

# Multi-Party Argument from Experience

Maya Wardeh, Trevor Bench-Capon, Frans Coenen

Department of Computer Science, The University of Liverpool, Liverpool, UK

**Abstract.** A framework, PISA, for conducting dialogues to resolve disputes concerning the correct categorisation of particular cases, is described. Unlike previous systems to conduct such dialogues, which have typically involved only two agents, PISA allows any number of agents to take part, facilitating discussion of cases which permit many possible categorizations. A particular feature of the framework is that the agents argue directly from individual repositories of experiences rather than from a previously engineered knowledge base, as is the usual case, and so the knowledge engineering bottleneck is avoided. Argument from experience is enabled by real time data-mining conducted by individual agents to find reasons to support their viewpoints, and critique the arguments of other parties. Multiparty dialogues raise a number of significant issues, necessitating appropriate design choices. The paper describes how these issues were resolved in the protocol and implemented in PISA, and illustrates the operation of PISA using an example based of a dataset relating to nursery provision. Finally some experiments comparing PISA with other classifiers are reported. **keywords** **Argumentation, Data mining, Dialogue Games.**

## 1 Introduction

In [19] PADUA (Persuasive Argumentation Using Association Rules), a protocol to enable two agents to argue about the classification of a case was presented. The distinguishing feature of PADUA was that the arguments used by the agents were derived directly from a database of previous examples using data mining techniques, with each participant drawing on a separate database of examples. The authors termed this *arguing from experience*, to contrast this style of argument with the typical form of persuasion dialogues, in which each agent uses a separate knowledge base in which domain expertise is represented as a set of rules [14]. Strategies for PADUA dialogues were discussed in [17] and PADUA was evaluated on a substantial dataset relating to welfare benefits in [18]. In this paper we present PISA (Pooling Information from Several Agents) which extends PADUA to allow any number of software agents to engage in the dialogue. This is particularly useful when the classification is not binary, since each possible classification can then have its own champion.

To date research into persuasive argumentation dialogues have largely been confined to scenarios with two agents: very few examples of dialogue with several agents can be found in the literature. The main issues regarding multi-party

dialogues in general were identified in [8] and [16]. In this paper we will address the issues identified in [8] and [16] and resolve them in ways appropriate to the style of dialogue supported by PISA.

The rest of this paper is organized as follows: in Section 2 details of the notion of multiparty argument from experience are considered. The PISA framework is introduced in section 3 and solutions to issues raised in Section 2 presented. In Section 4 we describe a series of experiments which we have performed to explore the operation of PISA. Finally Section 5 concludes the paper.

## 2 Multi Party Argument from Experience

### 2.1 Arguing From Experience

The aim of PISA is to allow a number of agents, each with their own private database of examples, to debate the correct classification of a new case. The classification is not intended to be binary: the examples can be classified in one of a number of ways, and each possible classification will be championed by one of the parties to the dialogue. Because the agents will use their database of examples directly, rather than drawing on a set of rules engineered in some previous analysis stage, we term this *arguing from experience*. PISA is based on previous work [19] which established a set of basic speech acts for argument from experience dialogues between two parties. These speech acts are inspired by work on case based reasoning in law, which forms arguments on the basis of precedent cases, especially the work carried out by Ashley [3] and Aleven [2]. A key difference, however, between arguing from experience and case based reasoning is that all of an individual agent's experiences (represented by a dataset of previous examples) are used collectively, rather than a single case being identified as a precedent. Unlike legal decisions authority comes from the frequency of occurrence in the set of examples rather than endorsement of a particular decision by an appropriate court.

When arguing from experience, rather than drawing rules from a knowledge base, the agents use data mining techniques to discover associations between features of the case under consideration and the appropriate classification according to their previous experience. This approach features several advantages:

1. Such arguments are often found in practice: many people do not develop a theory from their experience, but when confronted with a new problem recall past examples;
2. It avoids the knowledge engineering bottleneck that occurs when belief bases must be constructed;
3. There is no need to commit to a theory in advance of the discussion: the information can be deployed as best meets the need of the current situation;
4. It allows agents to share experiences that may differ: one agent may have encountered types of case that another has not. This is why it important that each agent uses its own database.

Using their distinct databases PISA agents produce reasons for and against classifications by mining Association Rules (ARs) from their datasets using Association Rule Mining (ARM) techniques [1], [6]. ARs [1] are probabilistic relationships expressed as rules of the form  $X \rightarrow Y$  read as *if X is true then Y is likely to be true*, or *X is a reason to think Y is true* where X and Y are disjoint subsets of some global set of attributes. Likelihood is usually represented in terms of a *confidence value* expressed as a percentage. This is calculated as  $support(XY) \times 100 / support(X)$  where the support of an itemset  $I$  is the number of records in the data set in the itemset  $I$  occurs. To limit the number of rules generated only itemsets whose support is above a user specified *support threshold*, referred to as *frequent itemsets*, are used to generate associations. To further limit the number of associations only those rules whose confidence exceeds a user specified confidence threshold are accepted. The process of identifying appropriate ARs is called Association Rule Mining (ARM).

