Item Response Theory in AI: Analysing Machine Learning Classifiers at the Instance Level

Fernando Martínez-Plumed[a], Ricardo B. C. Prudêncio[b], Adolfo Martínez-Usó[c], José Hernández-Orallo[a]

[a]*Dept. of Computer Systems and Computation, Universitat Politècnica de València, 46022 Valencia, Spain*
[b]*Centro de Informática, Universidade Federal de Pernambuco, Recife (PE), Brasil*
[c]*Universitat Jaume I de Castelló, Spain*

Abstract

[1] AI systems are usually evaluated on a range of problem instances and compared to other AI systems that use different strategies. These instances are rarely independent. Machine learning, and supervised learning in particular, is a very good example of this. Given a machine learning model, its behaviour for a single instance cannot be understood in isolation but rather in relation to the rest of the data distribution or dataset. In a dual way, the results of one machine learning model for an instance can be analysed in comparison to other models. While this analysis is *relative* to a population or distribution of models, it can give much more insight than an isolated analysis. Item response theory (IRT) combines this duality between *items* and *respondents* to extract latent variables of the items (such as discrimination or difficulty) and the respondents (such as ability). IRT can be adapted to the analysis of machine learning experiments (and by extension to any other artificial intelligence experiments). In this paper, we see that IRT suits classification tasks perfectly, where instances correspond to items and classifiers correspond to respondents. We perform a series of experiments with a range of datasets and classification methods to fully understand what the IRT parameters such as discrimination, difficulty and guessing mean for classification instances (and their relation to instance hardness measures) and how the estimated classifier ability can be used to compare classifier performance in a different way through classifier characteristic curves.

Keywords: Artificial intelligence evaluation, item response theory, machine learning, instance hardness, classifier metrics.

1. Introduction

Experimental evaluation is vital for AI research, especially for those problems whose theoretical evaluation is elusive. Not only are AI researchers interested in the performance of a particular method for a particular problem, but also in comparing that performance with other problems and against other methods. Research in AI usually reports aggregate measures for a benchmark of problems or performs pairwise comparisons between techniques, which make it possible to say that method A is better than method B.

[1] *Note for editor and reviewers:* This paper is a significantly extended version of [1], awarded as best paper at ECAI2016. This submission follows an invitation by AIJ for publication in the fast track scheme. The paper has been significantly rewritten and extended in terms of the use of IRT libraries, more and larger datasets, more figures and examples and a more consolidated view of how IRT has to be applied in machine learning. There are two new sections. One is dealing with the relation between the IRT parameters and many instance hardness measures in the classification literature. The other presents an analysis of IRT results with different model configurations, 2PL, 3PL and 3PL when the datasets are balanced on purpose, in order to better correspond with the original IRT situation where possible answers (here classes) are usually balanced. We have also covered recent related work and expanded on the discussion of potential applications. This is the revised version of the journal submission, according to the reviewers' comments.

Email addresses: `fmartinez@dsic.upv.com` (Fernando Martínez-Plumed), `rbcp@cin.ufpe.br` (Ricardo B. C. Prudêncio), `auso@uji.es` (Adolfo Martínez-Usó), `jorallo@dsic.upv.com` (José Hernández-Orallo)

However, when we want to improve a method or develop new ones, we need to know *where* methods fail and how, and what problems are more challenging for current state-of-the-art AI methods.

Item response theory (IRT) comprises a group of modelling and statistical tools borrowed from psychometrics that are designed to provide a precise characterisation of items and respondents (subjects), through the analysis of their responses [2, 3, 4]. By considering AI problems as items and AI methods as respondents, we can apply IRT to any area in AI. As usual in IRT, we can re-understand the proficiency (or ability) of a method as the difficulty level whose problems the method is able to solve.

In this paper we focus on the use of IRT in supervised machine learning, and classification in particular, where items are equated to the instances of a dataset and responses are equated to predicted classes. Not only is this the most similar machine learning scenario to many of the dichotomous problems addressed in IRT, but it also brings insightful parallelisms to many interesting questions in machine learning such as instance hardness, noise handling, outliers, meta-learning, borderline areas, risk aversion, etc., with a particular re-understanding under this theory.

For instance, it has been recently demonstrated that incorporating instance hardness into the learning process can significantly increase classification performance [5, 6]. However, existing instance hardness measures give a very limited perspective of what is happening with an instance. Apart from difficulty, a very interesting parameter in IRT is the *discrimination* of an instance. In machine learning, as we will see, the discrimination parameter can be seen as a measure of how effective each instance is for differentiating between strong or weak models for a certain dataset. These instances are usually solved by more proficient classifiers and misclassified by the rest. But there are other difficult instances for which good and bad classifiers behave equally poorly. These are not discriminating. Are they noise? Or are they systematically neglected by some families of classifiers? What can IRT tell us about these instances?

But, also interestingly, IRT shows a dual behaviour between instance difficulty and classifier ability. For instance, some classifiers can solve all the simple instances but none of the difficult ones, whereas other classifiers can solve most (but not all) of the simple ones and some of the difficult ones. Are these riskier classifiers better? What does this say about their limitations and the way to improve these classifiers? With IRT we can see what makes certain classifiers proficient. We can also analyse dominant regions depending on the difficulty parameter or the discrimination parameter. For instance, given a new instance, if we expect or estimate that it is going to be hard, one classifier may be preferable, while for another easy instance a different classifier may be more robust. These *classifier characteristic curves*, which we introduce here, may be a very insightful way to analyse machine learning models.

Starting with classification problems at the level of instances, we show that IRT presents itself as a well-developed theory that can be applied to machine learning and other areas of artificial intelligence (such as reinforcement learning algorithms and other search methods for planning as recently presented in [7]). Nevertheless, the application is not always straightforward and requires a careful understanding of the parameters and a right choice of IRT models, classifier populations and estimation approaches. After an increasing realisation of all these issues in [8] and [1], we now give an extended and more consolidated interpretation of the item parameters (difficulty, discrimination and guessing), its relation to instance hardness, the relevance of imbalance handling, and the unravelling of overall abilities linked to a more detailed view using classifier characteristic curves. After these first applications of IRT in AI, some recent works have taken other directions. In [9, 10], IRT has been used to select the best items from an NLP benchmark. However, the models are estimated from responses generated by humans using AMT (Amazon Mechanical Turk), and not from a set of artificial NLP methods. Of course, there is an important dilemma here about whether the parameters should be estimated from a population of human individuals or from a population of AI systems. Perhaps in some areas of AI it might make sense to take humans as a reference. However in most areas and machine learning in particular we think that current algorithms behave quite differently (and in many cases much worse or much better than humans), so we want to analyse what is hard and easy, discriminating or not, for state-of-the-art machine learning systems, as these are the systems we will ultimately use, and not humans. This is actually one of the key issues we have been analysing in previous works and especially in this one: how to select the appropriate population of machine learning techniques to get meaningful IRT parameters.

The rest of the paper is organised as follows. Section 2 discusses why a more detailed analysis of artificial intelligence results, and machine learning results in particular, is necessary and why IRT can be an appropriate tool for this. Section 3 describes the experimental methodology used in terms of classifier techniques, artificial classifiers and datasets used, as well as the particular estimation methods for the IRT models. Section 4 analyses the interpretation of the inferred instance parameters: the discrimination parameter (especially when close to zero or negative), the guessing parameter (whether it relates to the class distribution) and difficulty (whether they are at the boundaries or associated with noise). The three parameters are compared with instance hardness measures. Section 5 explores different ways of building IRT models in our context: 2PL models, 3PL models and 3PL models where classifiers are trained with the original data but IRT is applied after balancing the responses. Section 6 focuses on the inferred classifier ability, how it relates to accuracy and the effect of removing the instances with negative discrimination. It shows that plotting simple classifier characteristic curves can be insightful about the behaviour of a classifier. Finally, Section 7 discusses the findings of the previous sections and gives a global interpretation of what the IRT parameters mean and how they should be used. We enumerate a range of applications and areas of future work.

2. Artificial Intelligence and Item Response Theory

In any area of artificial intelligence, some problems are more difficult than others, and some techniques are more capable than others. But what is the relation between difficulty and ability? Is it a monotonic one, i.e., better techniques usually get better results on more difficult problems and usually solve the easier ones? Should we focus our efforts on developing or improving our techniques such that they address the more difficult and challenging problems? Or such that they are more robust with the easier, and perhaps more common, problems?

These questions are critical for the progress and evaluation of the techniques in any AI discipline [11, 12], from planning to machine translation. Of course, each discipline has a set of benchmarks and a group of state-of-the-art techniques, which are used to analyse and compare any new proposal, either as particular research papers or open competitions [13]. We can rank techniques according to their *overall* results, or even do pairwise comparisons and show that method A is better than B. The results may even say that the difference is statistically significant. However, what we seldom analyse is how the overall result for a collection of benchmark problems is distributed. Are these systems better on the most difficult problems at the cost of failing at some easy problems? Also, as the discipline progresses, new challenging problems are included and, sometimes, the easy problems are removed from the benchmark. The analysis of problem difficulty or hardness is then very relevant to understand not only whether, but how, AI methods are improving.

An area where the analysis of difficulty, or hardness, has been investigated recently is machine learning. Machine learning has a long tradition of evaluating different techniques with many problems, but the use of difficulty is not so common. In the area of meta-learning [14], it is common to analyse the features of classifiers and datasets in order to see which ones go well with a particular dataset. However, the notion of 'difficulty' of a dataset is not considered by common (aggregative) measures such as accuracy or cost [15]. There have been some recent analysis of repositories [16, 17], but the notion of difficulty is elusive in this context.

