软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(6):1488−1497 [doi: 10.13328/j.cnki.jos.005229]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

小模型大数据:一种分析软件行为的代数方法*

俞一峻[1], 刘 春[2]

[1](School of Computing and Communications, The Open University, Milton Keynes MK76AA)

[2](河南大学 计算机与信息工程学院,河南 开封 475001)

通讯作者: 刘春, E-mail: liuchun@henu.edu.cn

摘 要: 问题框架方法分析软件需求时需要通过借助领域知识及其之间的结构关系来论述用户的需求是可以被软件系统满足的.这类定性的可满足性论述支持早期需求决策,选择合理的软件体系结构和设计方案.但是,当前的移动软件需求方是偏好各异的用户个体,需求差异化明显,而且根据应用场景,这些需求会动态地发生变化.在这种情况下,现有的定性分析方法不再适用.大数据分析提供一种数据驱动的深度学习机制,为很多实践者采用.但依靠数据驱动的软件分析往往就事论事,仍然不能从根本上提供一个合理的论述来说明大量软件用户的需求到底是什么,也无法对可信软件的安全性和私密性提供可靠的论证.再多的数据也只能提供统计意义的表象,而无法彻底防范借用漏洞的攻击.尝试从提炼软件抽象目标行为的角度进一步深化问题框架的研究思路,针对各类个体行为建立概率模型,提出一种基于模型代数分析的方法,以避开纯粹数据驱动思路的大数据分析盲点.通过对安全和隐私性问题的分析,对所提出的方法可用性及局限性进行探讨,对未来大数据软件需求研究给予一定的启示.

关键词: 软件需求分析;概率模型;运营需求;代数分析;安全和私密性

中图法分类号: TP311

中文引用格式: 俞一峻,刘春.小模型大数据:一种分析软件行为的代数方法.软件学报,2017,28(6):1488−1497. http://www.jos.org.cn/1000-9825/5229.htm

英文引用格式: Yu YJ, Liu C. Little model in big data: An algebraicapproach to analysing abstract software behaviours. Ruan Jian Xue Bao/Journal of Software, 2017,28(6):1488−1497 (in Chinese). http://www.jos.org.cn/1000-9825/5229.htm

Little Model in Big Data: An Algebraic Approach to Analysing Abstract Software Behaviours

YU Yi-Jun[1], LIU Chun[2]

[1](School of Computing and Communications, The Open University, Milton Keynes MK76AA)

[2](School of Computer and Information Engineering, He'nan University, Kaifeng 475001, China)

Abstract: The problem frame method typically uses domain knowledge in order to demonstrate that a software system can satisfy the requirements of stakeholders by specifying how machine relates to stakeholders' problems. Qualitatively, satisfiability discourse can guide a software engineer to make early decisions on what the right solution is to the right problem. However, mobile apps deployed to app stores often not only need to accommodate millions of individual users whose requirements have subtle differences, but also may change at runtime under varying application contexts. Requirements of such apps can no longer be analyzed qualitatively to cover all situations. Big data analysis through deep learning has been increasingly adopted in practice to replace deep requirements analysis. Although effective in making statistically sound decisions, the conclusions of pure big data analysis are merely a set of unexplainable parameters, which cannot be used to show that individual users' requirements are satisfied, nor can they reliably validate the trustworthiness and dependability in terms of security and privacy. After all, training with more datasets could only improve statistical significance, but cannot

* 基金项目: 欧洲研究理事会高级研究基金(291652); 国家自然科学基金(61300035)
Foundation item: European Research Council Advanced Grant (291652); National Natural Science Foundation of China (61300035)
收稿时间: 2016-10-09; 修改时间: 2016-10-26; 采用时间: 2016-12-22; jos 在线出版时间: 2017-02-20
CNKI 网络优先出版: 2017-02-20 15:06:04, http://www.cnki.net/kcms/detail/11.2560.TP.20170220.1506.024.html

prevent software systems from the malicious exploitation of outliers. This paper attempts to follow Jackson's teaching of abstract goal behaviors as intermediate between requirements and software domains, and proposes an algebraic approach to analyzing the consequences of probabilistic software behavior models, so as to circumvent some blind spots of purely data-driven approaches. Through examples in security and privacy areas, the challenges and limitations to big data software requirement analysis are discussed.

Key words: software requirement analysis; probabilistic model; runtime requirement; algebraic analysis; security and privacy

软件需求分析的基本思路是了解和获取利益相关者对问题的陈述,把人的抽象需求跟机器实现的解决方案对接,并用关于现实世界的领域知识(比如目标、功能、约束)诠释为什么这些抽象的需求可以被具体的解决方案满足.20 年前,Zave 和 Jackson 总结出的可满足性论述(satisfaction argument)形成了软件需求的基本定义[1],并且指出:作为软件工程的一个分支,需求工程的主要目的,是用获取的问题描述进一步准确刻画软件的行为及其演化.之后 Jackson 正式提出了称为问题框架的一整套方法[2],其基本思想是分而治之,合而治之,把大的问题分解为子问题,再把子问题的解决方案组合在一起.如果用图形的方式,这样的结构可以展示为问题框架图,基本结构如图 1 所示.不同类型的问题,相应的领域知识呈现不同的框架,展现不同的关注点.对此感兴趣的读者可以参考文献[2]以及 Jackson 的一系列著述.值得一提的是,需求基本问题或者其语义是可以独立于图形的形式化描述的.

