

软件外包决策的模糊专家系统研究与开发^①

The Study and Development of Fuzzy Expert System for Software Outsourcing Decision

吴明晖 (浙江大学 计算机学院 浙江杭州 310027)
 (浙江大学城市学院计算机与计算科学学院 浙江杭州 310015)
 章欣 (浙江大学城市学院计算机与计算科学学院 浙江杭州 310015)

摘要: 根据软件开发外包的特点,提出了相关评价因子。基于模糊专家外壳程序 FuzzyJ Toolkit,针对软件开发外包承包商选择问题,设计和开发了用于决策支持的模糊专家系统。

关键词: 模糊专家系统 软件外包 FuzzyJ

1 引言

软件业是一个高速变化、新技术层出不穷的行业,同时又是人力资源成本相对较高的行业。企业通过外包和采购形式来获取待开发产品的部件,可以最大限度地从社会分工合作、资源共享中获益。但如何针对外包做出正确决策是一个难题^[1]。在软件外包领域,知识和信息大多是模糊而粗糙的,这些不确定、不明确、不精确的信息在人类专家的思维中占据着绝大部分。因此,有必要开发了一个基于模糊知识、模糊推理和模糊决策的专家系统来模拟这一思维过程,从而更好的提供决策支持。

从承包商数据库取出选定承包商的评价细节,从规则数据库中取出所有的规则,将评价细节和规则进行匹配和推理,最终得出一个模糊评价值;

2 软件外包承包商选择决策系统

2.1 系统体系结构

软件外包承包商选择决策支持系统的体系结构如图1所示,主要包括4部分:

- (1) 规则管理模块:实现对模糊规则的增加、删除、修改、检索和查看等功能;
- (2) 承包商管理模块:实现对承包商评价的增加、删除、修改、查看等功能;
- (3) 模糊规则推理模:本模块是系统的核心,它将

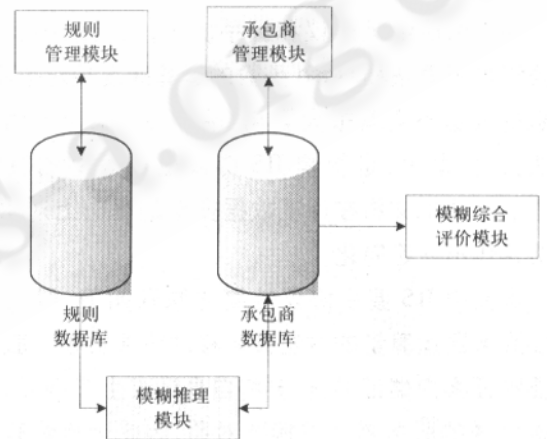


图 1

- (4) 模糊综合评价模块:本模块是系统的决策辅助,它从数据库中取出同等级别的承包商,对他们进行综合评价,给出评价值并进行排序,为外包商选择提供参考。

^① 基金项目:浙江大学城市学院教师科研基金(No. J52107011);

2.2 评价因子

通过对领域专家的调查并结合所获取到的承包商评估系统实例,包括“联想集团软件事业部子承包商选择和评价表”^[1]、“评估候选承包商的综合能力”^[5]、“承包商评估调查问卷”^[6]、“Outsourcing Goals”^[1]等,最终,我们综合考虑管理和技术方面,确定 17 项评价因子,如表 1 所示:

表 1 综合评价表

管理方面的考虑	技术方面的考虑
能否提供较好的服务(维护)	软件开发经验
承包商的信誉如何	可用人力资源
承包商是否已经取得业界认可的证书(如 ISO 质量认证、CMM2 级以上认证)	是否具有同类项目的开发经验以及与本项目相关的技术能力
承包商的可支配资源是否充足而稳定(总体实力和发展潜能)	技术方案的可行性
承包商的开发能力与管理能力如何	项目开发进度安排合理性
承包商以前开发的产品是否有良好的质量	工作量估计合理性
是否可以经常面对面交流	费用要求合理性
承包商是否有长期客户	承包商的工作质量提高潜力
外包经验	

