Information Sciences 506 (2020) 258–272

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Argumentation based reinforcement learning for meta-knowledge extraction

Junyi Xu[a], Li Yao[b,*], Le Li[c], Ming Ji[a], Guoming Tang[b]

[a] Center for Assessment and Demonstration Research, Academy of Military Science, Beijing, China
[b] College of Information System and Management, National University of Defense Technology, Changsha, China
[c] Audit Office of Central Military Commission, Beijing, China

ARTICLE INFO

Article history:
Received 9 August 2016
Revised 23 July 2019
Accepted 29 July 2019
Available online 5 August 2019

Keywords:
Argumentation
Reinforcement learning
Knowledge integration
Distributed data mining

ABSTRACT

Knowledge integration in distributed data mining has received widespread attention that aims to integrate inconsistent information locating on distributed sites. Traditional integration methods become ineffective since they are unable to generate global knowledge, support advanced integration strategy, or make prediction without individual classifiers. In this paper, we propose an argumentation based reinforcement learning method to handle this problem. Inspired by meta-learning, we integrate distributed knowledge and extract meta-knowledge that is agreed-upon knowledge consistent to all the agents. Specifically, two learning stages are introduced: *argumentation based learning* stage integrates and extracts meta-knowledge, and *reinforcement learning* stage evaluates and refines meta-knowledge. The two learning stages run alternately to extract global meta-knowledge base, which can be used to make prediction directly. The results from extensive experiments demonstrate that our method can extract refined meta-knowledge with a much satisfied performance.

1. Introduction

With the fast development and pervasive applications of data acquisition technologies, we have gain much more information than before and entered the era of big data. Traditional data mining techniques cannot handle the challenges of volume, velocity, variety and veracity (so called "4V") in big data. As an alternative, distributed data mining (DDM) [5,39,48] has received widespread attention recently, which aims to use distributed computing technology to extract knowledge from distributed data sources. With the advantages of better security and powerful computing capability, DDM can deal with data mining tasks on large-scale datasets. However, there are still some challenging problems in DDM, such as fragmentation, communication cost, integration and data skewness [20,38,48], in which the knowledge integration problem is currently a hotspot.

Knowledge integration in DDM focuses on integrating distributed inconsistent knowledge to obtain global knowledge [31,37]. In recent years, a variety of integration methods have been proposed to handle this task, such as meta-learning, stacking, boosting [18,19,42,46]. According to different strategies, these methods build diverse individual classifiers and combine their results to make the final prediction (which are essentially ensemble methods). However, there are still some limitations: i) most ensemble methods only integrate distributed classification results rather than knowledge; ii) classification

* Corresponding author.
E-mail addresses: xujunyi0923@163.com (J. Xu), arena_yao@163.com (L. Yao).

https://doi.org/10.1016/j.ins.2019.07.094

knowledge is stored dispersedly in individual classifiers without extracting global knowledge; iii) the classification process requires the participation of all individual classifiers, resulting in a high computational cost.

Also considering the rapid and continuous growth of data volume (one of the key characteristics of big data), many data-intensive computing applications require the learning algorithm to be capable of incremental learning from large-scale dynamic data, and to build up the knowledge base over time to benefit the future learning and decision-making process [26]. Therefore, in dynamic environment, how to mine high-quality knowledge in real-time and improve the classification ability, are still challenges in distributed knowledge integration field.

To tackle the above problems, we need to find a way to cope with distributed knowledge integration and extraction in large-scale dynamic environment. Inspired by the idea of meta-learning [42], we extract the global meta-knowledge from inconsistent distributed knowledge bases. By constructing arguments for and against a hypothesis, argumentation technique can effectively deal with inconsistent information, making it particularly useful for knowledge presentation, knowledge extraction and reasoning within expert systems [8,12]. Therefore, we introduce argumentation technology to integrate distributed knowledge from different agents. Through adequate interactions between agents, learning from argumentation can effectively resolve conflicts and extract meta-knowledge.

Furthermore, in dynamic data environment, we aim to improve the classification ability of meta-knowledge in real-time. With the capability of learning from dynamic environment, Reinforcement Learning (RL) [11,41] can accumulate experience over time, use such knowledge for future learning and thus improve the prediction performance. By learning from trial-and-error estimation and delayed rewards, reinforcement learning is able to cope with learning problems that are complex and difficult to be described. Well-understood algorithms with good convergence and consistency properties are available for solving the single agent reinforcement learning task [11], where the agent knows the state of environment and the reward function. Therefore, we use reinforcement learning to improve the classification ability of meta-knowledge. By utilizing feedbacks from environment, reinforcement learning is able to refine meta-knowledge, making it suitable for data analysis on dynamic data environment.

In this work, we combine reinforcement learning and argumentation, and propose a novel Argumentation based Reinforcement Learning method (ArgRL) to achieve meta-knowledge extraction in distributed data mining. Specifically, ArgRL framework consists of two learning stages: (1) *argumentation based learning* stage, implementing meta-knowledge integration and extraction; and (2) *reinforcement learning* stage, realizing meta-knowledge optimization. With the analysis on dynamic data, the above two stages alternate and interrelate, which can achieve global meta-knowledge extraction and meta-knowledge spiral process [35].

In summary, we make the following contributions:

(1) We design a novel argumentation based reinforcement learning framework to deal with the task of knowledge integration in distributed data mining, and study how to integrate, in a single framework, the capabilities of *argumentation based learning* and *reinforcement learning*, to realize meta-knowledge extraction.

(2) We apply the independent meta-knowledge base, which is extracted from distributed data and can be directly applied to solve the classification task without involving individual classifiers.

(3) We propose a framework that is flexible to be paired with other argumentation schemes or learning strategies, and hence, the framework can be used to extract high-quality knowledge in various domains.

(4) We perform comprehensive experiments over the public datasets. The results demonstrate that our method has good convergence property, and can extract refined meta-knowledge base with a better classification performance.

The rest of the paper is organized as follows: We review the existing work in Section 2, and propose the argumentation based reinforcement learning framework in Section 3. Designing and Implementation of ArgRL are presented in Section 4, and the argumentation framework: Arena is introduced in Section 5. Extensive experimental results are demonstrated and discussed in Section 6, followed by conclusion in Section 7.

2. Related work

As a special communication mechanism in MAS (Multi-Agent System), argumentation technology is with powerful description, rigorous reasoning abilities and other advantages [13]. A wide range of argumentation frameworks have been proposed in recent years, such as Dung's argumentation framework [16], preference-based argumentation framework [1], value-based argumentation framework [7] and so on. A comprehensive review of argumentation techniques for multi-agent systems can be found in [13].

Conventional argumentation techniques focus on knowledge representation and logical reasoning, which used to handle inconsistent information [2,28]. In recent years, researchers propose to combine machine learning techniques with concepts of argumentation technology, mainly for the following purposes.

(1) Generating and using arguments. Mozina et al. [33] try to integrate arguments into machine learning framework, which assumes the arguments as the input of the learning process. While Ontañón and Plaza [36] present an argumentation approach to generate arguments by induction and use them to reach agreements among agents. They focus on integrating the capabilities of three processes: the inductive learning process, the argumentation process and the hypothesis revision process.