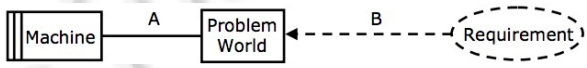

Fig.1 Conceptual diagram of problem frame approach

图 1 问题框架概念图

图 1 中,如果用 W 代表问题领域,S 代表软件系统解决方案,R 代表需求,三者之间存在如下的可满足性论述公式:

$$W,S\vdash R \tag{1}$$

该可满足性公式所表达的是:当 S 所代表的软件解决方案嵌入到 W 所代表的问题领域时,使得用户的需求 R 得到满足.在该公式中,关于$\vdash$语义的逻辑解释是,把包括需求在内的这些符号理解为逻辑性质的交集.这样得到的理解通常是定性的,即非真即假.这样做的好处是:通过对不同领域性质的取舍和描述,软件工程的视野扩大到软件程序世界之外的现实世界,能够在项目的早期有效地看到什么是合适的问题以及合适的方案(比如软件体系结构和设计).

在问题框架的需求分析实践中,由于突出了可满足性形式化定性论述的逻辑性,反而对软件和系统行为本身需要借助于其他的建模工具(比如 UMLStatecharts 状态图)在设计后续阶段加以分析.同时,非形式化的需求描述(比如自然语言)通常相对模糊和抽象,不利于分析软件实现和需求之间的联系,消除那些由于表达的原因而存在的含糊性.有鉴于此,Jackson 在近期的论述中[3]更加强调了抽象目标行为(abstract goal behaviour)的重要性,如图 2 所示.

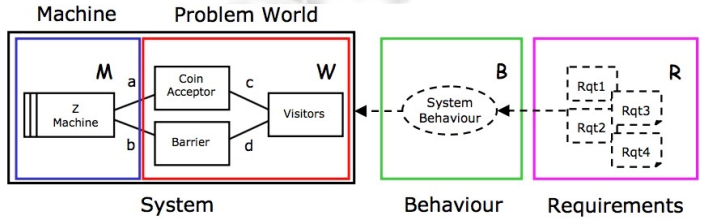

Fig.2 Conceptual diagram after introducing the concept of abstract goal behavior

图 2 引入抽象目标行为之后的问题框架概念图

解释抽象目标行为需要从软件需求的目标模型说起.目标是软件期望达成的状态,为了达成这个状态,软件系统必须从当前状态通过若干步的迁移,经过 0 到多个中间状态[4].当然,也可能存在障碍暂时或者一直无法达

到理想中的目标状态.从对已有软件程序抽取目标模型的逆向工程实践中[4],我们发现目标的时序分解和状态图的层次结构之间存在着天然的联系.这种对应可以帮助我们提炼软件行为中的设计目标:功能需求的“与”分解可以体现为对应状态迁移的串行或者并行组合,而非功能需求的分解可以落实为行为模型的可变性和或然性.当软件程序行为的目标模型可以抽取时,我们看到 Jackson 所述的抽象目标行为是可以落实为软件实现的.那么,当现实世界的领域实体的行为也能够用抽象行为刻画时(除了状态机,文献[3]中还采用了 Jackson 结构化正则表达树描述那些相对规范化的状态模型),软件实现和领域描述之间、领域描述和抽象目标之间就有了完全统一的形式化模型表示,便于进一步形式化分析和验证[3].

通过引入抽象目标行为(B),问题框架结构要素行为之间的关系可以更精确一致地表达,从而具体地揭示软件需求.问题领域行为 W 和抽象目标行为 B 之间的一个重要区别在于:问题领域行为是独立于软件机器实现行为 M 存在的;而抽象目标行为是伴生于软件系统行为的,随之启动,也随之终结.

有了统一的行为模型,相对而言,问题框架分析更容易形式化了.但是仅仅定性地分析是不够的.这主要体现在近年来出现的小众软件如移动 APP 的普及,从而大众可以随时根据个性化的需求选择差异化的解决方案,而不是依赖传统意义的大而全的软件包.这就直接给软件需求分析带来一个新的问题——怎样随时随地分析不同个人的个性化和差异化的需求?从移动 APP 软件的软件开发者的角度看来,这个问题已经超出了软件系统本身的范畴:从业务的角度出发,也必须考虑 1%用户的特定需求,只要这 1%的用户是更可能带来收益的;同时,如果另外 1%的用户提出的功能特性需要大量的工作量,或者造成安全和私密性的风险,即使听上去是可行的,也需要三思而后行.尤其是,以上的考量会随着软件行为的迁移而变化,在使用软件的不同阶段会有不同的效果.比如,一个简单的例子是植入 Google AdSense 广告的播放可以是用户可跳过的或者可忽略的(条幅式),也可以是不可跳过必看的(全屏间歇插入式),在 APP 的不同阶段,插入这些广告可能产生截然不同的效果.

