关系等，便于将这些设计相关信息均存储在软件知识库中。通常在对软件系统框架理解中会从框架源代码入手，采用假设 - 比较 - 调查 - 再研究这一循环研究方式对软件框架进行解读，最终达到完全理解框架的目的。在不同的框架解读环节，需要做的工作不同。

首先，假设阶段需要对软件系统总系统与子系统交互关系进行假设。软件开发人员可以利用陈旧的系统文档或者同高级技术人员进行交流，对相似软件系统结构进行推测，并根据推测形成对软件系统框架的初步假设与理解 [4]。例如，在对操作系统分析中，技术开发人员会对系统可以实现的功能，对各个子系统之间交互关系进行验证分析。其次，比较阶段对已经提出的假设进行求证。这一环节需要对比系统框架建设与系统实际功能实现的统一性。再次，调查阶段属于软件框架解构最为耗时的阶段。此时开发人员以各子系统之间差异与关系分析为目的，对预测的子系统之间交互关系进行测试，并在测试中准确判断两个子系统或者多个子系统之间存在的关系的真实性，通常此阶段主要运用的方式包括查看设计文档、源代码、向高级技术人员询问等方式。为了节省开发人员在调查过程中所需要的时间，研究中使用的数据挖掘技术采用软件反射框架方法，并在此基础上增加源代码启发式方法，以实现快速帮助开发人员了解软件系统框架的目的 [5]。

2.2 反射框架分析

反射框架分析为软件框架解构主要应用办法，此处对数据挖掘与应用更为深入，具体如图 2 所示。

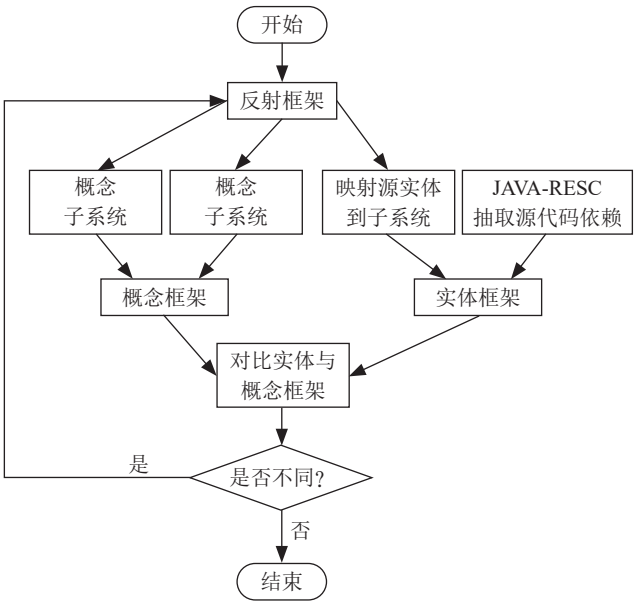

图 2 反射框架应用于软件框架结构的过程

图 2 表明软件开发技术人员可以利用从软件中获取的相关软件系统认知，对软件各子系统之间交互关系进行初步判断，并建立实际系统的目录中源代码、系统文件的映射关系，后对形成的概念框架与图实体框架进行对比，分析二者之间存在的差异，进而分析软件各子系统之间已经被证实存在的交互关系、未预测的交互关系、缺失的交互关系等。通过这种反射框架有利于软件技术开发人员更为深入掌握软件系统框架实际情况，并不断掌握软件各子系统之间准确的交互关系 [6]。在实际软件设计过程中根据软件系统各个功能的深入开发，对当前软件系统各子模块交互关系进行调整，以使得子系统交互可以达到预定功能的目的。因此在子系统关系探究上，除了是对当前系统框架进行掌握外，也是对系统功能拓展开发的重要环节。可以通过删除或者增加软件各子系统的交互关系实现对整体软件系统功能的进一步完善 [7]。具体分析源代码为：

```
// 真正的 make 方法，它直接调用了 resolve 继续去实现 make 的功能
// $abstract = 'HelpSpot\API'
public function make($abstract, array $parameters = [])
{
 // $abstract = 'HelpSpot\API'
 return $this->resolve($abstract, $parameters);
}
...
protected function resolve($abstract, $parameters = [])
{
 ...
 // 判断是否可以合理反射
 // $abstract = 'HelpSpot\API'
 if ($this->isBuildable($concrete, $abstract)) {
  // 实例化具体实例 ( 实际并不是实例化 , 而是通过反射“解刨”了 )
  $object = $this->build($concrete);
 } else {
  $object = $this->make($concrete);
 }
 ...
}
public function build($concrete)
{
 // $concrete = 'HelpSpot\API'
 if ($concrete instanceof Closure) {
   return $concrete($this, $this->getLastParameterOverride());
 }
 // 实例化反射类
 $reflector = new ReflectionClass($concrete);
 // 检查类是否可实例化
 if (! $reflector->isInstantiable()) {
  return $this->notInstantiable($concrete);
```

```
}
$this->buildStack[] = $concrete;
// 获取类的构造函数
$constructor = $reflector->getConstructor();
if (is_null($constructor)) {
 array_pop($this->buildStack);
 return new $concrete;
}
$dependencies = $constructor->getParameters();
$instances = $this->resolveDependencies(
 $dependencies
);
array_pop($this->buildStack);
// 从给出的参数创建一个新的类实例。
return $reflector->newInstanceArgs($instances);
}
```

