Multi-Agent System for Credit Scoring

Fábio Silva[1], Cesar Analide[2]

Abstract The use of multi-agent systems to solve complex problems in today's world is not a new approach. Nevertheless, there has been a growing interest in using its properties in conjunction with machine learning and data mining techniques in order to build smarter systems. A multi-agent system able to classify and recommend attribute values in an instance of a dataset is presented and intended to provide to the end user a better understanding of both the classification in the dataset and client possibilities to obtain a good classification. The multi-agent system presented will have the ability to classify a user credit application and suggest different values for its attributes under assessment.

Keywords: Credit Scoring, Client Evaluation, Machine Learning, Intelligent Agents

1 Introduction

Nowadays, the use of multi-agent systems (MAS) to solve complex real world problems already exists. Agent collaboration to perform complex tasks is a concept used by many systems.

One of the most commonly accepted definitions for the term agent by the scientific community is from Wooldridge and Jennings, where they define "An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives" (Wooldridge and Jennings, 1995).

Differences between agents may exist as some might be more reactive, responding quickly to some input, or more deliberative, using information from its sensors to build an internal representation of the world in order to plan and act upon it or,

[1] Fábio Silva
Department of Informatics, University of Minho, Braga, Portugal
e-mail: f.aandree@gmail.com
[2] Cesar Analide
Department of Informatics, University of Minho, Braga, Portugal
e-mail: analide@di.uminho.pt

even, have a hybrid approach joining the reactivity with deliberation (Wooldridge, 2002).

Wooldridge and Jennings (1995) define two different notions of agents, a weak notion of agent and a strong notion of agent.

The weak notion of agent includes those exhibiting some or all of the following characteristics:

- Autonomy;
- Social Ability;
- Reactivity;
- Pro-activeness.

These properties assure the agent can perform autonomous actions, communicate with other agents and its surroundings, react to events, and exhibit a goal-directed behavior take actions after perceiving its environment.

On the other side, a strong agent, apart from the weak agent properties, may also exhibit human-like properties like:

- Mobility;
- Rationality
- Veracity;
- Benevolence.

A strong agent is then able to exhibit human-like behavior like emotions, beliefs, intentions and obligations (Analide et al. 2004).

A MAS uses agents and its interactions in order to achieve a global objective. With respect to its architectures, a MAS may be open or closed. The former does not consider a formal architecture and, for instance, its agents do not know how many of them there are in the system or their functions. Consequently, a discovery service is fundamental to obtain agent interaction. The latter may define the number of agents present in the system, its interaction and relationships. Additionally properties of a MAS may concern:

- Coordination, whether or not agents compete or collaborate between each other;
- Communication, whether information is passed through messages, as with the JADE framework (Bellifemine et. al., 2007), or stored in a global memory unit or blackboard system, like in SICStus PROLOG's Linda library (Deransart et. al., 1996);
- Organization, whether or not there is a flat organization where all agent are equal or hierarchical.

On the other hand, the use of data mining techniques in the bank services is common, specifically on loan applications, and is a subject of study by the research community (Dan, 2008).

The development of MAS together with data mining and distributed data mining has been an emerging area of research. There has been an increasing interest to combine these approaches, making intelligent systems able to learn and react better (Cao, et. al., 2007). These systems solve problems through collaborative agents, exchanging information and knowledge with the objective of reaching a solution to these problems. It also allows the interoperation of different applications through the use of agent interfaces in a multi-agent system (Xiang, 2008).

A recent study also point the fact that data mining evolution is being made by moving to soft computing techniques, with the help of cloud computing and MAS, used in fields like services or research analysis fields, for example (Venkatadri and Reddy, 2011).

This work presents a MAS able to classify instances of a dataset, in this case financial loan applications, regarding the value of the classifier attribute, also performing suggestions based on the classifiers interpretation of the dataset.

In the proposed MAS, the credit scoring and the suggestion on loan application problems are solved with the help of agents communicating with each other performing their respective objectives. These agents react to new data as it becomes available, using machine learning and data mining techniques to represent their knowledge in order to classify and recommend end users on their loan applications.

