Agent-based Decision Support System

Team 1

Sergi Albiach, Anna Garriga, Benet Manzanares and Ramon Mateo

UNIVERSITAT ROVIRA i VIRGILI

Index

Introduction

- Collaborative classifier for firm fraud detection
- AUDIT dataset - 767 instances - 25 characteristics
- Train 10 classifiers
- Coordinate them to obtain unique test predictions

Agents Description

User Agent & Coordinator Agent & Classifier Agent

User Agent

- **Type:** User Interface
- **Role:** It is in charge of loading the dataset, splitting the data into training and test, starting the training and managing user queries. Its communication is limited to the coordination agent.
- **Number of agents:** 1

Coordinator Agent

- **Type:** Facilitator
- **Role/s:** It is responsible for splitting the training dataset between the agents and selecting the attributes that each of them will analyze. It is also in charge of receiving the output classifications from the classifier agents and aggregating them using the normalized accuracies from the training step as weights This result will be sent to the user agent.
- **Number of agents:** 1

Classifier Agent

- **Type:** Wrapper
- **Role/s:** It receives a part of the training data and creates a Decision Tree based on it. It also receives test cases and sends the output classification result to the coordinator agent.
- **Number of agents:** 10

User Agent: properties, functions, messages

Properties:

- Social ability
- Rationality
- Proactive
- Temporal Continuity

Messages:

- test predictions

Functions:

- setup
- readSettings
- readDataset
- startTrainingOrTestingMS
- generateTestInstances
- filterAttributes

Coordinator Agent: properties, functions and messages

Properties:

- Social ability
- Rationality
- Reasoning capabilities
- Autonomy
- Temporal Continuity

Messages:

- train
- test
- accuracy
- predict

Functions:

- setup
- workingCallback
- initialization
- createClassifiers
- getClassifiersAIDs
- classifierNameToIdx
- train
- filterAttributes
- trainingCallback
- test
- testingCallback

Classifier Agent: properties, functions and messages

Properties:

- Social Ability
- Rationality
- Learning
- Reasoning Capabilities
- Temporal Continuity

Messages:

- train
- test

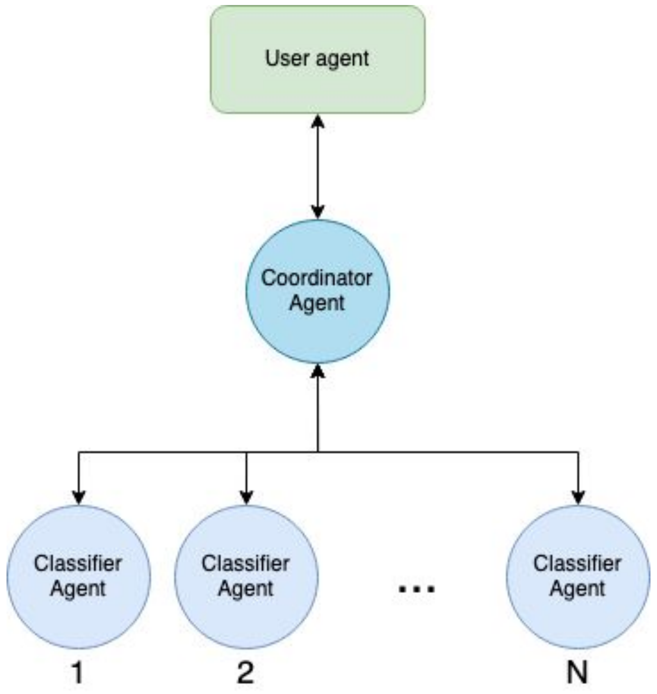
Functions:

- train
- createClassifier
- classifyInstance
- computeAccuracy

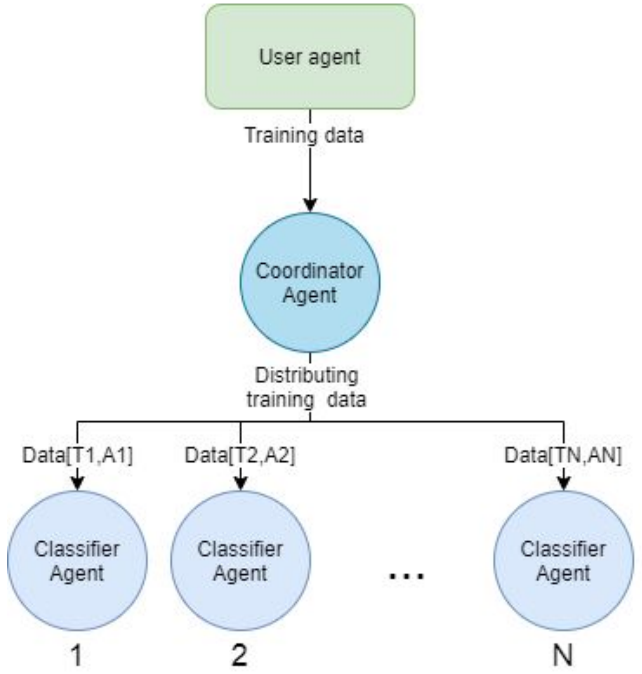
Reasoning capabilities

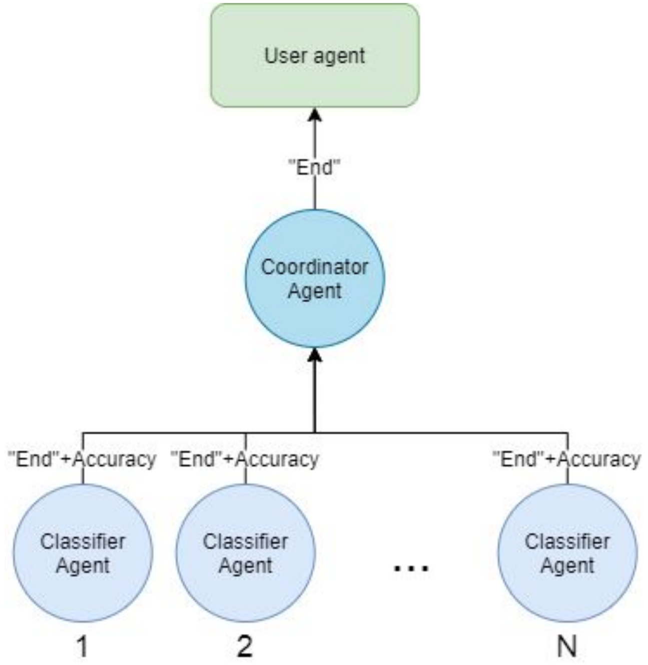
System Architecture and Phases

General Architecture

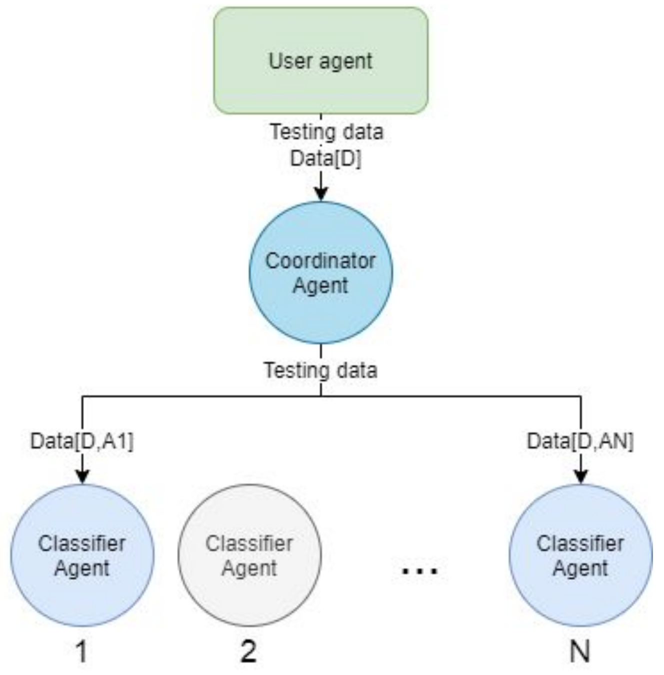


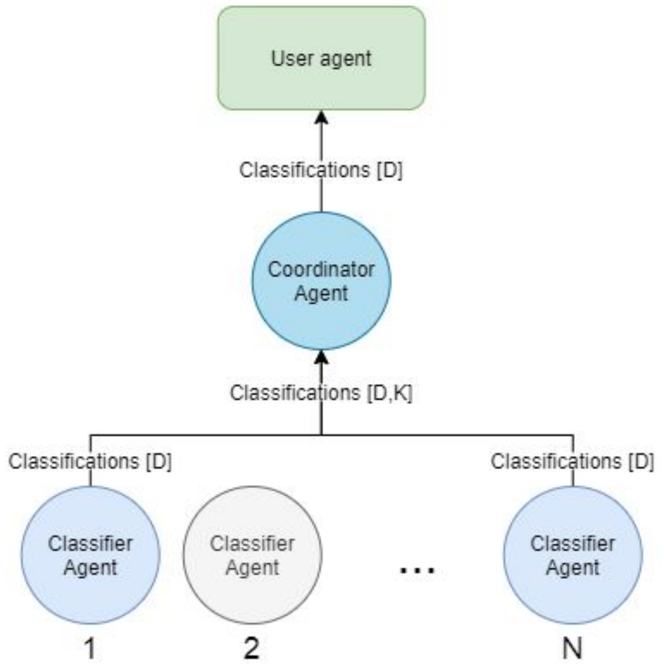
Training phase





Testing phase





Communication Protocols and Behaviours

Communication protocols

- **AchieveREInitiator**: It is used when:

  -the *UserAgent* asks the coordinator to start the training or test process

  -the *CoordinatorAgent* requests a *ClassifierAgent* to perform a training or testing procedure

- **AchieveREResponder**: It is used:

  -by the *CoordinatorAgent* to manage the training and testing requests from the *UserAgent*

  -by the *ClassifierAgent* for the training and testing requests from the *CoordinatorAgent*

Variation of these protocols that accept more parameters:

- **OurRequestInitiator** extends **AchieveREInitiator**: accepts a string as a task name or descriptor (printed *onStart* and *onEnd*), a callback, in case we need to answer the returned message in any specific way.

- **OurRequestResponder** extends **AchieveREResponder**: accepts a function that will be invoked when an *INFORM* response to the request is received (independently if the “agree” message is sent or not before).

Code Implementation

OurAgent

Father class of the other agents, extends from class Agent. All agents extend from this class.