(2) Applications on classification problems. Amgoud and Serrurier [3,4] focus on the classification problem and propose a general formal argumentation-based model that constructs arguments for/against each possible classification of an exam-

ple. By exploiting the idea of active learning, Napierala and Stefanowski [34] introduce an argument based rule induction algorithm with a specialized classification strategy for imbalanced classification tasks.

(3) Building expert systems. By using arguments as means for experts to elicit some of their knowledge through explanations of the learning examples, Groznik et al. [23] develop a neurological decision support system to help the neurologists differentiate between three types of tremors.

More recently, data mining technology has received widespread attention, making the combination of argumentation and data mining technology become a new hot topic [24]. Wardeh et al. propose argumentation from experience in [43], which combines the argumentation theory with data mining techniques. By using association rules as arguments in the knowledge base of agents, PADUA (Protocol for Argumentation Dialogue Using Association Rules) argumentation model can effectively achieve two party argumentation process and resolve uncertainties in classification problems. And then, Wardeh et al. [45] describe an approach for multi-agent classification using an argumentation from the experience paradigm: PISA (Pooling Information from Several Agents). The concept of collaborative group of agents is also proposed for arguing from experience in [44].

Both ArgRL and PISA use data mining technology to obtain distributed rule bases for each agent, while our argumentation strategy in this paper is different from PISA. PISA tries to handle the classification task by using argumentation, while our method aims to evaluate the distributed local knowledge by argumentation and to extract high-quality meta-knowledge based on RL. In PISA, the overall argumentation process presents a competitive ensemble approach, whereas the inter-group decision making procedure is more like a cooperative ensemble approach. In our method, the relationship between agents is totally cooperative, and the purpose of argumentation is to extract global knowledge, which we believe is more suitable to solve distributed knowledge integration problem in DDM.

Researchers also investigate the integration of reinforcement learning and argumentation, but most of them use RL to optimize the capability of argumentation itself. For instance, Georgila and Traum [21] learn argumentation policies for negotiation by using RL. Monteserin and Amandi [32] apply reinforcement learning to improve the argument selection strategy. In contrast, the reinforcement learning applied in our model is used to evaluate and refine meta-knowledge, rather than modifying argument selection or learning appropriate negotiation policy as shown in the previous work.

3. Argumentation based reinforcement learning

3.1. The n-Armed Bandit model

Our method assumes that the dynamic data environment has instant feedback. In dynamic data environment, when the system selects a rule from the meta knowledge base to classify the current case, the environment will immediately return a numeric feedback, 1 or 0, representing correct or incorrect classification. The instant feedback is only related to the current case, and has nothing to do with historical cases. Cases to be classified are independent of each other, i.e., classification of the current case does not affect the next case to be classified in the environment.

In our method, we can only get a simple feedback, 1 or 0 to show that the classification result is success or not, rather than the actual label. For example, in the field of online advertisement serving and evaluation, a website aims to select the most suitable one from a large number of resources given by advertisers and serves to users who are browsing the website, in order to increase corresponding advertisement click rate. Depending on whether the user clicked on an advertisement or not, the website receives a feedback immediately. Through multiple feedbacks from users, the website needs to develop an optimal strategy for advertisement serving so that users will click on more advertisements.

In addition, in dynamic environment, the data to be classified is continuously arriving and the scale of data keeps growing. Therefore, we need an online learning strategy to learn how to select the best rule from the meta knowledge base, to enhance the classification ability of our model for dynamic data classification tasks.

In response to the above questions, we use reinforcement learning method to improve the online learning ability. In the above learning task, once an agent performs a classification action, it can immediately get the environment feedback, rather than obtaining the feedback after multiple classification actions. As it can be seen, this learning task corresponds to a one-step, non-episodic reinforcement learning task.

In fact, one-step reinforcement learning task can be corresponding to the classic theoretical model, n-Armed Bandit. The problem can be described as follows. Supposing there is an n-armed bandit, and given a limited number of times, each time the gambler has the option of pulling one arm of the bandit. Each arm spits coins with a certain probability that the gambler does not know. And n-arms of the bandit are not exactly the same. The goal of the gambler is to learn a strategy to maximize the reward, i.e., earning the most coins. Inspired by the n-Armed Bandit problem and its corresponding model, in the process of action selection, we can treat n actions as n arms in the bandit. So the question turns to be, how to select actions to get the maximum expected reward.

We denote the true value of action a as $Q^*(a)$, and the estimation value of action a at the t-th time is $Q_t(a)$. The true value of an action is the expectation reward received by the action, so a natural way to estimate it is to average the rewards that the action has received. In other words, if the action a has been selected k_a time before the t-th selection, and the rewards obtained by the action a are $rwd_1, rwd_2, \dots, rwd_{k_a}$ respectively, then the estimation value of the action a can be calculated as follows:

$$Q_t(a) = \frac{rwd_1 + rwd_2 + \ldots + rwd_{k_a}}{k_a}. \tag{1}$$

After the estimation value of an action is defined, we can design the action selection strategy. The simplest action selection strategy is to choose the action with the largest estimation value, that is, select the greediest action a^* in the t-th time of selection. The estimation value of the action a^* is $Q_t(a^*) = \max_a Q_t(a)$. This strategy always uses the current knowledge to maximize immediate rewards and does not take time to explore better behaviors. A better alternative is epsilon greedy strategy, which is used to select rules from meta knowledge base to make predictions in our method.

3.2. The Process of ArgRL

Inspired by the idea of meta-learning, our method (ArgRL) aims to perform meta-level learning based on distributed data mining to extract global meta-knowledge.

Our model consists of two learning stages: *argumentation based learning* stage and *reinforcement learning* stage. Firstly, on the basis of distributed data mining, our model uses argumentation technology to extract meta-knowledge from historical datasets. Then, in dynamic data environment, we apply reinforcement learning technique to evaluate the quality of meta-knowledge and improve the classification ability of meta-knowledge in real-time. During the application, both of the two learning stages run alternately to extract and evaluate meta-knowledge in dynamic environment. The framework of ArgRL is shown in Fig. 1.

In reinforcement learning, the data perceived from the external environment (e.g., using the sensor) will be treated as *state* [30]. Similarly in ArgRL, for the classification task, the case to be classified can be seen as the data perceived from the external environment, which is treated as the current *context* of the classifier. When handling the classification task, all the cases to be classified are treated as the *environment* in the reinforcement learning stage of our model.

Given a *context*, our model makes prediction on the basis of *meta-knowledge base*, which stores global knowledge extracted from local knowledge base, and correspondingly, *action* refers to predicting a specific case on the basis of meta-knowledge base. *Reward* refers to a feedback from environment according to an action behavior. As traditional setting in RL, environment feedback is a reward signal, 1 or 0, to represent whether the classification is correct or not, rather than the correct label.

Given the current *context* as an input, an action will be selected by meta-knowledge base according to specific strategies. Then, environment transfers to a new context, and returns a scalar reward signal that evaluates the quality of this action.