In the context of this paper the antecedent of an AR represents a set of reasons for believing the example should be classified as expressed in the consequent. There are six speech acts (moves) used in PADUA, which can be seen as falling into three categories as follows:

1. **Propose Rule.** Move that allows generalizations of experience to be cited, by which a new rule with a confidence higher than a certain threshold is proposed.
2. **Attacking moves.** These moves argue that the reasons given in a proposed rule are not decisive in this case. This can be achieved using one of three speech acts as follows:
  - **Distinguish.** When a player plays a distinguish move, it adds some new premise(s) to a previously proposed rule, so that the confidence of the new rule is lower than the confidence of the original rule.
  - **Counter Rule.** This is similar to the “propose rule” move, but is used to cite a generalization leading to a different classification.
  - **Unwanted Consequences.** Using this move a player suggests that certain consequences (conclusions) of the proposed rule do not match the case under consideration.
3. **Refining moves.** These moves enable a rule to be refined to meet objections. This can be achieved using one of two speech acts:
  - **Increase Confidence.** Here a player adds one or more premises to a rule that has been previously been proposed to increase the confidence of the rule.
  - **Withdraw unwanted consequences.** Here a player excludes the unwanted consequences of a rule that has been previously proposed, while maintaining a certain level of confidence.

For each of the above six moves a set of legal next moves (i.e. moves that can possibly follow each move) is defined. Table 1 summarizes the rules for “next moves”, and indicates where a new set of reasons is introduced to the discussion.

While the above model can be applied to a number of application areas such as legal argumentation and other classification problems [18], where agents can

**Table 1. Speech acts (moves) in PISA**

Move	Label	Next Move	New Rule
1	Propose Rule	3,2,4	Yes
2	Distinguish	3,5,1	No
3	Unwanted Cons	6,1	No
4	Counter Rule	3,2,1	Yes
5	Increase Conf	3,2,4	Yes
6	Withdraw Unwanted	3,2,4	Yes

benefit from differences in experience to solve a case, PADUA is restricted to two players only. The work presented here significantly extends this model to provide an environment in which three or more participants can argue with the others using the set of speech acts listed above, drawing their arguments from individual databases. Since extension from two players is not straightforward, before describing this multi-player framework in detail the main issues to be addressed in multiparty dialogues in general and multiparty argument from experience in particular, will be reviewed.

## 2.2 Issues in Multiparty argument from experience

The main concern facing multiparty argument from experience is how to allow for an indefinite number of agents to engage in the debate without jeopardizing the basic argument from experience protocol proposed in [19]. There are a number of issues of relevance in any multiparty dialogue. The following summary of these issues is based on the issues identified in [8] and [16]:

1. *System openness*: Multiparty argument from experience dialogues can either be closed or open. A closed dialogue starts with  $N$  players and continues with the same  $N$  players until the end; no new participants are allowed to join the dialogue once it has started, and players cannot leave the dialogue while it is in progress. Open dialogues are the opposite: players are free to join or leave at any time.
2. *Roles*: In two-party argument from experience there are only two roles, the proponent and the opponent, but in multiparty dialogues the situation is more complicated. There may be several proponents and several opponents of a thesis. Alternatively there may be several participants each with their own distinct options. Also, some parties within the dialogue can take a neutral point of view, or stand as mediator between the opposing parties. Also, linguistically speaking, in two player dialogues one (and only one) player

can speak per turn (the *speaker*) while the other listens (the *listener* or *hearer*). In multiparty dialogues there can be more than one hearer per turn. Moreover, one can argue that there can be more than one speaker per turn, since in real life people may start talking at the same time, interrupt each other or compete with each other for attention (the loudest wins).

3. *Addressing*: Multiparty argument from experience should implement a clear addressing policy which can take one of two forms: either public broadcasting where all the players listen to what the speaker is saying; or targeted broadcasting of the speech act to some players (but not all of them).
4. *Turn taking*: The turn taking policies of two-party dialogues is straightforward: the listener becomes the speaker when the current speaker finishes. This is not the case in multiparty dialogues, and hence such a simple policy cannot be applied to multiparty argument. In persuasion dialogues the decision whether all players are given permission to talk when they want, or they have to wait for their designated turns, can greatly influence the final outcome of the dialogue.
5. *Termination*: In [19] the argument from experience dialogue terminated once one party could not argue any further (has nothing more to say). This simple rule is not sufficient for multiparty dialogues where the dialogue may be terminated either when all the other players are convinced or once the majority of them are. Another issue regarding termination is that sometimes players may fail to convince each other and could end up playing for ever, if there were no mechanism to end such dialogues. Finally in some scenarios the game may end without one single player dominating the others: in these cases there should also be a mechanism to determine the winner of the game or simply to allow ties to take place.

The above are issues that must be addressed in any system for multiparty dialogue. There are no right or wrong answers: the questions must be resolved appropriately for the particular context. We will next describe how these issues are resolved in the PISA system.