There is a more significant analysis of the difficulty or hardness of *instances*, given a dataset. In [5], Smith et al. provide an empirical definition of instance hardness based on the average behaviour of a set of diverse classifiers (e.g., the average error produced by the pool of classifiers for that instance). This has several potential applications, as mentioned above, for the detection of where different classifiers fail and how they can be improved. Somehow related we also find the area of outlier detection [18] in machine learning and data mining where outliers may represent anomalies, border points or minority classes which can be hard to classify correctly. For instance, in [19] authors provide an average path length of a given instance as a good measure of whether the instance is an outlier. However, these hardness and anomaly measures miss important information about instance difficulty as it might be the case that the instance is difficult for all classifiers homogeneously (only 10% of the classifiers get it right with no correlation to their accuracy) or is difficult especially for most but some classifiers (only 10% of the classifiers get it right but these are the

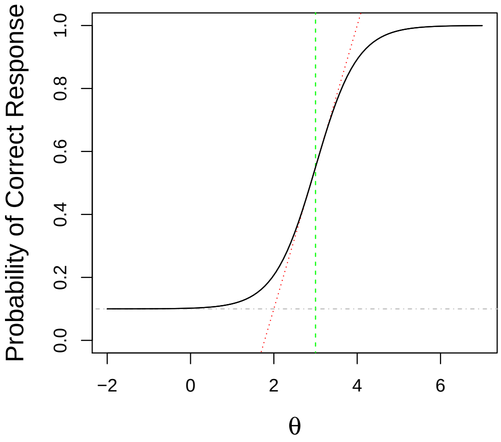

Figure 1: Example of a 3PL IRT model (in black), with slope $a = 2$ (discrimination, in red), location parameter $b = 3$ (difficulty, in green) and guessing parameter $c = 0.1$ (chance, in grey).

most competent ones for the dataset). This information is key to understand what the instances really are and how the classifiers are really behaving. Also, instance hardness alone does not say much about whether a few instances can be used to tell between good and bad classifiers, in a model selection situation.

Interestingly, all of these issues have been addressed in the past by item response theory, yet in very different contexts. Item response theory (IRT) [2, 4] considers a set of models that relate responses given to items to latent abilities of the respondents. IRT models have been mainly used in educational testing and psychometric evaluation in which examinees' ability is measured using a test with several questions (i.e., items).

In IRT, the probability of a correct response for an item is a function of the examinee's ability (the person's underlying level of the construct that is being measured) and some item's parameters. There are models developed in IRT for different kinds of responses, but we will focus on the dichotomous models. In dichotomous models the response can be either correct or incorrect. That does not mean that there are only two possible answers to a question. There might be more than two, as usually in multiple-option questionnaires.

Let U_{ij} be a binary response of a respondent j to item i, with $U_{ij} = 1$ for a correct response and $U_{ij} = 0$ otherwise. Let θ_j be the ability or proficiency of j. If ability θ equals item difficulty, there are even odds of a correct answer. However, the greater the ability is above (or below) an item difficulty, the more (or less) likely a correct response. Now, assuming that the result only depends on the ability and no longer on the particular respondent, we can express the response as a function of i alone, i.e., U_i. For the basic 3-parameter (3PL) IRT model, the probability of a correct response given the examinee's ability is modelled as a logistic function:

if ability equals difficulty b, there are even odds (1:1, so logit 0) of a correct answer, the greater the ability is above (or below) the difficulty the more (or less) likely a correct response

$$P(U_i = 1|\theta_j) = c_i + \frac{1 - c_i}{1 + exp(-a_i(\theta_j - b_i))} \tag{1}$$

The above model provides for each item its *Item Characteristic Curve (ICC)* (see Figure 1 as an example), characterised by the parameters[2]:

- Difficulty (b_i): it is the location parameter of the logistic function and can be seen as a measure of item difficulty. When $c_i = 0$, then $P(U_i = 1|b_i) = 0.5$. It is measured in the same scale of the ability;

[2] Following the convention of some IRT libraries, the three parameters will be usually abbreviated in many plots and tables as "Dffclt", "Dscrmnn" and "Gussng" respectively.

- Discrimination (a_i): it indicates the steepness of the function at the location point. For a high value, a small change in ability can result in a big change in the item response. Alternatively we can use the slope at location point, computed as $a_i(1 - c_i)/4$ to measure the discrimination value of the item;

- Guessing (c_i): it represents the probability of a correct response by a respondent with very low ability ($P(U_i = 1| - \infty) = c_i$). This is usually associated to a result given by chance.

The basic IRT model can be simplified to two parameters (e.g., assuming that $c_i = 0$), or just one parameter (assuming $c_i = 0$ and a fixed value of a_i, e.g. $a_i = 1$).

The ability of an individual is not measured in terms of the number of correct answers but it is estimated based on his/her responses to discriminating items with different levels of difficulty. Respondents who tend to correctly answer the most difficult items will be assigned to high values of ability. Difficulty items in turn are those correctly answered only by the most proficient respondents.

Straightforward methods based on maximum-likelihood estimation (MLE) can be used to estimate either the item parameters (when examinee abilities are known) or the abilities (when item parameters are known). A more difficult, but common, situation is the estimation when both the item parameters and respondent abilities are unknown. In this situation, an iterative two-step method (Birnbaum's method [20]) can be adopted:

- Step (1) Start with initial values for abilities θ_j (e.g., random values or the number of correct responses) and estimate the model parameters;

- Step (2) Adopt the estimated parameters in the previous step as known values and estimate the abilities θ_j.

In this method, item parameters and respondent abilities are simultaneously estimated only based on a set of observed responses to items, with no previous knowledge about the true ability of the respondents.

In our adaptation of IRT, an item in IRT can be identified with a problem in AI, and an individual (or subject) can be identified with an AI method, technique or system. In the case of machine learning, an item can be a dataset (the whole problem) or it can be an instance (an example in a dataset). While we think that the equating of items with datasets can be very interesting, we leave this as future work, with this paper focusing on the analysis of items as instances. Recently, there has been more understanding about how IRT should be applied to classification at the level of instances in [8, 1]. This work completes this analysis with a wide range of datasets, a comparison with instance hardness measures and the analysis of class imbalance.

3. Methodology

Our mapping of IRT to the instance-wise analysis of classification problems means that items are instances and respondents are classifiers. Hence, for a particular application scenario the analysis will focus on how a range of classifiers behave for the set of instances in a dataset. In other words, given a dataset with N examples, the procedure is as follows:

1. Generate a diverse set of M classifiers (e.g., with any machine learning library) using a training subset.
2. Evaluate those M classifiers with the examples in the test subset and get the responses (whether the class predictions are correct) for all of them. Therefore, the binary results from the classifiers are always obtained by using the test "fold" (training sets are never used to obtain the responses, but to train the classifiers).
3. If using cross-validation in the previous two steps, there will be an $N \times M$ matrix U with all binary responses $U_{i,j}$.
4. With this matrix U, use an IRT inference technique to derive the response models $P(U_i = 1|\theta)$ for all the examples as well as the abilities θ_j for all classifiers.
5. Use the model parameters and the abilities to understand the instances and how classifiers behave.

In practice, a particular study will be done for a single dataset. This will reveal many details and, as we will see, can be accompanied by several plots each. Nevertheless, in order to determine how IRT works in general, which is one of the goals of this paper, we consider several datasets.

We have used one artificial and 12 real datasets. Cassini[3] is a 3-class bivariate toy dataset composed of 200 instances with a 10% of random noise we put on purpose (see Figure 2). This artificial dataset is mostly used for illustrative purposes. The 12 real datasets are from the UCI repository [21], as shown in Table 1. For instance, Figure 5 shows one of these datasets, "Heart-statlog", a binary dataset that has 270 instances and 13 attributes containing heart disease data.

ID	Dataset	# Examples	# Attributes	# Classes	Class %
1	Echocardiogram	131	10	2	(67.2%, 32.8%)
2	Hepatitis	155	19	2	(79.4%, 20.6%)
3	Heart-statlog	270	13	2	(44.4%, 55.6%)
4	Ionosphere	351	33	2	(64.1%, 35.9%)
5	Parkinsons	195	22	2	(75.4%, 24.6%)
6	Balance-scale	625	4	3	(7.8%, 46.1%, 46.1%)
7	Energy	600	8	3	(18.2%, 46.5%, 35.3%)
8	Seeds	210	7	3	(33.3%, 33.3%, 33.3%)
9	Teaching	151	5	3	(34.4%, 33.1%, 32.5%)
10	Vertebral-column	310	6	3	(19.4%, 48.4%, 32.3%)
11	Ecoli	327	6	5	(43.7%, 23.5%, 10.7%, 6.1%, 15.9%)
12	Flags	178	29	5	(20.2%, 8.4%, 33.7%, 22.5%, 15.2%)
13	Cassini	200	2	3	(40.0%, 40.0%, 20.0%)

Table 1: Datasets used for the analysis. The first 12 are UCI datasets [21] and the last one is an artificial toy dataset.

As the datasets generate a sufficiently large number of examples (items) for applying IRT, we also need a good number of classifiers (respondents). In order to achieve this, we started with 11 different families (decision trees, rule-based methods, discriminant analysis, Bayesian, neural networks, support vector machines, bagging, random forests, nearest neighbours, partial least squares and principal component regression). All the classifiers are implemented in R. Some use a particular package while others use the classifier through the interface provided by the caret[4] package. We modified their parameters for all families in such a way that we obtained a heterogeneous set of 121 classifiers, as can be seen in Table 2. For instance, a pool of classifiers was produced by Random Forest (RF) trained with different numbers of trees. We learned and evaluated the models adopting 10-fold cross-validation. As mentioned above, this procedure generates one result per example and technique, $N \times M$ in total.

Apparently, this looks appropriate. However, when we derived the first IRT models we found that for many examples (very easy examples), all techniques were right. As a result, the models (and the curves) became horizontal or singular, and hence useless, even if they should not be horizontal and placed at very low difficulties. In order to solve this, we need to ensure that results for every instance have a minimum of diversity. Consequently, apart from the classifiers generated by machine learning techniques, we also introduced some artificial classifiers:

- Three random classifiers (*RndA*, *RndB*, *RndC*). The three use the prior class probabilities.
- Majority/minority classifier (*Maj*, *Min*), which always return the majority/minority class of the dataset.
- Two idealistic (not feasible in practice) classifiers, using the test labels: the optimal/pessimal classifier (*Opt*, *Pess*) which always succeeds/fails respectively.

Jointly with the 121 original classifiers, this totals 128 classifiers (see Table 2).

These seven artificial classifiers are also very useful as indicators (e.g., to see how calibrated difficulty and ability are), as we have a clear intuition about their abilities, and we can check where they are located

[3] Provided by the mlbench R package (see `https://cran.r-project.org/web/packages/mlbench/`).

[4] See `http://caret.r-forge.r-project.org`.