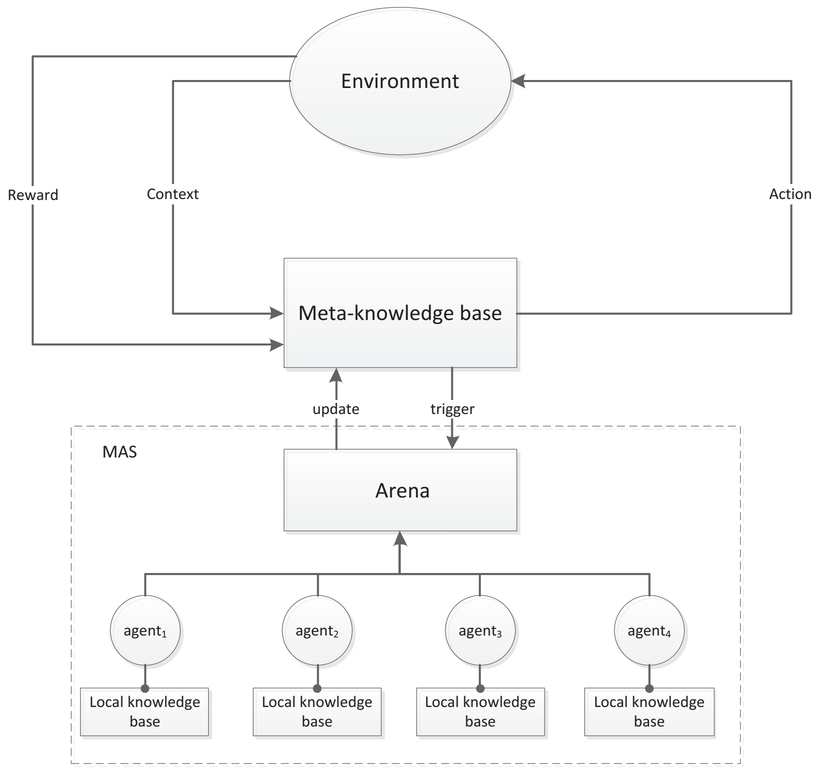

Fig. 1. The framework of ArgRL.

Our model records this reward and updates the average reward value of the meta-knowledge. Depending on the reward value, multi-agent system will decide whether or not to trigger the argumentation based learning process.

In argumentation based learning stage, multi-agent system performs data mining on distributed historical datasets to construct independent local knowledge bases. Triggered by the feedback from environment, multi-agent system will start an argumentation process. In this situation, our model treats the environment *context* (i.e., the case to be classified) as an argumentation topic, and delivers it into the multi-agent argumentation framework (*Arena* platform). The argumentation topic perceived from the external environment is an instance that requires to be discussed by participating agents according to their experiences, that is, a specific case to be classified.

During the argumentation, agents argue on the current topic, and use local knowledge to construct arguments to communicate with other agents. Then the winning rule will be regarded as the optimal meta-knowledge, and added into the meta-knowledge base to realize meta-knowledge supplement. For a specific topic, the winning rule of argumentation is superior to the local knowledge of each agent. Due to different processes of argumentation, the winning rules on the same topic may be different. Therefore, the optimal meta-knowledge on a specific topic is not the global optimal solution, and it can only be regarded as a local optimal solution.

In order to make classification effectively in dynamic environment, we need to refine the meta-knowledge base. Simply performing accumulation and de-duplication is not enough, and only considering the overall performance of meta-knowledge base is also ineffective. Instead, it is necessary to evaluate the classification ability of each rule in the meta-knowledge base. To this end, in the reinforcement learning stage, our model not only calculates the overall performance metric, but also records the average reward of each rule in meta-knowledge base. Average reward value represents the classification ability of a rule to some extent, which can be used to evaluate the meta-knowledge. Then, the meta-knowledge with lower reward values will be eliminated in application scenario, in order to extract a more refined meta-knowledge base.

In the bottom layer, argumentation based learning stage can realize preliminary knowledge extraction from the distributed historical data and give supplement to meta-knowledge base, which provides support for the reinforcement learning process in the upper layer. Reinforcement learning evaluates the meta-knowledge extracted from argumentation, and applies them in dynamic data analysis directly, which achieves meta-knowledge evaluation towards practical applications. With the analysis of large-scale dynamic data, the two learning stages coordinately accomplish the meta-knowledge integration and extraction, thus realizing the spiral ascending process of meta-knowledge base.

4. Implementation

4.1. Generalization of state space

Generalization is a technique commonly used in reinforcement learning that makes informed decisions in the states that have never been met in the past. The states in our model are generalized to share the historical classification experiences with future classification tasks.

In the process of rule selection, the best action for classification should be based on the most valuable rules. By generalization, it is possible to use the most valuable rules to make classification based on historical experience in a state that has never been met before. Since different cases may use the same rule for classification, these cases may be attributed to the same state type. Therefore, based on this assumption, this model achieves the sharing of historical experiences by generalizing the state space. The generalization method is defined as follows.

Definition 1. (Generalization of State Space): Let a state s' be a to-be-classified case satisfying the prerequisites of the global rule gk, and obtain a state-action pair $<s', gk>$. Due to a large number of satisfied states for gk, multiple state-action pairs can be generated. Think of satisfied states of gk as a category, that is, generalize states according to their satisfied global rules. Use s to denote the states that satisfied gk, and then the state-action pair after state generalization is denoted as $<s, gk>$.

From Definition 1, we can see that after the generalization of state space, different cases to be classified by the same rule are considered as the same state type. In this way, to a specific state, the classification correctness of rules is not certain, but a probability distribution.

4.2. Average reward

The most popular option among the included studies is the use of *rules* that explain the arguments, facilitating its understanding [13]. In our model, rule gk in the meta-knowledge base is defined as the following tuple: Given an environment state s, rule gk is represented as $gk = (attri, con, conf, \delta_s^{gk})$, where *attri* refers to the premises of rule *gk; con* refers to the consequent of rule gk, i.e., conclusion; *conf* refers to the confidence of rule gk; δ_s^{gk} is the average reward of rule gk until state s.

In order to choose the best rules from the meta knowledge base for classification, we need to evaluate the quality of each global rule in the meta knowledge base. Therefore, in reinforcement learning, we use the average reward δ^{gk} as the utility value in the learning process, and through the reinforcement learning on dynamic data, we continuously update the average reward values of global rules to estimate their true qualities.

In the ArgRL model, we can calculate the average reward of the global rule under state s, i.e., the cumulative reward of global rule at state s, divided by the number of times the global rule participates in classification. Due to dynamic data environment, the average reward value of the global rule gk changes with the classification process of dynamic data.

The reward function is formally defined as follows. According to the classification result, the reward at the i-th classification rwd_i is:

$$rwd_i = \begin{cases} 1, correctly\_classified \\ 0, incorrectly\_classified \end{cases} \tag{2}$$

Let the average reward value of global rule gk be δ^{gk}. If global rule gk is used t times during the classification process, and the obtained rewards are $rwd_1, rwd_2, \ldots, rwd_{\mathrm{t}}$ respectively, then the average reward value δ^{gk} is:

$$\delta^{gk} = \frac{1}{t}\sum_{j=1}^{t} rwd_j \tag{3}$$

In the initial state, the average reward of global rule gk is $\delta_0^{gk} = 0$.

In the process of reinforcement learning, we use the Monte Carlo method to update the average reward value δ^{gk}. Assuming that global rule gk is used t times in classification, the update of the average reward of global rules gk is as follows.

$$\delta_t^{gk} = \delta_{t-1}^{gk} + \frac{1}{t}(rwd_t - \delta_{t-1}^{gk}) \tag{4}$$

where t is the number of times that the current global rule gk has participated in the classification, rwd_t is the instantaneous reward obtained by global rule gk selected to make classification in the t-th time, δ_t^{gk} is the average reward value of global rule gk after participating t times classification, and δ_{t-1}^{gk} is the average reward value of global rule gk after participating t-1 times classification.

