Multi-agent Based Classification Using Argumentation From Experience

Maya Wardeh⋆, Frans Coenen, Trevor Bench-Capon, and Adam Wyner

Department of Computer Science, The University of Liverpool,
Liverpool L69 3BX, UK
{maya. wardeh, coenen, tbc, A. Z. Wyner}@ liverpool. ac. uk@ liverpool. ac. uk

Abstract. An approach to multi-agent classification, using an *Argumentation from Experience* paradigm is describe, whereby individual agents argue for a given example to be classified with a particular label according to their local data. Arguments are expressed in the form of classification rules which are generated dynamically. The advocated argumentation process has been implemented in the PISA multi-agent framework, which is also described. Experiments indicate that the operation of PISA is comparable with other classification approaches and that it can be utilised for *Ordinal Classification* and *Imbalanced Class* problems.

Key words: Classification, Argumentation, Multi-agent (Data Mining), Classification Association Rules.

1   Introduction

Argumentation is concerned with the dialogical reasoning processes required to arrive at a conclusion given two or more alternative viewpoints. The process of multi-agent argumentation is conceptualised as a discussion, about some issue that requires a solution, between a set of software agents with different points of view; where each agent attempts to persuade the others that its point of view, and the consequent solution, is the correct one. In this paper we propose applying argumentation to facilitate classification. In particular, it is argued that one model of argumentation, *Arguing from Experience* ([23, 22]), is well suited to the classification tasks. Arguing from Experience provides a computational model of argument based on inductive reasoning from past experience. The arguments are constructed dynamically using Classification Association Rule Mining (CARM) techniques. The setting is a "debate;; about how to classify examples; the generated Classification Association Rules (CARs) provide reasons for and against particular classifications.

The proposed model allows a number of agents to draw directly from past examples to find reasons for coming to a decision about the classification of an unseen instance. Agents formulate their arguments in the form of CARs generated from datasets of past examples. Each agent's dataset is considered to

---

⋆ Corresponding author: maya.wardeh@liverpool.ac.uk

encapsulate that agent's *experience*. The exchange of arguments between agents represents a dialogue which continues until an agent poses an argument for a particular classification that no other agent can refute. The model has been realised in the form of argumentation framework called PISA: *Pooling Information from Several Agents*. The promoted argumentation-based approach is thus a *multi-agent classification* technique that offers a number of practical advantages: (i) dynamic generation of classification rules in a *just in time* manner according to the requirements of each agent, (ii) easy-to-understand explanations, in the form of dialogues, concerning a particular classification, and (iii) application to ordinal classification and imbalanced class problems as well as standard classification. The approach also provides for a natural representation of agent "experience" as a set of records, and the arguments as CARs. At the same time the advocated approach also preserves the privacy of the information each agent *knows*, therefore it can be used with sensitive data.

The rest of this paper is organised as follows. Section 2 provides an overview of the PISA Framework. Section 3 details the nature of the Classification Association Rules (CARs) used in PISA. In Section 4 details and empirical analysis are provided of three different applications of PISA to classification problems: (i) standard classification, (ii) ordinal classification and (iii) the imbalanced class problem. Finally, we conclude with a summary of the main findings and some suggested for further work.

2   Argumentation-Based Multi Agent Classification: The PISA Framework

The intuition behind PISA is to provide a method whereby agents argue about a classification task. In effect each agent can be viewed as a dynamic classifier. The overall process thus leads to a reasoned consensus obtained through argumentation, rather than some other mechanism such as voting (e.g. [2]). It is suggested that this dialogue process increases the acceptability of the outcome to all parties. In this respect PISA can be said to be an ensemble-like method. Both theoretical and empirical research (e.g. [19]) has demonstrated that a good ensemble is one comprising individual classifiers that are relatively accurate but make their errors on different parts of the input training set. Two of the most-popular ensemble methods are: (i) Bagging [3] and (ii) Boosting [13]. Both techniques rely on varying the data to obtain different training sets for each of the classifiers in the ensemble.

PISA is viewed as a bagging-like multi-agent ensemble, whereby the dataset is equally divided amongst a number of participants corresponding to the number of class values in the dataset. Each participant applies the same set of algorithms to mine CARs supporting their advocated class. To this end, each participant can be said to correspond to a single classifier. The argumentation process by which each participant advances moves to support its proposals corresponds to voting methods by which ensemble techniques assign class labels to input cases. But rather than simple voting, PISA applies an argumentation debate (dia-

logue). PISA also differs from Boosting techniques in that it does not generate a sequence of classifiers; instead the desired classification is achieved through the collaborative operation of several classifiers. Furthermore, PISA classifies unseen records by (dynamically) producing a limited number of CARs sufficient to reach a decision without the need to produce the full set of CARs.