In section 2, an initial state of the art in these multi-agent systems is presented showing recent research in the area, available tools and theories to build such systems. Over section 3, a case study is presented contextualizing the use and the benefits of a MAS approach to solve it. Section 4 details the proposed MAS and the available results are discussed in section 5. Conclusions and future work are presented in section 6.

2 State of the Art

Multi-agent systems and data mining tasks have been developed in the past as two separate emerging areas. However, recently, there has been noticed a change in this regard. Research in agent mining systems presents itself as an opportunity to develop new approaches to solve challenges taking advantage of the agent interaction and learning algorithms from data mining (Cao, et. al., 2007). Recent surveys also consider the interest for data mining solutions in the banking services as a mean to solve some of its classical problems such as credit scoring, risk management and customer relationship (Dan, 2008).

The bank industry has also been a case study before with approaches trying to make a multilayered multi-agent data mining architecture for the banking domain (Xinli and Chosler, 2007). Other uses of a multi-agent system associated with data mining are related to their distributing computing properties. The collaborative agent work in a multi-agent system allows agents to share their internal information when needed, while performing their own work for the greater benefit of the system. In this context distributed data mining tasks can be successfully implemented in a multi-agent system (Zhou et. al., 2010). Distributed data mining systems are also used to tackle complex knowledge discovery problems, such as supply chain finance whereas there are heterogeneous data sources which have to be taken into account (Xiang, 2008).

The use of a multi-agent system as the base for a distributed data mining system has been observed in some research papers over the years. The idea is to combine the intrinsic properties of multi-agent systems and their agent communication to develop distributed and collaborative data mining tasks. One of the uses for such systems is the discovery of relevant pattern in a data warehouse through the use of intelligent data mining agents, acting in collaboration and performing collaborative action (Khan, 2008).

Several distributed data mining systems implemented through agent-based systems have been proposed in the literature (Klusch, et. al., 2003), some even granted special attention to multi-agent systems as a mean to integrate different data sources from heterogeneous applications, making use of specialized agents (Kahn, 2008), (Zhou et. al. 2010), (Cao et. al. 2007) and (Xinli and Chosler, 2007).

Today's availability of tools to build such systems include data mining and machine learning tools as well as many multi-agent platforms. With respect to data mining tools, Weka (Witten and Frank, 2006) and Rapidminer (Mierswa et. al., 2006) are presented as open source tools, also connected to data mining projects on the bank industry (Silva and Analide, 2010). For multi-agent system platforms, references to Jade, AgentBuilder and Zeus could be presented as examples. The intention is not to provide a complete list of tools and platforms but rather a list of examples from which multi-agent data mining systems may be developed.

3 Case Study for Loan Applications

In this paper, the problem of client classification for loan applications will be considered as a case study. This problem is often referred to as credit scoring and has been studied over the years by many researchers in order to assess client risk and predict future behavior repaying loans to financial institutions (Eletter et. al., 2010), (Islam, et. al., 2009), (Silva and Analide, 2011).

The main objective of such system is to build updatable decision mechanisms that use data and information from past events. While doing so, the system is intended to learn new trends from new data and help, both financial institutions and clients, with their loan inquiries or requests in an autonomous manner.

Additionally, a suggestion model will interact with both clients and loan providers, helping them perceive the most advantageous conditions for their loan applications. This mechanism is intended to discover which characteristics are desired in clients to grant them with a loan application, even when considering that some client attributes may be immutable. The suggestion model may also be used to investigate client types and promote new financial products and services.

The case study used here respects to a credit scoring dataset (available at the UCI repository, in http://archive.ics.uci.edu/ml). The choice fell upon a German credit dataset, where each client is characterized by a set of 20 attributes, followed by the classification of each customer, as depicted in Table 1.