4.3. Action selection strategy

In this model, a rule that can be selected for classification must satisfy several constraints. Firstly, the global rules for classification must exist in the meta knowledge base, and moreover, it should match the information of the current case. Therefore, given a case to be classified, there will be only a small part of the global rules that can be candidates. In this case, we use the following selection strategy for the set of candidate rules.

If the candidate rule set is not empty, the model uses ε greedy strategy for rule selection. This strategy achieves exploration with a small probability ε, and achieves exploitation with the probability 1-ε. That is, we select randomly by a certain probability ε in the candidate rule set; we select the rule with the highest average reward with probability 1-ε. Generally, ε is chosen as a smaller value, such as 0.1. In our model, ε is chosen dynamically to balance the probability of exploration and exploitation in different periods, so it can be defined as $\varepsilon = \frac{1}{\sqrt{i}}$, where i is the number of times of the current classification.

In the above process, if there are multiple rules with the highest average reward at the same time, the global rule with the highest confidence is selected. This is because the average reward is calculated during the reinforcement learning process according to the feedback obtained in dynamic data classification, and it represents the classification ability of global rule in application process. Correspondingly, the confidence of global rule, obtaining from historical data in data mining stage, represents the quality of global rule in historical experience. Therefore, in the application process, in order to better measure the quality of global rules, we consider the average reward at first, and make use of confidence to compare different global rules if multiple average rewards are the same.

In addition, if the candidate rule set is empty, the classification action will be null, indicating that the model is incapable of classifying the current case. In this situation, the model will trigger the argumentation process. The current case will be put into the Arena platform as an argumentation topic, and the meta knowledge suitable for the current case will be extracted from distributed local knowledge bases to supplement the meta knowledge base. In addition, when the environment feedback is 0, that is, the current case is incorrectly classified, the system will also trigger the multi-agent argumentation. In the future learning, the new meta knowledge extracted from argumentation may be chosen to participate in classification, and to some extent achieve the exploration of knowledge.

4.4. Algorithm of ArgRL

The main control algorithm of ArgRL explains how ArgRL choose the best rule from meta knowledge base to handle the dynamic data classification task. The pseudocode of the algorithm is shown in Algorithm 1.

Algorithm 1 Main control algorithm of ArgRL.

```
input: dynamic_dataset E; meta_Knowledge_Base GK;
output: meta_Knowledge_Base GK;
for each(gk ∈ GK) do
    gk.δ = 0; /*initialization*/
    gk.count = 0; /*record the times of classification*/
end for i = 0;
while E ≠ ∅ do
    e = E[i];
    i++;
    ε = 1/√i;
    GK' = matching(GK, e); /*Get candidate rules that match the current case e*/
    if GK' = ∅ then
        gk = Arena(e); /*Use Argumentation to supplement global rules*/
        GK = GK ∪ {gk};
    else
        rand = random();
        if rand < ε then
            gk is chosen randomly from the candidate rule set GK';
        else
            if there is only one rule with the max δ then
                gk = arg max_{gk∈GK'} gk.δ;
            else
                gk = arg max_{gk∈{gk|gk.δ=max}} gk.conf;
            end if
        end if
        label = classifier(e, gk);
        rwd = perceive(label);
        gk.count = gk.count + 1 ;
        gk.δ = gk.δ + (1/gk.count)(rwd − gk.δ) ; /*Update the average reward*/
        if rwd = 0 then
            gk = Arena(e); /*Use Argumentation to supplement global rules*/
            GK = GK ∪ {gk};
        end if
    end if
end while
return GK;
```

5. Argumentation framework: Arena

5.1. Definition and structure of Arena

Arena is a dialectic analysis model for multiparty argument games, which provides a competition platform for participating agents. Formally the Arena framework can be described as follows:

$$Arena = (Referee, \mathcal{P}, \mathcal{L}, \mathcal{O}, \mathcal{R}oles, Rules, \mathcal{H}) \tag{5}$$

where: (i) *Referee* refers to the agent that manages the argumentation process according to dialogue rules of Arena; (ii) $\mathcal{P}$ is the set of agents participating in argument game on Arena; (iii) $\mathcal{L}$ is the language used in Arena, which includes the language of content $L_{Content}$, a set L_c of speech acts for administrating match, and a set L_A of speech acts for arguing between Master and Challenger; (iv) $\mathcal{O}$ is the Ontology which represents the common knowledge believed in all the participants of Arena; (v) $\mathcal{R}oles$ is the set of roles played by the participating agents in Arena, which consists of Master, Challenger and Spectator ; (vi) *Rules* is the set of the game rules abided by all the participating agents in Arena, and (vii) $\mathcal{H}$ records the history of the multiparty dialogues happened in Arena using the grid of dialectic analysis trees.

The basic structure of Arena is shown in Fig. 2.

In Arena, there can be only one Master and one Challenger in an argumentation, while other participants are just Spectators which are not allowed to propose arguments. Therefore, during an argumentation, participating agents need to compete for Master or Challenger continually to propose their opinions. At the beginning of an argumentation, Referee broadcasts the discussion topic, and the first agent who proposes an argument about the current topic will be selected as the Master. All

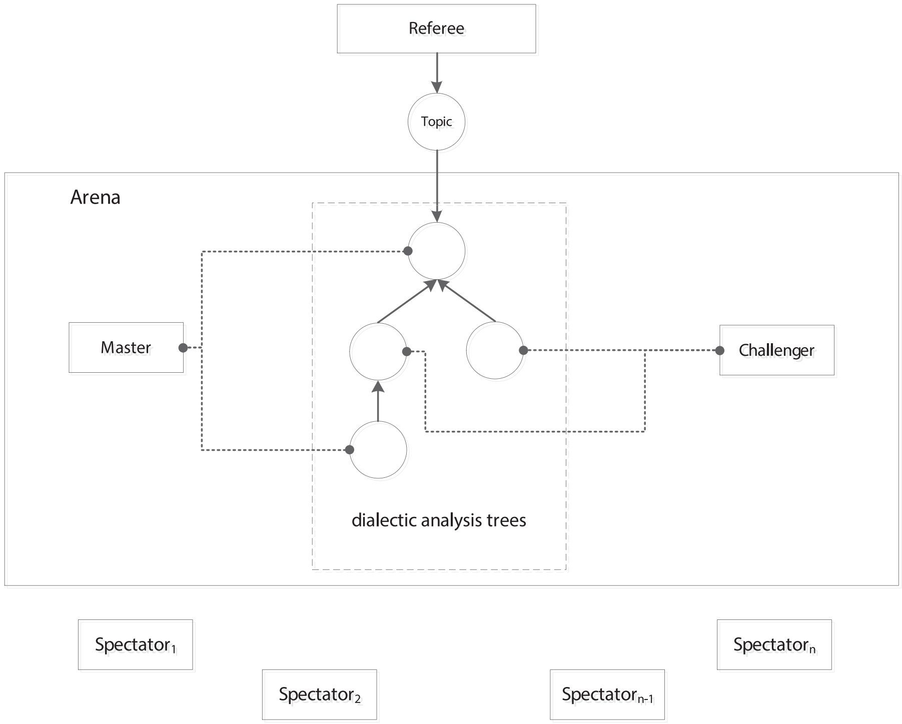

Fig. 2. The basic structure of Arena model.

the other participants whose options are different from the Master can challenge the Master and form the queue of challengers, and the first participant in the queue will be selected as the Challenger. All the other participant agents except Master and Challenger are Spectators of Arena.