The PISA framework comprises three key elements:

1. **Participant Agents**. A number of Participant Agents, at least one for each class in the discussion domain such that each advocates one possible classification.
2. **Chairperson**. A neutral mediator agent which administers a variety of tasks aimed at facilitating PISA dialogues.
3. **Set of CARs**. The joint set of CARs exchanged in the course of one PISA dialogue. This set is represented by a central argument structure, called the *Argumentation Tree*, maintained by the Chairperson. Participant Agents have access to this tree and may use it to influence their choice of move. The agents consult the tree at the beginning of each round and decide which CAR they are going to attack. Full Details of this data structure can be found in [23]. Once a dialogue has terminated the status of the argumentation tree will indicate the *winning*' classification. Note that the dialogues produced by PISA also explains the resulting classifications. This feature is seen as an essential advantage offered by PISA.

Each Participant Agent has its own distinct (tabular) local dataset relating to a classification problem (domain). These agents produce reasons for and against classifications by mining CARs from their datasets using a number of CARM algorithms (Section 3). The antecedent of every CAR represents a set of reasons for believing the consequent. In other words given a CAR, $P \rightarrow c$, this should be read as: *P are reasons to believe that the case should classify as c*. CARs are mined dynamically as required. The dynamic mining provides for four different types of *move*, each encapsulated by a distinct category of CAR. Each Participant Agent can employ any one of the following types of move to generate arguments: (i) Proposing moves, (ii) Attacking moves, and (iii) Refining moves. The different moves available are discussed further below. Note that each of these moves has a set of legal next moves (see Table 1).

Proposing Moves. There is only one kind of proposing move:

1. **Propose Rule**: Allows a new CAR, with a confidence higher than a given threshold, to be proposed. All PISA dialogues commence with a Propose Rule move.

Attacking Moves. Moves intended to show that a CAR proposed by some other agent should not be considered decisive with respect to the current instance. Two sub-types are available: (i) Distinguish and (ii) Counter Rule, as follows:

2. **Distinguish**: Allows an agent to add new attributes (premises) to a previously proposed CAR so that the confidence of the new rule is lower than the confidence threshold, thus rendering the original classification inadmissible.

3. **Counter Rule**: Similar to Propose Rule but used to cite a classification other than that advocated by the initial Propose Rule move.

Refining Moves. Moves that enable a CAR to be refined to meet a *counter attack*. For the purposes of using PISA as a classifier, one refining move is implemented:

4. **Increase Confidence**: Allows the addition of new attribute(s) to the premise associated with a previously proposed CAR so as to increase the confidence of the rule, thus increasing the confidence that the case should be classified as indicated.

Move	Label	Next Move		Move	Label	Next Move
1	Propose Rule	2, 3		3	Counter Rule	2, 1
2	Distinguish	4, 1		4	Increase Conf	2, 3

Table 1. Legal next moves in PISA.

3 PISA Dynamic CAR Mining

Having introduced, in the foregoing, the legal moves in PISA dialogues, the realisation of these moves is described in this section. The idea is to mine CARs according to: (i) a desired minimum confidence, (ii) a specified consequent and (iii) a set of candidate attributes for the antecedent (a subset of the attributes represented by the case under discussion). Standard CARM techniques (e.g.[6, 15]) tend to generate the complete set of CARs represented in the input data. PISA on the other hand utilises a *just in time* approach to CARM, directed at generating particular subsets of CARs, and applied such that each agent mines appropriate CARs as needed. The mining process supports two different forms of dynamic ARM request:

1. Find a subset of rules that conform to a given set of constraints.
2. Distinguish a given rule by adding additional attributes.

In order to realise the above, each Participant Agent utilises a T-tree [5] to summarise its local dataset. A T-tree is a *reverse* set enumeration tree structure where nodes are organised using *reverse* lexicographic ordering, which in turn enables direct indexing according to attribute number; therefore computational efficiency gains are achieved. A further advantage, with respect to PISA, is that the reverse ordering dictates that each sub-tree is rooted at a particular class attribute, and so all the attribute sets pertaining to a given class are contained in a single T-tree branch. This means that any one of the identified dynamic CARM requests need be directed at only one branch of the tree. This reduces the overall processing cost compared to other prefix tree structures (such as FP-Trees [15]). To further enhance the dynamic generation of CARs a set of algorithms that work directly on T-trees were developed. These algorithms were able to mine CARs satisfying different values of support threshold. At the start of the dialogue each player has an empty T-tree and slowly builds a *partial* T-tree from their data set, as required, containing only the nodes representing

attributes from the case under discussion plus the class attribute. Note that no node pruning, according to some user specified threshold, takes place; except for nodes that have zero support. Two dynamic CAR retrieval algorithms were developed: (i) Algorithm A which finds a rule that conforms to a given set of constraints, and (ii) Algorithm B which distinguishes a given rule by adding additional attributes. Further details of these algorithms can be found in [24]

4   Applications of PISA

Arguing from Experience enables PISA agents to undertake a number of different tasks, mainly:

1. *Multi-agent Classification*: Follows the hypothesis that the described operation of PISA produces at least comparative results to that obtained using traditional classification paradigms.
2. *Ordinal Classification*: Follows the hypothesis that PISA can be successfully applied to datasets with ordered-classes, using a simple *agreement* strategy.
3. *Classifying imbalanced data using dynamic coalitions*: Follows the hypothesis that dynamic coalitions between a number of participant agents, representing *rare* classes, improves the performance of PISA with imbalanced multi-class datasets.