Table 1 Dataset attributes

	Attribute		Attribute		Attribute
1	Status	8	Installment rate	15	Housing
2	Duration	9	Personal status	16	Existing credits
3	Credit History	10	Debtors	17	Job
4	Purpose	11	Residence	18	Liable people
5	Credit amount	12	Property	19	Telephone
6	Savings	13	Age	20	Foreign worker
7	Employment duration	14	Installment plans	21	Classification

The dataset presented in Table 1, is a combination of personal, social and financial information about bank clients. This dataset was chosen due to its attribute completeness and soundness, which could provide realistic and relevant results to this work.

Generally, at least four types of client attributes are used: demographic, financial, employment and behavioral indicators. Each of these categories are helpful characterizing the client's spatial distribution, their expenses and incomes, their financial behavior over time, and current and future employment status (Vojtek and Koenda, 2006). These indicators are important because of their soundness in helping to estimate a client's risk probability and their explanatory power when a credit-scoring method is employed to analyze a loan application.

4 Multi-Agent System

The multi-agent system development present in this section complies with the initial agent and multi-agent considerations referred to in section 1.

4.1 The system

The proposed MAS here discussed is composed of different types of agents with specialized characteristics which, when working together, are able to assess clients and suggest alternatives in a classification system. More specifically the system is intended to be open, with a hierarchical organization and uses messages to perform communication between agents.

The system implements a total of six agent types, with different functions:

- Feeder
- Model
- Decision
- Suggestion
- Monitor
- Configuration

Each agent can contribute with more than one instance in the multi-agent system.

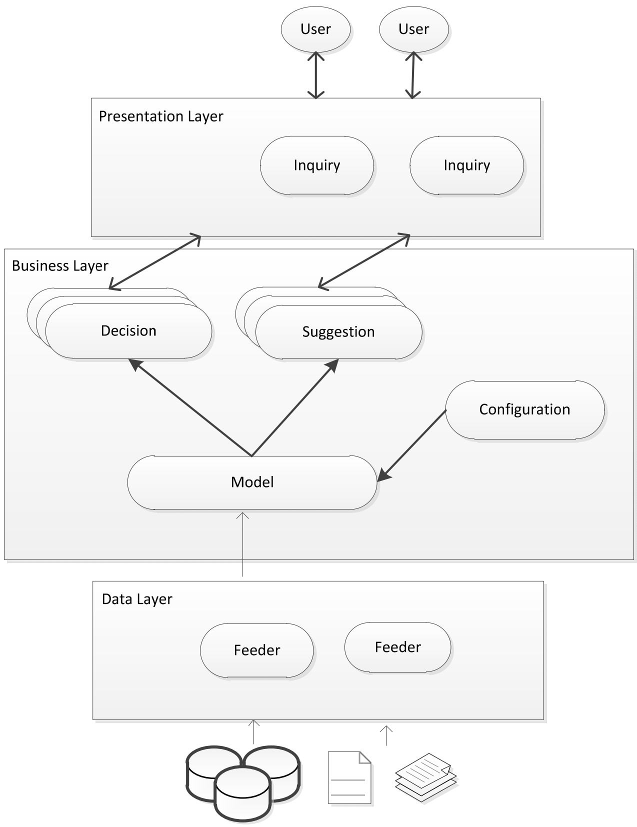

Fig. 1 Multi-Agent System for Credit Scoring.

The interaction between agents and system users is represented in Figure 1. This representation comprehends the classical three layer architecture, used in most software engineering process to separate data, business logic and user interface, and applies it to the multi-agent context.
One of the goals of this system is to classify people according to loan perspectives, bad or good client. The Suggestion agent provides a complementary feature

to the classification by analyzing incomplete client information and looking for changes in their attributes that would grant them the loan applications or, even, describing the decision process according to some types of clients.
In the following subsections, the specific behavior of each agent is described and contextualized in the system.