Family	ID	Technique	R Package	Tuning Parameters	#
Decision Trees	C5.0	C5.0	C50, plyr	winnow	2
	J48	J48	RWeka	unpruned	2
	LMT	Logistic Model Trees	RWeka	C, A	4
	rpart	CART	rpart		1
	ctree	Conditional Inference Tree	party	mincriterion	3
Rule-based methods	jRip	Rule-Based Classifier	RWeka	E	2
	PART	PART decision lists	rpart		1
Discriminant analysis	sda	Shrinkage Discriminant Analysis	sda	diagonal, lambda	3
	fda	Flexible Discriminant Analysis	earth, mda	degree, nprune	3
	mda	Mixture Discriminant Analysis	mda	subclasses	3
Bayesian	NB	Naive Bayes	RWeka		1
	naiveBayes	Naive Bayes	e1071	laplace	1
Neural Networks	rbf	Radial Basis Function Network	RSNNS	negativeThreshold	1
	mlp	Multi-Layer Perceptron	RSNNS	size	5
	avNNet	Model Averaged NN	nnet	size, decay, bag	3
	pcaNNet	NN with Feature Extraction	nnet	size, decay	1
	lvq	Learning Vector Quantization	class	size, k	3
Support Vector Machines	SMO	Sequential Minimal Optimization	RWeka		1
	svmRadialCost	SVM (RBF Kernel)	kernlab	C	4
	svmLinear	SVM (Linear Kernel)	kernlab	C	6
	svmPoly	SVM (Polynomial Kernel)	kernlab	degree, scale, C	9
	gbm	Stochastic Gradient Boosting	gbm, plyr	n.trees, interaction.depth, shrinkage, n.minobsinnode	9
Bagging	treebag	Bagged CART	ipred, plyr, e1071		1
	bagFDA	Bagged Flexible Disc. Analysis	earth, mda	degree, nprune	4
Random Forests	rf	Random Forest	randomForest	mtry	7
	RRF	Regularized RF	randomForest, RRF	mtry	7
	cforest	Conditional Inference RF	party	mtry	7
	parRF	Parallel Random Forest	e1071, foreach, randomForest	mtry	7
Nearest neighbor methods	knn	k-Nearest Neighbors		k	6
	IBk	k-Nearest Neighbors	RWeka	K	6
Partial least squares	pls	Partial Least Squares	pls	ncomp	2
	simpls	Partial Least Squares	pls	ncomp	2
Principal component regression	gcvEarth	Multivariate Adapt. Reg. Splines	earth	degree	3
Base Lines	OptimalClass	Optimal Classifier			1
	PessimalClass	Pessimal Classifier			1
	MajorityClass	Majority Classifier			1
	MinorityClass	Minority Classifier			1
	RandomClass	Random Classifier			3

Table 2: Pool of classifiers used: families, techniques, tuned parameters and number of models per variant to obtain 128 classifiers.

according to IRT models. Furthermore, the use of these classifiers helps with the interpretability of easy instances (avoiding flat curves). Ignoring these instances in a first analysis would not show us what happens with these non-discriminating cases (the models give an undefined value for the discrimination parameter). In the end, when we are given new examples or new datasets we can never know in advance if an example can get systematically right answers from all classifiers. By adding these special classifiers we are certain that we do not need to make a two-stage procedure (estimate, clean the non-discriminating cases, and estimate again).We have to take into account that when we add the special classifiers, some metrics of fit (e.g., the item or model fit statistics[5]) get corrupted by divisions by zeros (because of the optimal and pessimal classifiers are right and wrong respectively, all the time). As a result, we will not use the item-fit statistics as the quality of the estimation. Instead we can look at the correlations. Note that IRT is expected to produce very different results from what we get if we calculate the percentage of examples that are misclassified (e.g., accuracy) or the percentage of classifiers that are wrong with an example. What we just check is whether the correlations between abilities and performance, and between difficulty and the percentage of classifiers that are wrong with an example, are positive.

As we have seen, there are logistic IRT models with one, two and three parameters. Also, they can be

[5] Fit statistics determine whether an IRT model fits the data well enough to use the examinee estimates.

estimated with several libraries. In this paper, we use some publicly IRT packages in R [6]. In particular, we do most of the experiments using 3PL models and an MLE approach to estimate the models' parameters of all instances and the classifiers' ability simultaneously, as usual in IRT.

We use the `ltm` [22] R package, which implements the previously mentioned Birnbaum's method, to be consistent with our previous work [1, 8]. The `MIRT` R package [23] has also been used to check whether *(a)* the estimated parameters are consistent, and *(b)* both methods obtain similar maximum likelihood locations. It should also be noted that the IRT estimation methods have, in general, some requirements and limitations in terms of minimum sample size and maximum number of items in order to obtain stable item parameters. In general, samples of at least 20-30 examinees (classifiers) are required for dichotomical models. Nevertheless, many IRT libraries (including `ltm`) output indicators of the goodness of fit, which can be used as a criterion to know whether the number of techniques or examples might be insufficient. Hence, occasionally, if the results from `ltm` are unstable (according to fit/goodness measures) we use the results from the `MIRT` package instead, which has shown to be more robust.

Furthermore, these and other common methods used in IRT can be trapped in local minima or may not converge, with zero estimates for some guessing parameters or a non-positive definite Hessian matrix at convergence. This seems to be a methodological issue since other well-known IRT packages (using maximum likelihood to estimate the IRT parameters) have similar limitations (mostly when it comes to datasets with more than a thousand items [24]). Further tuning of some parameters, such as the starting values, the parameter scaling vector and the optimiser, may lead to successful convergence. In any case, it is important to highlight that IRT is applied to the test split, not to the training split. For very large datasets, with millions of examples, one could train with a large proportion of the examples and test with a very small part ($<$1000 instances), and still get a good estimation for them. This could be done as part of the cross-validation process and ultimately get the difficulties for all instances (and average the abilities). In all cases, we derive the 3 model parameters that characterise an instance: difficulty, guessing parameter and discrimination power, as we will analyse in the following section[7]. The ability of a classifier is also estimated at the same time. We will study the ability parameter in Section 6.

4. Instance parameters: discrimination, guessing and difficulty

The 3-parameter logistic model is composed of three parameters per instance: difficulty, discrimination and guessing. In order to understand these parameters, we can use the toy Cassini dataset first (see Figure 2). In this case, 200 IRT models were derived (one per instance) and 128 values of ability for the set of classifiers were estimated. The parameters for six items are shown in Table 3. The examples of item characteristic curves[8] (ICCs) are presented in Figure 3. What we see is that some instances are more difficult than others: instance "2" has difficulty 0.96 while instance "3" has difficulty -2.1. The slopes (discrimination parameter) for instances "1" to "4" are positive but with very different difficulties. We also see that the guessing parameters are very different, ranging from 0.67 for item "2" to 0 for many others.

Let us explore these parameters in more detail.

4.1. The difficulty parameter

The item parameter that seems easiest to understand is difficulty. Intuitively, easy items are solved by almost all classifiers, and difficulty items are those that are only solved by very able classifiers. This is clear in the examples in Figure 3, items "1" and "3" are easier than "2" and "4". Note that difficulties go unbounded from $-\infty$ to ∞. However, depending on the assumptions made for the estimation of the parameters, the *magnitude* of difficulty can be more or less meaningful. If the particular IRT technique

[6] An overview of all IRT packages is given in `https://cran.r-project.org/web/views/Psychometrics.html`.

[7] The whole list of classifiers parameters as well as the data used in the experiments, plots, configuration files and the full code can be downloaded from `https://github.com/nandomp/IRT4ML`.

[8] All ICCs for both the Cassini and the UCI datasets can be analysed in our Shiny application *IRT4ML* in `https://nandomp.shinyapps.io/IRT4ML/`. This web application has been developed to help readers understand and analyse visually the IRT parameters for each dataset and experiment.

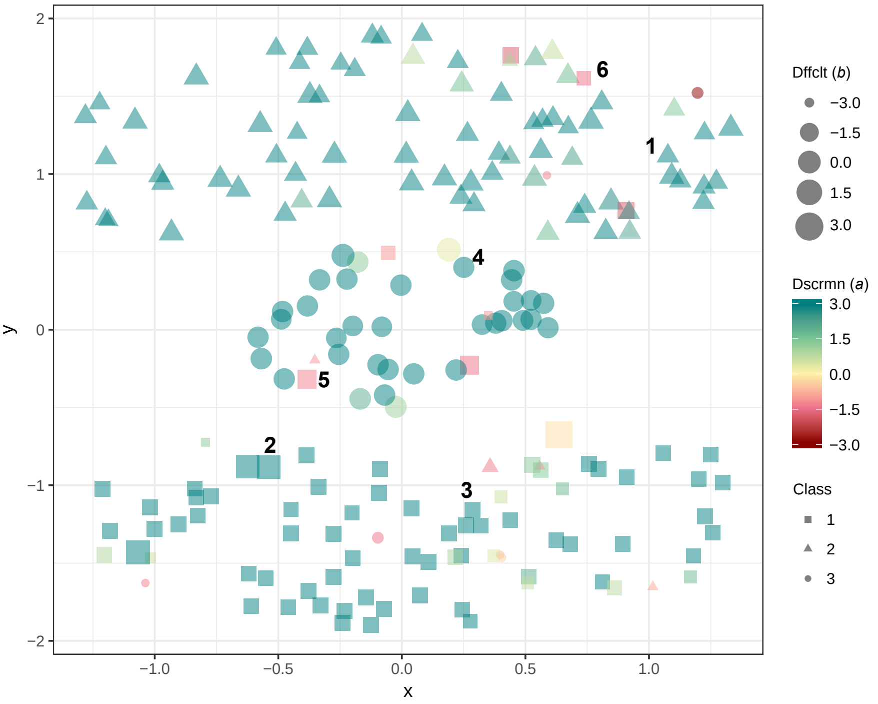

Figure 2: Visualisation of the Cassini toy dataset. Different shapes represent different classes. Those instances with positive slope are represented with different shades of blue to yellow, while negative slope are represented with different shades of orange to red. ICCs of those instances labelled with a number are shown in Figures 3 and 4.

Item	Gussng (c)	Dffclt (b)	Dscrmn (a)
1	0.0000013	-1.780543467	3.0209284
2	0.6736535	0.962307474	4.3968907
3	0.0000005	-2.131842918	5.7883337
4	0.0550886	-0.131519369	4.7300441
5	0.0000000	-1.225138876	-1.3837845
6	0.0000000	-2.393864040	-1.5230734

Table 3: ICC parameters for the plots in Figures 3 and 4.

assumes that abilities are distributed normally and centred at 0, we expect difficulties to be around 0 as well, because difficulty and abilities are in the same units and subtracting in the 3PL model, see Eq. 1 —although difficulties may not follow a normal distribution. The best understanding is that, according to Eq. 1, we need an ability of b_i to have a probability of correct response (classification) of $\frac{c_i+1}{2}$.

But if we look at Table 3 we see that item "6" is very easy, while apparently it is just noise (a "square" example at one far edge of the an area dominated by triangles). Is this example difficult or is it just noise (and hence the notion of difficulty should not be applied)? In order to answer the question we have to look at several general questions about instance difficulty. What happens with items that are not solved by any classifier? Are they extremely difficult or just noise (and hence solving them would be overfitting and hence bad)? And what if there is no relation between classifier ability and success for a particular instance, or this relation is negative?