In this section the above applications of PISA are empirically evaluated. For the evaluation we used a number of real-world datasets drawn from the UCI repository [4]. Where appropriate continuous values were discretised into ranges. The chosen datasets (Table 2) display a variety of characteristics with respect to number of records (R), number of classes (C) and number of attributes (A). Importantly, they include a diverse number of class labels, distributed in a different manner in each dataset (balanced and unbalanced), thus providing the desired variation in the experience assigned to individual PISA participants.

Name	R	C	A	Bal	Name	R	C	A	Bal
Hepatitis	155	2	19(56)	no	Ionosphere	351	2	34(157)	no
HorseColic	368	2	27(85)	no	Congressional Voting	435	2	17(34)	yes
Cylnder Bands	540	2	39(124)	yes	Breast	699	2	11(20)	yes
Pima (Diabetes)	768	2	9(38)	yes	Tic-Tac-Toe	958	2	9(29)	no
Mushrooms	8124	2	23(90)	yes	Adult	48842	2	14(97)	no
Iris	150	3	4(19)	yes	Waveform	5000	3	22(101)	yes
Wine	178	3	13(68)	yes	Connect4	67557	3	42(120)	no
Lymphography	148	4	18(59)	no	Car Evaluation	1728	4	7(25)	no
Heart	303	5	22(52)	no	Nursery	12960	5	9(32)	no
Dematology	366	6	49(49)	no	Annealing	898	6	38(73)	no
Zoo	101	7	17(42)	no	Automobile (Auto)	205	7	26(137)	no
Glass	214	7	10(48)	no	Page Blocks	5473	7	11(46)	no
Ecoli	336	8	8(34)	no	Solar Flare	1389	9	10(39)	no
Led7	3200	10	8(24)	yes	Pen Digits	10992	10	17(89)	yes
Chess	28056	18	6(58)	no					

Table 2. Summary of data sets. Columns indicate: domain name, number of records, number of classes, number of attributes and class distribution (approximately balanced or not.

4.1   Application 1: PISA-Based Classification

The first application of PISA is in the context of multi-agent classification based on argumentation. In order to provide an empirical assessment of this application we ran a series of experiments designed to evaluate the hypothesis that PISA produces at least comparative results to that obtained using traditional classification paradigms. In particular, ensemble classification methods. The results presented throughout this sub-section, unless otherwise noted, were obtained using *Ten-fold Cross Validation* (TCV). For the purposes of running PISA, each training dataset was equally divided among a number of Participant Agents corresponding to the number of classes in the dataset. Then a number of PISA dialogues were executed to classify the cases in the test sets[1]. In order to fully assess its operation, PISA was compared against a range of classification paradigms:

1. *Decision trees*: Both C4.5, as implemented in [14], and the Random Decision Tree (RDT)[7], were used.
2. *CARM*: The TFPC (Total From Partial Classification) algorithm [6] was adopted because this algorithm utilises similar data structures [5] as PISA.
3. *Ensemble classifiers*: Table 3 summarises the techniques used. We chose to apply Boosting and Bagging, combined with decision trees, because previous work demonstrated that such combination is very effective (e.g. [2, 19]).

Ensemble	Technique	Base	Ensemble	Technique	Base
Bagging-C4.5	Bagging[3]	C4.5 (S=1%)	Bagging-RDT	Bagging[3]	RDT (S=1%)
ADABoost-C4.5	ADABoost.M1 [13]	C4.5 (S=1%)	ADABoost-RDT	ADABoost.M1 [13]	RDT (S=1%)
MutliBoostAB-C4.5	MultiBoosting [25]	C4.5 (S=1%)	MultiBoostAB-RDT	MultiBoosting [25]	RDT (S=1%)
DECORATE	[18]	C4.5 (S=1%)			

Table 3. Summary of the Ensemble Methods used. The implementation of these methods was obtained from [14]. (S=Support, RDT=Random Decision Trees)

For each of the included methods (and PISA) three values were calculated for each dataset: (i) classification *error rate*, (ii) *Balanced Error Rate* (BER) using a confusion matrix obtained from each TCV[2]; and (iii) *execution time*. These three values then provided the criteria for assessing and comparing the classification paradigms.

---

[1] For each evaluation the confidence threshold used by each participant was 50% and the support threshold 1%.

[2] Balanced Error Rates (BER) were calculated, for each dataset, as follows:

$$BER = \frac{1}{C\sum_{i=1}^{C}\frac{F_{ci}}{F_{ci}+T_{ci}}}$$

C = the number of classes in the dataset, T_{ci} = the number of cases which are correctly classified as class ci, and F_{ci} = the number of cases which should have been classified as ci but where classified under different class label.

The results are presented in Table 4. From the table it can be seen that PISA performs consistently well; out performing the other association rule classifier, and giving comparable results to the decision tree methods. Additionally, PISA produced results comparable to those produced by the ensemble methods. Moreover, PISA scored an average overall accuracy of 93.60%, higher than that obtained from any of the other methods tested (e.g. Bagging-RDT (89.48%) and RDT (90.24%))[3].