4.2 System Behavior

With these six different agent types, the system is built and acts in order to achieve its global objectives: to classify and suggest successful instances in the credit scoring dataset, and suggest loan application to clients, detailing alternatives to their loan applications and also explanation on why some application might be refused. An information flow can be observed, in figure 1, from the moment data is imported into the system to the moment where knowledge is shared to the Inquiry agent.
The Feeder agent is used to import a data into the system and is also responsible for monitoring the data source for new data. A Model agent, upon receiving the new data, updates its decision algorithms according to its internal classifier algorithm, which can be unique to each agent. These trained models are then shared with Decision and Suggestion agents in order to keep them informed about the latest and more accurate models. Both Suggestion and Decision agents use their classifying models to classify and suggest instances. These actions are performed upon request by the Inquiry agents present in the multi-agent system, which are the point of contact between the system and its end-users. A configuration agent exists to overview the multi-agent system, enforce some logical rules on the system, such as the maximum number of agents of each type, and enforce business policies, such as forcing classifiers to deny loans to specific instances with certain values even when the classifier considers it a good client.
All communication between agents is supported by an Agent Management System (AMS) internal to the multi-agent platform chosen, developed in Jade (Bellifemine et. al., 2007). It is requested that, upon creation and deletion, all agents in the system are registered or unregistered in the AMS service so they can be found by other agents.

4.3 Agents

In this section, each agent will be detailed and their function and purpose in the multi-agent system explained.

Feeder Agent
In order to build a data mining multi-agent system, there is a need to find and collect initial data, from which machine learning and artificial intelligence techniques will be applied. The Feeder agent is a sensor type agent, responsible to monitor the data sources environment and retrieve new data to the system. Then it must pass the data into the Model agent, who is responsible to use it to create or update his models. At any given time, there could be multiple Feeder agents, each one monitoring a data source, which can be a data warehouse cube, a logical table or, even, simple raw data files.

Model Agent
The Model agent is responsible for building classifier models from machine learning, artificial intelligence and data mining techniques. Its aim is to learn from past experiences how to assess each situation. Furthermore the Model agent is required to train a classifier algorithm which should be updatable to decrease the training time. The agent itself is only responsible for the training of a classifier, and different model agents can train different classifiers. All operations regarding the data preparation, necessary for each classifier, is handled by the specific agent.
After training or update of the classifier, the agent is responsible for passing the internally trained model, as well as information regarding the filters applied to the data to the Decision and Suggestion agents, represented in figure 2, which will use this information to complete their tasks. In terms of agent types, this agent is a deliberative agent which analyses the environment and builds the most advantageous interpretation of it as a classifier model. In order to build its internal classifiers a data mining and machine learning algorithms library from Weka (Witten and Frank, 2005), was used. Furthermore some optimization upon classifiers is possible, building hybrid algorithms connected to the credit scoring problem (Silva and Analide, 2010), which can also be used as classifier models in this agent. The Configuration agent, in figure 2, assures that business rules are passed into the classifier models, enforcing business policies on the predicted values.
The main advantage of this Model agent is to be able to make isolated training of the decision system components, meaning that other agents, dependent on this knowledge representation, can continue to operate successfully while the system itself is being updated.

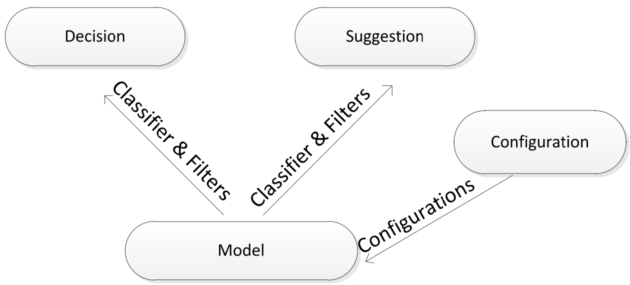

Fig. 2 Model Agent Interaction**.**

Decision Agent

The Decision agent has reactive characteristics, as it waits for requests from other models to perform its own tasks. Whenever a new model becomes available, the agent will temporally halt its execution to substitute its old representation for the new one, sent by a Model agent.

After receiving the first model, trained by a Model agent, this Decision agent is able to respond to requests from other agents requesting an evaluation of an individual with the complete list of attributes filled. Each response to requests carries not only the predicted value, but also the model version in which it was obtained. Consequently future agents might be able to justify the reason why evaluation may differ over time. This ability to explain decisions is another important motivation to develop a multi-agent based solution.