考虑到不同用户在不同时刻的不同需求,目前的通用做法是:通过 Data Analytics 搜集日志数据分析,然后通过大数据深度学习找到规则性的规律,从而自动调整软件的设计.但是,正如我们前面阐述的,这样数据驱动的学习结果通常是基于统计规律的参数化模型(比如神经网络的非线性算子的参数),很难加以解释,得到的模型也不易在不同软件之间重用.因此,我们需要一个能够用软件和用户的状态模型解释的“小”模型,来论述解释软件行为和后果的可满足性.这个“小”模型高度抽象,因此不必枚举所有的状态序列,而是根据专家的经验,估算状态变迁环节的发生概率,并且在代数分析时不必填入具体的数值,直到有了具体实际观察数据再加以调整.这样保证在需求分析阶段,即使没有大数据对现象也可以有一定的预判.当安全和私密性需求不能简单地进行统计分析时,也可以不受统计数据的干扰做最坏情况判定.

1 软件行为的基本概率模型

1.1 基本定义

首先把软件行为定义为一个输入/输出状态自动机,见定义 1.

定义 1. Moore 状态自动机是六元组$(S,\Sigma,\Delta,\delta,\lambda,s_0)$,$S$ 是状态集合,其中包括 s_0 是初始状态;Σ是输入字母集合;Δ是输出字母集合;$\delta: S\times\Sigma\to S$ 是状态迁移函数;$\lambda: S\to\Delta$是状态结果输出函数.

为了量化表示行为的出现和结果,我们在自动机基础上进一步引入概率和效果函数,见定义 2.

定义 2. 定量化的软件行为模型是八元组$(S,\Sigma,\Delta,\delta,\lambda,s_0,\pi,I)$,其中,在定义 1 基础上增加了两个元素:$I: \lambda\to R$ 是状态效用实函数;$\pi: \delta\to[0,1]$是一个状态迁移的概率函数,并且所有迁出的概率和为 1 或 0.

$$\sum_{s'|(s,s')\in\sigma}\pi(s,s')=\{0,1\}.$$

注意:当 R 为$[0,\infty)$时,我们分析的是软件行为的价值;当 R 为$(-\infty,0]$,我们分析的是软件行为的风险;而当 R 为$(-\infty,\infty)$时,我们分析软件行为的成本绩效.

定义 3. 定量化软件行为模型的效用函数定义为遍历所有状态的概率加权平均总效用:给定一个起于 s_0 的状态迁移序列,其抵达最后状态的概率是遍历的所有状态迁移的概率之乘积,而总效用函数是所有可能状态迁

移序列到达最终状态的概率乘以其效用之加权总和.

$$p(s)=\prod_{k=0}^{n-1}\pi(s_k,s_{k+1}),$$

$$r^n(s)=i(s)\times p(s),$$

$$r^*(s)=\sum_{1}^{\infty}\sum_{(s_0,s)\in\delta^n} r^n(s).$$

定义 4. 定量化软件行为模型的效用函数定义为遍历所有状态的概率加权平均总效用:给定一个起于 s_0 的状态迁移序列,其抵达最后状态的概率是遍历的所有状态迁移的概率之乘积,而总效用函数是所有可能状态迁移序列到达最终状态的概率乘以其效用之加权总和.

$$p(s)=\lim_{n\to\infty}p(s_n), r^*(s)=\lim_{n\to\infty}r^n(s).$$

1.2 定量计算软件行为的效用函数

由于状态迁移图允许有循环回路,穷举所有状态迁移序列是不可能的.因此,传统上的做法是模拟足够多、足够长的序列,并且期待这样得到的效用函数是收敛的即可.但是实际上,这样的收敛性是无法保证的.

为了计算软件行为模型的效用函数,我们开发了一种符号代数分析的时间复杂度 $O(N^3)$的算法(如图 3 所示),使得:(1) 状态迁移图可以有任意循环回路;(2) 状态迁移概率函数可以是符号变量而不是常数;(3) 状态效用函数也可以是符号变量而不是常数;(4) 当计算无法收敛时,给出判定条件;(5) 总效用函数的全局最优值可以在所有约束条件下通过基于差分演化的优化算法找到.鉴于篇幅限制,以上算法,性质证明做了一定的简化,详见文献[5].所有的实现参见开源工具 https://github.com/yijunyu/demo-riskexplore.

Algorithm 1: Computing risks by simulating a stochastic decision making process through random walks.

Data: An exploration machine $(S, \Sigma, \Delta, \delta, \lambda, s_0, \mathcal{I}, \pi)$ per Definition 2;
t, maximal number of traces to simulate;
n, maximal length of traces to simulate;
Result: Approximated risks $r^n(s)$ per Definition 3 from the simulated traces

for c:=1 *to* t **do**
  for $l := 1..n$ **do**
    if $\sum_{(s_{l-1},s_k)\in\delta}\pi(s_{l-1},s_k)=0$ **then**
      break;
    end
    $r :=$ random$(0, \sum_{(s_{l-1},s_k)\in\delta}\pi(s_{l-1},s_k))$ where k indexes outgoing transitions of s_{l-1}; ▷ uniformly distributed random number
    Let $s_l = s_k$ where k is the minimal number so that $r < \sum_{(s_{l-1},s_k)\in\delta}\pi(s_{l-1},s_k)$;
    $r^*(s_l) := r^*(s_l) + i(s_l)$;
  end
end
$r^n(s) := r^n(s)/t$ for each state s;
return $r^n(s)$;

