Abstract:
Performance and user-friendliness are big challenges for data mining tools. In this demo, we introduced an agent-based distributed data mining platform. User can use it to manage and share the data mining-related resources conveniently. An example is introduced to illustrate how the whole platform works in distributed environment. It has satisfactory performance even when data is highly distributed and large.

Keyword: data mining, software package, multi-agent system

1. INTRODUCTION
Large and distributed data sets are a big challenge to data mining tools. Huge amounts of digital data are collected and stored for varieties of purposes. Normally, these data are stored in different database servers which locate in distributed sites, but some of them are relevant to each other [1]. Therefore MADDM (Multi-Agent Distributed Data Mining framework is proposed to integrate data mining and Multi Agent System in one platform [2] [3] [4].

In addition, the user-friendliness is another big challenge to a data mining tool. Data mining is complex because there are many resources related with data mining. Such as data source, data set, algorithm, model, report, etc. these resources should be integrated together very well in the data mining tool so that user can use and share them conveniently. There are some existed data mining systems which are working on user-friendliness such as SAS[5] and WEKA[6].

In this demo, we introduced an Agent-Based Distributed Data Mining Platform named i-Analyst. We use a case study to illustrate how agents enhance the performance of data mining.

2. SYSTEM ARCHITECTURE
As shown in Fig.1, i-Analyst can be divided into resource management layer, execution layer and common APIs. In resource management layer, algorithm SDK allows algorithm developer to generate new algorithm framework, define algorithm interface and provide data manipulation APIs. The algorithm can be plugged into i-Analyst seamlessly.

And there are algorithm management, data & visualization management, project management, case & instance management, data mining model workflow designer, project report designer, and user and privilege management. Algorithm management is used to manage self-developed algorithms created based on i-Analyst SDK, and build-in algorithms which are wrapped from 3rd party systems such as WEKA and RapidMiner. i-Analyst maintains the data in hierarchical view, data source and data set. A data source refers to the source of data, e.g. file, database, and each source may contain multiple data sets. Data & Visualization management can be used to register new data source and new data set, browse data source structure and view data set in table view or chart view. Project management, case management and instance management are used to manage data mining projects. Data mining model workflow designer is used to
generate data mining model for data mining project (shown in Fig. 2). Project report designer is used to generate project report from model execution result. All these resources are shared and can be accessed according to users’ privilege.

The execution layer is DAP (Distributed Agent Platform). The DAP is built from WADE [7]. WADE is an extension of JADE [8], that eases distributed multi-agent system development. The platform comprises of multiple agent containers. The main container hosts a persistent main service agent, known as DAP Service Agent (DSA), as it is the starting point of autonomous process. Activity Runner (AR) containers perform data mining task as specified in the data mining model. Distributed AR containers can be pre-configured or on-spot configured to join a platform. DAP maintains a platform resource database which contains system resource, such as data sets, data mining models, etc.

Some common libraries such as algorithm API, data API and Model API are shared between these two layers. Algorithm API provides an abstract layer to construct new algorithm and access to its parameters and data sets. Data API allows access to data by providing an abstract layer in which developers do not have to deal directly with the specific mechanism. Model API allows dynamic construction of the data mining model in the workflow designer to converts the graphical workflows into the data mining model.

The main characters in this system are agents. They represent most complex logic of the system. Besides WADE built-in agents, we now focus on the three DAP specific agents.

1. **DAP Service Agent (DSA)** – the agent is the starting point of the autonomous activity. The agent receives forwarded message from WSIG regarding the requested service; then it determines for an appropriate action to take. On request for instance execution, the agent verifies the request for the particular instance and forward to a Case Mediator agent.

2. **Case Mediator Agent (CMA)** – the agent receives the information about an instance, it mediates the resources whether it is local or remote accessible, then it spawns a set of Activity Runner agents to perform the actual model execution. The CMA monitors the status of the execution, collects results, and notifies DSA.

3. **Activity Runner Agent (ARA)** – the agent runs an activity, which is a component of a data mining model. ARA is the actual worker that performs the action defined in activity assigned by a CMA. After it finishes, it returns the result to the CMA.