2.3 系统实现

2.3.1 模糊推理模块

本模块实现规则的模糊推理,在应用 FuzzyJ Toolkit 模糊专家系统外壳程序所提供的功能基础上,还根据需求进行了部分算法的改进。下面结合系统中的具体应用来进行详细描述。

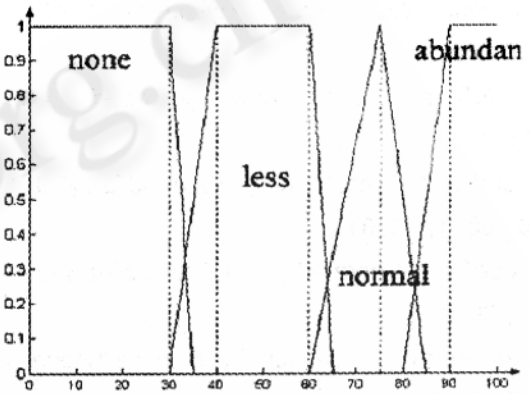


图 2

(1) 模糊变量。所定义的表达方式用来描述一个模糊概念,例如:温度,压力,年龄等。模糊变量有相对应的论域和一个基本术语集合,对于特定的模糊变量,可以具有单位。而模糊术语以模糊集的方式,通过构

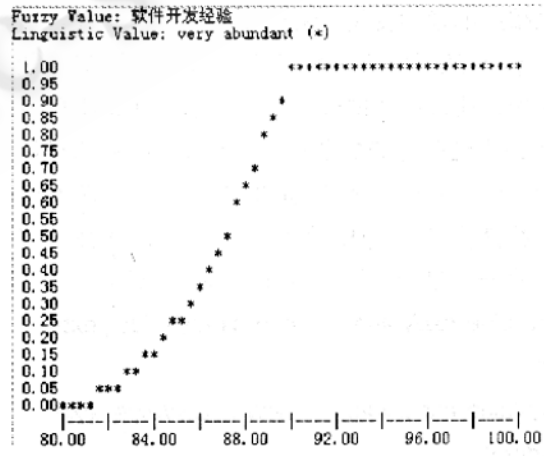
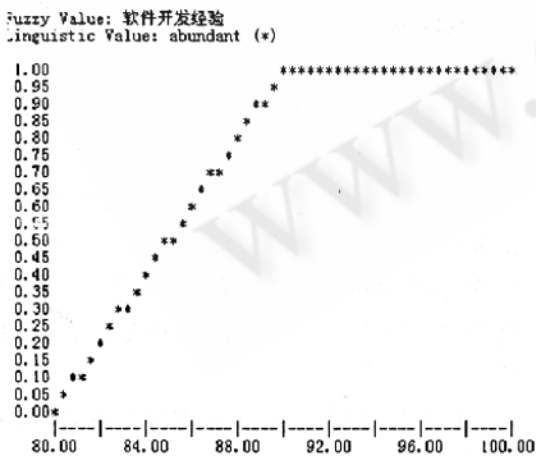


图 3 模糊值——“软件开发经验丰富”以及“软件开发经验非常丰富”的隶属函数图

造隶属函数映射到论域中。因为隶属函数的无限性,我们可以利用多种形态的模糊集来组成一个隶属函数。在 FuzzyJ Toolkit 中为用户提供了 15 种形式的模糊集。

对应于系统中的 17 个评价项,需要分别构造它们的模糊变量。以下是构造一个“软件开发经验”模糊变量的代码,它由四个不同的模糊集组成,也就是对应四个基本模糊术语集——“无、少、一般、丰富”。

```
FuzzyVariable exp = new FuzzyVariable("软件开发经验", 0, 100.0, "");
exp.addTerm("none", new RightLinearFuzzySet(30.0, 35.0));
exp.addTerm("less", new TrapezoidFuzzySet(30.0, 40.0, 60.0, 65.0));
exp.addTerm("normal", new TriangleFuzzySet(60.0, 75.0, 80.0));
exp.addTerm("abundant", new LeftLinearFuzzySet(80.0, 90.0));
```

