

基于多 Agent 的信贷风险预测实现技术研究*

彭岩¹ 涂序彦²

(首都师范大学信息工程学院 北京100037)¹ (北京科技大学信息工程学院 北京100083)²

Study on Agent-Based Credit Risk Prediction Implementation Technology

PENG Yan¹ TU Xu-Yan²

(Colleg of Information Engineering, Capital Normal University, Beijing 100037)¹

(College of Information Engineering, Beijing University of Technology and Science, Beijing 100083)²

Abstract Risk early Warning is very important to Banks. This dissertation puts forward a Agent-Based Credit Risk Prediction System, the main parts of the system such as model training, prediction, results analyzing are realized with agent that have the characteristic of autonomy, social ability, reactivity and pro-activeness etc. And also use agent technology to make the Risk Early Warning more effective.

Keywords Credit risk, Prediction, Agent

1 概述

信贷决策过程是一个对数据进行逻辑分析与综合判断的过程,随着经济的飞速发展,各商业银行对信贷过程中的各种风险越来越重视,其高层管理者面对的是大型、复杂和非结构化的决策问题。要使用计算机为决策者提供有价值的辅助决策信息,就得对软件和软件设计人员提出更高的要求,同时软件系统也变得越来越复杂。为了满足多种功能要求,软件系统中包含了大量的相互关联的部件,部件之间的不断进行的复杂交互、协作与冲突成为软件设计人员的难题。在这种情况下,使用具有自主交互能力的 Agent,可以使系统能够在一定程度上实现自我管理。Agent 中包含了如何解决某类问题的知识,而不再用程序明确地指示它所处理可以预见的事件和处理每一事件的方法。信贷风险预警系统正是一个复杂的软件系统,因此,我们运用 Agent 体系理论,提出面向信贷风险预警的多 Agent 的预测实现方法。

2 信贷风险预测的使能技术——多智体系统 MAS

Agent 技术源自人工智能,基于 Agent 的系统的开发和应用是目前计算机科学领域中一个非常重要和活跃的研究方向,在分布式系统、软件工程及通信等各个领域得到广泛的应用,代表了一种新的方式和途径。

2.1 Agent 模型

Agent 的模型结构可分为三类:慎思型 Agent、反应型 Agent 和混合型 Agent。

(1)慎思型 (deliberative architecture)。该结构主要基于符号人工智能方法,包括客观世界的一个符号标示模型,自动规划和推理机构,能处理一些复杂的智能问题,但其中一些符号人工智能相关的问题还没有很好地解决。

(2)反应型 (reactive architecture)。在该结构中,Agent 行为的复杂性可以是 Agent 所处环境复杂性的反映。Agent 只是简单地对外界刺激产生反应,不保存内部状态,其优点是简单而实用性好。

(3)混合型 (hybrid architecture)。该结构结合前两种方法

的优点。将两种结构混合的方法是采用分层方式:内层是一个基于符号 AI 的认知、推理、决策层。外层是一个能快速响应和处理环境中突发事件的反应层,它以刺激-回答方式对环境的改变做出反应,并被给予较高的优先级。

2.2 多 Agent 系统模型

多 Agent 系统 (Multi-Agent System, MAS) 是分布式人工智能的一个新的研究领域。多 Agent 系统是将独立的 Agent 系统,通过某种特定的联系,组成一个社会的环境^[1]。其中各 Agent 通过相互协商与合作来实现多 Agent 系统的功能。因此,多 Agent 体系结构分为个体 Agent 体系结构和 Agent 群体社会结构两个部分。其中个体 Agent 体系结构是指在 Agent 系统中单个 Agent 的构造结构。Agent 群体社会结构通过定义角色和相互关系为 Agent 提供了一种交互框架。一个多 Agent 系统可以由多个已存在的 Agent、多主体间的联合意图 (Joint Intention)、常识 (Common Sense) 即主体的公共知识和 Agent 的局部规划/协调器等组成。其中智能体的局部规划/协调器是多智能体系统的重要组成部分,因为单个智能体的资源、知识和能力等往往是非常有限的,协调和合作在多智能体系统中就变得非常重要。由多个 Agent 相互协调实现对多库系统的协调控制,便构成基于多 Agent 的多库协同系统。

2.3 Agent 通信

在多 Agent 系统中,每个 Agent 都是一个自治或半自治系统,Agent 之间以及 Agent 与环境之间进行交互与通信一般分为黑板系统和消息/对话系统^[2]。

·黑板系统。是传统人工智能系统和专家系统的议事日程的扩充,通过使用合适的结构支持分布式问题求解。在多 Agent 系统中黑板提供公共工作区,主体可以交换信息、数据和知识。黑板系统可以用在任务共享和结果共享系统中。