Once Master and Challenger are identified, agents will use the speech acts for constructing Master-Challenger dialogues to perform argumentation in Arena. The whole process of argumentation is stored in dialectic analysis trees. Note that the argumentation between Master and Challenger only focuses on Master's opinion and experience, and it has nothing to do with Challenger's opinion. That is to say, only Master is entitled to propose his opinion about the current topic, and Challengers just argue on flaws in the Master's opinion.

5.2. Argument scheme

Supposing that x represents a topic under discussion, agents can produce arguments from their individual knowledge base. Argument scheme is defined as follows: Premises: $l_1(x), l_2(x), \ldots, l_n(x) \to$ Conclusion: $w(x)$; Confidence: c; Exceptions: $e_1, \ldots, e_k$; Conditions: $u_1(x), u_2(x), \ldots, u_s(x)$; $\neg v_1(x), \neg v_2(x), \ldots, \neg v_t(x)$.

Such argument scheme for experience can be read as follows: In my experience, if anything x doesn't belong to $\{e_1, \ldots, e_k\}$, with features $u_1, u_2, \ldots, u_s$ and not with features $v_1, v_2, \ldots, v_t$, then x with features $l_1, l_2, \ldots, l_n$, are w with probability c.

5.3. Speech acts

The speech act is the language of participant to communicate with others in argumentation, which collectively forms the basic building blocks for constructing Master-Challenger dialogues. In Arena, there are six speech acts: *ProposeOpinion*, *DistinguishRule*, *CounterRule*, *BeInapplicable*, *BeAnException*, and *Defeated*, which can be divided into three categories according to the function.

(1) Applying for Master.

ProposeOpinion: Agent A_i uses this speech act to propose the opinion about the topic t by selecting the rule k which matches the topic t ($k \in AR_i \wedge premises(k) \subset t$) and has the highest confidence from his local knowledge AR_i ($conf(k) = \max\{conf(k') | k' \in AR_i\}$).

(2) Proposing different opinion.

CounterRule: Agent A_i uses this speech act to attack the adversary's argument k' by selecting the rule k with higher confidence ($k \in AR_i \wedge premises(k) \subset t \wedge conf(k) > conf(k')$).

Table 1
Speech acts in Arena.

Move	Label	Next move	Attack with new rule
1	ProposeOpinion	2,3,4,5	yes
2	CounterRule	3,4,5	yes
3	DistinguishRule	4,5	yes
4	BeInapplicable	0	no
5	BeAnException	0	no

DistinguishRule: Agent A_i uses this speech act to attack the adversary's argument k', and allows the addition of some new premise(s) to a previously proposed rule k', so that the confidence of the new rule k is lower than k' ($k \in AR_i \wedge premises(k) \subset t \wedge premises(k) \supset premises(k') \wedge conf(k) < conf(k')$). It means the adversary's argument k' is unreasonable because of not considering enough information about the topic t.

BeInapplicable: Agent A_i uses this speech act to state that the adversary's argument k' is inapplicable to this topic t, since the Conditions $\{u_1, u_2, \ldots, u_s;$
$\neg v_1, \neg v_2, \ldots, \neg v_t\}$ of k' in his own knowledge base AR_i doesn't satisfy topic t ($(\{u_1, u_2, \ldots, u_s\} \not\subset t) \vee (t \cap \{v_1, v_2, \ldots, v_t\} \neq \emptyset)$), which means agents have different experience for k'.

BeAnException: Agent A_i uses this speech act to state that the topic t is in exception $\{e_1, \ldots, e_k\}$ of adversary's argument k' with the same Premises and Conclusion in his own knowledge base AR_i ($t \in \{e_1, \ldots, e_k\}$).

(3) Conceding defeat.

Defeated: agent A_i uses this speech act to state that he is defeated.

In Arena, each agent only uses the above speech acts from local knowledge base to attack the adversary according to Table 1.

5.4. Termination condition

If the Master is defeated by the Challenger, then the Challenger will become the new Master, and he can propose his opinion about the current topic from his own knowledge base. Then all the other participants decide whether or not to challenge this new opinion, and form a new challenger queue. Otherwise, if Challenger is defeated, the next participant in the challenger queue will be selected as the new Challenger, and the argumentation continues.

Termination conditions of Arena are as follows: If Master can defeat all the challengers, Master wins the argumentation and Master's winning rule will be considered as the optimal knowledge for the current topic; if there is no agent applying for Master, the argumentation is tie. Since the number of arguments produced by participants is finite and the defeated arguments are not allowed to use repeatedly, the termination of argumentation can be guaranteed.

5.5. Argument evaluation in reinforcement learning

In our method, each agent will mine association rules from distributed historical data at first, to get the local association rule base AR_i (AR_i only depends on the historical data, so it will not change with the environment once it has been generated). In the argumentation process, association rules will be transformed into arguments, and the strength of argument is defined as the confidence of association rule. Throughout the classification process, the unknown data will be converted to the environment context (i.e., cases to be classified), and input to the model for prediction. It should be noticed that, we strictly follow the scenario of reinforcement learning, that is, the environment feedback is a reward signal, 1 or 0, to represent whether the classification is correct or not, rather than the correct label.

When meta-knowledge base fails to make correct prediction, the model will trigger the argumentation process to further evaluate the strength of arguments. At this point, only the attributes of the cases are input into the multi-agent system. According to arguments in local association rule base AR_i, each agent needs to evaluate the strength of arguments by using specific argument schemes and speech acts. And then, the argument with the highest strength for the current case will be extracted and supplemented to the meta-knowledge base.

5.6. Algorithm of Arena

The pseudopod of Arena is shown in Algorithm 2.

6. Experiment

6.1. Datasets

In order to empirically evaluate our method, we use seven public machine learning datasets from UCI Machine Learning Repository [6](The URL of datasets is http://archive.ics.uci.edu/ml/datasets.html). Detailed description of the datasets and specific parameters of our method are shown in Table 2.

Algorithm 2 Algorithm of Arena.

input: Argumentation topic s; Agents AS;
output: winning rule k;
Broadcast s; /* s is the topic of argumentation */
Q_p = Generate_Participants(AS, s); /* Q_p is the queue of participants*/
for each($P_i \in Q_p$) **do**
    Propose_Argument (P_i, s, AR_i); /*AR_i is the local knowledge base*/
    if P_i == silence **then** /*P_i is defeated*/
        select next participant P_{i+1}
    end if
    if only P_i == active **then**
        P_i = winner;
        k = Main_Argument(P_i);
        return k
    end if
end for

Table 2
Datasets for experiment.

	Number of attributes	Number of instances	Number of items	Missing values?	Support	Confidence
Balance-Scale	4	625	23	No	10%	70%
Tic-Tac-Toe	9	958	29	No	5%	50%
Nursery	8	12,960	31	No	2%	50%
Voting	16	435	34	Yes	40%	70%
Lymph	18	148	63	No	35%	60%
Breast-Cancer	9	286	43	Yes	10%	70%
Bank-Marketing	16	45,211	78	Yes	70%	90%

In Table 2, two datasets (*Breast-Cancer* and *Voting*) contain missing values. In order to avoid the impact of missing values on experimental results, we remove the instances with missing values in *Breast-Cancer* and *Voting*. In addition, we preform discretization on the numeric attributes for the Bank-Marketing dataset.