这里“软件开发经验”模糊变量 exp 的取值范围为 0 到 100,利用不同隶属函数来设定四个模糊术语集。比如,我们设计“一般”这一模糊术语的隶属函数为两条线段,其顶点坐标分别为 (60, 0), (75, 1) 和 (80, 0)。其它几个模糊属于可以构造类似直线性函数(甚至曲线性函数)。

(2) 模糊值。模糊值是基于模糊变量构造的一个概念。可以用“有点热”,“很冷”这样的语言作为模糊值来描述相应的模糊变量“温度”。对于“冷”、“热”这样的基础术语,还可以用“很”、“有点”、“非常”等等的修饰语来使这些模糊值的表达更加丰富。FuzzyJ Toolkit 提供了关于这些修饰的函数,方便构造更多的模糊值。例如,依据上面构造的模糊变量“软件开发经验”,我们构造两个它的模糊值,一个是“软件开发经验丰富”,另一个是“软件开发经验非常丰富”:

```
FuzzyValue expValue1 = new FuzzyValue(exp, "abundant");
FuzzyValue expValue2 = new FuzzyValue(exp, "veryabundant");
System.out.println(expValue1.plotFuzzyValue("*", 80, 100));
System.out.println(expValue2.plotFuzzyValue("
```

```
*", 80, 100));
```

对应于两个模糊值,我们输出他们的隶属函数图示如图 3。当我们给出一个评价值为 84 时,对于左图,它的隶属度为 0.3;对于右图,它的隶属度为 0.15。也就是说,评价值 84 表示“软件开发经验丰富”的置信度为 0.3;表示“软件开发经验非常丰富”的置信度为 0.15。

(3) 规则。系统模拟的人类思维方式是推理,而推理的基础就是规则。人类常常采用“如果……那么……”的推理方式,因而,映射到模糊专家系统中,也使用了类似的语言形式,即产生式规则。规则中的条件部分称为“规则前件”,结果称为“规则后件”,一条规则中可以有多个规则前件和多个规则后件。前件和前件,后件和后件之间可以用“而且”,还可以用“或者”来连接。如果在规则中出现“或者”的时候,它显然可以分解成两条不含“或者”的规则。

下面我们也列举一条规则以及用代码来构造这一规则:

如果软件承包商的“软件开发经验”——“非常丰富”,而且,“可用人力资源”——“丰富”,那么,对他技术方面的评价是“满意”。

```
FuzzyRule rule = new FuzzyRule();
rule.addAntecedent(expValue1); // 添加规则前件 1, expValue1 为之前构造的模糊值
rule.addAntecedent(expValue2); // 添加规则前件 2, expValue2 为之前构造的模糊值
rule.addConclusion(new FuzzyValue(Re, "satisfy")); // 添加规则后件
```

构造模糊变量采用静态方式,即在程序初始化时进行构造,而模糊值因为包含修饰语和基本模糊值两个部分,采用动态构造的方式。在本系统中,每条规则的前件不同,不仅仅是数值不同,前件的个数也可能会不同,但后件只有一个,即评价结果。在构造规则时,将动态添加前件和后件的模糊值,以及相应的模糊输入。

(4) 模糊推理。模糊推理的实质将模糊输入同模糊规则前件按照一定的推理方法进行运算,得出结论。之前构造规则时,对应于每个规则前件,必须添加相应的模糊输入, RuleEx.addInput(new FuzzyValue(FV, STechMV)) 函数就是用于添加模糊输入。

```
FuzzyValueVector resultVe;// 结论模糊向量
FuzzyValue resultVa;// 结论模糊值
resultVe = rule.execute ( new LarsenProduct-
MaxMinRuleExecutor ( ) );
```

```
resultVa = resultVe.fuzzyValueAt ( 0 );
```