·消息传送。采用消息通信是实现灵活复杂的协调策略的基础。使用规定的协议,Agent 之间交换的消息可以用来建立通信和协作机制。与黑板系统不同,两个 Agent 之间消息是直接交换。为了支持协作策略,通信协议必须明确规定通信过程、消息格式和选择通信语言。

*)本课题研究得到国家863高技术项目(863-511-944-019)的资助。彭岩 博士研究生,讲师,主要研究方向:计算机网络、人工智能。涂序彦 博士生导师,主要研究方向:人工智能及其应用。

3 基于多 Agent 的风险预测系统

3.1 模型结构分析

本文所设计的基于多 Agent 的风险预测模型 (Multi-Agent based Risk Prediction Model, MARP) 将用于信贷决策系统, 该领域决策者提出的预测任务往往涉及多行业、多领域及多学科的知识, 这就要求模型具有相对独立的求解特定问题能力的 Agent 实体, 同时, 多个 Agent 之间又能实现协作与合作, 以处理单个 Agent 无法解决的复杂预测问题。因此, 本文的多库协同模型中 Agent 的应用主要侧重于利用其自治性、能动性和社会性:

(1) 用 Agent 的自治性。将单个 Agent 作为信息集成场所和决策工具。Agent 能自行控制其状态和行为, 能在没有人或其他程序介入时操作和运行。

(2) 运用 Agent 的能动性。Agent 主动表现出目标驱动的行为, 能自行选择合适时机采取适宜行动。

(3) 运用 Agent 的社会性。Agent 可以在多 Agent 环境中协同工作和消解冲突, 执行和完成一些相互受益且自身无法求解的复杂任务。即运用 Agent 的协作、合作、协同及协商的能力, 将信贷风险的预警过程通过多个 Agent 来管理, 通过 Agent 之间的通讯能力和分布协调的特性实现对各类信贷风险的快速反应和对银行利益的优化。

MARP 由多个 Agent 实体组成, 对于复杂问题的预测不仅需要 Agent 运用各自的能力与知识进行推理和预测, 而且需要多个 Agent 对任务协作。我们将预测过程中的预测问题生成系统 PPGS、预测模型训练系统 PMTS、风险预测系统 RPS、结果分析系统 RAS、结果生成系统 RGS、资源管理系统 RMS (包括: 数据库管理系统 DBS、模型库管理系统 MBS、方法库管理系统 WBS、知识库管理系统 KBS) 设计成对应的 PPGSAgent、PMTSAgent、RPSAgent、RASAgent、RGSAgent、RMSAgent 以及对各个 Agent 进行总控与协调的“主管”——管理 Agent (MAgent)。这样, 每个 Agent 能作用于自身和环境, 并与其他 Agent 进行通信。通过与其他 Agent 的通信, 可以开发新的规划或求解方法来处理不完全、不确定的知识。通过 Agent 之间的合作, 可以改善每个 Agent 的基本能力, 提高整个系统的风险预警能力。

3.2 个体 Agent 的结构模型

在本模型中, 将具有独立行为能力的 Agent 作为基础实体。由于在经济领域中, 经济个体有着相当多的非理性行为, 不是认知和推理等智能因素所能解释的, 因此, 我们选用混合型系统结构作为 Agent 的基本结构。

Agent 的结构模型是一个八元组 Agent = 〈感知模块, 协作模块, 任务求解与推理机, 执行模块, 思维状态, 知识库, 行为规则, 通讯模块, 控制模块〉。如图 1 所示。其中:

·通讯模块: 是多 Agent 间的通信接口, 完成 Agent 与其它 Agent 间的信息传递。以 KQML 语言系统的消息传递方式进行传递请求, 通过消息发送、消息接收控制信息传输和数据变换。

·感知模块: 是 Agent 感知外部环境的模块, Agent 通过它感知环境的变化, 并对环境信息做出一定的抽象。

·控制模块: 是 Agent 的中枢, 负责协调、控制、选择相应的功能模块, 按一定运行模式完成任务。

·思维状态: 是三元组〈信念, 目标, 意图〉。

·协作模块: 根据本 Agent 所知的其他 Agent 的能力信息, 决定与哪个 Agent 合作完成当前的任务。

·任务求解和推理机: 是 Agent 利用知识库和行为规则

按照当前的思维状态, 进行推理和决策, 做出行动规划, 以完成任务。

·知识库: 存储事件解释和分类知识、控制行为知识、问题求解知识、领域知识及通讯知识等。

·数据库: Agent 的内部数据库包括 Agent 对自身的描述、Agent 对世界状态的描述及对其他 Agent 状态的描述等。每个 Agent 都有一个描述, 形成 Agent 数据中的数据集合。