Dataset	PISA	Ensembles							Decision Trees		TFPC
		Bagging		ADABoost.M1		MultiBoost		Decorate			
		C4.5	RDT	C4.5	RDT	C4.5	RDT		C4.5	RDT	
Hepatitis	**13.33**	18.06	14.84	15.48	21.29	13.55	18.71	16.13	16.13	23.23	18.00
Ionosphere	3.33	7.69	6.84	7.12	10.83	6.27	10.83	7.41	8.55	**2.57**	14.29
HorseColic	**2.78**									3.89	22.78
Congress	1.78	3.01	2.31	2.08	3.01	2.08	3.01	2.77	4.16	**0.00**	9.30
CylBands	**15.00**	42.22	27.04	42.22	34.81	42.22	34.81	39.81	42.22	36.48	30.37
Breast	**3.91**	5.01	4.86	4.86	4.86	4.86	5.43		4.86	5.07	10.00
Pima	**14.47**	27.21	25.26	25.26	23.83	25.13	24.87	25.66	26.69	16.18	25.92
TicTacToe	2.84	7.20	5.43	**2.19**	20.35	**2.19**	20.35	5.85	15.45	20.77	33.68
Mushrooms	0.41	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.06	1.05
Adult	14.49								**13.09**	19.19	
Iris	**2.67**	4.67	5.33	6.00	7.33	6.00	7.33	4.67	4.00	8.00	6.00
Waveform	**2.16**	17.97	11.98	21.48	21.48	13.62	11.96	21.48	21.48	2.42	33.32
Wine	1.18								**0.00**	25.29	
Connect4	5.01								**4.31**	34.17	
Lympho	**6.23**	18.92	19.59	14.86	29.73	15.54	29.73	19.59	22.97	25.00	24.29
Car Eval	4.11	4.51	**1.24**	2.43	6.25	2.60	6.25	4.28	5.09	5.90	30.00
Heart	5.05	20.07	19.73	22.79	21.09	19.05	19.73	20.41	19.05	**4.67**	46.67
Page Bloc	**2.24**	6.93	6.93	7.02	6.93	7.02	6.93	6.93	7.02	6.94	9.95
Nursery	6.37	2.08	3.09	**0.38**	3.09	0.35	3.09	1.91	2.62	3.72	22.25
Dematology	4.96	4.10	3.55	3.83	15.30	3.28	15.30	**1.64**	6.01	5.28	25.00
Annealing	9.55	1.22	0.67	0.45	1.78	**0.56**	1.78	1.34	1.56	1.67	11.80
Zoo	9.90	7.92	4.95	3.96	19.80	3.96	19.80	6.93	7.92	**0.00**	8.00
Auto	**12.00**	15.12	15.61	14.15	21.46	15.61	21.46	16.10	18.05	17.00	29.00
Glass	**14.69**	27.10	21.50	22.43	29.91	25.23	29.91	29.91	33.18	29.91	33.81
Ecoli	5.17	13.99	15.18	16.37	24.70	14.88	24.70	13.10	15.77	8.79	37.27
Flare	6.09	**2.48**	3.41	3.41	3.41	3.41	3.10	3.10	**2.48**	8.03	14.74
Led7	**12.00**	24.81	24.16	24.84	24.28	24.91	24.34	24.75	24.84	24.25	31.03
Pen Digit	2.75	4.47	**1.35**	1.58	2.51	5.07	1.87	2.51	5.65	1.08	18.24
Chess	9.13									18.58	15.73

Table 4. Test set error rate (%). Values in **bold** are the lowest in a given dataset.

Table 5 shows the BER for each of the given datasets. From the table it can be seen that PISA produced reasonably good results overall, producing the best result in 14 out of the 39 datasets tested.

Table 6 gives the execution times (in milliseconds) for each of the methods. Note that PISA is not the fastest method. However, the recorded performance is by no means the worst (for instance Decorate runs slower than PISA with respect to the majority of the datasets). Additionally, PISA seems to run faster than Bagging and ADABoost with some datasets.

4.2   Application 2: PISA-Based Ordinal Classification

Having established PISA as a classification paradigm, we now explore the application of PISA to ordinal classification. In this form of multi-class classification the set of class labels is finite and ordered. Whereas traditional classification paradigms commonly assume that the class values are unordered. For many practical applications class labels do exhibit some form of order (e.g. the weather can be cold, mild, warm and hot). Given ordered classes, one is not only concerned