In the experiment, we use 4 agents to demonstrate the knowledge integration process in distributed data mining. Firstly, the training set is divided into 4 parts equally for each agent. Then agents will use Apriori-TFP algorithm [15] to generate association rules on their own datasets, which are used as local knowledge to argue with each other. The confidence and support threshold in Apriori-TFP algorithm are shown in Table 2, which are selected according to the dimensions of different datasets. For the dataset with a higher dimension, we set a relative higher support threshold, and vice versa.

In association rule mining, *item* is a common term to represent the attribute value of instances, so the rules can be described as some items that is associated to another set of items. The number of items represents the number of attribute values the dataset contains. Therefore, the number of items has a direct impact on the computation complexity of association rule mining.

6.2. Convergence analysis

In reinforcement learning, convergence to equilibria is a basic stability requirement [22]. It means the agents' strategies should eventually converge to a coordinated equilibrium [11]. In this section we analyze the convergence of reinforcement learning process of ArgRL. To demonstrate it more clearly, we compared ArgRL with the offline joint learning model AMAJL [47]. Both of them extract meta knowledge from the training data through the argumentation technique and use meta knowledge base to make classification. The difference is that in the application process, ArgRL combines with reinforcement learning technology, and uses the environmental feedback information to optimize meta knowledge base. In addition, ArgRL complements new knowledge by triggering the argumentation according to environmental feedbacks. In contrast, AMAJL, after obtaining meta knowledge base, only uses its knowledge to classify the unknown data. In addition, the meta knowledge base of ArgRL is dynamically changing whereas the meta knowledge base of AMAJL is fixed during testing.

In the following experiment, we evaluate our method with the Tic-Tac-Toe dataset. First, randomly selecting 50% of the original data set as training data for underlying distributed data mining and argumentation to get an initial meta knowledge base. On this basis, the remaining 50% is taken as the test data, and every continuous 15 test cases will be treated as a group to be input into the two models. As it can be seen, each test case eventually gets used exactly once. We record the classification accuracy for each group to compare the convergence process in dynamic data environment, which are shown in Fig. 3. The horizontal axis represents the group number of the test data, and the vertical axis represents the classification accuracy.

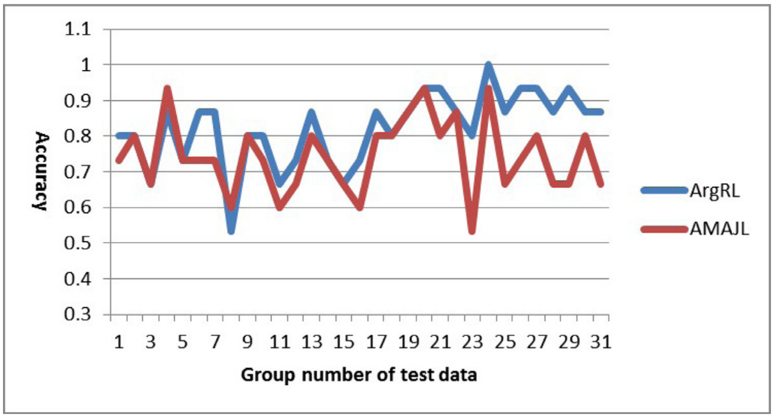

Fig. 3. The convergence process of classification accuracy on *Tic-Tac-Toe* dataset.

As it can be seen from Fig. 3, both ArgRL and AMAJL fluctuate to some extent with the input of the test data, while there is a clear difference between the two models. ArgRL fluctuates significantly in the early stage. With the continuous input of test cases, the fluctuation range of ArgRL tends to decrease and there is a clear upward trend. The classification performance of ArgRL starts to surpass AMAJL completely in the later stage until it converges. In contrast, the classification performance of AMAJL has more significant fluctuations. And as the classification process continues, there is no obvious upward trend. Therefore, although its performance is similar to that of ArgRL at the initial stage, the classification performance at the later stage is significantly weaker than that of ArgRL.

The above process clearly demonstrates the convergence of reinforcement learning in ArgRL. ArgRL has the same meta knowledge base as AMAJL at the beginning of the test. With dynamic input of test cases, reinforcement learning in ArgRL continuously optimizes meta knowledge base according to environmental feedbacks, ensuring its continuous learning ability and improvement of classification performance. With the learning process on large amounts of data, the classification performance of meta knowledge base will eventually reach the convergence state.

6.3. Classification performance

In order to validate the performance of our method, we compare ArgRL with multiple representative baseline methods, including three powerful ensemble learning methods: Bagging [40], AdaBoost [17,19] and RandomForest [9], and five widely used single classifiers: CBA (Classification Based on Associations) [29], SVM [14] (linear, radial), NaiveBayes [27], and CART (Classification Regression Trees) [10]. All these baseline methods are implemented on the data mining software WEKA [25], and Bagging and AdaBoost choose the CBA algorithm as the individual classifier. For better comparison, we use the same confidence and support threshold in Table 2 for baseline methods based on association rule (CBA, Bagging and AdaBoost). The resting baseline methods choose the default parameters in WEKA.

In this experiment, we randomly select 50% of the original data as the training data, the remaining 50% as the test data. As mentioned earlier, in ArgRL, the training data are used to build the initial meta knowledge base. The remaining 50% of the data serve as dynamic data (the test data) during the reinforcement learning process. In contrast, other comparisons simply generate classification models based on the training data and classify the test data. Each classification process is repeated 50 times, and the average performance (accuracy and standard deviation) is reported in Table 3.

Our method uses association rules as the underlying classification knowledge scheme, so we also focus on four rule-based classifiers that use association rules for classification: AMAJL, CBA, Bagging and Adaboost. It can be seen from Table 3 that, our method has shown satisfied performance over all the seven datasets.

Compared to all the rule-based classifiers, ArgRL significantly outperforms AMAJL, CBA, Adaboost and Bagging on *Balance-Scale*, and outperforms AMAJL, CBA and Bagging on *Voting, Lymph* and *Bank-Marketing*, and has similar performance with that of AMAJL, CBA and Bagging on *Nursery*.

Furthermore, on *Tic-Tac-Toe* dataset, our method can achieve significantly higher performance than NavieBayes and SVM-r. On *Breast-Cancer* dataset, our method significantly achieves comparable performance with SVM-l, CART and RandomForest.

As it can be seen, ArgRL shows good classification performance on multiple data sets. This is because that ArgRL can effectively extract high-quality meta knowledge from distributed historical data on the one hand. On the other hand, the reinforcement learning process can dynamically update and optimize meta knowledge and finally get satisfactory classification performance.

Moreover, in the above classification process, we also record the average number of rules in meta-knowledge base after convergence, and the number of association rules mined by all the agents, which are shown in Table 4.

It can be seen from Table 4, compared with the number of association rules obtained by distributed mining, the number of rules in the meta knowledge base finally obtained by ArgRL is much less. For example, in *Nursery* dataset, ArgRL extracts

Table 3
Accuracy of ArgRL and baseline methods.