2.3 静态依赖图（源代码）应用

无论是计算机软件亦或是手机 APP 等软件，在开发过程中软件系统是不断演化的，以满足不断变化的用户需求，保持软件始终具有较强的竞争力，尽可能延长软件的生命周期。因此，在软件更新过程中软件开发技术人员需要对软件源代码进行更新，而一般一款软件在源代码更新上并非是一人或者同一批技术人员可以完成的，往往需要诸多软件开发技术人员参与，为了避免不同软件开发技术人员在软件开发中出现个体化差异而影响软件代码更新，在实际软件工程中会运用 SVN 对软件版本进行控制，以明确记录软件系统源代码修改及更新的具体情况，这些记录软件系统更新、修改等信息会存储在 SVN 库中 [8]。考虑到 SVN 存储的文件信息并不能准确描述系统层次关系，因此使用相适宜的源代码实体，可准确判断是删除依赖关系或者是添加依赖关系。之后将特征属性关联到相应的映射源代码实体上，如此一来再对系统进行更新更加简单易于操作 [9]。

为了实现静态依赖关系图自动添加，对 SVN 库中数据进行分析。主要分为两个步骤：

（1）对源代码文件不同历史版本进行识别，并准确识别源代码实体（已经定义），记录已经定义源代码实体的内容与名字。例如，在库中存在两个 B 版本的文件，初始版本中该文件有函数 4 个，第二个版本中函数增加了一个新的函数，有 5 个函数。进而通过这些源代码实体，比对初始版本可以准确掌握软件系统更新与修改情况。

（2）采用历史符号标记表，对相同版本源代码文件的不同历史版本进行重新定义，进而实现对系统不同版本源代码实体的多个历史版本快照。通过对 SVN 库中源代码文件不同历史版本记录比对，将源代码注解到某一个历史版本删除或者新增交互关系上，进而通过源代码注解情况分析不同历史版本之间交互关系，如图 3 所示。再之后，逐渐分析不同历史版本交互关系图，获得一个软件工程项目周期内各系统历史交互关系图。借助软件系统历史交互关系图，便于软件技术开发人员掌握软件各系统具体交互关系，并通过源代码注解读取，使得软件技术开发人员可更为快速掌握软件系统框架层次情况 [10]。

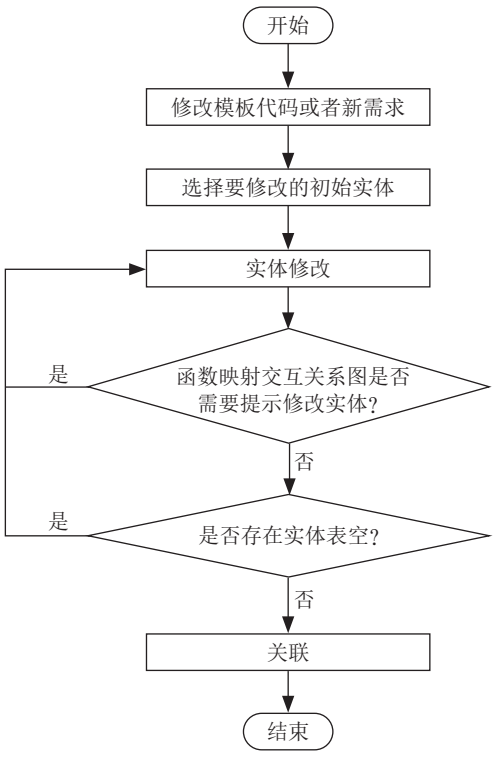

图 3 静态依赖关系图自动添加流程

3 GBOMC 软件系统简化案例

3.1 GBOMC 软件静态依赖图应用过程

为了尽可能节省软件系统代码简化过程中可能出现的重复工作等，提高软件系统开发与优化效率，对 GBOMC 软件系统进行实验，由于这个软件此系统代码库数据繁多，利用如上静态依赖图自动添加流程逐渐构建历史交互图用时 1.2 小时，并将最终生成的历史系统交互图结果存储在 XML 文件中。在对该 GBOMC 软件系统交互关系调查中，一直重复使用 XML 文件，大大节省了系统层次结构所需要的时间。随着软件系统不断地发展，为初始版 XML 文件添加相对应源代码注解，并在 XML 文件 SVN 库更新上，使用优化的 gSpan 算法，有效降低重复图出现的概率，确保依赖图完全集，具体代码为：

```
1:retrieve XMIL to sort the labels in by modify time
2: label dependent fie vertices and edges;
3: add the sourcenotes to edges;
4: s1-all link: 1-edge graphs in;
5: sortsl in DFS lexicographic order by time;
6: S ← s1:
7: for each edge e ∈ 1 do
8:    initialize s with e,ets  D by graphs which contains e
9:       grapMining(D,S,s);
10:      add the source notes to edges:
11: if S,contains(s)
12:      remove sl;
13:  graphMining(D,S, s);
```