·执行模块: 进行事务处理, 用于执行 Agent 所要完成的任务。

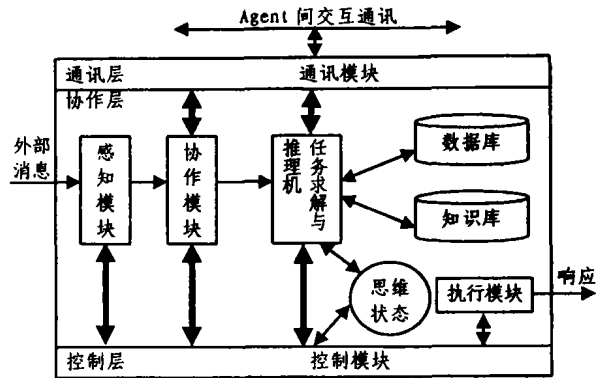


图1 Agent 的结构模型

3.3 多 Agent 间的通讯机制

实现多 Agent 之间协作的关键是多 Agent 之间的通讯机制, 目前国际上最著名的 Agent 通讯语言是 ARPA 的知识共享计划 (knowledge sharing effort) 中提出的两个相关语言: KQML (Knowledge Query and Manipulation Language)^[3] 和 KIF (Knowledge Interchange Format)。KQML 定义了一种 Agent 之间传递信息的标准语法以及一些“动作表达式” (performative)。KIF 则给信息的内容提供一种语法。本文采用 KQML 实现 Agent 之间的通信。

我们根据需求和各 Agent 的特征, 设计了各种消息, 每条消息表示为一个七元组: 〈Performative, sender, receiver, MessageID, MessageSeq, Reply-with, Content, Ontology〉。其中, Performative 为行为原语, sender 为消息发送者, receiver 为消息的接收者, Language 为语言的合法表达, MessageID 为消息标识, MessageSeq 为消息序列, Reply-with 为消息不为空时, 表明该消息是需要回答的, Content 为消息的内容, Ontology 指出本体所定义的常量子集。下面是消息发送和响应实例:

RGSAgent 向 RPSAgent 询问向 Company A 贷款的风险, 要求回应 A 是否具有偿付能力, 见表 1。在这个消息中, ask-if 为行为原语, 表示发送者想知道接收者虚拟知识库的内容, 以下各行是行为原语的参数及其值。

表1 RGSAgent 向 RPSAgent 询问

```
(ask if
:Sender RGSAgent
:Receiver RPSAgent
:MessageID RGSAgent-RPSAgent-2
:MessageSeq NULL
:Reply with one of
(solvency, bankruptcy)
:Language LPROLOG
:Content "?risk(CompanyA, medium-term,
return-level,
RETURN)")
```

RPSAgent 向 RGSAgent 返回的响应消息见表 2:

表2 RPSAgent 向 RGSAgent 返回的响应消息

```

(reply
:Sender RPSAgent
:Receiver RGSAgent
:MessageID RPSAgent-RGSAgent-2
:MessageSeq RPSAgent-RGSAgent-2
:Reply with one of
      (solvency,bankruptcy)
:Language LPROLOG
:Content "risk(CompanyA,medium-term,
      return-level,bankruptcy")
    
```

表2所示的消息包含一条新的消息序列,因此 MessageSeq 值为 NULL;Reply-with 及 Content 组元使用的是 KIF,Reply-with 告诉接收 Agent 用其所列的项来回答询问。由于 Content 组元使用了 RETURN,则接收 Agent 返回的消息是对原始消息处理后的回应。即 RPSAgent 应返回的消息是用“bankruptcy 或 solvency”取代原消息中的 RETURN 作为对询问消息的响应。在整个预测过程中,表1所示的消息会促使 RPSAgent 调用资源信息中的数据、模型或方法对 Company A 的财务状况进行分析预测,并在必要时对资源信息进行变更。

3.4 基于多 Agent 的风险预测系统

系统的体系结构是一个七元组,即 (PPGSAgent、PMTSAgent、RPSAgent、RASAgent、RGSAgent、RMSAgent、MAgent)。如图2所示,每个组元都是一个 Agent 类,针对具体问题系统通过 Magent 创建 Agent 实例。