In order to fully understand the difficulty parameter, we need to look at the discrimination parameter a_i and also the baseline guessing parameter c_i.

4.2. The discrimination parameter

The discrimination parameter (slope) is a measure of the capability of an item to differentiate between individuals (classifiers). Therefore, when applying IRT to evaluate classifiers, the slope of an instance can

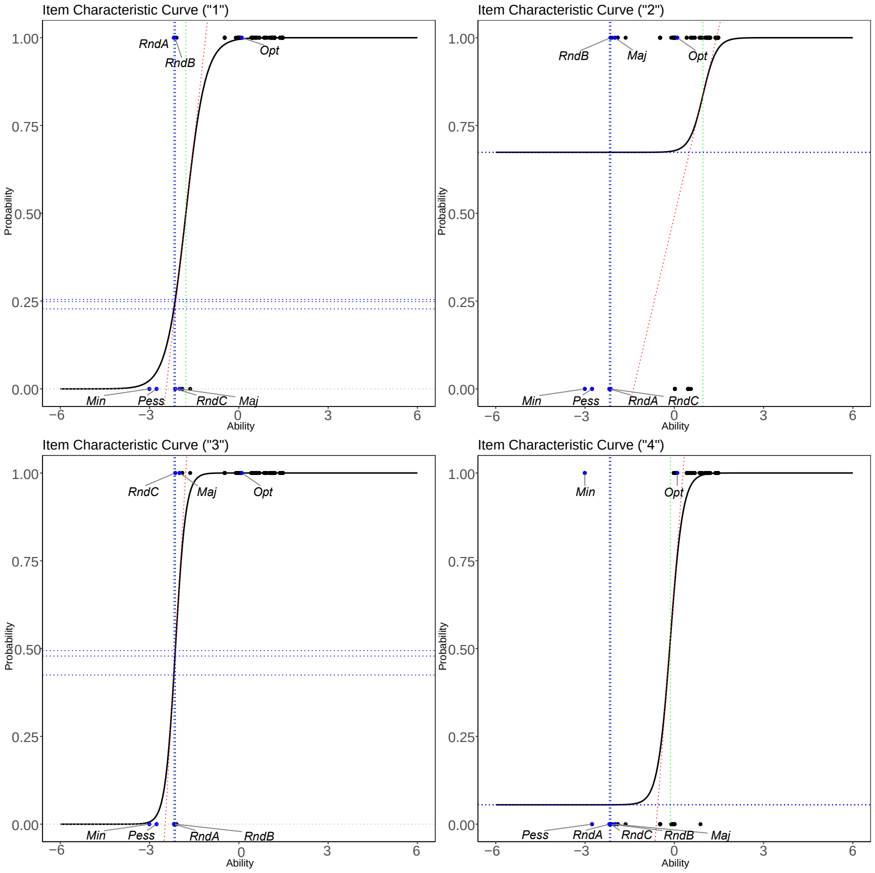

Figure 3: Examples of ICCs (with positive slope) of the points labelled in Figure 2 from "1" to "4". Classifier are also shown around the ICCs as dots, placed at the top of the plot ($y = 1$) if the classifier is correct for the instance, and at the bottom ($y = 0$) otherwise, with the x value of the dot being the ability of the classifier. Artificial classifiers are named. The probability of correct response for the ability values of the three random classifiers are plotted with blue dotted lines (cut points) usually appearing together or very close.

be used to indicate if the instance is useful to distinguish between strong or weak classifiers for a problem.

From the 200 instances in Cassini, 180 had positive slopes (i.e., positive discrimination values), matching the common assumption of IRT and the nice ICCs in Figure 3. In these cases, the probability of correct responses is positively related to the estimated ability of the classifiers. But negative discrimination values were observed for 20 instances. We can identify them in Figure 2 as those with an yellowish or reddish colour. Figure 4 shows two ICCs examples: cases "5" and "6". Since the discrimination is negative, this means that these instances are most frequently well classified by the weakest classifiers. These cases are anomalous in IRT (usually referred to as "abstruse" or "idiosyncratic" items). But in the context of machine learning, these are precisely the instances that may be most useful to identify particular situations. For example, if two instances 1 and 2 in a binary classification problem have exactly the same features but belong to different classes, then $P(U_{1j} = 1|\Theta_j) = 1 - P(U_{2j} = 1|\Theta_j)$. In this situation, one of the instances may have been wrongly labelled, which can result in a negative-slope ICC. Focusing on the Cassini dataset, all noisy instances put on purpose have negative slope (plus a very few others located on the boundaries between classes).

The same applies when using the Heart-statlog dataset (Figure 5), but in this case, where no noisy instances are introduced on purpose (but there might be noise originally), negative slopes usually appear for instances that are in regions of the instance space dominated by the other classes. Note that the visualisation by only two principal components adds some distortion here. Still, these mislabelled examples, for the purpose of evaluation, have to be considered ground truth and hence they usually have medium-high

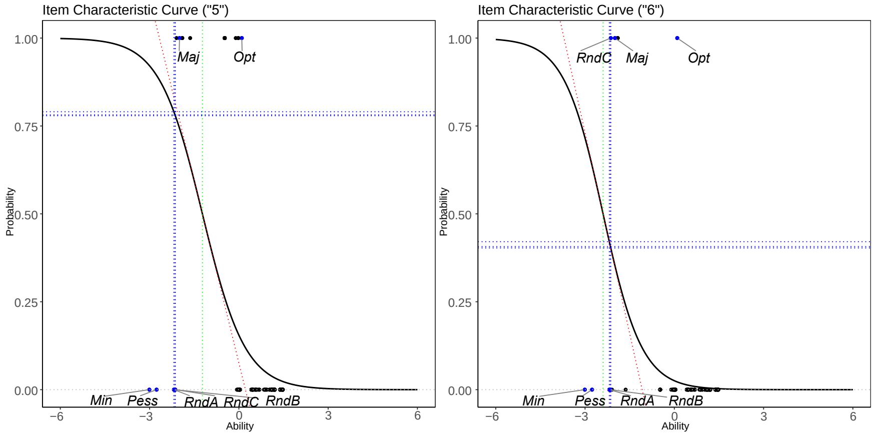

Figure 4: Examples of ICCs (negative slope) of the points labelled in Figure 2 as "5" and "6". Classifier abilities are plotted at $y = 1$ if the classifier is correct, otherwise, at $y = 0$. Artificial classifiers are named.

difficulty levels, as Figure 5 shows. Nonetheless, the negative discrimination is really giving us this extra dimension to determine when an instance is actually well aligned with ability (1, with positive discrimination, and only the good classifiers can solve it); when it has a high misclassification ratio happening with good and bad classifiers equally (2, discrimination close to 0), or when it is not aligned with ability (3, negative discrimination, with more good classifiers misclassifying it than bad classifiers). These three cases do not correspond to borderline, overfitted or outlying instances necessarily. There is no sufficiency correspondence either. For instance, the only clear outlier in Figure 5 seems to be located at the top left area of the plot, with negative discrimination and low difficulty, but other examples with similar parameters exist in other areas. The parameters explain the role of the examples when evaluating different classifiers. For instance, according to IRT, the more instances we have of type 1 the more robust the evaluation seems to be.

By looking at the discrimination parameter for several instances, we now clearly see that difficulty alone is insufficient to understand what is going on with an instance, and that the discrimination parameter, especially when negative, can highlight the key instances in a dataset. We will analyse this further jointly with the difficulty parameter at the end of this section. But we look at the guessing parameter first.

4.3. The guessing parameter

In IRT, the pseudo-guess (or guessing) parameter (characterising the lower asymptote of the ICCs) tells us how likely the examinees are to obtain the correct answer by (random) guessing. Namely, even if the examinee does not know anything about the matter (has an ability equal to $-\infty$), he or she can still have some chance to succeed. For instance, on a multiple choice testing item with four possible answers, the guessing parameter should be 0.25.

However, we now find that, when applying IRT in machine learning, the guessing parameter has nothing to do with the original meaning for psychometrics. Following the above definition, our intuition would tell us that the guessing parameter should be equal to one divided by the number of classes. But we see it is not the case when evaluating classifiers with datasets. An illustrative example of this can be seen again if we go back to Figure 3, which plots some examples of ICCs for the Cassini dataset (3 classes). We see that the lower asymptotes of the ICCs take different values which, although helping the logistic model to be more flexible, are very different from what one would expect for this dataset (which would depend on the class distribution). If we plot the probability of correct response for the ability values of the three random classifiers as a cut point (dashed blue lines) in Figure 3, we get values around 0.25, 0.67, 0.45 and 0.06, which in some cases differ very much with respect to the guessing parameters, in this case 0.0, 0.67, 0.0 and 0.06. From the rest of experiments with UCI datasets used we concluded exactly the same —initially surprising— fact.

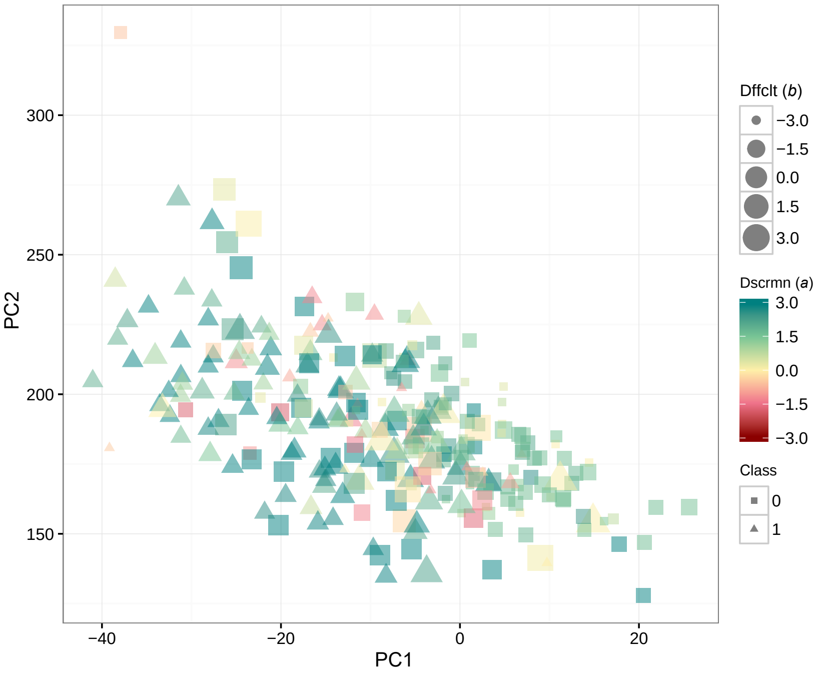

Figure 5: Visualisation of the Heart dataset using the first two principal components. Different shapes represent different classes. Those instances with positive slope are represented with different shades of blue to yellow, while negative slope are represented with different shades of orange to red.