---

[3] These accuracies were calculated from Table 4

to maximise the classification accuracy, but also to minimise the distances between the actual and the predicted classes. The problem of ordinal classification is often solved by either multi-class classification or regression methods. However, some new approaches, tailored specifically for ordinal classification, have been introduced in the literatures (e.g. [12, 21]). PISA can be utilised for ordinal classification by the means of *biased agreement*. Agents in PISA have the option to *agree* with CARs suggested by other agents, by not attacking these rules, even if a valid attack is possible. PISA agents can either *agree with all* the opponents or with a *pre-defined set* of opponents that match the class order. For instance, in the weather scenario, agents supporting the decision that the weather is hot, agree with those with the opinion that the weather is warm, and vice versa. Whereas agents supporting that the weather is cold or mild agree with each other. We refer to the latter form of agreement by the term *biased agreement*. In which the agents are equipped with a simple list of the class labels that they could agree with (the *agreement list*). Here, we have two forms of this mode of agreement:

Dataset	PISA	Ensembles							Decision Trees		TFPC
		Bagging		ADABoost.M1		MultiBoost		Decorate			
		C4.5	RDT	C4.5	RDT	C4.5	RDT		C4.5	RDT	
Hepatitis	**12.00**	27.41	20.63	23.37	33.69	19.89	25.05	24.60	23.38	38.19	36.44
Ionosphere	4.58	7.08	6.63	6.42	11.43	5.31	11.43	7.08	8.17	**2.19**	13.41
HorseColic	**2.80**									3.71	28.63
Congress	2.35	3.43	2.66	2.27	3.19	2.27	2.27	3.05	4.69	**0.00**	9.71
CylBands	**14.50**	46.10	24.48	46.10	35.63	46.10	35.63	40.14	46.10	34.56	32.78
Breast	**4.75**	6.03	6.20	6.20	6.20	6.07	6.20	6.71	6.20	4.71	12.89
Pima	**13.94**	28.88	26.86	26.86	25.18	26.72	26.12	27.16	28.34	24.47	33.67
TicTacToe	2.14	6.71	5.35	**2.25**	22.46	2.25	22.46	5.25	16.98	22.94	47.44
Mushrooms	0.59	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.06	1.04
Adult	**8.80**								17.75	39.89	
Iris	**2.90**	4.61	5.29	5.93	7.32	5.93	7.32	4.69	3.96	7.96	6.07
Waveform	3.93	18.00	11.99	21.51	21.51	13.64	11.97	21.51	21.51	2.39	33.35
Wine	1.42									**0.00**	24.05
Connect4	11.90									**5.33**	66.67
Lympho	**15.95**	30.12	9.74	25.90	43.43	38.66	43.43	39.35	35.97	47.11	16.09
Car Eval	8.21	11.25	6.77	**4.79**	10.43	5.29	10.43	10.24	16.55	10.67	75.00
Heart	8.25	9.16	8.97	9.93	7.98	**7.85**	8.97	9.43	7.97	9.82	48.02
Page Bloc	**9.45**	21.46	22.85	27.89	21.46	27.87	21.46	22.85	27.89	21.48	19.89
Nursery	5.47	4.14	2.28	1.05	5.82	**0.76**	5.78	4.55	5.98	5.74	40.10
Dematology	8.49	4.68	3.89	3.93	19.41	3.31	19.41	1.84	7.00	**3.25**	61.67
Annealing	16.13	6.76	3.92	**2.57**	4.31	3.25	4.31	7.16	6.83	4.43	33.51
Zoo	13.23	12.78	10.71	10.71	36.51	10.71	36.51	15.71	17.50	**0.00**	17.14
Auto	12.26	**11.43**	15.98	10.55	18.92	15.98	18.92	12.84	17.04	13.57	19.60
Glass	**16.09**	24.57	19.42	24.33	29.56	23.18	29.56	29.56	37.98	29.56	48.55
Ecoli	16.18	36.66	40.18	41.18	51.89	37.92	51.89	24.16	43.35	**9.42**	23.23
Flare	17.18	12.77	12.66	10.91	12.66	12.66	12.62	12.62	12.54	**7.59**	14.74
Led7	**11.84**	24.56	24.07	24.87	24.31	25.09	24.23	24.92	24.72	24.36	31.39
Pen Digit	3.47	4.48	**1.51**	1.59	2.23	4.92	1.89	2.23	5.57	3.73	18.38
Chess	**9.63**									16.38	24.53

Table 5. Test set BER (%). Values in **bold** are the lowest in a given dataset.

1. **No Attack Biased Agreement (NA-BIA)**: In which agents consult their agreement list before mining any rules from their local datasets and attempt only to attack/respond to CARs of the following shape: $P \rightarrow Q : \forall q \in Q, q \notin agreement_list$

2. **Confidence Threshold Biased Agreement (CT-BIA)**: Here, if the agents fail to attack any CARs that contradict with their agreement list, then they try to attack CARs $(P \rightarrow Q : \exists q \in Q, q \in agreementlist)$ if and only if they fail to mine a matching CAR, with same or higher confidence, from their own local dataset $(P \rightarrow Q` : Q` \supseteq Q)$.