	ArgRL	AMAJL	CBA	Bagging	Adaboost	NavieBayes	SVM-r	SVM-l	CART	RandomForest
Balance-Scale	0.75 ± 0.015	0.735 ± 0.036	0.679 ± 0.058	0.722 ± 0.043	0.708 ± 0.040	0.887 ± 0.014	0.891 ± 0.013	0.901 ± 0.015	0.774 ± 0.024	0.827 ± 0.023
Tic-Tac-Toe	0.877 ± 0.051	0.797 ± 0.043	1.000 ± 0.002	0.999 ± 0.002	0.989 ± 0.006	0.703 ± 0.017	0.795 ± 0.022	0.983 ± 0.004	0.911 ± 0.024	0.915 ± 0.014
Nursery	0.906 ± 0.008	0.905 ± 0.006	0.906 ± 0.005	0.904 ± 0.005	0.951 ± 0.006	0.903 ± 0.004	0.955 ± 0.002	0.930 ± 0.002	0.988 ± 0.002	0.974 ± 0.003
Voting	0.939 ± 0.02	0.928 ± 0.034	0.896 ± 0.016	0.902 ± 0.017	0.942 ± 0.018	0.901 ± 0.015	0.954 ± 0.009	0.955 ± 0.011	0.955 ± 0.011	0.960 ± 0.011
Lymph	0.707 ± 0.092	0.673 ± 0.117	0.704 ± 0.062	0.674 ± 0.086	0.727 ± 0.074	0.830 ± 0.040	0.806 ± 0.031	0.812 ± 0.034	0.744 ± 0.039	0.805 ± 0.040
Breast-Cancer	0.682 ± 0.031	0.655 ± 0.041	0.710 ± 0.029	0.715 ± 0.020	0.687 ± 0.032	0.722 ± 0.024	0.706 ± 0.008	0.689 ± 0.024	0.697 ± 0.025	0.687 ± 0.032
Bank-Marketing	0.884 ± 0.001	0.876 ± 0.009	0.883 ± 0.002	0.883 ± 0.002	0.883 ± 0.002	0.879 ± 0.001	0.887 ± 0.001	0.888 ± 0.002	0.896 ± 0.113	0.885 ± 0.001

Table 4
Average number of rules in meta-knowledge base.

	ArgRL	AMAJL	Num_of_Association_Rules
Balance-Scale	13.5	13.3	28.72
Tic-Tac-Toe	92.5	91.4	2997.64
Nursery	101.6	99.8	2263.94
Voting	7.5	7	3454.28
Lymph	10.7	9.4	1097.44
Breast-Cancer	36.1	33.1	1241.72
Bank-Marketing	15.3	15.3	244

Table 5
Average running time (millisecond) on test data.

	ArgRL	CBA	Bagging	Adaboost	NavieBayes	SVM-r	SVM-l	CART	RandomForest
Balance-Scale	8833.45	1.96	2.3	0.42	0.51	12.03	5.27	0.27	8.99
Tic-Tac-Toe	4298.4	1.34	9.25	9.12	0.29	28.04	27.24	0.31	8.43
Nursery	482441.4	9.39	70.09	9.79	5.84	1934.38	844.28	5.44	232.05
Voting	2350.35	0.43	3.11	0.71	0.17	5.17	4.12	0.07	3.72
Lymph	8920	0.16	1.32	0.43	0.11	2.45	2.11	0.04	1.01
Breast-Cancer	8879	0.21	1.39	0.18	0.09	3.64	3.34	0.05	2.13
Bank-Marketing	2327609.6	40.197	135.165	137.836	73.320	3837.599	1728.480	13.873	1333.808

101.6 global rules from 2263.94 local association rules to achieve the 90.6% classification accuracy. This shows that ArgRL can effectively extract high-quality knowledge and deal with the application of dynamic data scenarios efficiently.

In addition, we also compare the number of rules between ArgRL and AMAJL. At the beginning of the test, both models have the same knowledge base, and as the classification process continues, ArgRL evaluates the global rules through reinforcement learning and complements new rules with argumentation techniques, so the size of its meta knowledge base is dynamically increasing. Thus as it can be seen from Table 4, ArgRL gets more rules than AMAJL. However, a closer comparison shows that the difference between the two models is not obvious. Rules of ArgRL is only slightly more than AMAJL, but its classification performance is obviously higher, as shown in Table 3. This further illustrates the effectiveness of the reinforcement learning process in ArgRL.

In addition, we also record the running time of ArgRL and baseline methods in the above classification process, and the average results for predicting the test set are shown in Table 5.

It is seen from Table 5 that the efficiency of ArgRL is acceptable, but weaker than that of the baseline methods. The reasons mainly lie in the following aspects.

1. Baseline methods are implemented on weka software, where the related data is stored in memory, and thus, in the classification process they do not need to read the external storage; on the contrary, the data of our program is stored in the database, it requires frequent connections and access to the database during association rule mining and argumentation, and thus, consuming substantial amount of I/O time.
2. Without reinforcement learning mechanism, baseline methods do not learn new knowledge in application process, i.e., they make prediction for the test data directly; on the contrary, our method has a stronger learning ability. In order to handle the unknown environment, argumentation process will be triggered frequently during the prediction process, and thus, taking more time.
3. Baseline methods do not have the knowledge integration process for distributed datasets; our method aims to deal with the distributed knowledge integration problem. Thus, to extract globally consistent knowledge, a large number of interactions between agents need to be conducted during integration, which requires more time.
4. Baseline methods are implemented on weka, which has been made a lot of optimizations, and thus, they run much faster; we use java program language to implement ArgRL by ourselves, so there are a lot of operations (e.g., loop calculation, database connection and multi-table query) to be further optimized.

In future research, instead of using database, we may change the program framework, so that it can manipulate many operations in-memory, e.g., mining association rules, constructing arguments and performing argumentation, to further reduce the time cost.

7. Conclusion

In order to handle the knowledge integration task in distributed data mining, we propose an argumentation based reinforcement learning method, which consists of two learning stages. In argumentation based learning stage, agents use distributed local knowledge as arguments to communicate and negotiate with each other, which achieves distributed knowledge integration and meta-knowledge extraction effectively. In reinforcement learning stage, we design a unique meta-knowledge evaluation and elimination mechanism, which greatly compresses the knowledge scale and obtains a refined meta-knowledge base. The two learning stages alternate and interrelate to realize the spiral process of the meta-knowledge base. Experimental results show that our method has good convergence property, and outperforms the state-of-art baseline methods over multiple public datasets.

The argumentation technology can handle the inconsistent information effectively. However, most argumentation based approaches use rules as arguments. In future research, it is necessary to integrate other knowledge representation schemes into the argumentation framework. Moreover, we are also interested in designing more speech acts to make the fullest use of the local knowledge. In the reinforcement learning process, we may introduce more strategies to make further evaluation of the meta-knowledge, and conduct more efficient knowledge extraction.

Declaration of Competing Interest

We declared that we have no conflicts of interest to this work.

Acknowledgments

This study is supported by the National Natural Science Foundation of China (no. 71371184).