Instead, we can compute the average conditional probability of success of all classifiers of a given ability, denoted by θ, for all instances, i.e.,

$$pSuccess(\theta) = \frac{\sum p_i(U_i = 1|\theta)}{N} \tag{2}$$

where N is the number of instances.

And now, interestingly, if we take the abilities θ of the three random classifiers for Cassini, which are -2.2, -2.1 and -2.1, we get the values of $pSuccess(\theta)$ equal to 0.35, 0.36 and 0.36 respectively. As the class proportions for this dataset are 0.4, 0.4 and 0.2 and the random classifiers use the prior distribution, we have $0.4^2 + 0.4^2 + 0.2^2 = 0.36$ as expected accuracy, which explains these values.

As a conclusion, the guessing parameter has to be interpreted as an extra degree of freedom to fit the logistic models, but not linked to the class distribution. Interestingly, as we have introduced the pessimal classifier, it is even clearer that linking the guessing parameter to the number of classes or their distribution does not make sense, as there can be models, at least in theory (e.g., the pessimal classifier), that have 0 accuracy even for two classes.

This means that if we look at instance “2” on Figure 3, we see that a better fit is performed against the intuition that the worst possible classifier should have a probability of 0 of getting the instance right, and not about 0.67 as we see on the y-axis. This suggests that 2PL models, only using difficulty and discrimination, might be more intuitive, despite having less flexibility. Overall, however, the number of examples with a guessing parameter that is not close to zero is not very high. In what follows, we still use the 3PL models because we want to further analyse whether the guessing parameter might be indicative of some special examples.

4.4. IRT parameters and instance hardness measures

In the previous subsections, we have seen that some instances, the most interesting ones (noisy ones and those on the boundaries) could be characterised by looking at difficulty and discrimination (and to a lesser extent, perhaps, guessing). This suggests that one single factor is insufficient to characterise items properly. Also, it is important to remind that in IRT the value of difficulty is not equal to 1 minus the proportion of classifiers that predict the example correctly, a common value for population-dependent instance hardness:

$$IH_i = 1 - \frac{1}{M} \sum_j U_{ij} \tag{3}$$

This leads us to a tradition of methods estimating difficulty or hardness by other (simpler) methods. Smith et al. [5] perform a very comprehensive analysis of instance hardness metrics. Table 4 includes a range of instance hardness measures. They are based on different factors that are thought to influence hardness, such as how well surrounded they are of instances of the same class, overlaps or complexities of boundaries, information required to classify the example, likelihood and class imbalance issues[9]. Also, in [5] they compared these measures with the results of several machine learning methods individually and in aggregation, by calculating one minus the average probability of correctly classifying the instance for a set of classifiers. This populational measure is what they call the instance hardness using the indicator function, denoted by them by IH_{ind}, which is Eq. 3 on expectation (they also introduce the same function but using the predicted class scores or probabilities for the true class, i.e., as soft classifiers).

Abbr.	**+/-**	**Measure**	**Description**
kDN	$+$	k-Disagreeing Neighbours	Overlap of an instance using all of the data set features on a subset of the instances
DS	$-$	Disjunct Size	Complexity of the decision boundary for an instance.
DCP	$-$	Disjunct Class Percentage	Overlap of an instance using a subset of the features and a subset of the instances.
TD_U	$+$	Tree Depth (unpruned tree)	The description length of an instance in an induced C4.5 decision tree without pruning.
TD_P	$+$	Tree Depth (pruned tree)	The description length of an instance in an induced C4.5 decision tree with pruning.
CL	$-$	Class Likelihood	Overlap of an instance using all of the features and all of the instances.
CLD	$-$	Class Likelihood Difference	Relative overlap of an instance using all of the features and all of the instances.
MV	$+$	Minority Value	Class skew.
CB	$-$	Class balance	Class skew.

Table 4: Instance hardness measures from [5].

Smith et al. [5] compare the population-dependent IH with all the population-independent hardness measures in Table 4. In order to do the comparison they use Spearman correlation, which is reasonable since the measures of instance hardness have different magnitudes and we are interested in monotonic (but possibly non-linear) relations. They show that kDN is the measure with highest (absolute) correlation with IH (0.830) and TD_P is the one with lowest (absolute) correlation (0.324).

We are here interested in how the three IRT parameters relate to the populational instance hardness value IH and the non-populational measures. In order to do this we calculate all the instance hardness measures in Table 4 for the 12 datasets in Table 1. We also calculate the IH values for all the instances in these datasets by averaging the classification mistakes of the 128 classifiers described in Section 3 (as we use a uniform weighting for all classifiers this is the same as one minus the proportion of classifiers that predict the example correctly, as for Eq. 3). The IRT parameters are also estimated as also explained in Section 3. The Spearman correlation matrices are shown in Figure 6. For those measures for which higher magnitudes correspond to less hardness, we have changed the sign of the measure (those shown with $+/-$ in Table 4), so that the direction of the magnitude is well-aligned for all measures. This makes the analysis easier to see, especially as we are using green colours for positive correlations and red colours for negative correlations.

We first look at the IRT instance parameters. We see that discrimination and difficulty usually have a moderate positive correlation (from 0.65 for Echocardiogram, except for Hepatitis, with -0.06). The

[9] We found that two measures, MV and CB, are exactly the same for all datasets and combinations. This is correct and hence makes one of them redundant. We have kept both for completeness and to make the comparison with [5] easier.

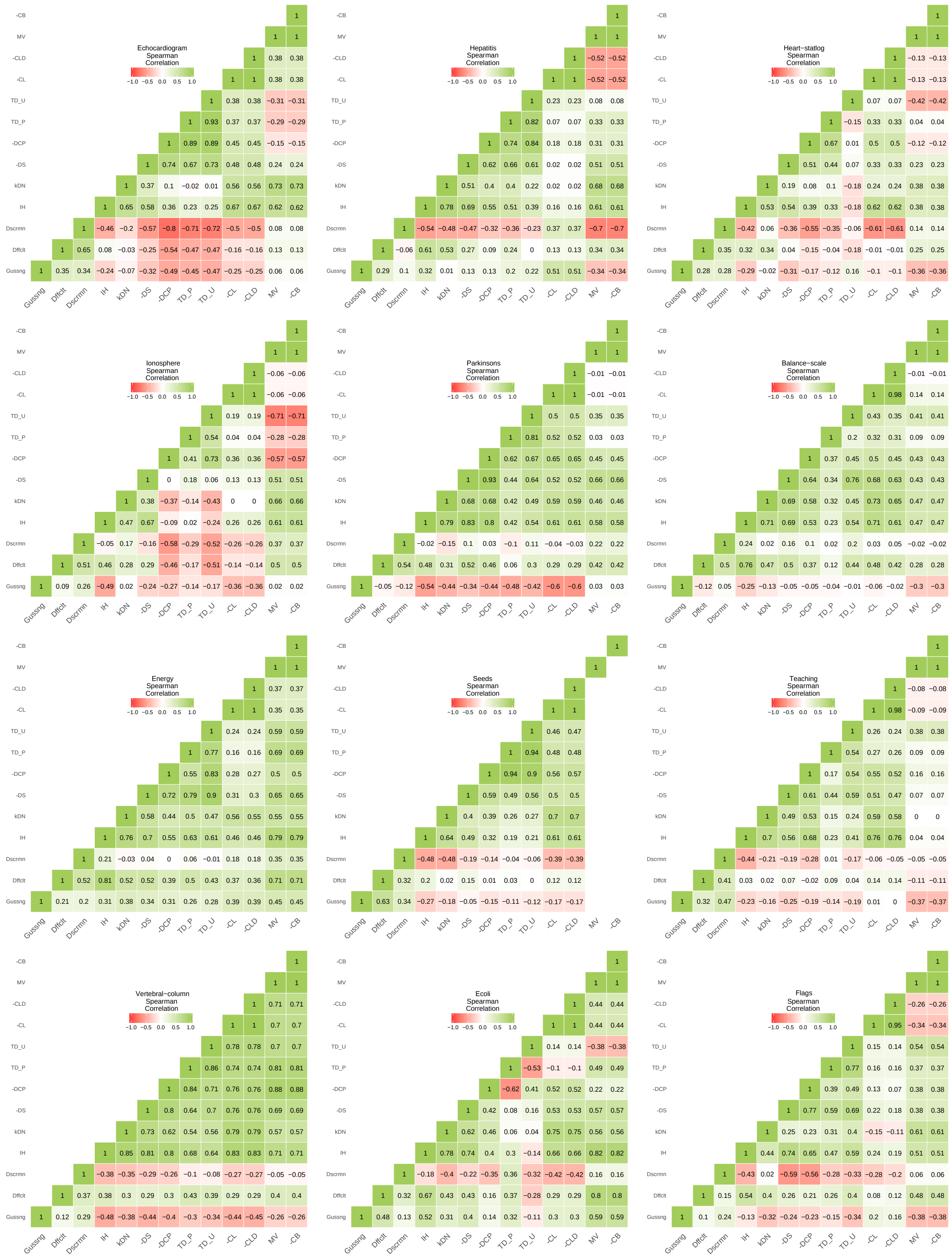

Figure 6: Correlations between population-dependent IRT parameters and IH values and population-independent instance hardness measures for the 12 datasets in Table 1. Some cells are empty because all the values are equal for all instances. Discrimination (a) is shortened as 'Discrmn', difficulty (b) is shortened as 'Dffclt' and guessing (c) is shortened as 'Gussng'.

correlation between discrimination and guessing is usually small. Finally, the correlation between difficulty and guessing is small too, except for Seeds (0.63) and Ecoli (0.48). This suggests that the only two parameters for which there is a very important interaction is discrimination and difficulty, but perhaps not sufficient as to conflate both into a single hardness measure.

If we look at IH and compare it with the other populational-based parameters, we see that IH has a very different range of correlations with guessing. However, when we look at discrimination and IH we find that most datasets have a negative correlation. On the contrary, these correlations are always positive between IH and difficulty (for some datasets as high as 0.81), which was expected, as hardness and difficulty were designed to account for similar concepts. However, the overall picture suggests that even if IH is more aligned with difficulty, it still conflates what discrimination and difficulty are separating.