### 3 PISA

This section introduces a proposed model for multiparty argument from experience, PISA (Pooling Information from Several Agents), a protocol designed to enable these multiagent dialogues. The scenarios envisaged by the PISA model are those where there are several options for a particular point of view (a classification), and each one of these options is advocated by one or more software agents. Additionally in PISA there is one software agent, the chairperson, which does not advocate any position, but rather manages the dialogue and facilitates communication between the advocates.

This style of dialogue thus determines the *roles* of the players: a chairperson, and, for every option, at least one player acting as its advocate. If more than one player shares the same position, then these players join forces and form a group

(see sub-section 3.6). Each participant or group is in principle the defender of its own thesis, and an opponent of the rest of the players or groups; although PISA does allow players to temporarily defend another player’s thesis where appropriate for strategic reasons.

PISA dialogues are *open* dialogues, in the sense that participants (other than the chair) may enter or leave when they wish. For *turn taking*, a structure with rounds is adopted, rather than a linear structure where agents are selected as the next speaker in turn. In each round, any agent who can make a move can do so: the chair then updates a central argument structure, and another round occurs. The central argument structure acts as a coordination artifact as proposed by Oliva et al [13]. In any round there are a number of *speakers* (players that participate in that round) and a number of *addressees* (the players whose positions are under attack). The remaining players (i.e. those who did not participate and were not attacked in the given round) need to be aware of the developments in the dialogue and are therefore assumed to be passive *listeners*.

There is no limitation on the number of groups (players) that can participate in any round; but, in order to simplify the game, each group (player) is limited to one move per round. This turn taking policy gives the participants a rich context to explore strategy issues in the best way possible. It also simplifies the game, allowing players to skip rounds, i.e. remain silent, if they think that making a move at this stage does not provide them with an advantage. Finally, this structure allows agents to play their attacks and counter attacks as soon as they wish and they do not have to wait for their turn to contribute.

The above is not perhaps the most usual structure for human meetings, but it can be found, for example, in some board games. We suggest that the structure is particularly appropriate for the PISA context in order to achieve fairness, particularly since every advocate is playing for themselves, and has to regard every other advocate as an opponent (even though they may temporarily focus their efforts on a particular opponent, or effectively form temporary coalitions against a particular opponent). For *addressing*, every move after the first move attacks a move of some other agent and so, as noted above, that agent can be regarded as the “addressee” of that move, and the others as “listeners”. The game will *terminate* when no agent makes a contribution for two rounds or after some limiting number of rounds have been played. Two silent rounds are necessary to ensure that the agents have really finished and not silent for tactical reasons. The termination of the game is thus guaranteed (see sub-section 3.2 for further discussion). The model is essentially that of a facilitated discussion, with the chairperson acting as facilitator. In the following subsections the realization of this model is described in more detail.

### 3.1 Control Structure

The suggested control structure for PISA consists of a dialogue facilitator, the chairperson, who is responsible for monitoring and guiding an indefinite number of groups of players, where each group comprises all of the players who have joined the game and who share the same objective. The chairperson is also

responsible for the central argument structure used to coordinate the dialogue: namely the argumentation tree (see sub-section 3.3 for further detail).

When a new game commences the chairperson randomly orders the groups participating in the game, the first player proposes a new rule and the chairperson adds it to the argumentation tree. This is called the first argumentation round. In each of the subsequent rounds all the players who can and wish to attack any of the arguments played in the previous rounds are allowed to offer their arguments and the chairperson adds them to the argumentation tree. The suggested facilitated discussion protocol using the argumentation tree as a mediating artifact enjoys the following advantages:

- It increases the flexibility of the overall PISA system. By assigning the majority of protocol surveillance to the chairperson the system gains great flexibility. For example the system could be switched from closed to open (and vice versa) by applying a few limited changes to the chairperson behaviour, while the rest of the players remain unaffected.
- It is a very simple structure: there is no complicated turn taking procedure involving a choice of the next player, allowing the internal implementation of the player agents to be kept as simple as possible.
- It provides a fair dialogue environment: the organizational configuration of the dialogue is neutralized by restricting the control tasks to the chairperson who is not allowed to take sides in the dialogue. This means that no one player is privileged to speak while the others must remain silent.

### 3.2 Game Termination

The chairperson terminates a PISA game when two rounds have passed and the argument state has not changed. The reason behind waiting two rounds is that sometimes some players may choose to skip some rounds for strategic reasons. Therefore, if a round is passed without any move being played, the chairperson has to warn all the players that if they have anything to say the next round is their last chance to do so if they wish to prevent the game from ending.

Termination, as described above, is called *legal termination*; however there are also cases in which the game should be exceptionally terminated (i.e. *exceptional termination*). In this case the chairperson has the authority to terminate the game if only one player remains active after all the other players have withdrawn (in which case the surviving player wins the game). Also, if the game has taken more than  $N$  rounds, the chairperson will end the game; the assumption is that if the parties can not agree in  $N$  rounds, where  $N$  is sufficiently large, then they will never agree. In this case no one wins the game. Some further discussion relating to termination, regarding how the winner is determined, is given in sub-section 3.6.