To test the hypothesis that the above approach improves the performance of PISA when applied to ordinal classification a series of TCV tests, using a number of datasets from Table 2 which have ordered classes, were conducted. PISA was run using the NA-BIA and CT-BIA strategies, and the results were compared against the use of PISA without any agreement strategy. Additionally, to provide better comparison the Mean Squared Error (MSE) and the Mean Absolute Error (MAE) rates for the included datasets and methods were calculated. [10] notes that little attention has been directed at the evaluation of ordinal classification solutions, and that simple measures, such as accuracy, are not sufficient. In [10] a number of evaluation metrics, for ordinal classification, are compared. As a result MSE is suggested as the best metric when more (smaller) errors are preferred to reduce the number of large errors; while MAE is a good metric if, overall, fewer errors are preferred with more tolerance for large errors. Table 7 provides a summary of the results of the experiments. From the table it can be seen that the NA-BIA produces better results with datasets with ordinal classes.

cm Dataset	PISA	Ensembles							Decision Trees		TFPC
		Bagging		ADABoost.M1		MultiBoost		Decorate			
		C4.5	RDT	C4.5	RDT	C4.5	RDT		C4.5	RDT	
Hepatitis	115	110	**40**	190	70	200	60	610	**40**	60	213
Ionosphere	437	1130	210	1170	20	1210	20	4090	80	**12**	109
HorseColic	17									**4.8**	108
Congress	34	50	20	20	140	130	20	590	30	**15**	154
CylBands	83	110	130	40	20	40	20	1190	40	**17**	936
Breast	31	110	110	140	110	170	170	330	**8.1**	**8**	11
Pima	75	160	90	80	130	80	110	500	20	21	**11**
TicTacToe	71	80	70	250	30	280	10	620	20	**6.1**	61.4
Mushrooms	313	750	380	110	**50**	60	**50**	6400	80	117	630
Adult	3019									**706**	1279
Iris	42	40	50	60	50	50	10	110	10	13	**2**
Waveform	1243	1840	380	4400	830	1650	560	4730	200	**102**	862
Wine	136									**106**	163
Connect4	4710									**3612**	6054
Lympho	15	80	50	90	10	70	10	140	**5**	**5**	29
Car Eval	74	300	110	370	20	20	310	1580	80	24	**17**
Heart	343	250	80	480	20	430	10	620	20	**5**	183
Page Bloc	159	130	430	430	130	280	130	430	120	**55**	60
Nursery	965	1790	720	3130	60	3760	**10**	1449	110	139	204
Dematology	194	160	40	230	20	20	20	480	20	**7**	169
Annealing	750	1090	120	850	**10**	1170	**10**	3340	50	28	689
Zoo	43	40	10	20	10	30	10	110	10	**5**	85
Auto	210	440	70	320	10	350	10	520	20	**5**	43
Glass	180	260	120	340	**10**	430	**10**	1060	20	**10**	43
Ecoli	139	240	150	360	10	340	10	1510	10	**3**	4
Flare	239	30	20	60	40	20	20	140	10	27	23
Pen Digits	1345	2300	460	5810	820	2790	800	2300	290	**80**	1606
Led7	78	730	360	260	130	1150	480	3380	110	90	**25**
Chess	2412									334	**226**

Table 6. Test set execution times (milliseconds). Values in bold are the lowest in a given dataset.

Datasets	ER			BER			MSE			MAE		
	PISA	CT-BIA	NA-BIA	PISA	CT-BIA	NA-BIA	PISA	CT-BIA	NA-BIA	PISA	CT-BIA	NA-BIA
Lympo	6.21	4.76	**3.38**	15.95	20.73	**13.94**	0.199	0.046	**0.015**	2.07	1.36	**0.84**
Car Eval	4.11	5.00	**4.03**	**9.53**	10.09	10.61	0.863	1.220	**0.708**	1.02	1.32	**1.01**
Page Bloc	**2.67**	3.64	3.91	13.43	10.42	**10.06**	**1.250**	5.164	4.757	**0.49**	0.78	0.83
Nursery	6.37	6.27	**5.83**	11.79	13.57	**7.88**	7.450	7.071	**6.725**	1.61	1.57	**1.46**
Dema	**4.96**	7.95	6.87	8.49	8.74	**7.53**	0.144	0.143	**0.100**	1.46	1.37	**1.24**
Zoo	9.90	7.92	**6.86**	13.23	14.67	**12.17**	**0.223**	0.230	0.232	2.26	2.26	**1.96**
Ecoli	6.03	5.52	**4.34**	16.81	**6.72**	6.91	0.008	0.008	**0.005**	8.23	7.92	**4.63**

Table 7. The application of PISA with datasets from Table 2 with ordered classes.

4.3 Application 3: PISA-Based Solution to the Imbalanced Class Problem

Another application of PISA is using dynamic coalitions between different agents to produce better performance in the face of imbalanced class problem. It has

been observed (e.g.[16]) that class imbalance (i.e a significant differences in class prior probabilities) may produce an important deterioration of the performance achieved by existing learning and classification systems. This situation is often found in real-world data describing an infrequent but important case (e.g. Table 2. There have been a number of proposed mechanisms for dealing with the class imbalance problem (e.g. [9, 20]). [11, 16] note a number of different approaches:

1. Changing class distributions: by "upsizing" the small class at random (or focused random), or by "downsizing" the large class at random (or focused random).
2. At the classifier level by either: manipulating classifiers internally, cost-sensitive learning or one-class learning.
3. Specially designed ensemble learning methods.
4. Agent-based remedies such as that proposed in [17] where three agents, each using a different classification paradigm, generate classifiers from a filtered version of the training data. Individual predictions are then combined according to a voting scheme. The intuition is that the models generated using different learning biases are more likely to make errors in different ways.