We now compare these three indicators with the populational-independent instance hardness measures. The instance hardness measures against discrimination have a rather independent behaviour, only a few datasets (Echocardiogram, Hepatitis) show a strong (negative) correlation with some of of the measures. Difficulty, which we would expect to positively correlate strongly, has some moderate positive correlations with some of the instance hardness measures, but very small or even negative for others, although this varies from dataset to dataset. For instance, for kDN we find 0.53, 0.47, 0.52 for the datasets Hepatitis, Balance-scale and Energy, but very low positive correlations for others. For MV (and CB) we find 0.5, 0.71, 0.8, 0.48 for Ionosphere, Energy, Ecoli and Flags, but very low positive correlations for others. Finally, the guessing parameter seems generally independent of instance hardness measures (only Ecoli gives a surprising MV and CB correlation of 0.59).

In summary, IRT gives some information that is not comparable or reducible to any of the measures in the literature. IH can actually disguise more than one component, which appears as discrimination and difficulty here. Also, difficulty could sometimes be more strongly related to a notion of locality (kDN) but sometimes heavily affected by class balance. The imbalance problem issue will be revisited in the following sections.

5. Choosing the right IRT configuration

Now that we have a better understanding of the IRT item parameters, their interaction and, most especially, their distinctive character with respect to other previous measures, we can analyse in more depth the IRT models for the 12 real datasets we have chosen for our study. We are focusing on two questions we raised in the previous section about the best model configuration.

The first question is about the guessing parameter. We understood that this parameter could not capture the prior class distribution and was simply an extra parameter that gave more flexibility to the models. Consequently, there was a dilemma: using a 2PL model assumed a 0 guessing parameter (allowing models to be wrong for all instances), which was also simpler, but a 3PL model was more flexible and could allow for a better fit. In order to resolve this dilemma, we are going to analyse the IRT parameters for these two configurations (2PL and 3PL models) for the 12 datasets.

The second question is also motivated by the prior class distribution. In a classification problem, some classes are more frequent than others. However, in IRT, we do not expect some responses to be more frequent than others. In principle, in a questionnaire, if students knew that answer “a” is, for instance, much more frequent than answer “b”, this would automatically make students favour answers “a” or even choose “a” systematically. As a result, items whose solution is “b” would be more difficult, as almost no student would take the risk of answering them correctly. This phenomenon is not covered by the guessing parameter in IRT either. This suggests that in order to get rid of this phenomenon, we could apply the IRT estimation process to a balanced version of the dataset. More precisely, the classifiers are learnt with the original data distribution, but the IRT parameters are estimated with balanced classes. This is achieved by oversampling the minority classes. Note that this is performed during cross-validation and does not affect the classifiers, it just makes IRT inference methods pay the same attention to examples of all classes.

So now we have three different configurations: imbalanced (original) 2PL models, imbalanced (original) 3PL models and balanced 3PL models. Figures 7, 8 and 9 show the 12 datasets. The left column shows the difficulty and discrimination parameters with the 2PL model, the middle column shows them when obtained with a 3PL model and the right column shows them with a 3PL model where classes are balanced for IRT estimation, as explained above.

If we compare the two first columns for the 12 datasets we see that 2PL models usually generate more negative discriminations. For instance, Hepatitis, Ionosphere, Parkinsons and Teaching are full of examples with negative discrimination with the 2PL models. Actually, in Hepatitis, Ionosphere and Parkinsons, all the examples of the minority class have negative discrimination. This suggests that some good classifiers are not able to get good results for the minority class and this is interpreted by the estimation procedures as that these examples should have negative discrimination. This would make the analysis mostly useless. On the contrary, by including the guessing parameter the percentage of examples with negative discrimination is reduced very significantly. The extra parameter accommodates these cases in a different way, by moving up the curves and, in the large majority of cases, inverting the slope. The only case where the situation is not fully solved is Hepatitis, where the negative discriminations are less steep but still covering the whole minority class.

If we look at the difficulties, even if we are showing only two principal components and the boundaries are not the actual boundaries between classes, we see two things. First, the difficulties are slightly smaller in general for the 3PL models with respect to the 2PL models, which is consistent with the addition of a baseline for the guessing parameters. Second, and more importantly, we see that the highest difficulties are usually around boundary areas and minority classes.

We want to see whether the balanced version of the 3PL model shown in the third column is able to address some of the remaining issues. Effectively, we see that most negative discriminations also disappear for Hepatitis. About difficulty, we do not see a significant change. Note that the classifiers are still trained with the original data distribution. Consequently, they still pay more attention to the majority classes and neglect the minority classes. And this is still seen by IRT, but no longer confused with an abundance of negative discrimination parameters.

Finally, there is another good reason why a 3PL model and its balancing could be advantageous over the 2PL model. Here we pay attention to the datasets that had many negative discrimination values: Ionosphere, Hepatitis and Parkinsons. Figure 10 shows that for these three datasets the relation between ability and accuracy is counterintuitive for the 2PL model. The large number of negative discrimination examples spoils everything. Things improve significantly for Ionosphere and Parkinsons when we move to the 3PL models. We now see a monotonically increasing relation between ability and accuracy. However, for Hepatitis, the relation is still inappropriate. This last case is solved by the 3PL model with balancing, as we see in the right column. Also, the right column usually places the extreme classifiers (Optimal, Pessimal, Majority and Minority) at more reasonable locations.

Overall, we see that for some datasets the model configuration can be important. Our recommendation for a single dataset just follows what we have seen in the previous figures. If the 2PL model gives a large number of negative discrimination instances, we should try with a 3PL model. If we still find the phenomenon and the classes are imbalanced, we can try a 3PL model where the test examples are balanced on purpose. This also affects the only classifier parameter, ability, as we have seen in Figure 10. We have not paid much attention to this parameter yet. The next section fully explores its meaning and relevance.

6. Classifier analysis: ability

As we mentioned in the introduction, IRT has a dual character in the way that we get information about the items (instances) but also about the respondents (classifiers). IRT estimates a value of ability θ for each classifier. How is this indicator interpreted? This is what we see next.

6.1. Estimated abilities and actual classifier performance

At the end of the previous section we saw that ability and accuracy are positively related but not completely linearly. Of course, complete linearity is not necessarily what we want or expect, as IRT weights examples differently. For instance, if a classifier can solve very difficult examples but fails on some easy ones, IRT can consider that the former have more value than the latter, depending on their item parameters.

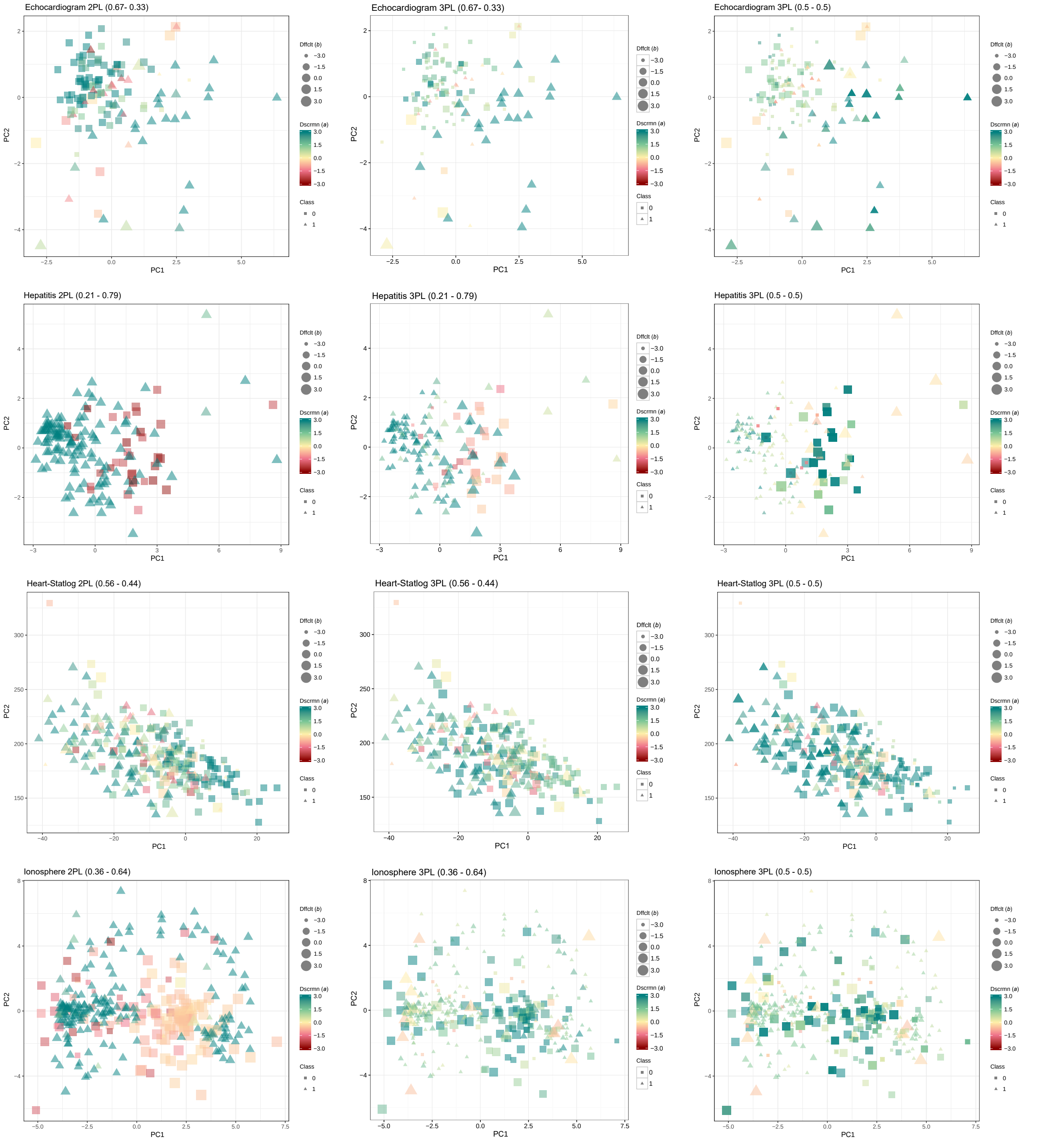

Figure 7: Visualisation of the datasets Echocardiogram, Hepatitis, Heart-Statlog and Ionosphere using the first two principal components. Difficulty and discrimination are shown with size and colours respectively. The left column shows a 2PL model, the middle column shows a 3PL model and the right column shows a 3PL model where classes are balanced for IRT estimation.