### 3.3 Argumentation Tree

The notion of an *Argumentation Tree* is used to describe the central data structure representing the game's arguments, and the attack relations between those

arguments. This tree acts as a mediating artifact for the dialogue as described in [13]. The tree structure used in PISA differs from other argumentation structures used in the literature as it consists of arguments played by several players. It uses four colours to mark the status of the arguments played so far, and two types of links: *explicit links* (edges) representing direct attacks, and *implicit links* representing indirect attacks. The issue of addressing is solved via the direct links: speech acts are addressed to the participants who played the argument attacked by those acts, except for the first move in the game which is addressed to all the other players. This type of addressing is a public broadcast as all the participants can “listen” to what “speakers” are saying by consulting the tree.

The PISA Argumentation Tree data structure consists of *Nodes*, *Links* and the *Green Confidence* value. Nodes represent the speech acts made in the game. Each node has a colour (green, blue, red or purple) representing the status of this node (and hence the argument represented by it) in the current round of the game. Nodes are either green or blue when introduced, depending on whether they propose a new rule or only attempt to undermine an existing one. Red nodes are those directly under attack and purple nodes are those indirectly under attack. Nodes change their colour as defined in Table 2. Links represent the explicit attack relationships between nodes. Finally the Green Confidence value is a global value associated with the tree and is the confidence value of the undefeated green node with highest confidence.

**Table 2. The Argumentation Tree Colours**

Colour New Rule	Meaning	Changes To
Green	A (1, 4, 5 or 6) move node, undefeated in the given round.	(To red) If attacked by at least one undefeated node. (To purple) if indirectly attacked by an undefeated green node with higher confidence.
Red	The node is defeated in the given round.	(To green) if all attacks against the node are successfully defeated and the original node colour was green. (To blue) if all attacks against the node are successfully defeated and the original node colour was blue.
Blue	A (2 or 3) move node undefeated in the given round	(To red) If attacked by at least one undefeated node.
Purple	A (1, 4, 5 or 6) move node indirectly attacked by a higher confidence green node, played by a different player (group).	(To green) if all attacks against the node are successfully defeated, and if the move indirectly attacking this node was defeated. (To red) if successfully attacked by at least one undefeated node.



When a player plays some move ( $m$ ), the move must satisfy a number of conditions in order to be added as a node to the argumentation tree, otherwise the move will be rejected. The conditions of acceptance are as follows:

- Move  $m$  is added to the tree only if it changes the colouring of the tree (consequently a player cannot add an argument that enhances its position in the game if such a move does not change the tree colouring).
- A player can put forward one move only per round (deciding which rule to play is a strategy issue).
- Moves (1, 4, 5, 6) implicitly attack all the other (1, 4, 5, 6) moves played by other players which have lower confidence.
- Moves (2, 6) affect only the nodes they directly attack.
- Moves (1, 4, 5, 6) explicitly attack the (2, 6) nodes they are associated with (if any).
- Participants should not play moves that weaken their position (i.e. a player should not put forward moves that change the colouring of the argumentation tree such that another player would take the lead). This condition holds when a player tries to attack blue nodes that were originally made to attack an argument proposed by other players.

### 3.4 Winner Announcement/Tie Break

Once a game has terminated, the chairperson consults the argumentation tree to determine the winner. The winner should satisfy one of the following conditions:

1. All the green nodes belong to the winner.
2. If there are no green nodes, and all the blue nodes were played by the same player, that player wins.

There are two cases where a game will end with no clear winner. The first is where there is more than one green node with the same confidence on the argumentation tree at the end of the game belonging to different participants and no green nodes with higher confidence. The second is where the game ends without any green nodes on the argumentation tree but with blue nodes belonging to more than one player.

The first case is considered a *strong tie* situation, as the players have actually proposed classifications within the game. One possible solution may be to start a new game between the tying parties only and see how this game ends. But there is nothing to guarantee that this game will not also end up in a tie. In this case the chairperson may be forced to announce a tie (after the second game or after a suitable number of games between the tying parties).

The second case is considered a *weak tie* situation, as the tied players did not actually have any proposed classifications at the end of the game, but has simply prevented anyone from establishing a satisfactory classification. In such cases starting a game between the tied players may be of benefit, but with the requirement that the players should propose as many reasons for their classification as they can.

### 3.5 Players' Strategies

In PISA players' strategies can be a key factor in determining the winner of the dialogues, as the more sophisticated the strategy used the greater the agent's chances of winning against their opponents. PISA strategies are categorized according to the understanding of the status of the argumentation tree involved. The status of the argumentation tree is the state of the tree at the start of each round. Players can either take this status into consideration or not. Those who pay attention to this status can generally be considered "smarter" than those who do not, as they are more responsive to the changes in the game. PISA strategies fall into three main *strategy types* (types 1, 2 and 3) which are derived from the basic fact that players have the freedom to choose when and how frequently they are going to attack their opponents. Each is discussed below.