Algorithm 2: Computing the risks of an exploration machine algebraically by solving linear recurrence equations through LDU decomposition

Data: An Exploration Machine: $(S, \Sigma, \Delta, \delta, \lambda, s_0, \mathcal{I}, \pi)$
Result: A strictly lower-triangular matrix P' in the recurrence equations with updated constant vector $\vec{c}'$ corresponding to Equation (12); Likelihood $\vec{p}$ and risk $\vec{r^*}$ vectors;

n:=$|S|$;
Let I, J be respectively identity and anti-identity matrix;
$P_{i,j} := \pi(s_i, s_j)$ for $(s_i, s_j) \in \delta \wedge i, j \in [1..n]$; ▷ Equation (8)
let $MLDU$:=$J(I-P)J$; ▷ LDU decomposition Eqn.(13)
let $P' = I - JUJ$ be the new transition probability matrix, and $\vec{c}' = JD^{-1}L^{-1}M^{-1}J\vec{c}$ be the new initial probabilities; ▷ Equations (17) and (18)
$\vec{p} = JU^{-1}J\vec{c}'$; ▷ Equation (10)
for i:=1 *to* n **do**
  $r^*(s_i) := \mathcal{I}(\lambda(s_i))$ * p_i; ▷ Equation (11)
end
return $P', \vec{c}', \vec{p}, r^*(s)$;

Fig.3 Two algorithms for computing risks

图 3 计算风险收益的两种算法

算法 1 给出了如何通过模拟随机过程的方法计算总的风险收益,算法 2 给出了如何通过代数计算的方法给出随机过程收敛时总的风险收益.假设随机过程收敛,下面的推导证明两者是等价的.

首先,收敛后的或然性可以建立如下线性递推关系:

$$p(s_j)=\sum_{(s_i,s_j)\in\delta}(\pi(s_i,s_j)p(s_i)).$$

以上关系可以用线性代数方程概括:

$$\vec{p} = P\vec{p} + \vec{c}, \vec{p} \geqslant \vec{0},$$

其中,P 是概率迁移矩阵,c 是初始状态向量$(1,0,\ldots,0)^T$.这个方程可以改写为

$$(I - P)\vec{p} = \vec{c}.$$

其对应的解为

$$\vec{p} = (I - P)^{-1}\vec{c}.$$

最后,风险收益值为

$$\vec{r} = \vec{i}^T(I - P)^{-1}\vec{c}.$$

为了求得代数解而不是数值解,算法 2 对 $I–P$ 使用了符号 $M'U'D'L'$分解(即一种 LU 分解的变体),其中,M'为置换矩阵,U'为上三角矩阵,D'为对角矩阵,L'为下三角矩阵.逐步求得等价的方程如下(推导步骤从略,详见文献[5]).

$$\vec{p} = P'\vec{p} + \vec{c}',$$

$$P'=I–L'=I–JUJ,$$

$$\vec{c'} = D'^{-1}U'^{-1}M'^{-1}\vec{c} = JD^{-1}L^{-1}M^{-1}J\vec{c}.$$

其中,M'为置换矩阵,L'为下三角矩阵,U'为上三角矩阵,D'为对角线矩阵,J 为倒单位矩阵($J_{i,n+1-i}$=1,其他元素为 0).

1.3 定量分析软件行为后果的需求分析方法

图 4 给出了定量分析软件行为后果的需求分析过程:首先,这里需求获取的目标是获得软件的抽象目标行为模型和程序实现的行为模型,根据机器学习或者专家经验给出状态迁移的概率和状态效用函数,允许使用变量参数,也允许状态迁移的循环回路;其次,根据风险(或成本收益)计算算法,获得最小化(或最大化)的配置参数,如果无法通过调整系统做到最优,那么调整变量约束,返回第 1 步重新计算,直到能够达到可满足性需求(比如最小风险或者最大成本收益).

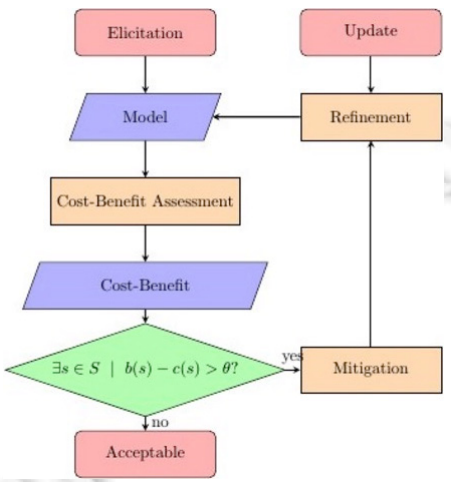

Fig.4 Requirements analysis process by quantitative analysis of software behavior effects

图 4 定量软件行为模型后果的需求分析的迭代过程

另外,系统的行为模型,包括抽象目标模型不是一成不变的,当不同的用户提出不同的需求时,事实上我们对软件行为的概率模型随时进行调整和适应.这也是一种数据驱动的方法,只不过我们考虑的因素不仅仅局限于用户的数据,而是包括了软件的行为本身以及用户自己的主观意愿.