Due to service level issues, when the agent becomes stressed by the number of requests received, it has the option to launch a clone agent in the system, in order to maintain service levels, distributing incoming requests by other Decision agents.

Suggestion Agent

In order to make suggestions, the Suggestion agent uses the available knowledge to look for more advantageous solutions to incomplete evaluation requests. In the proposed multi-agent system, the Suggestion agent receives the model created by a Model agent and uses it to perform searches in the decision model, in order to obtain the most advantageous situations for the request presented.

As the decision model might not be simple to understand, a genetic algorithm was developed, which is able to use an incomplete set of attributes and search in the available domain for the values of the missing attributes, granting successful evaluations in the decision model. The suggestion is a ranked list of possibilities for the missing attributes.

The suggestive algorithm is an adaptation of genetic algorithms, in figure 3, to solve this specific suggestion problem. The idea is to use genetic algorithms to perform a search in the global space of possible solutions and deliver the positive answers to the client. The algorithm used to search such responses is a set of steps:

- Select each missing client attribute as a gene in a chromosome;
- If not created, randomly create the initial population of chromosomes; otherwise, select the best clients from the set generated earlier;
- Apply the selection operator and select pairs of chromosomes;
- In selected pairs of chromosomes, apply the crossover operator by calculating a split point to exchange genes between each pair of chromosomes;
- Apply the mutation operator and assign a random value to one gene in selected chromosomes;
- Join the gene information with the known immutable client attributes, and use the decision model as the objective function;
- If the maximum time of calculation is not exceeded or if there are still negative client classifications or if the number of desired alternatives is not met, start from the beginning; otherwise, the algorithm ends here.

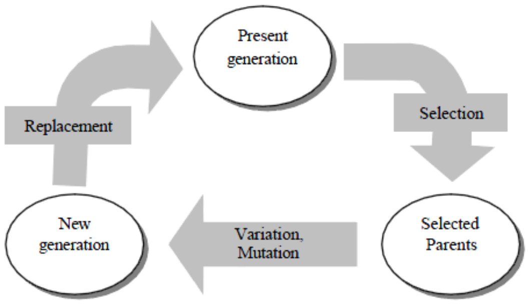

Fig. 3 Genetic algorithm evolution cycle.

In the case study, credit data system, each individual in the population will be the set of attributes that were not specified by a client. Those attributes are then generated randomly between the space of possible solutions for each attribute type. After applying the selection and mutation operators, the attributes are joined with the immutable client attributes, and a classification of each pseudo-client is done, retaining the raw classification value as the client score to select the chromosome population for the next iteration and choose the best classified clients from the possible set. The classification algorithm used in this algorithm is supposed to be already trained by the Model agent and to have an initial filter that normalizes the client set of attributes according to the rules created in the training step of the classification algorithm.

When the algorithm reaches the end of a stage, the population selected for the next iteration is the set of chromosomes that achieved better classification, that have a different combination of attributes, from the previous or the present generation. Good suggestions should provide different alternatives, so the user has different contexts to choose from. This last step assures that the answers to the initial problem are all different (Silva et. al., 2011).

Due to service level issues, as it happens with the Decision agent, when it becomes stressed by the number of requests received, the agent has the option to launch a clone agent incoming requests with another Suggestion agent.

Configuration Agent

The Configuration agent is responsible for the multi-agent system supervision. This agent enforces logical rules to the maximum number of agents in the system and business policies, such as business rules to grant or deny loans directly, without the need to go through the evaluation of a classifier trained by a Model agent. When applied, these policies could, for example, deny loan based on attributes values like unemployed people or people with bad loan history.

Inquiry Agent

The Inquiry agent, which interacts with the final end-user, receives classifying and suggestive requests and passes them to the available Suggestion and Decision agents. The human-agent interaction is possible through a graphical user interface, which reacts to the users requests, delegating the actions to other agents in the system. The decision is made when the full list of attributes is available and suggestion is performed when some values are missing.