References

[1] L. Amgoud, C. Cayrol, On the acceptability of arguments in preference-based argumentation, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, in: UAI'98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, pp. 1–7.
[2] L. Amgoud, H. Prade, Using arguments for making and explaining decisions, Artif. Intell. 173 (3–4) (2009) 413–436.
[3] L. Amgoud, M. Serrurier, Arguing and explaining classifications, in: Argumentation in Multi-Agent Systems, International Workshop, Honolulu, Hi, Usa, 2007, pp. 164–177.
[4] L. Amgoud, M. Serrurier, Agents that argue and explain classifications, Autonom. Agents Multi-Agent Syst. 16 (2) (2008) 187–209.
[5] L.M. Aouad, N.A. Le-Khac, T. Kechadi, Variance-based clustering technique for distributed data mining applications (2017) 111–117.
[6] K. Bache, M. Lichman, Uci Machine Learning Repository, 2013.
[7] T. Benchcapon, Value based argumentation frameworks, Artif. Intell. 171 (1) (2002) 444–453.
[8] I. Bratko, M. Mozina, J. Zabkar, Argument-based machine learning, in: Foundations of Intelligent Systems, 16th International Symposium, Bari, Italy,Proceedings, 2006, pp. 11–17.
[9] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[10] L.I. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and regression trees (cart), Biometrics 40 (3) (1984) 17–23.
[11] L. Busoniu, R. Babuska, B. De Schutter, A comprehensive survey of multiagent reinforcement learning, IEEE Trans. Syst. Man Cybern.Part C 38 (2) (2008) 156–172.
[12] D.V. Carbogim, D. Robertson, J. Lee, Argument-based applications to knowledge engineering, Knowl. Eng. Rev. 15 (2) (2000).
[13] Á. Carrera, C.A. Iglesias, A systematic review of argumentation techniques for multi-agent systems research, Artif. Intell. Rev. 44 (4) (2015) 509–535.
[14] Chang, ChihChung, Lin, ChihJen, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 389–396.
[15] F. Coenen, P. Leng, S. Ahmed, Data structure for association rule mining: T-trees and p-trees, IEEE Trans. Knowl. Data Eng. 16 (6) (2004) 774–778.
[16] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games, Artif. Intell. 77 (2) (1995) 321–357.
[17] Y. Freund, Experiments with a new boosting algorithm, in: Thirteenth International Conference on Machine Learning, 1996, pp. 148–156.
[18] Y. Freund, Boosting a weak learning algorithm by majority, Inf. Comput. 121 (2) (1997) 256–285.
[19] Y. Freund, R.E. Schapire, A Desicion-theoretic Generalization of On-line Learning and an Application to Boosting, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 23–37.
[20] Y. Fu, Distributed data mining: an overview, Newslett. IEEE Tech. CommitteeDistrib. Process. (2001) 866–883.
[21] K. Georgila, D.R. Traum, Reinforcement learning of argumentation dialogue policies in negotiation, in: The 12th Annual Conference of the International Speech Communication Association, Florence, Italy, 2011, pp. 2073–2076.
[22] A. Greenwald, K. Hall, Correlated q-learning, in: Machine Learning, Proceedings of the Twentieth International Conference, Washington, DC, USA, 2003, pp. 242–249.
[23] V. Groznik, M. Guid, A. Sadikov, M. Mozina, D. Georgiev, V. Kragelj, S. Ribaric, Z. Pirtoek, I. Bratko, Elicitation of neurological knowledge with argument-based machine learning., Artif. Intell. Med. 57 (2) (2013) 133–144.
[24] I. Habernal, I. Gurevych, Argumentation Mining in User-generated Web Discourse, MIT Press, 2017.
[25] M.A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explor. 11 (1) (2009) 10–18.
[26] H. He, S. Chen, K. Li, X. Xu, Incremental learning from stream data., IEEE Trans. Neural Netw. 22 (12) (2011) 1901–1914.
[27] G.H. John, P. Langley, Estimating continuous distributions in bayesian classifiers, in: UAI '95: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, Montreal, Quebec, Canada, 1995, pp. 338–345.
[28] A. Kakas, P. Moraitis, Argumentation based decision making for autonomous agents, in: International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2003, Melbourne, Victoria, Australia, Proceedings, 2003, pp. 883–890.
[29] B. Liu, W. Hsu, Y. Ma, Integrating classification and association rule mining, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York City, New York, USA, 1998, pp. 80–86.
[30] T.M. Mitchell, J.G. Carbonell, R.S. Michalski, Machine Learning, McGraw-Hill, 2003.
[31] C. Moemeng, V. Gorodetsky, Z. Zuo, Y. Yang, C. Zhang, Agent-Based Distributed Data Mining: A Survey, Springer US, Boston, MA, pp. 47–58.
[32] A. Monteserin, A. Amandi, A reinforcement learning approach to improve the argument selection effectiveness in argumentation-based negotiation, Expert Syst. Appl. 40 (6) (2013) 2182–2188.
[33] M. Mozina, J. Zabkar, I. Bratko, Argument based machine learning, Artif. Intell. 171 (2007) 922–937.
[34] K. Napierala, J. Stefanowski, Addressing imbalanced data with argument based rule learning, Expert Syst. Appl. 42 (24) (2015) 9468–9481.
[35] I. Nonaka, H. Takeuchi, The knowledge-creating company: how japanese companies create the dynamics of innovation, J. Int. Bus. Stud. 29 (4) (1996) 5651–5661.
[36] S. Ontañón, E. Plaza, Coordinated inductive learning using argumentation-based communication, Autonom. Agents Multi-Agent Syst. 29 (2) (2015) 266–304.
[37] A.C. Paula, E. Scalabrin, Using distributed data mining and distributed artificial intelligence for knowledge integration, in: International Workshop on Cooperative Information Agents, 2007, pp. 89–103.
[38] V.S. Rao, S.V. Ramaswamy, G., Distributed data mining and agent mining interaction and integration: a novel approache, Int. J. Res. Rev. Appl. Sci. 4 (4) (2010).

[39] J.D. Silva, C. Giannella, Distributed data mining and agents, Eng. Appl. Artif. Intell. 18 (7) (2005) 791–807.
[40] M. Skurichina, R.P.W. Duin, Bagging for linear classifiers, Pattern Recognit. 31 (7) (1998) 909–930.
[41] R.S. Sutton, A.G. Barto, Reinforcement learning: an introduction, IEEE Trans. Neural Netw. 9 (5) (1998) 1054.
[42] R. Vilalta, Y. Drissi, A perspective view and survey of meta-learning, Artif. Intell. Rev. 18 (2) (2002) 77–95.
[43] M. Wardeh, T. Bench-Capon, F. Coenen, Padua: a protocol for argumentation dialogue using association rules, Artif. Intell. Law 17 (3) (2009) 183–215.
[44] M. Wardeh, T. BenchCapon, F. Coenen, Arguing from experience using multiple groups of agents, Argument Comput. volume 2 (2) (2011) 51–76.
[45] M. Wardeh, F. Coenen, T.B. Capon, Pisa: a framework for multiagent classification using argumentation, Data Knowl. Eng. 75 (2) (2012) 34–57.
[46] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (2) (1992) 241–259.
[47] J. Xu, L. Yao, L. Li, Argumentation based joint learning: a novel ensemble learning approach, Plos One 10 (5) (2015) e0127281.
[48] L. Zeng, L. Li, L. Duan, K. Lu, Z. Shi, M. Wang, W. Wu, P. Luo, Distributed data mining: a survey, Inf. Technol. Manag. 13 (4) (2012) 403–409.