1.4 小模型大数据需求分析方法的效果和局限性

最后必须指出的是:一切基于数据驱动的方法,包括我们提出的小模型大数据需求分析的方法,都受制于数

据本身的质量和数量.除此之外,我们的模型之所以称之为小模型,是相对于大数据而言的.我们的代数分析算法的复杂度决定了模型不可能非常大,但是由于充分地符号化和参数化,这样的小模型对潜在无限可能的数据也能有相当程度的代表性.

2 小模型大数据需求分析方法实践的初步实例

为了说明小模型大数据需求分析方法,这里列举了 3 类实际运用的例子.为了可读性做了一定的简化,以便于理解.

2.1 安全需求分析

信息安全需求主要分为保密性、完整性和可用性等.这里举例,说明行为模型中循环回路是很普遍的,因此,传统的概率模型检测算法无法一步到位地给出答案.

图 5 给出了一个登录猜测密码的软件行为模型.用 p 这个符号代表猜中的概率,那么当没有猜中时,用户以 $1-p$ 的概率停留在系统的开始状态.

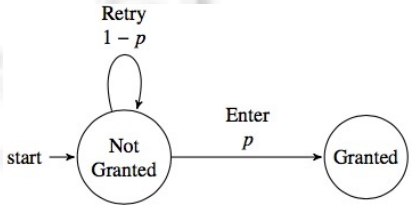

Fig.5 The probabilistic model of the behavior of inputting password

图 5 注册用户密码输入行为的概率模型

分析这个模型的风险,假设登录前为 $-O$(用户的开销),登录后为 V(比如价值 V 的银行账号),那么,使用算法 2 计算发生损失的风险估算值为 $-O/p+V$.因此,如果 p 足够小,$O/p>V$ 就能够保证系统的相对安全性.这也是对 DoS 攻击防范的一个策略,让登录者必须花费一定的时间代价,从而避免其无限尝试.当然,登录设计也可以采取最多 3 次尝试的行为模型,其计算就不再有循环回路了(这里不加赘述).

从这样小的一个模型可以看到:即便是采用传统的大数据方法,也需要获得足够多的日志才能分析出风险的大小,比如搜集用户过去一万次登录中有多少次用户无法输入正确的密码来估算概率 p.正如前文所示,由于黑客的存在概率是无法根据过去的数据预测的,这个概率的估算更多的是凭借经验和对加密算法的信心.而采用小模型大数据需求分析的方法可以很快地找到极端条件下的答案,基本原理是:采用代数方式表示概率为变量后,能够表达极端情况的数值,因此不需要对变量值用实际训练集数据中的出现频率加以估计,因此能够用于对极端情况做定量的需求分析.虽然这种分析的准确程度还不能取代大数据分析的结果,但好处是:如果搜集的大数据带有片面性,也不易被偏差误导(比如美国大选中基于大数据学习得到的社交网络 Facebook 推荐假新闻为人诟病的问题).更为复杂的例子见文献[5].

2.2 私密性需求分析

其实对私密性,尤其是社交网络软件上信息共享的私密性也牵涉到行为分析.而且,这里的行为分析更多的是侧重在用户之间的行为上,分析其利弊权衡[6].这里,泄露隐私的风险和获得朋友点赞的好处是一对矛盾,因为朋友圈的行为是不受控制的,一旦其中的一个人把消息再分享出去,就会导致秘密不再局限在用户指定的小圈子里,从而违反了隐私需求;反之,从来不分享消息也就无从得到朋友的点赞和信任,进而无法获得更多的信息.

为了分析这对矛盾,我们可以考虑一个行为概率模型,如图 6 所示.当 Facebook 用户不同的操作行为的概率是变量时,我们可以看到,各种可能性都有.其中,一旦到达 reshare 状态时,用户的隐私可能泄露.因此,需要分析所有状态迁移序列的概率.如果使用 PRISM[7],这样的计算必须罗列所有变量的所有可能性.不仅如此,$S_9 \rightarrow S_2$ 和

$S_8 \rightarrow S_9$的迁移构成了循环回路,导致当*Pagain*=1和*Preply*=1时,总的风险效用函数计算是发散的,因此必须排除.