In the following we present a refinement of the basic PISA model which enables PISA to tackle the imbalance-class problem in multi-class datasets, using Dynamic Coalitions between agents representing the rare classes. Unlike the biased agreement approach (Sub-section 4.2), coalition requires mutual agreement among a number of participants, thus a preparation step is necessary. However, for the purposes of this paper we assume that the agents representing the rare classes are in coalition from the start of the dialogue, thus eliminating the need for a preparatory step. The agents in a coalition stop attacking each other, and only attack CARs placed by agents outside the coalition. The objective of such coalition is to attempt to remove the agents representing dominant class(es) from the dialogue, or at least for a pre-defined number of rounds. Once the agent in question is removed from the dialogue, the coalition is dismantled and the agents go on attacking each others as in a normal PISA dialogue. In the following we provide experimental analysis of two coalition techniques:

1. **Coalition (1)**: The coalition is dismantled if the agent supporting the dominant class does not participate in the dialogue for two consecutive rounds.
2. **Coalition (2)**: The coalition is dismantled if the agent supporting the dominant class does not participate in the dialogue for two consecutive rounds, and this agent is not allowed to take any further part in the dialogue.

To test the hypothesis that the above approaches improves the performance of PISA when applied to imbalanced class datasets we ran a series of TCV tests using a number of datasets from Table 2, which have imbalanced class distributions. The results were compared against the use of PISA without any coalition strategy. Four measures were used in this comparison: error rate, balanced error rate, time and geometric mean (g-mean)[4]. This last measure was used to quan-

---

[4] The geometric mean is defined as $g - mean = (\prod_{i=1}^{C} p_{ii})^{\frac{1}{C}}$ where p_{ii} is the class accuracy of class i, and C is the number of classes in the dataset.

tify the classifier performance in the class [1]. Table 8 provides the result of the above experiment. From,the table it can be seen that both coalition techniques boost the performance of PISA, with imbalance-class datasets, with very little additional cost in time, due to the time needed to dismantle the coalitions.

Datasets	ER			BER			G-Mean			Time		
	PISA	Coal(1)	Coal(2)	PISA	Coal(1)	Coal(2)	PISA	Coal(1)	Coal(2)	PISA	Coal(1)	Coal(2)
Connect4	5.02	4.18	**3.78**	11.90	9.68	**8.70**	87.47	89.96	**91.00**	4710	5376	5818
Lympo	6.21	5.02	**4.03**	15.95	**11.90**	14.64	69.31	82.60	**92.81**	15	65	55
Car Eval	4.11	**3.73**	4.22	9.53	7.24	**4.47**	79.42	88.40	**92.52**	74	163	158
Heart	5.05	**4.95**	**4.95**	8.25	**2.54**	3.17	84.44	87.67	**89.97**	343	531	612
Page Bloc	2.24	1.43	**1.14**	13.43	**7.96**	9.63	68.17	**85.43**	84.02	159	207	222
Derma	4.96	3.91	**3.60**	8.49	4.95	**4.48**	75.79	84.27	**90.14**	194	119	107
Annealing	9.55	4.24	**4.01**	16.13	7.72	**4.24**	63.57	86.20	**91.52**	750	980	881
Zoo	9.90	8.00	**7.00**	13.23	8.33	**3.92**	67.19	85.42	**85.51**	43	93	85
Auto	12.00	6.37	**5.77**	12.26	**6.53**	6.64	79.74	87.88	**90.87**	210	336	293
Glass	14.69	12.02	**5.74**	16.09	7.45	**5.81**	80.12	**93.60**	93.24	180	178	171
Ecoli	6.03	**5.15**	5.64	16.18	10.93	**3.92**	74.16	87.31	**96.01**	139	86	81
Flare	**6.09**	7.10	6.86	17.18	5.58	**5.15**	77.41	91.21	**95.76**	2393	2291	6267
Chess	9.13	8.47	**6.28**	9.63	5.91	**5.82**	76.70	91.26	**92.22**	2412	3305	3393

Table 8. The application of PISA with imbalanced *multi-class* datasets from Table 2.

5   Conclusions

The PISA Arguing from Experience Framework has been described. PISA allows a collection of agents to conduct a dialogue concerning the classification of an example. The system progresses in a round-by-round manner. During each round agents can elect to propose an argument advocating their own position or attack another agent's position. The arguments are mined and expressed in the form of CARs, which are viewed as generalisations of the individual agent's experience. In the context of classification PISA provides for a "distributed" classification mechanism that harnesses all the advantages offered by Multi-agent Systems. The effectiveness of PISA is comparable with that of other classification paradigms. Furthermore the PISA approach to classification can operate with temporally evolving data. We have also demonstrated that PISA can be utilised to produce better performance with imbalanced classes and ordinal classification problems.