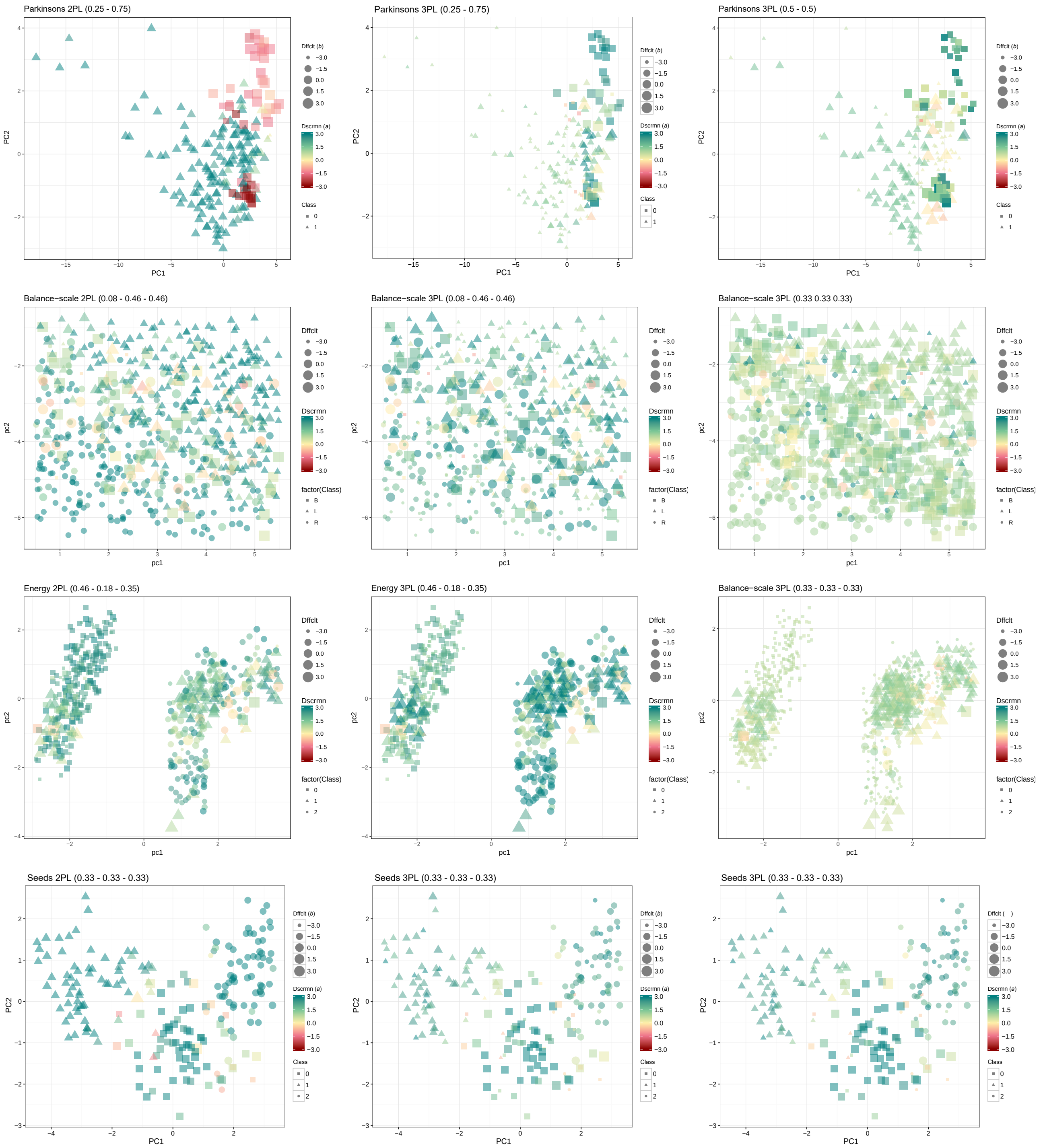

Figure 8: Visualisation of the datasets Parkinsons, Balance, Energy and Seeds using the first two principal components. Difficulty and discrimination are shown with size and colours respectively. The left column shows a 2PL model, the middle column shows a 3PL model and the right column shows a 3PL model where classes are balanced for IRT estimation. Note: Some jitter has been added in the plots of “Balance-scale” and “Energy” for a clearer visualisation.

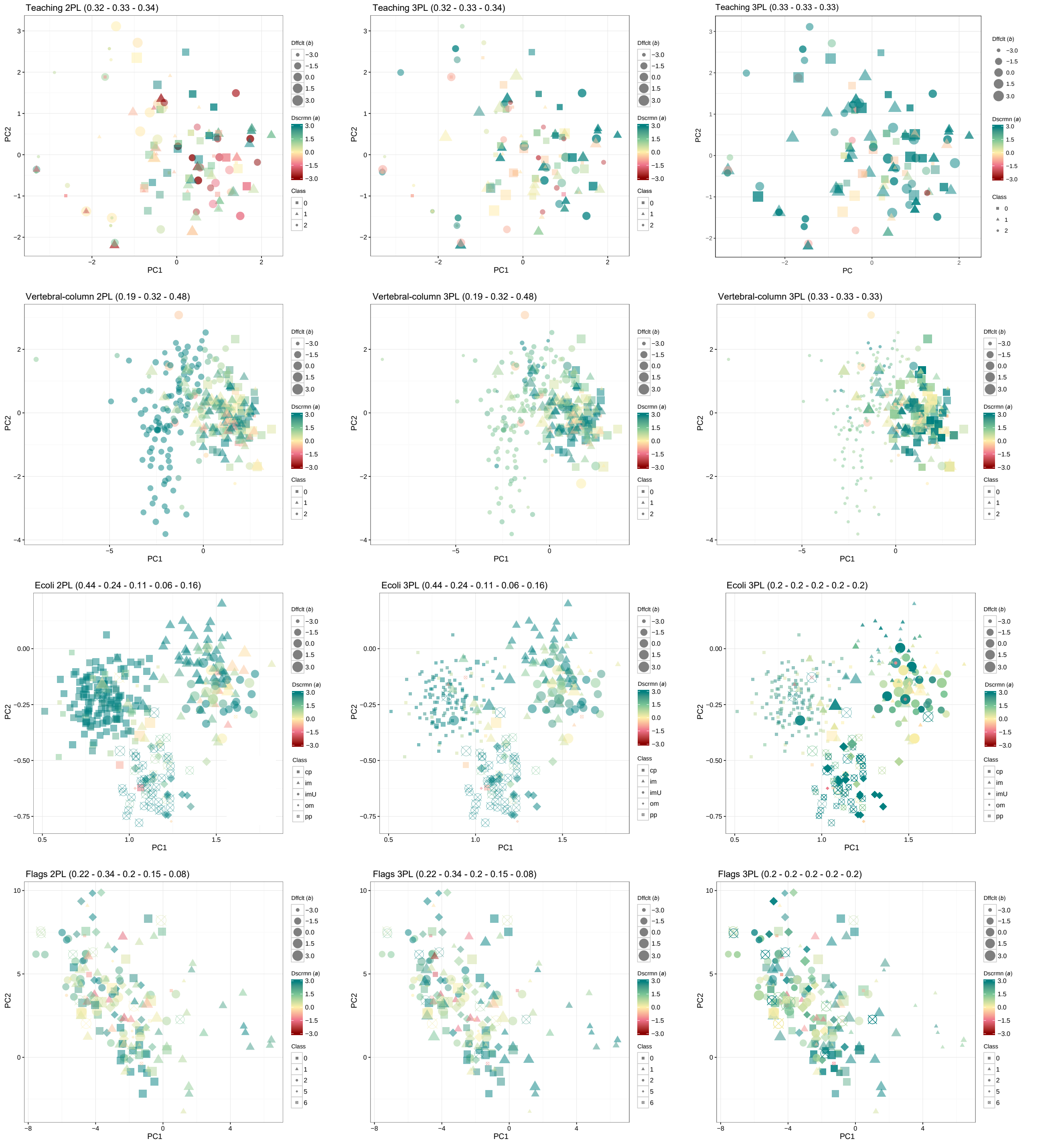

Figure 9: Visualisation of the datasets Teaching, Vertebral, Ecoli and Flags using the first two principal components. Difficulty and discrimination are shown with size and colours respectively. The left column shows a 2PL model, the middle column shows a 3PL model and the right column shows a 3PL model where classes are balanced for IRT estimation.

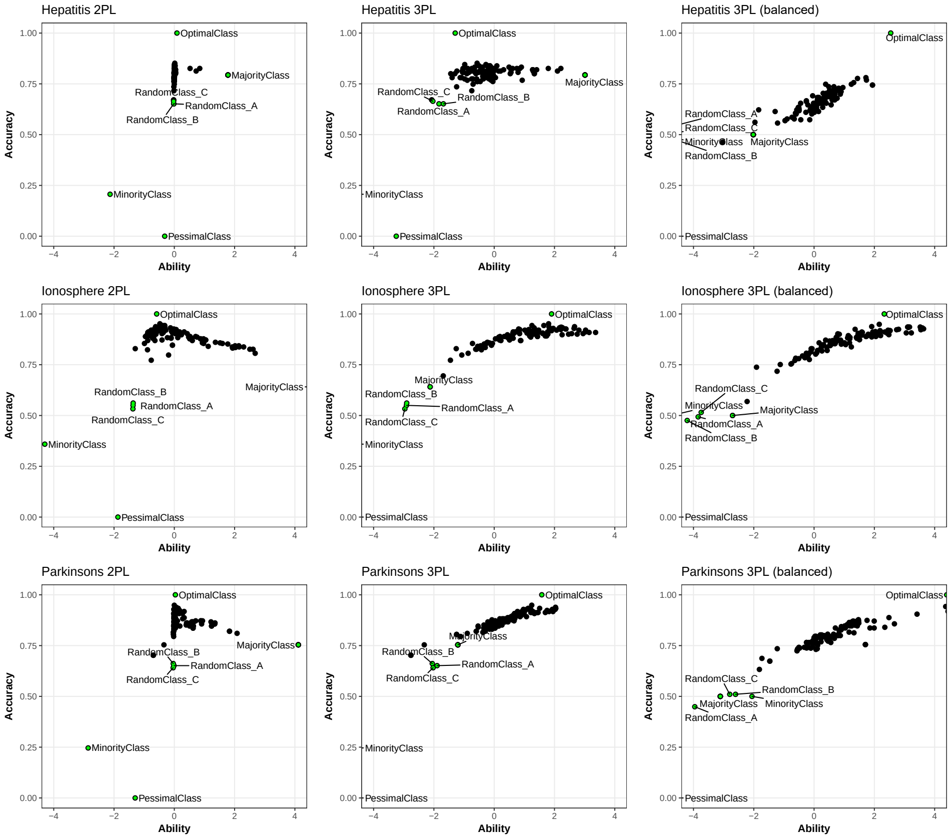

Figure 10: Scatter plot showing the relationship between the ability parameter θ and the classifier accuracy for Hepatitis, Ionosphere and Parkinsons. The left column shows a 2PL model, the middle column shows a 3PL model and the right column shows a 3PL model where classes are balanced for IRT estimation.