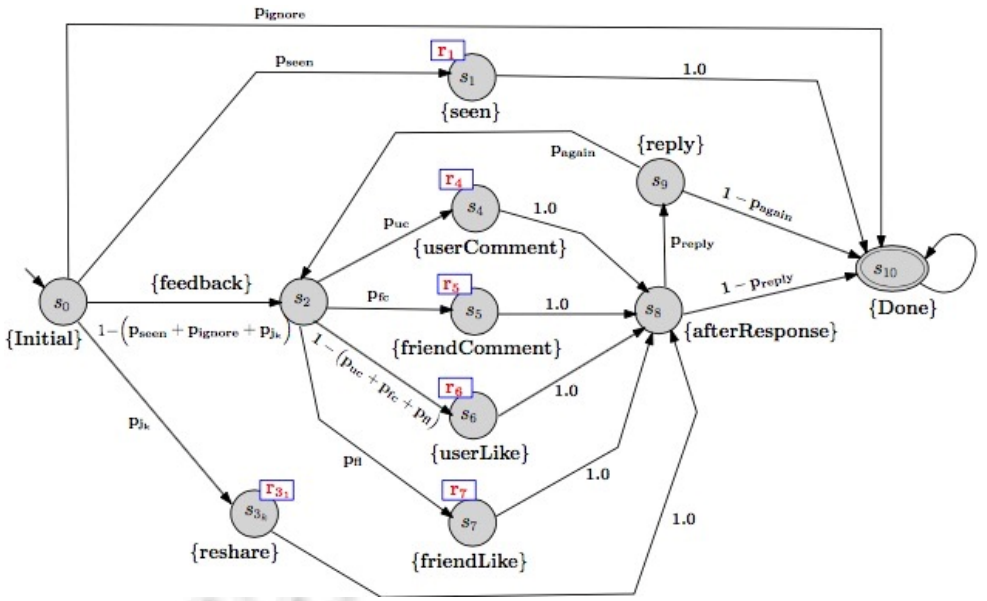

Fig.6 The probabilistic model of the behavior of sharing messages in social network

图 6 社交网络分享消息行为的概率模型

经过我们的符号代数分析,用算法 2 计算得出的私密泄露风险值为函数:

*pseen***r*1+*pjk***r*3–(*r*4**puc**(*pjk*–(1–*pseen*–*pignore*)–*pagain***preply***pjk*))/
(1–*pagain***preply***pfl*+*pagain***preply**(*pfl*–(1–*puc*-*pfc*))–*pagain***preply***pfc*–*pagain***preply***puc*)–
(*r*5**pfc**(*pjk*–(1–*pseen*–*pignore*)–*pagain***preply***pjk*))/
(1–*pagain***preply***pfl*+*pagain***preply**(*pfl*–(1–*puc*–*pfc*))–*pagain***preply***pfc*–*pagain***preply***puc*)+
(*r*6*(*pfl*–(1–*puc*–*pfc*))*(*pjk*–(1–*pseen*–*pignore*)–*pagain***preply***pjk*))/
(1–*pagain***preply***pfl*+*pagain***preply**(*pfl*–(1–*puc*–*pfc*))–*pagain***preply***pfc*–*pagain***preply***puc*)–
(*r*7**pfl**(*pjk*–(1–*pseen*-*pignore*)–*pagain***preply***pjk*))/
(1–*pagain***preply***pfl*+*pagain***preply**(*pfl*–(1–*puc*–*pfc*))–*pagain***preply***pfc*–*pagain***preply***puc*).

结论是:当敏感信息传播的概率为特定值时,最终的风险可以控制到最低.比如,假定所有非零影响因子*r*1=*r*3=*r*4=*r*5=*r*6=*r*7=1,利用 DEoptim 优化算法可以很快算出:当 *pseen*=0.006239582,*pignore*=0.987266657,*pjk*=0.003001324,*puc*=0.115949677,*pfc*=0.446085095,*pfl*=0.131866686,*preply*=0.003728548,*pagain*=0.048901284 时,总的风险可以降到 0.01273453.

如果采用数据驱动分析的方法,一个局限性在于 Facebook 社交网络的数据是不对外开放的,这样,作为外部用户很难直接度量其他用户的可信度.但是有了基于用户行为的小模型,即便没有实测的数据,也能对极端情况做定量的需求分析.

2.3 移动应用软件商业价值需求分析

这里,我们考虑一个游戏的广告效果分析.首先,假设游戏拥有两个版本:一个是收费版不提供广告,另一个是免费版提供广告.在免费版的界面上提供条幅广告条(只要执行程序总是显示),但是按照占据屏幕的面积有一定的概率会带来点击.另外,在免费版进行到通关的时候或者关闭窗口前,会按照 1/2 的概率弹出全屏广告窗口,推销公司的其他游戏.图 7 显示了这个状态迁移图的符号概率模型.同样,当用户重复完成固定的任务时,行为模型也是循环反复的.

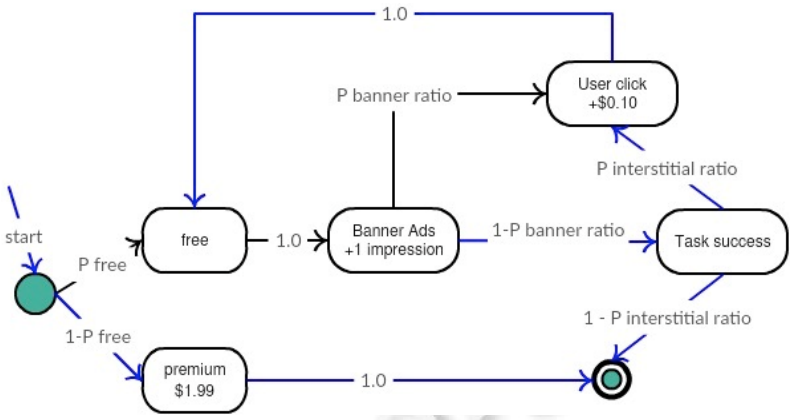

Fig.7 The probabilistic model of the advertising behavior of mobile applications

图 7 移动 APP 广告收费行为的概率模型

经过符号代数分析,用算法 2 计算得出的广告价值函数为

$$(1-pfree)*Premium+(Impression*pfree)/(1-(pbanner-pinterstitial*(pbanner-1)))-$$
$$(Click*(pinterstitial*(pbanner-1)-pbanner)*pfree)/(1-(pbanner-pinterstitial*(pbanner-1))).$$