References

1. Alejo, R., Garcia, V., Sotoca, J., Mollineda, R., Sanchez, J.: Improving the Performance of the RBF Neural Networks with Imbalanced Samples. In Proc. 9th Intl. Conf. on Artl. Neural Networks. Springer. pp. 162–169 (2007).
2. Bauer, E. and Kohavi, R: An empirical comparison of voting classification algorithms: Bagging, Boosting and variants. In J. Machine Learning, vol.36. pp. 105 – 139 (1999).
3. Brieman,L.: Bagging predictors. In J. Machine Learning, vol.24, Springer. pp. 123–140 (1996).
4. Blake, C.L., and Merz.,C.J.: UCI Repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`: University of California (1998).
5. Coenen,F., Leng,P., and Ahmed,S.: Data structure for association rule mining: T-trees and p-trees. In J. IEEE Trans. Knowl. Data Eng., vol.16(6). pp. 774 – 778 (2004).
6. Coenen, F. and Leng, P.: Obtaining Best Parameter Values for Accurate Classification. In Proc. ICDM'05, IEEE. pp. 597–600 (2005).
7. Coenen, F.: The LUCS-KDD Decision Tree Classifier Software, `http://www.csc.liv.ac.uk/~frans/KDD/Software/DecisionTrees/decisionTree.html`, Dept. of Computer Science, The University of Liverpool, UK, (2007).

8. Dietterich, T.: Ensemble methods in machine learning. In: Lecture Notes in Computer Science, vol.1857, Springer. pp. 1–15 (2000).
9. Elkan, C. : The Foundations of Cost-Sensitive Learning. In Proc. IJCAI'01, vol.2. pp: 973–978 (2001).
10. Gaudette, L., and Japkowicz, N.: Evaluation Methods for Ordinal Classification. In Yong, G., and Japkowicz, N. (editors), Advances in Artificial Intelligence, LCNS, vol.5549, Springer. pp. 207–210 (2009).
11. Guo, X., Yin, Y., Dong, C. and Zhou,G.: On the Class Imbalance Problem. In Proc. ICNC'08, IEEE. pp. 192–201(2008).
12. Frank, E., Hall, M.: A simple approach to ordinal classification. In Proc. ECML'01, Springer. pp. 145–157 (2001).
13. Freund, Y. and Schapire,R: Experiments with a new boosting algorithm. In Proc. ICML'96. pp. 148 – 156 (1996).
14. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten,I. H: The WEKA Data Mining Software: An Update. In J. SIGKDD Explorations, vol.11(1), (2009).
15. Han, J., Pei, J. and Yiwen, Y.: Mining Frequent Patterns Without Candidate Generation. In Proc. SIGMOD'00, ACM Press. pp. 1–12 (2000).
16. Japkowicz, N., and Stephen, S.: The Class Imbalance Problem: A systematic study. In J. Intelligent Data Analysis, vol.6(5). pp. 429–449 (2002).
17. Kotsiantis, S., and Pintelas, P. :Mixture of Expert Agents for Handling Imbalanced Data Sets. In Annals of Mathematics, Computing & TeleInformatics, vol.1. pp. 46–55 (2003).
18. Melville, P., and Mooney,R.: Constructing Diverse Classifier Ensembles Using Artificial Training Examples. In Proc. IJCAI'03. pp.505–510 (2003).
19. Opitz, D., and Maclin, R.: Popular Ensemble Methods: An Empirical Study, In J. Artif. Intell. Research, vol.11. pp: 169 – 198 (1999).
20. Philippe., L., Lallich, S., Do., T. and Pham, N.: A comparison of different off-centered entropies to deal with class imbalance for decision trees. In proc. PAKDD'08. pp. 634–643 (2008).
21. Blaszczynski, J., Slowinski, R. and Szelag, M.: Probabilistic Rough Set Approaches to Ordinal Classification with Monotonicity Constraints. In Hllermeier, E., Kruse, R. and Hoffmann, F.,(editors): Computational Intelligence for Knowledge-Based Systems Design. LNCS vol.6178, Springer Berlin. pp. 99–108 (2010)
22. Wardeh, M., Bench-Capon, T., and Coenen, F.: Multi-Party Argument from Experience. In Proc. ArgMAS'09. Budapest, Hungary. Springer. (2009).
23. Wardeh, M., Bench-Capon, T., and Coenen, F.: Arguments from Experience: The PADUA Protocol. In Proc. COMMA'08. Toulouse, France. IOS press. pp. 405–416 (2008).
24. Wardeh, M., Bench-Capon, T., and Coenen,F.: Dynamic Rule Mining for Argumentation Based Systems. In Proc. 27th SGAI Intl. Conf. on AI (AI'07). Cambridge, UK. Springer, London. pp. 65–78 (2007).
25. Webb, G.: MultiBoosting: A Technique for Combining Boosting and Wagging. In J. Machine Learning, vol.40(2). pp. 159–196 (2000).