Functions:

- showMessage
- showErrorMessage
- createOurMessageRequest
- RegisterInDF
- getFromDF
- blockingGetFromDF

OurMessage

An auxiliary serializable class which acts as the equivalent to a C structs and can be set as content object of an ACLMessage.

It contains:

- *String* **type**: Usually used for different used for differentiating between training and test requests.
- *Serializable* **content**: Usually used for the dataset

OurRequestInitiator

Functions:

- OurRequestInitiator(Agent, request, taskName)
- OurRequestInitiator(Agent, request, taskName, consumer)
- onStart()
- onEnd()
- getSenderName(message)

Handlers:

- handleAgree
- handleNotUnderstood
- handleRefuse
- handleOutOfSequence
- handleInform
- handleFailure

OurRequestResponder

Functions:

- OurRequestResponder
- onStart
- onEnd
- getSenderName
- prepareResultsNotification

Handlers:

- handleRequest

Results

Accuracy, Precision, Recall & F1 Score

● **Accuracy:** 0.944	● **Precision:** 0.948
● **F1 Score:** 0.928	● **Recall:** 0.912

Conclusions

Main Conclusions

- The architecture is providing an effective and efficient model for the studied problem
- Some attributes are quite more powerful than others
- MAS must be programmed very carefully
- Not a commonly used technology

Thank you for your attention

Team 1

Sergi Albiach, Anna Garriga, Benet Manzanares and Ramon Mateo

UNIVERSITAT ROVIRA i VIRGILI