假如购买(premium APP)收入为 1.99 美元,印象(impression)收入为 0.01 美元,点击(click)收入为 0.10 美元,那么为了最大化广告价值,DEoptim 优化结果为 *pfree*=1,*pbanner*=1,*pinterstitial*=1,亦即所谓 freemium 模式.根据计算结果得出的建议是:应该尽量争取提供免费的 APP,利用重复的广告收费.但是当指定 *pbanner*=0.1(即,用户只有十分之一的概率点击广告条)并且 *pinterstitial*=0.1(即,用户只有十分之一的概率容忍插播广告)时,优化结果为 *pfree*=0,亦即 premium 收费模式.

当游戏的收费用户比例较多时,尽量减少广告的投放;而当没有用户愿意付费时,应该增大广告的投放量.但是,一味地显示广告也会给用户造成困扰,需要在软件行为的特定阶段投放有效的广告,这些阶段应该是在满足用户的阶段性需求时.当然,这时的软件抽象行为模型和广告收入商务模型会比图 7 复杂,比如 In-App Purchase 就更加依赖于 APP 的具体业务逻辑和行为了.

如果只采用大数据分析的方法,首先必须获得数据.在 APP 完成提交以前,即使通过 Canary 测试,也只能获取 1%用户的真实数据,而不能有效地对影响广告收入的行为提前作出决策.采用小模型虽然不能完全符合实际,也可以对软件行为的定制决策的效果提前定量分析.

3 相关工作比较

3.1 概率模型检查

Kwiatkowska 等人的概率模型检测工具[7]能够对概率行为模型的决策作出判定,也能够计算状态的回报.但是由于简化了循环回路的处理,PRISM 计算强连通分枝可达性,而不追究每一个强联通分支状态点的概率叠加,因此经常过早地结束分析,无法判定模拟的状态迁移序列计算是否发散.我们的代数分析算法很好地解决了这个问题,使得任意的循环回路都可以处理,也能判定什么时候状态迁移序列会发散,约束优化算法,以免在搜索空间里找不到全局最优解.但是代数分析算法假设的概率模型是离散时序的马尔可夫链,因此还不能处理连续时序的问题.

关于大数据分析方法,Google TensorFlow 的深度学习系统是个典型的例子(http://github.com/tensorflow).TensorFlow 的计算模型是借助数据流的概率分布计算递推拟合的结果,这与我们的线性递推的计算模型有一定的相似性.主要的区别在于:基于需求分析阶段统计数据不充分的假设,我们并不追求用小模型完美地反映大数据分布的情况,而是通过线性的拟合得到可用的定量决策依据.

3.2 目标驱动的需求分析定量化

Letier 等人[8]提出,用贝叶斯概率网络模型描述不确定性需求的量化软件体系机构的风险属性.Liaskos 等人[9]提出,用 AHP 方法从成对比较的因素中抽取特征量化目标关联度,从而控制定性目标关联的主观性.这两类方法提示,需求分析定量化是一个趋势.只不过在需求端量化容易脱离问题框架的范畴,从而无法将软件程序行为和抽象目标行为紧密结合起来.如果从目标模型生成行为模型[10]固然可以连带分析对应的行为模型的风险,但是由于模型驱动方法的抽象层次落差,在缺乏细节变换支持的前提下,仍然无法像从面向对象设计模型到 Java 代码那样保证完全自动的双向变换[11].

3.3 移动AppStore大数据分析

数以百万计的移动 APP 为软件工程提供了大量的数据.对 APP 需求的行为分析目前还在起步阶段,但是这方面代码分析的工作已经存在不少.除了代码之外,AppStore 的用户反馈机制也为需求工程提供了很多自然语言需求的素材.目前,已经有一些研究工作关注这些自然语言描述中体现出来的需求以及跨 APP 需求之间的主题上的联系,值得关注[12].在一定意义上,这也是从用户的角度分析需求的间接反映.

3.4 安全和私密性的风险和论证

安全需求分析方面的工作很多[13],其中,定量的相对较少.即使是侧重于安全需求风险分析论证的近期工作[14],也需要借助预先分类的领域知识,如 CAPEC,CWE 等,由领域专家引导.Pasquale 等人将安全需求的效用函数定义为由模糊因果网络导出[15],从而为自适安全性提供环境依据,这和基于概率模型的需求分析有所异同,但其本质上也是一种需要专家建立的小模型.

Yang 等人[6]确实用理论概率模型来提供私密性风险和社交好处权衡的依据,但是还没有针对个人和软件行为分析和论证私密性需求.Calikli 等人[16]从个人和集体行为的差异和模式,通过自动归纳推理推导出一些启发式规则,用于推荐合理的私密性策略.这些方法跟统计机器学习的区别在于,这些学习到的规则是可以加以合理解释的.