**Type1: Attack whenever possible:** Players attack whenever they can do so with a legal move. Players may adopt this strategy to enhance their chances of winning the game by being as aggressive as possible. This type of strategy is divided into three sub types:

1. Attack whenever possible by proposing new rules.
2. Attack whenever possible by undermining the opponent.
3. Tree dependent attack whenever possible: players using this strategy try to read the tree and anticipate the best possible attack. This strategy combines the previous two giving the players the freedom to use either of them depending on the position of other players in the game. For example if proposing a counter rule would not yield the best undefeated rule, the player will first attempt to reduce the confidence of the current best rule.

**Type2: Attack only when needed:** Players attack their opponent only when needed, typically when all their proposed rules so far have been successfully attacked by other players, or when their attempts to undermine all the other players have failed (because the other players successfully defended their positions). In other words the strategy is for the player to wait as long as its current proposal is secure (not under attacked). Also by waiting and monitoring other players' moves; players get a better chance to predict their opponents' moves and plan their actions accordingly. This type of strategy is also divided into three sub types:

1. Attack only when needed by proposing rules: players using this strategy will only attack when they no longer have any green nodes on the tree, by proposing a new rule (or any other equivalent move). If they have a green node these players will "wait and see".
2. Attack only when needed by undermining the opponent: These players do not move if they have a green node, or all the blue nodes belong to them.
3. Tree dependent attack only when needed: Players following this strategy try to use the tree to determine their best next move. This strategy combines the previous two giving the players the freedom to use either of them depending on the position of other players in the game.

**Type3: Attack to prevent forecasted threat:** Using this type of strategy a player may be able to anticipate the upcoming attacks against its proposals and therefore either change these proposals (i.e. the moves to be played) or attack the other players before they have the chance to attack its positions.

Where attack takes the form of proposing new rules we say that the player is in *build* mode, whereas if the attack uses undermining we say the player is in *destroy* mode.

### 3.6 Groups

In PISA players advocating the same thesis are required to act as a *single group of players*. Every group is allowed one and only one argument per round. This restriction aims to simplify PISA dialogues. The PISA notion of groups prevents participants sharing the same objective from arguing without consulting each other and consequently causing contradictions amongst themselves or attacking each other. This may, however, lead to a situation in which the weaker parties (within the groups) are forced to withdraw from the game and the remaining stronger members no longer have sufficient shared experience to win.

Each group has a leader, the player applying the best strategy, where strategies are ranked according to their level of understanding of the argumentation tree, and the dialogue process. The leader guides the inter-group dialogue, and decides which one of the moves suggested by all the members is the best to be played next in the game. In this the inter-group dialogue is a variation of targeted broadcasting, in which only group members can listen to what they are saying, while other players are completely unaware of these conversations. The leader can also redirect other members' moves against different opponents, or advise them to follow its own strategy, an act that makes the group benefit from both the smartness of the leader and the experience of other more straightforward players.

## 4 Example

PISA has been fully implemented in Java, and uses the tool described in [5] to mine its association rules (ARs). We have applied PISA to several datasets including the welfare benefits application used to evaluate PISA [18] and a dataset drawn from an application to process applications for a nursery school in Ljubljana [12]. In this section we will illustrate experimentally the kinds of dialogues produced by PISA, using the nursery dataset from the UCL data repository [17].<sup>1</sup>

The Nursery data set was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It was used during the 1980s when there was excessive enrollment to these schools in Ljubljana,

---

<sup>1</sup> UCI machine learning repository: <http://archive.ics.uci.edu/ml/datasets/Nursery>

Slovenia, where there were often two applications for every place. The final decision depended on eight factors forming three sub problems: the occupation of parents and the child’s current nursery provision; the family structure and its financial standing; and the social and health picture of the family. The model was developed using the DECMAK expert system shell for decision making [4].

The original data set consisted of 12960 records (past cases) classified into five levels of priority: (i) not recommended, (ii) recommended, (iii) highly recommended, (iv) priority and (v) special priority. The distribution of the classes in terms of records was as follows: 4320, 2, 328, 4266 and 4044 (33.33%, 0.015%, 2.53%, 32.91% and 31.204%). Note that the highly recommended classification is also rather rare: it is this kind of rarely encountered case which is often misclassified. For the purpose of this experiment, and because of the small number of examples in the class, the two records for the recommended class were removed from the dataset so that four player games could be designed using this revised data set. The dataset has also 8 attributes other than the class attribute, which take the following values:<sup>2</sup>

- Parents occupation: usual, pretentious, of great pretension
- Childs nursery: proper, less proper, improper, critical, very critical
- Form of the family: complete, completed, incomplete, foster
- Number of children: 1,2,3, more than 3
- Housing conditions: convenient, less convenient, critical
- Financial standing of the family: convenient, inconvenient
- Social conditions: non-problematic, slightly problematic, problematic
- Health conditions: recommended, priority, not recommended

#### 4.1 A PISA Debate

For our example we will use a run of PISA with four agents, each representing one of the four classifications. We will refer to these agents as NR (not recommended), HR (highly recommended), PR (priority) and SP (special priority). We will use an example which should be classified as highly recommended, since this is the rarest, and hence the classification most likely to be in dispute. Specifically the case has attributes: the parents have a usual occupation, has less than proper nursery, completed family, two children, convenient housing, inconvenient finance, non problematic social conditions and recommended health conditions recommended. For our run we set the confidence level to 50% and the support level to 1%: both of these are deliberately low so as to get as detailed a dialogue as possible. These values are also well established as the default thresholds in the datamining community.