其中, LarsenProductMaxMinRuleExecutor () 就是模糊推理所应用的方法。

(5) 逆模糊化。在给出一个模糊变量的值后, 如何来判断其所表达的模糊概念需要进行逆模糊化。在 FuzzyJ Toolkit 中所提供的逆模糊化函数是用一个具体的实数来表示模糊值。除此之外, 将一个需要进行逆模糊化的模糊值同一些已静态定义的基础模糊值进行匹配, 从中挑选出最匹配的一个基础模糊值来解析当前的模糊值, 这也是一种适合的方法。匹配两个模糊值的相似性主要有距离度量法和贴近度法 [7], [8]。FuzzyJ Toolkit 中提供了 fuzzyMatch (FuzzyValue1, FuzzyValue2, threshold) 函数来匹配两个模糊值, 并且可以设定 threshold (阈值 $0 \leq \text{threshold} \leq 1$) 来控制匹配的程度。如果之前已经建立了一组比较完全的并且经过语言修饰的模糊值, 则可以将需要逆模糊化的模糊值和这些基础模糊值进行匹配, 找到最匹配的基础模糊值就是该模糊值所要表达的模糊概念。在本系统中, 我们还提供了数值型的逆模糊化, 结合概念性的逆模糊化, 供决策者判断。

2.3.2 综合评价排序模块

在评价体系中的多个因子并不是都具有相同重要程度, 因此, 用户在对每个因子所给与的评价还需要和因子自身的重要性结合起来才能真正反映决策意愿。对于每个评价项, 我们同样给出一组修饰语和一组模糊值来表示该评价项的重要性, 用户可以根据当前项目的需求, 给予评价项不同的重要性。

我们将“重要性”作为一个模糊变量, 定义其论域为“0—5”, 基础模糊术语为“重要、一般、不重要”。根据用户定义的评价项重要性, 我们动态构造其模糊值, 在得到所有评价项重要性的模糊值之后, 我们将它们归一化, 构造成权重模糊值。将权重模糊值乘以相应评价项的模糊值, 就得到了综合评价模糊值: $\underline{S} = \sum_{i=1}^n \underline{A}_i \times \underline{B}_i$

FuzzyJ Toolkit 所构造的模糊隶属函数是连续空间

的, 也未提供处理模糊代数运算的函数。考虑到连续空间代数运算的复杂性, 我们把构造的模糊值离散化, 在离散空间中计算模糊乘法并将结果存储在一个模糊集中。以下是在系统中添加的模糊乘法的算法公式:

$$\underline{A}_i * \underline{B}_i = \int_{z_i \in R} \frac{\bigvee_{z_i=x_i * y_i} (\mu_{\underline{A}_i}(x) \wedge \mu_{\underline{B}_i}(y))}{Z} \quad (6) \quad (\text{其中 } \mu_{\underline{A}_i}$$

(u) 为 u 对 \underline{A}_i 的隶属度)

最后将该模糊集逆模糊化为一个实数, 作为综合评价值。

3 系统改进方向

专家系统的推理是基于规则的, 规则的数量和质量决定着系统的效能。现实中, 规则库中所存储的知识还达不到我们的要求。既然模糊专家系统模拟的是人类的语义结构和思维方式, 那么, 同样可以模拟人类的自我学习能力。将模糊系统与神经网络相结合, 把模糊规则和隶属函数用神经网络表示出来, 用于模糊推理。利用神经网络的特性, 修改隶属函数, 求精模糊规则, 并将它们提取出来更新知识库, 这是系统进一步改进的方向。

参考文献

- 1 Minghui Wu, Hareton Leung, A Knowledge - Based Framework For Software Development Outsourcing Decision.
- 2 FuzzyCLIPS, http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyClips/fuzzyCLIPSIndex2.html, NRCC.
- 3 FuzzyJ Toolkit Web site, http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit.html, NRCC.
- 4 杨一平, 现代软件工程技术 with CMM 的融合, 人民邮电出版社, 2002. 11, P261 - 262.
- 5 林锐, 软件工程与项目管理解析, 电子工业出版社, 2003. 10, P313 - 314.
- 6 斯蒂夫·迈克康奈尔 著, 席相霖 等译, 快速软件开发, 电子工业出版社, 2002. 1.
- 7 曹谢东, 模糊信息处理及应用, 科学出版社, 2003. 10.
- 8 李登峰, 模糊多目标多人决策与对策, 国防工业出版社, 2003. 4.