·Magent 为管理 Agent:负责管理整个体系中的 Agent,协助整个体系 Agent 的正确运行,说明各 Agent 的功能和使用方法,建立 Agent 间通信和协作。控制并协调其它 Agent。MAgent 包括两方面的知识:一是有关各 Agent 的能力的知识;二是有关预测系统自身的动态信息,包括当前的预测请求、当前的调度计划、当前的协调规划、当前调度的执行情况等等。主要功能是根据用户的需求,对求解任务进行分解与细化,形成面向用户实际问题的规划。根据解题规划,完成其他各智体的任务指派,并协调其他智体。

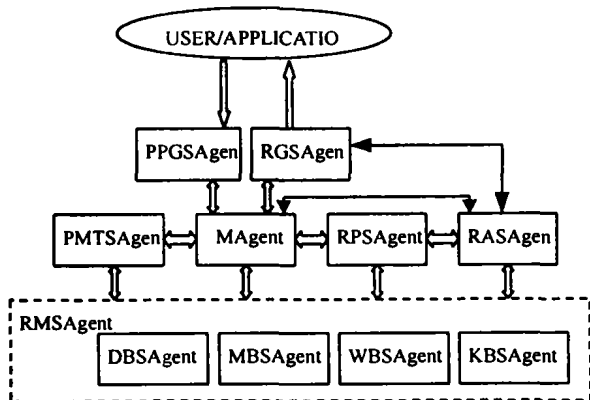


图2 基于 Agent 的风险预测系统

·PPGSAgent 为问题生成系统 Agent:用户提出的任务往往是贷款的决策意向而不是决策问题,PGSAgent 的任务首先是将决策意向转变成决策问题,产生决策意向的问题描

述框架和基本决策过程类型。

·PMTSAgent 为预测模型训练 Agent:提供对神经网络模型进行自动训练的功能。尤其是在信贷政策、税务政策及企业状况发生重大变化时;当有特殊破产案例或贷款过程异常结束时,能及时加入新样本对预测模型进行重新训练与调试。

·RPSAgent 为风险预测系统 Agent:为预测系统的核心,是完成预测任务的主体,其中封装了本文所提出的 SOM 预测模型。其功能是根据 PGSAgent 发来的需求,向 RMSAgent 要求相关数据与知识,利用自身封装的模型对企业进行风险评估。同时,RPSAgent 具有学习功能,能够从成功或失败的预测中不断学习,调整模型参数,并记录调整过程,作为将来进行同类预测问题时的参考。

·RASAgent 为结果分析系统 Agent:对预测生成的结果进行进一步的分析、处理与解释,使最终结论清晰、明确,并在一定程度上增加输出的可视化效果。

·RGSAgent 为结果生成系统 Agent:与需求用户进行交互,将预测结果输出。

·RMSAgent 资源管理系统 Agent:包括数据库管理系统 DBSAgent、模型库管理系统 MBSAgent、方法库管理系统 WBSAgent、知识库管理系统 KBSAgent。其功能是完成对各种资源的管理、协调与调用任务。

整个系统的运行过程如下:

(1)用户提出预测任务后,PPGS 对任务进行分类、分解,生成预测问题。(2)为完成相应的预测问题,MAgent 将问题发送给 RPSAgent 进行预测,或发送给 PMTSAgent 进行预测模型再训练。(3)RPSAgent 根据 PPGS 发送的消息,使用其相应的功能模块,调用 KBSAgent 的知识进行推理,运用 WBSAgent 的方法、MBSAgent 的模型和 DBSAgent 的数据进行风险预测;求解过程中通过 Magent 发送各类消息实现多 Agent 间的交互与协作。(4)RPSAgent 将预测结果发送给 RASAgent 进行结果分析。(5)RGSAgent 将从 RASAgent 的预测结果,形成决策结果通过通信模块提供给外部用户或应用。

小结 Agent 技术在各领域的应用已成为多学科交叉研究的热点。本文在分析 Agent 系统体系结构的基础上,利用 Agent 的自治性、能动性和社会性等特点,将信贷风险预测过程中预测模型训练、预测、预测结果分析等各主要步骤用 Agent 技术实现,提出了基于多 Agent 的风险预测系统技术方案。希望该系统的实施,能够在商业银行控制风险,防范风险和贷款效益中起到积极的作用。

参考文献

- 1 Oragani A S, Giorgini P. Multi-agent environment, Lecture Note in Artificial Intelligence. ECAI'96 Workshop,1996. 103~116
- 2 史忠植. 智能主体及其应用. 科学出版社,2000
- 3 Labrou Y,Finin T. A proposal for a New KQML Specification. In:TR CS-97-03,UMBC, Internet URL. http://www.cs.umbc.edu/~jklabrou, 1997
- 4 Kraus S. Negotiation and cooperation in Multi-agent environments. Artificial Intelligence, 1997 (94): 79~97