5 Experiments

The results provided use the dataset presented in section 3 and consider the described multi-agent system with a neural network, as a classifier in a Model agent and, consequently, this is the classifier algorithm passed to the Suggestive and Decision Agents.
The instances of agents used are:

- 1 Feeder Agent
- 1 Model Agent
- 2 Decision Agents
- 2 Suggestion Agents
- 2 Inquiry Agents
- 1 Configuration Agent

5.1 Data mining

In order to prove the utility of the proposed system, some data mining algorithms were experimented in order to assess their efficiency. From a machine learning and data mining standpoint, the agents perform with various algorithms with a success rate roughly between 70% and 80% obtained from a test split of 66% for training and 33% for evaluation on the dataset. Another algorithms and optimization techniques can be found in previous works (Silva and Analide 2010), and they can be launched in the system at anytime..It is possible to update the system in the future with new classifiers that might perform even better than the ones presented in Table 2. Comparing other approaches using the same dataset, they have shown similar results despite the use of different algorithms (Silva and Analide, 2011**).**

Table 2 Data mining results.

Algorithm	Correct Evaluation (%)	Error (%)
Multilayer Perceptron	73.5	26.5
J48	77.6	22.4
Naïve Bayes	75.6	24.6
OneR	72.4	27.6

5.2 Suggestive System

The use of a suggestive system helps both the financial institution clients and employees. From the employee perspective, by using the system he is able to search for ideal conditions to advise its customers on their loan applications (Silva et. al., 2011). For example, simulating a male married customer with a history of delays in paying its former loan installments, who's applying for a 5000€ loan for educational purposes, age 30, intending to repay it in 18 months, the multi-agent system was able to provide options such as:

- Demand a savings account at the bank between 500€ and 1000€;
- To have, at least, a rented house;
- Does not need guarantors;
- Other installment plans have to be at the same bank;
- Savings account between 100€ and 500€.

From the client perspective, when searching for loan conditions, a customer enters its initial attributes and sees under what conditions its saving and checking accounts are more favorable to have a loan accepted. Using another example, simulating a 40 year old customer with good credit history, self-employed, liable for 2 people with a rented house, who wants to buy a new car valued up to 30000€ and wants to pay it in 36 months, the system may suggest:

- Saving accounts between 0€ and 200€;
- Balance of checking account 100€ between 500€;
- Installments only can take up to 66% of his disposable income.

This procedure could be repeated, to meet both client and financial institution satisfaction with other suggestions. As the suggestive system is not intended to be exhaustive, different runs might give different results.

5.3 Experience and Practice with the Multi-Agent System

As a multi-agent system, it was successfully experimented a number of properties concerning its robustness.
First, crashing one Decision and one Suggestion agents from the testing system, did not result in a system failure, since the Inquiry agent was always redirected for the available Decision and Suggestion agents. Moreover, when stressed, these last agents cloned themselves, helping them cope with the increase in inquiries by users.
The model update function was also able to be done without affecting response time in any other agent, proving that the system can be updated while performing

decision and suggestive tasks. Once a new model becomes available it is transmitted to the other agents so they can use it in their future tasks.
The Feeder agent, upon noticing new data available, was able to retrieve it to the system, and, as a consequence, initiate an update of the classifier used in the system, so that the agents can use it.

6 Conclusions and Future Work

The purposed multi-agent system was built according to the specifications and met its global objectives. The results obtained show that the system performed as expected regarding availability, robustness and updatability. A number of simulations were described, and also some potential suggestions were described, presenting both the decision capability and usefulness of the suggestive system. For instance, tests showed that the system can tolerate failure of some agents, remaining functional. Due the agent’s ability to clone themselves when the demand is high, the system creates new agents to handle the increase in traffic or to substitute agents which crashed and would be necessary.
Moreover, the suggestion component interprets successfully the knowledge patterns developed by the data mining and machine learning algorithms. When the last stage of data mining is over, the suggestion algorithm is then able to use the classification models as a source of knowledge pattern and provide suggestions based on them, regardless of the classifier.
Future work may involve developing improvements on the suggestive and classifier algorithms in order to obtain better evaluations and suggestions. Additionally, the suggestions provided from the algorithm should be validated by financial experts to validate them and assure that they have business value.
More Feeder and Model agents may also be developed to increase the availability of data sources in the system and classification algorithms. In addition to the increase on the number of agents, it can be also developed a collaboration mechanism between agents so they can decide, based on classification of different classifier models, on different agents. This collaboration mechanism should involve communication between Decision agents with different classification models and a Coordinator agent responsible to aggregate individual assessment into a final classification.