HR is invited to propose a rule and suggests the following association: *usual occupation, less than proper nursery, convenient housing and recommended health* → *highly recommended* with a confidence of 52.38%. This rule is attacked by the other three agents in round two:

---

<sup>2</sup> The terminology for the attributes is that of [12]: the translation is sometimes quaint.

- NR: proposes a counter rule: usual occupation, complete family, 2 children, convenient housing and inconvenient finance → with confidence 55.55%.
- PR: also proposes a counter rule: recommended health → priority with confidence 55.72%.
- SP: distinguishes the original rule since where the family is complete the confidence of the proposed rule falls to only 20%.

Note that SP does not propose a rule of its own. Since the case falls into the narrow band of highly recommended we might expect to find reasons for the classifications on either side, but not the very different special priority. None the less, SP can play a useful role in critiquing the arguments of the other players.

At this stage PR is ahead as it has the best unattacked rule. (The reader might find it helpful to refer to the completed argument tree shown in Figure 1 as the debate develops.) In round three all four players move:

- HR: proposes a counter rule to attack the current best rule. In fact it has an excellent rule: usual parent, less than proper nursery, complete family, convenient housing and recommended health → highly recommended with confidence 85.71%.
- NR: distinguishes PR's argument in the previous round by pointing out that usual occupation and recommended health only gives priority with a confidence of 18.64%.
- PR: proposes a counter rule against NR's rule of round two: usual occupation and less than proper nursery and recommended health → priority with confidence 61.16%.
- SP: also distinguishes PR's second round rule by pointing to the usual occupation, but from their database the modified rule has confidence of 19.94%.

Now HR is back in the lead. Note that in fact the proposed rule is the same as the rule as modified by SP in round 2. This difference in confidence is explained by the fact that SP may have very few highly recommended cases in its database. In round four SP has no move. The other two agents can, however, make moves:

- NR: distinguishes PR's rule from round three, by pointing out that recommended health reduces the confidence of the priority classification to only 20%.
- PR: proposes a counter rule against HR's rule of round three: less than proper nursery, completed form, inconvenient finance and recommended health → priority with confidence 86.95%.

Now PR is winning, but in round five this can be distinguished by NR, since the addition of non-problematic social problems reduces the confidence to just 20%. In round six PR proposes another rule: *usual occupation and less than proper nursery and recommended health → priority* with confidence 65.95%. This, however, can be distinguished by HR since adding non-problematic social behaviour again reduces the confidence to 20%. This reinstates the argument of HR made in round 3. No more arguments are possible at this stage, and so the final classification is *highly recommended*.