14: if s ≠ min(S)

15: return;

16: S ← SU{s):

17:enumerate s in each graph in and count its link class;

通过以上优化的 gSpan 算法可以使得对软件系统各子系统交互关系理解更简便，同时通过一个系统源代码实体的更改可以实现对多个子系统相关源代码实体的更改，更为高效率更新软件系统代码。

3.2 GBOMC 软件应用静态依赖图后测试

测试环境为：

（1）软件：Windows 10，Eclipse 软件平台，Java 编写算法。

（2）硬件：1 GB 内存、PM 1.6 GHZ CPU、80 GB 硬盘、笔记本电脑。

测试结果如表 1 与表 2 所示，结果显示此种方式算法性能较优、修改历史记录完整，有一定深入研究与应用价值。

表 1 GBAM、GBOMC、CNMC 源代码实体修改历史记录表

软件系统	所有修改记录	一般维护记录	添加新实体记录	关联修改记录
GBAM	26 800（100%）	5 326（20%）	6 989（26%）	14 485（54%）
GBOMC	18 653（100%）	3 790（20%）	5 272（28%）	9 591（52%）
CNMC	29 809（100%）	6 953（23%）	7 096（24%）	15 760（53%）

表 2 GBOMC、CNMC、GBAM、Average、F−measure 算法性能表（测试三次）

软件系统	第一次		第二次		第三次	
	Recall	Precision	Recall	Precision	Recall	Precision
GBOMC	0.30	0.65	0.35	0.62	0.43	0.49
CNMC	0.26	0.63	0.32	0.60	0.40	0.54
GBAM	0.29	0.59	0.34	0.57	0.38	0.54
Average	0.27	0.62	0.33	0.56	0.41	0.50
F-measure	0.49		0.46		0.45	

4 结 论

笔者以软件知识库为例分析数据挖掘技术在软件知识库中的应用，分析软件解构、反射框架、静态依赖图（源代码）应用等，并对使用的静态依赖图应用方式进行测试，此种方式算法性能优越，准确性高，可以实现对系统的进一步开发，应用价值较高。

参考文献：

[1] 张勇．软件工程行业中数据挖掘的应用探讨 [J]. 中国新通信，2018，20（5）：89.

[2] 段彬，魏巍．数据挖掘在软件工程领域中的应用浅析 [J]. 信息系统工程，2018（4）：89.

[3] 李喆．数据挖掘技术在软件工程中的应用研究 [J]. 数码设计（下），2019（11）：216−217.

[4] 黄智聪．数据挖掘技术在软件工程中的应用 [J]. 数字化用户，2019，25（17）：104.

[5] 钱晓军，范冬萍，吉根林．物联网差异数据库中的故障数据快速挖掘仿真 [J]. 计算机仿真，2016，33（1）：301−304.

[6] 黄炜．基于数据挖掘技术的计算机网络病毒防御系统设计 [J]. 机电信息，2020（23）：140−141.

[7] 郑幸源，洪亲，蔡坚勇，等．基于 AJAX 异步传输技术与 Echarts3 技术的动态数据绘图实现 [J]. 软件导刊，2017，16（3）：143−145.

[8] 宁德军，叶培根，刘琴，等．基于存储库数据挖掘的开源软件成功度量方法 [J]. 电子学报，2018，46（12）：2930−2935.

[9] 张波，李舸．基于改进聚类算法的 Web 异常数据挖掘软件设计 [J]. 现代电子技术，2019，42（8）：73−76+81.

[10] 梁艺琼．基于数据挖掘技术的舆情分析系统的设计 [J]. 电脑知识与技术，2020，16（3）：1−2.

作者简介：张海霞（1979.12—），男，汉族，湖北荆州人，讲师，硕士研究生，研究方向：软件工程。

（上接 12 页） 上线正常运营。

参考文献：

[1] 李子若．新媒体时代下网络表情包的特征及传播功能 [J]. 今传媒，2020，28（2）：16−19.

[2] 胡远珍．网络社交中表情符号的表达与象征意义分析 [J]. 湖北大学学报（哲学社会科学版），2017，44（6）：147−154+169.

[3] 胡国强，周兆永，信朝霞．基于 SRS 的开源直播系统的设计与实现 [J]. 现代电子技术，2016，39（16）：36−39+43.

[4] 冯青．基于云计算的视频存储和播放系统设计与实现 [D]. 长沙：湖南大学，2017.

作者简介：邱慧玲（1991—），女，汉族，江西上饶人，助教，硕士，研究方向：人工智能、网页设计与制作。