3. **DISTRIBUTED DATA MINING**

Data mining models are presented as a workflow. Workflow is a de facto standard in most data mining software packages because of its power of visualization and clarity. In i-Analyst, a given workflow is converted into a data mining model (DMM) which can be interpreted by the execution layer. The execution is available in two options: local and distributed. When all required resources are accessible by the system, the system will by default
execute the DMM locally. But when the required resources are distributed and constrained with access permissions, DAP will analyze the DMM and form a sub system to carry on the task. A data mining task involves acquiring the resources and assigns to appropriate executors, in this system, agents. Agents are a small program unit that is capable to execute activities defined in DMMs. In each execution, DMM is translated into an execution plan for agents to follow the instructions, such as resource locations for agents to acquire. To converse the privacy constraint, agents move from one site to another which they have the i-Analyst agent platform installed. In this way, Data set can be shared but not exposed. Agents move to the site that has the data. This approach optimizes the resources because it minimizes data transmission.

4. EXAMPLE
This example shows the use of multi-agent in i-Analyst. The example is a scenario of smart trading which simulates a situation that a big broker invests 1 million dollars in three markets and he wants to maximize the profit. He assigns the money to three trading agents, and wants them to maximize the profit for him.

4.1 Setting
The example compares the performance of trading with and without agent collaboration. The trading game is defined as follow.

1. Each agent can choose a trading strategy provided to do trading on any one or up to three markets, the purpose to maximize his overall profit. The available trading strategies are Moving Average (MA), Filter Rule (FR), and On Balance Volume (OBV). A market is a data set of historical trading data of a product. The data set is from Yahoo Finance which covers from 1998 to 2008. We have more than 1,900 symbols, which we will pick only 3 to simulate each market.

2. The multi-agent system consists for three kinds of agents. (i) Broker Agent interacts with the user including receiving input and reporting the result. (ii) Simulator Agent (Game keeper) controls the sequence of the trading game. (iii) Trading Agent optimizes trading strategies (parameter tuning) and takes trading actions. During the game, It also observe other agents’ performance and behavior and decides to change its investment strategy or re-tunes strategy’s parameters

3. The agent is rewarded by commissions; commissions consist of two part, a global commission rate and a local commission rate Firstly, the global commission is to reward the overall performance for all agents equally, if the overall return is $[2\%, 5\%]$ 5% out of the overall profit will be distributed to each agent in terms of their profit shares; if more than 5%, 8%. Secondly, the local commission rate, every agent gets the same rate of commission out of his own profit made; if $[8\%, 15\%]$, 20% out of the profit he makes, if profit is negative, he has to pay back the broker on the commission rate, if $[15\%, 25\%]$, 25%; if more than 25%, 30%.

4. The agent can see other agents’ actions, in order to optimize his trading decision; if he finds another agent is in a good position of making money, he can borrow money to that agent by negotiating a commission. The commission scheme rewards to the best performing agent, so other agents can observe the difference and decide to collaborate. On the other hand, the simple execution without agent collaboration ignores the decision to invest with other agents. It only follows the trading strategy trading signal.

5. Trading performance is the total value of the portfolio. The portfolio is $<\text{fund, PROFILE}>$, where fund is the amount of money currently available and PROFILE is the set of investment profile. A profile is $<\text{instrument, units, price}>$, where instrument is a selected stock, units is the amount of units owned, and price is the last closing price.

The value of a portfolio is

$$\text{value(portfolio)} = \text{fund} + \sum \text{value(profile)}$$

Where the value of each profile is

$$\text{value(profile)} = \text{units} \times \text{price}$$

The example is implemented using the provided APIs to extend the functionality of DAP into the smart trading scenario.
4.2 Results

Fig. 3 Result

Fig. 3 shows a selected result comparing the portfolio value after 100 trading days. Multi-agent with collaboration (MAS) outperforms the single agent (SINGLE) that strictly follows the given trading strategy. MAS yields profit of 199036.61 dollars (19.9%), while SINGLE yields 71597.67 dollars (7.16%). The result shows that giving the multi-agent collaboration improves the performance of the data mining.

5. CONCLUSIONS

We have introduced a system that includes software packages and development kit that enhances data mining development to achieve better result and performance optimization and yet preserve the security and privacy of the data. The example result shows that the agents play an important role in boosting the performance and enable distribution of data and execution.

We are working on providing more evidence of the system performance measurement in more details in order to clarify the need of agents in the distributed data mining.

REFERENCE