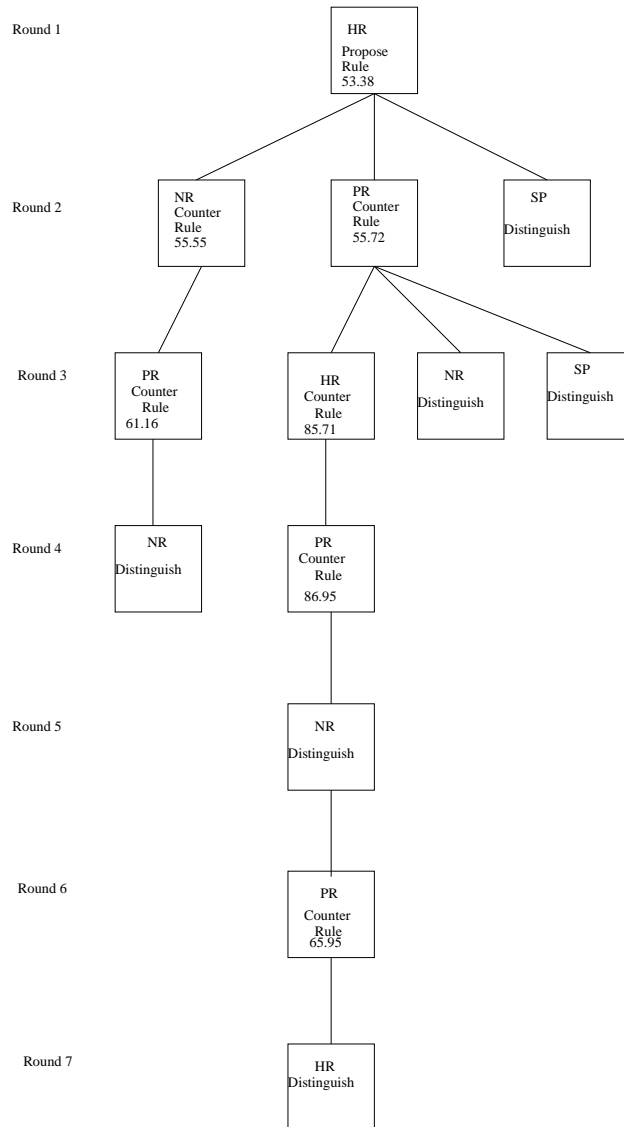


Figure 1. Completed Argument Tree

## 4.2 Evaluation

To evaluate the effectiveness of PISA as a classifier, PISA was evaluated against a variety of standard classifiers using a number of datasets varying in size and structure. A series of cross-validation tests were performed. Here we first present the results from the experiments with the nursery data used in the example debate.

Five other classifiers were used, operating on the union of the data sets:

- **TFPC**: TFPC, Total From Partial Classification [6], is a Classification Association Rule Mining (CARM) algorithm founded on the TFP (Total From Partial) Association Rule Mining (ARM) algorithm; which, in turn, is an extension of the Apriori-T (Apriori Total) ARM algorithm [1]. TFPC is designed to produce Classification Association Rules (CARs) whereas Apriori-T and TFP are designed to generate Association Rules (ARs). In its simplest form TFPC determines a classification according to given support and confidence thresholds. The nature of the selected thresholds is therefore the most significant influencing factors on classification accuracy. A more sophisticated version of TFPC uses a hill climbing technique to find a best accuracy given start support and confidence thresholds.
- **CBA**: CBA (Classification Based on Associations) is a Classification Association Rule Mining (CARM) algorithm developed by Bing Liu et al [10]. CBA operates using a two stage approach to generating a classifier:
  1. Generating a complete set of CARs.
  2. Prune the set of CARs to produce a classifier.
- **CMAR**: CMAR (Classification based on Multiple Association Rules) is another CARM algorithm developed by Wenmin Li et al [9]. CMAR also operates using a two stage approach to generating a classifier:
  1. Generating the complete set of CARs according to a user supplied support threshold to determine frequent (large) item sets, and confidence threshold to confirm CRs.
  2. Prune this set to produce a classifier.
- **Decision Trees**: Classification using decision trees was one of the earliest forms of data mining. Ross Quinlan’s C4.5 is arguably the most referenced decision tree algorithm [15]. One of the most significant issues in decision tree generation is deciding on which attribute to split. Various algorithms have been proposed in the literature. Two are used here: Most frequently supported (or Random) Decision Trees (**RDT**) and Information Gain Decision Trees (**IGDT**). The first selects the first attribute in a list of attributes order according to its support frequency within the entire data set. Information gain [11] is one of the standard measures used in decision tree construction.

To evaluate PISA using the Nursery dataset the data was distributed amongst four groups, comprising of one player each (each advocating one of the four possible classifications discussed in the previous subsection). 10 fold cross validation was then applied using 50% as the confidence level and 1% as the support threshold. PISA games were played among the four participants to classify 400 cases, 100 cases representing each of the four possible classifications. To evaluate the effect of groups, the nursery dataset was also sub-divided over four groups each of four players. The same 10 fold tests were then applied. The results of both tests are shown in Table 3.

As can be seen from these results, PISA performs consistently and well - out performing the other association rule classifiers, and giving comparable results to the decision tree methods, which are particularly well suited to the nursery dataset. Interestingly PISA performs better using groups than individual

**Table 3. Cross Validation tests on Nursery Data**

Test	4 players	4 groups	TFPC	CBA	CMAR	Rand	Igain
1	96.75	98.5	63.25	72	69.75	100	96.5
2	94	99.75	63	70	70.25	100	99
3	96.5	99	63.25	73.75	70.25	100	99
4	94.75	99.75	63	72.75	69.5	100	99
5	95.5	99	63.25	70.75	69.75	100	99
6	96	98.75	63	71.75	69.75	100	99
7	97.25	99.75	63.25	72.75	69.25	100	99
8	93.75	99.5	63.25	72.75	70.25	100	99
9	96	98	63.25	73.75	69.75	100	99
10	94	99	63	72.75	70	100	99
Summary	95.45	99.1	63.15	72.3	69.85	100	98.75

players: it seems that the greater the number of separate databases the greater the number of arguments found, enabling a more thorough exploration of the problem.

One motivation for using PISA as a classifier is that we believe that allowing agents with different points of view to debate their reasons will produce classifications that are robust in the face of noisy datasets, where the previous examples include a significant number of misclassifications. We explore this conjecture particularly in the context of the the welfare benefits data set [18], since decisions on such benefits typically exhibit high error rates, often as high as 30%. In [18] we found that PADUA was indeed able to perform well in the face of extensive noise. Although we do not have space to discuss the welfare benefits dataset in detail here, we will report the main results. Cross validation tests were performed using a dataset of 3000 records, of which 600 (20%) were incorrectly classified. PISA was tested, with four players and with four groups each of four players, against the same five classifiers as the nursery data. The results are shown in Table 4.