4 结语和展望

本文提出了大数据需求分析时常常被忽视的思路,即借助能够清楚解释的符号化代数模型来反映大数据的需求.由此,基于问题框架的思路,结合抽象目标状态,对软件行为和使用者行为统一建立定量概率模型.从分析参数化和个性化的需求入手,本文提出的代数分析算法能够根据符号化参数计算总体风险或成本绩效,并根据系统和环境的变化和约束前提下推荐最优的参数值.

展望当前软件正在向社会化发展,全面整合人、机、物,因此除了对软件本身,也需要对相关联的人、机、物的行为模式有全面定量的认识.传统需求分析的代数模型方法能否弥补单纯数据驱动的大数据分析的不足之处,也还需要进一步深入研究.

References:

[1] Zave P, Jackson M. Four dark corners of requirements engineering. ACM Trans. on Software Engineering and Methodology, 1997, 6(1):1−30. [doi: 10.1145/237432.237434]

[2] Jackson M. Problem Frames: Analyzing and Structuring Software Development Problems. ACM Press, 2001.

[3] Jackson M. System behaviours and problem frames: Concepts, concerns and the role of formalisms in the development of cyber-physical systems. In: Proc. of the Dependable Software Systems Engineering. 2015. 79−104.

[4] Yu Y, Wang YQ, Mylopoulos J, Liaskos S, Lapouchnian A, Leite JCSP. Reverse engineering goal models from legacy code. In: Proc. of the 13th Int'l Conf. on Requirements Engineering (RE 2005). 2005. 363−372. [doi: 10.1109/RE.2005.61]

[5] Nhlabatsi A, Tun TT, Khan N, Yu Y, Bandara AK, Khan KM, Nuseibeh B. Why can't I do that? Tracing adaptive security decisions. EAI Endorsed Trans. on Self-Adaptive Systems, 2015,1(1):1−16. [doi: 10.4108/sas.1.1.e1]

[6] Yang M, Yu Y, Bandara AK, Nuseibeh B. Adaptive sharing for online social networks: Atrade-Off between privacy risk and social benefit. In: Proc. of the 13th IEEE Int'l Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom 2014). 2014. 45−52. [doi: 10.1109/TrustCom.2014.10]

[7] Kwiatkowska M, Norman G, Parker D. PRISM 4.0: Verification of probabilistic real-time systems. In: Proc. of the 23rd Int'l Conf. on Computer Aided Verification (CAV 2011). 2011. 585−591. [doi: 10.1007/978-3-642-22110-1_47]

[8] Letier E, Stefan D, Barr ET. Uncertainty, risk, and information value in software requirements and architecture. In: Proc. of the 36th Int'l Conf. on Software Engineering (ICSE 2014). 2014. 883−894. [doi: 10.1145/2568225.2568239]

[9] Liaskos S, Jalman R, Aranda J. On eliciting contribution measures in goal models. In: Proc. of the 20th Int'l Conf. on Requirements Engineering (RE 2012). 2012. 221−230. [doi: 10.1109/RE.2012.6345808]

[10] Yu Y, Lapouchnian A, Liaskos S, Mylopoulos J, Leite JCSP. From goals to high-variability software design. In: Proc. of the 17th Int'l Symp. on Methodologies for Intelligent System (ISMIS 2008). 2008. 1−16. [doi: 10.1007/978-3-540-68123-6_1]

[11] Yu Y, Lin Y, Hu Z, Hidaka S, Kato H, Montrieux L. Maintaining invariant traceability through bidirectional transformations. In: Proc. of the 34th Int'l Conf. on Software Engineering (ICSE 2014). 2014. 540−550. [doi: 10.1109/ICSE.2012.6227162]

[12] Sarro F, Al-Subaihin AA, Harman M, Jia Y, Martin W, Zhang Y. Feature lifecycles as they spread, migrate, remain, and die in App stores. In: Proc. of the 23rd Int'l Conf. on Requirements Engineering (RE 2015). 2015. 76−85. [doi: 10.1109/RE.2015.7320410]

[13] Nhlabatsi A, Nuseibeh B, Yu Y. Security requirements engineering for evolving software systems: A survey. Int'l Journal of Social Sciences and Education, 2010,1(1):54−73. [doi: 10.4018/jsse.2010102004]

[14] Yu Y, Franqueira VNL, Tun TT, Wieringa R, Nuseibeh B. Automated analysis of security requirements through risk-based argumentation. Journal of Systems and Software, 2015,106:102−116. [doi: 10.1016/j.jss.2015.04.065]

[15] Pasquale L, Spoletini P, Salehie M, Cavallaro L, Nuseibeh B. Automating trade-off analysis of security requirements. Requirement Engineering, 2016,21(4):481−504. [doi: 10.1007/s00766-015-0229-z]

[16] Çalikli G, Law M, Bandara AK, Russo A, Dickens L, Price BA, Stuart A, Levine M, Nuseibeh B. Privacy dynamics: learning privacy norms for social software. In: Proc. of the 11th Int'l Symp. on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2016). 2016. 47−56. [doi: 10.1145/2897053.2897063]

俞一峻(1972－),男,上海人,高级讲师,主要研究领域为软件维护,需求工程.

刘春(1982－),男,讲师,CCF 专业会员,主要研究领域为需求工程.