7 References

Khan D (2008) CAKE – Classifying, Associating and Knowledge DiscovEry - An Approach for Distributed Data Mining (DDM) Using PArallel Data Mining Agents (PADMAs). IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

Liu X (2008) An Agent-Based Architecture for Supply Chain Finance Cooperative Context-Aware Distributed Data Mining Systems. Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services

Dan Z (2008) Data Mining Applications in the Banking Industry in China (1998-2007). International Conference on Information Management, Innovation Management and Industrial Engineering

Wooldridge M (2002) An Introduction to MultiAgent Systems. John Wiley & Sons

Silva F, Analide C (2010) Design of an Application for Credit Scoring and Client Suggestion. Proceedings of the IASK International Conference E-Activity and Leading Technologies & InterTIC

Silva F, Analide C (2011) Information Asset Analysis: Credit Scoring and Credit Suggestion. International Journal of Electronic Business (IJEB) Special Issue: "Expertise, Innovation and Technology: Are We in the e-XXI Century", Inderscience Publishers

Li H X, Chosler R (2007) Application of Multilayered Multi-Agent Data Mining Architecture to Bank Domain. International Conference on Wireless Communications, Networking and Mobile Computing

Witten I H, Frank E (2005) Data Mining: Practical Machine Learning Tools and Techniques, Elsevier, San Francisco

Wooldridge M, Jennings N R (1995) Intelligent agents: theory and practice. The Knowledge Engineering Review

Venkatadri M, Reddy L C (2011) A Review on Data mining from Past to Future. International Journal of Computer Applications vol 15, No. 7, 19-22

Vojtek M, Koenda E (2006) Credit-scoring methods. Czech Journal of Economics and Finance.

Silva F, Analide C, Novais P (2011) Credit Scoring Data for Information Asset Analysis. Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011

Eletter S F, Yaseen S G, Elrefae G A (2010) Neuro-Based Artificial Intelligence Model for Loan Decisions. American Journal of Economics and Business Administration

Zhou D, Rao W, Lv F (2010) A Multi-Agent Distributed Data Mining Model Based on Algorithm Analysis and Task Prediction. 2nd International Conference on Information Engineering and Computer Science (ICIECS)

Islam Md S, Zhou L, Li F (2009) Application of Artificial Intelligence (Artificial Neural Network) to Assess Credit Risk: A Predictive Model for Credit Card Scoring. Dissertation, School of Management Blekinge Institute of Technology

Bellifemine F L, Caire G, Greenwood D (2007) Developing Multi-Agent Systems with JADE. JohnWiley

Cao L, Luo C, Zhang C (2007) Agent-Mining Interaction: An Emerging Area. Autonomous Intelligent Systems: Multi-Agents and Data Mining, Lecture Notes in Computer Science

Mierswa I, Wurst, M, Klinkenberg R, Scholz M, Euler T (2006) Yale: rapid prototyping for complex data mining tasks. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining

Analide C, Novais P, Machado J, Neves J (2004) Anthropopathy and its Assessment in Virtual Entities. Innovations through Information Technology, Proceedings of the IRMA 2004 International Conference

Klusch M, Lodi S, Moro G (2003) Agent-based Distributed Data Mining: The KDEC Scheme. Intelligent Information Agents- The Agent Link Perspective Lecture Notes in Computer Science

Deransart P, Cervoni L, Ed-Dbali A (1996) Prolog: The Standard. Springer-Verlag, London