**Table 4. Cross Validation tests on Nursery Data**

Test	4 players	4 groups	TFPC	CBA	CMAR	Rand	Igain
1	92.46	97.46	64.26	69.31	61.57	93.24	90.74
2	91.93	98.50	63.29	69.31	60.60	93.38	90.05
3	91.77	98.39	63.29	69.31	61.57	93.38	90.05
4	92.08	98.15	58.36	69.31	61.67	93.38	90.05
5	92.33	97.31	64.55	69.31	61.57	93.38	90.05
6	93.76	98.62	60.40	69.71	61.60	93.38	90.05
7	94.03	98.15	59.59	69.71	61.57	93.38	90.05
8	91.17	98.35	59.71	69.83	61.57	93.38	90.05
9	93.65	90.46	59.71	69.31	61.60	93.38	90.05
10	93.27	96.63	64.55	68.93	61.62	92.95	90.02
Summary	92.64	97.20	61.77	69.40	61.49	93.32	90.11



Notice from these results that the decision tree techniques, which were able to perform almost perfectly on clean data, exhibit a falling off of performance when this significant degree of noise is introduced. PISA also has some deterioration, but, especially when groups are used, this is relatively slight. Indeed, in these experiments, the PISA classifier with groups achieved the highest level of performance. We find these results very encouraging and will explore this point further in future experiments.

## 5 Concluding Remarks

In this paper the PISA argumentation from experience framework has been described. We have described the mechanisms and strategies used to facilitate multi-party argumentation, as well as the design and implementation of PISA. We have illustrated the operation of the system through an example, and reported some evaluation results comparing PISA with conventional classifiers. We suggest that the PISA framework offers several advantages:

- It allows argumentation between any number of players rather than the more usual two. This requires consideration and resolution of a range of issues associated with dialogues with more than two participants
- It operates without the need for a Knowledge Base but instead allows players to generate arguments using data mining techniques.
- The resulting decision making appears to be robust in the face of misclassifications in some of the data sets.
- The process leads to a reasoned consensus; which is not obtained through, for example, voting, increasing the acceptability of the outcome to all parties.

## References

1. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *SIGMOD Conference*, pages 207–216. ACM Press, 1993.
2. V. Alevan. *Teaching Case Based Argumentation Through an Example and Models*. Phd thesis, University of Pittsburgh, Pittsburgh, PA, USA, 1997.
3. K. D. Ashley. *Modeling Legal Argument*. MIT Press, Cambridge, MA, USA, 1990.
4. M. Bohanec and V. Rajkovic. Expert system for decision making. *Sistemica*, 1(1):145–57, 1990.
5. F. Coenen. The lucs-kdd tpc classification association rule mining algorithm. Department of Computer Science, University of Liverpool, 2004.
6. F. Coenen, P. H. Leng, and S. Ahmed. Data structure for association rule mining: T-trees and p-trees. *IEEE Trans. Knowl. Data Eng.*, 16(6):774–778, 2004.
7. F. Dignum, editor. *Advances in Agent Communication, International Workshop on Agent Communication Languages, ACL 2003, Melbourne, Australia, July 14, 2003*, volume 2922 of *Lecture Notes in Computer Science*. Springer, 2004.
8. F. Dignum and G. Vreeswijk. Towards a testbed for multi-party dialogues. In Dignum [7], pages 212–230.

9. W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In N. Cercone, T. Y. Lin, and X. Wu, editors, *ICDM*, pages 369–376. IEEE Computer Society, 2001.
10. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD*, pages 80–86, 1998.
11. T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
12. M. Olave, V. Rajkovi, and M. Bohanec. An application for admission in public school systems. In *Expert Systems in Public Administration*, pages 145–160, 1989.
13. E. Olivia, M. Viroli, A. Omicini, and P. McBurney. Argumentation and artifact for dialogue support. In *ArgMAS*, pages 24–38, 2008.
14. H. Prakken. Formal systems for persuasion dialogue. *Knowledge Eng. Review*, 21(2):163–188, 2006.
15. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1998.
16. D. R. Traum. Issues in multiparty dialogues. In Dignum [7], pages 201–211.
17. M. Wardeh, T. J. M. Bench-Capon, and F. Coenen. Padua protocol: Strategies and tactics. In K. Mellouli, editor, *ECSQARU*, volume 4724 of *Lecture Notes in Computer Science*, pages 465–476. Springer, 2007.
18. M. Wardeh, T. J. M. Bench-Capon, and F. Coenen. Argument based moderation of benefit assessment. In *JURIX*, pages 128–37. IOS Press, 2008.
19. M. Wardeh, T. J. M. Bench-Capon, and F. Coenen. Arguments from experience: The padua protocol. In P. Besnard, S. Doutre, and A. Hunter, editors, *COMMA*, volume 172 of *Frontiers in Artificial Intelligence and Applications*, pages 405–416. IOS Press, 2008.