In order to better understand ability, Figures 11 (leftmost), 12 (leftmost) and 13 (leftmost) show the estimated abilities of all classifiers using 3PL models (without balancing) for all the datasets in Table 1 against accuracy. The second leftmost plots in the same figures show the average probability of success $pSuccess(\theta_c)$ given the ability of the classifier as we introduced in Eq. 2 against accuracy. In both cases, we see a strong relation, as expected, i.e., able classifiers have higher accuracy. It seems that the correlation is more linear in the case of $pSuccess(\theta_c)$, but basically the two leftmost plots in these figures portray a similar picture.

The interesting bit comes when we look at the extreme classifiers, such as Pessimal and Optimal. We should expect that they had the worst and best estimated abilities respectively, but this is not what we see. Actually, there are many classifiers with higher ability than Optimal.

As we already saw in the previous section, this is related to how many instances there are with a negative value of the discrimination parameter, because these greatly affect the estimation of the ability parameter of the classifiers. By using a 3PL model and balancing the classes for IRT estimation we can minimise the effect for some of the most flagrant cases, but we cannot eliminate all the negative discrimination cases.

In order to show this, we are going to recalculate all the parameters and abilities, but previously removing all instances with negative discrimination from the dataset. This is what we see in figures 11 (two rightmost plots), 12 (two rightmost plots) and 13 (two rightmost plots). Now the Optimal and Pessimal classifiers are (in most cases) the best and worst classifiers respectively. Also, we see that the accuracies and abilities have a much more monotonic and tight relationship. In fact, by removing the examples with negative discrimination, there are classifiers that can get almost 100% accuracy.

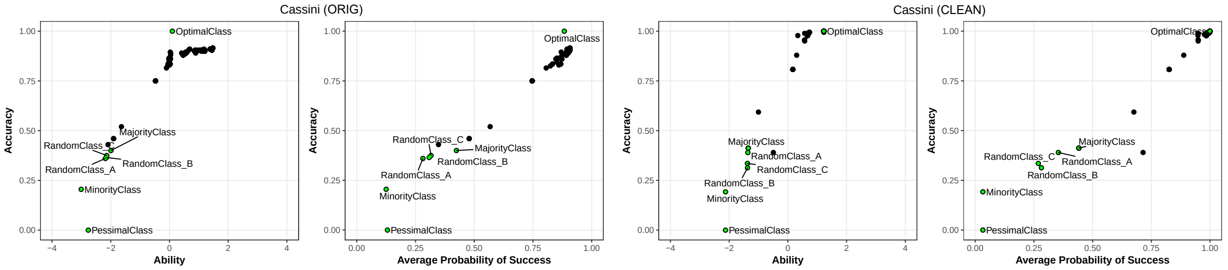

Figure 11: (Two leftmost plots) Original Cassini dataset. (Two rightmost plots) Cassini dataset where those instances with a negative discrimination parameter (a) have been removed. In both scenarios we display two scatter plots: one showing the relationship between the ability parameter θ and the classifier accuracy and; and the second one, showing the relationship between the average probability of success $pSuccess(\theta_c)$ given the ability of the classifier and their accuracy.

The outcome of this observation is that IRT penalises those classifiers that respond correctly to the instances with negative discriminations, as a good classifier should be *wrong* with items with negative discrimination. In other words, for instances with negative discrimination parameters, IRT considers that the classifiers that succeed may be less able, either because they overfit, underfit or are right by chance. This might sound counterintuitive but is consistent with the item parameters. This suggests the common practice in IRT of removing the items with low or negative discrimination, leaving only the items that are useful to evaluate respondents for exams and tests. If we do that for a dataset, we are not sure that we are removing noise or just odd instances that are well labelled, but we have an ability value that is more indicative of the quality of the classifier.

From a machine learning point of view, whether we have to remove the instances with negative discrimination is an important question, but it depends on what we want to do. If we want to learn models, it is more dubious whether they should be removed (but this should be analysed for each technique). However, if we want to evaluate models, over a benchmark of datasets, it seems that removing these instances (from the test set) can produce a more robust value of ability. We must also remember that, for a collection of classifiers, accuracy is not commensurate across datasets, but IRT puts all individual scores on a standardised and commensurate, scale; so it is more meaningful to compare between datasets. Accuracy and other related measures cannot do it right: estimate the “average accuracy” for many datasets does not make sense, and average rankings (e.g., Friedman test) actually turn a discrete value (a rank) into a quantitative one but without providing any insight about the quantitative differences between these values. On the other hand, when evaluating models over just one dataset, ability gives us the connection with the difficulty of the instances, unlike accuracy and other related measures. Before a more extensive analysis is done, we will not run into any conclusions.

6.2. Classifier characteristic curves

Once the different IRT parameters of each instance are estimated and understood, we propose to define a classifier characteristic curve (CCC) for each classifier of interest, inspired by the concept of person characteristic curve previously developed in IRT. A CCC is a plot for the response probability (accuracy) of a particular classifier as a function of the instance difficulty. Figure 14 presents the CCC of the subset of classifiers in Table 5 for the Heart dataset[10] using the difficulty parameter b_i as was estimated in the previous experiments with the population of classifiers. For producing the CCC, we divided the instances in bins of the same size according to the difficulty parameter. For each bin, we plot on the x-axis the average difficulty of the instances in the bin and on the y-axis we plot the frequency of correct responses of the classifier (accuracy). In this experiment, we excluded the instances with negative slopes.

In Figure 14 (top) we show the CCC obtained with the 70% of Heart-statlog, for which difficulties have been obtained as before, using IRT on the test data (adopting 10-fold cross-validation). Since we wanted to

[10] Classifier characteristic curves (CCC) for the rest of UCI datasets can be found at `https://github.com/nandomp/IRT4ML/tree/master/Experiments/`.

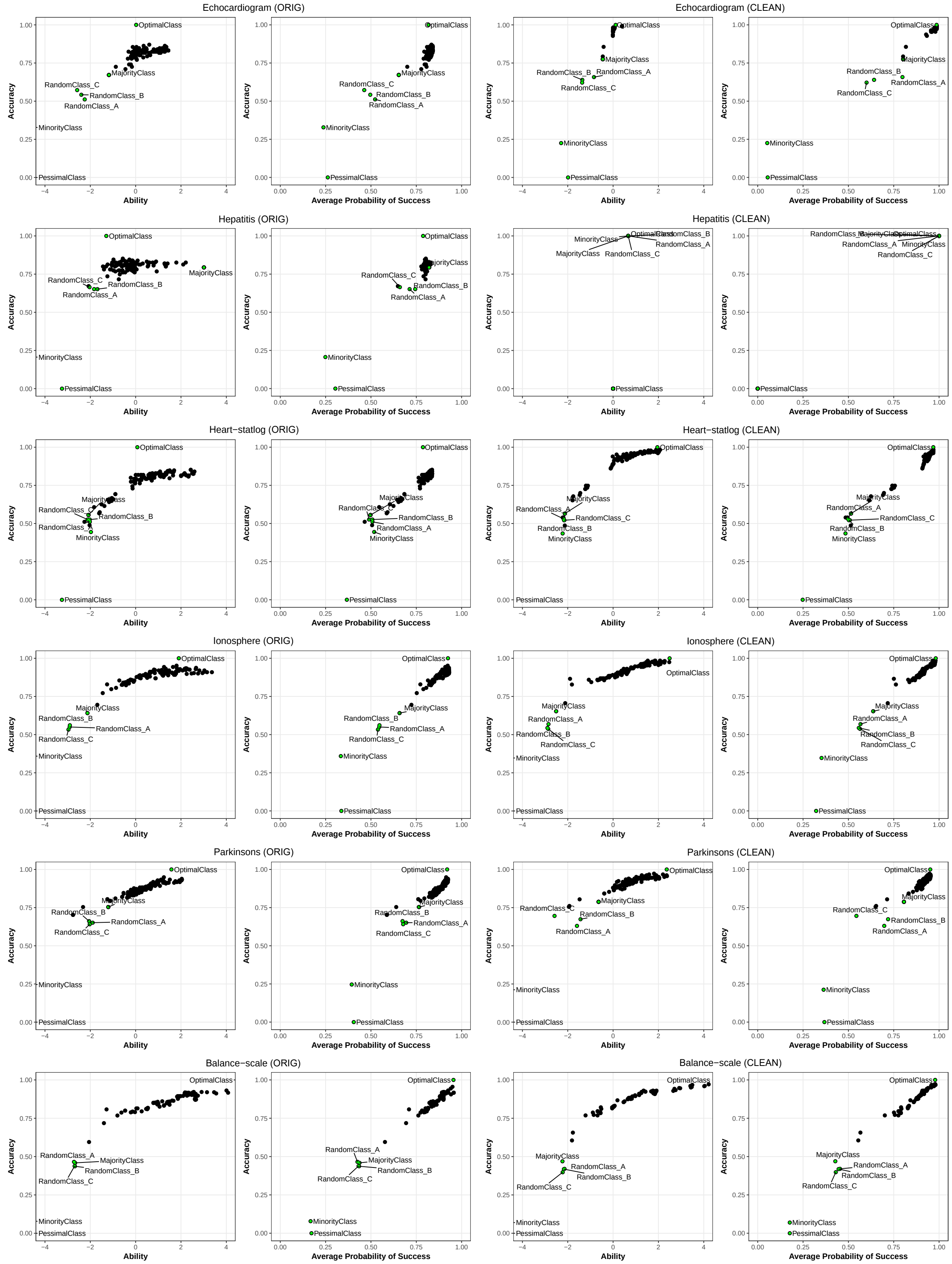

Figure 12: (Two leftmost plots) Original UCI datasets (1-6). (Two rightmost plots) UCI datasets (1-6) where those instances with a negative discrimination parameter (a) have been removed. In both scenarios we display two scatter plots: one showing the relationship between the ability parameter θ and the classifier accuracy and; and the second one, showing the relationship between the average probability of success $pSuccess(\theta_c)$ given the ability of the classifier and their accuracy.

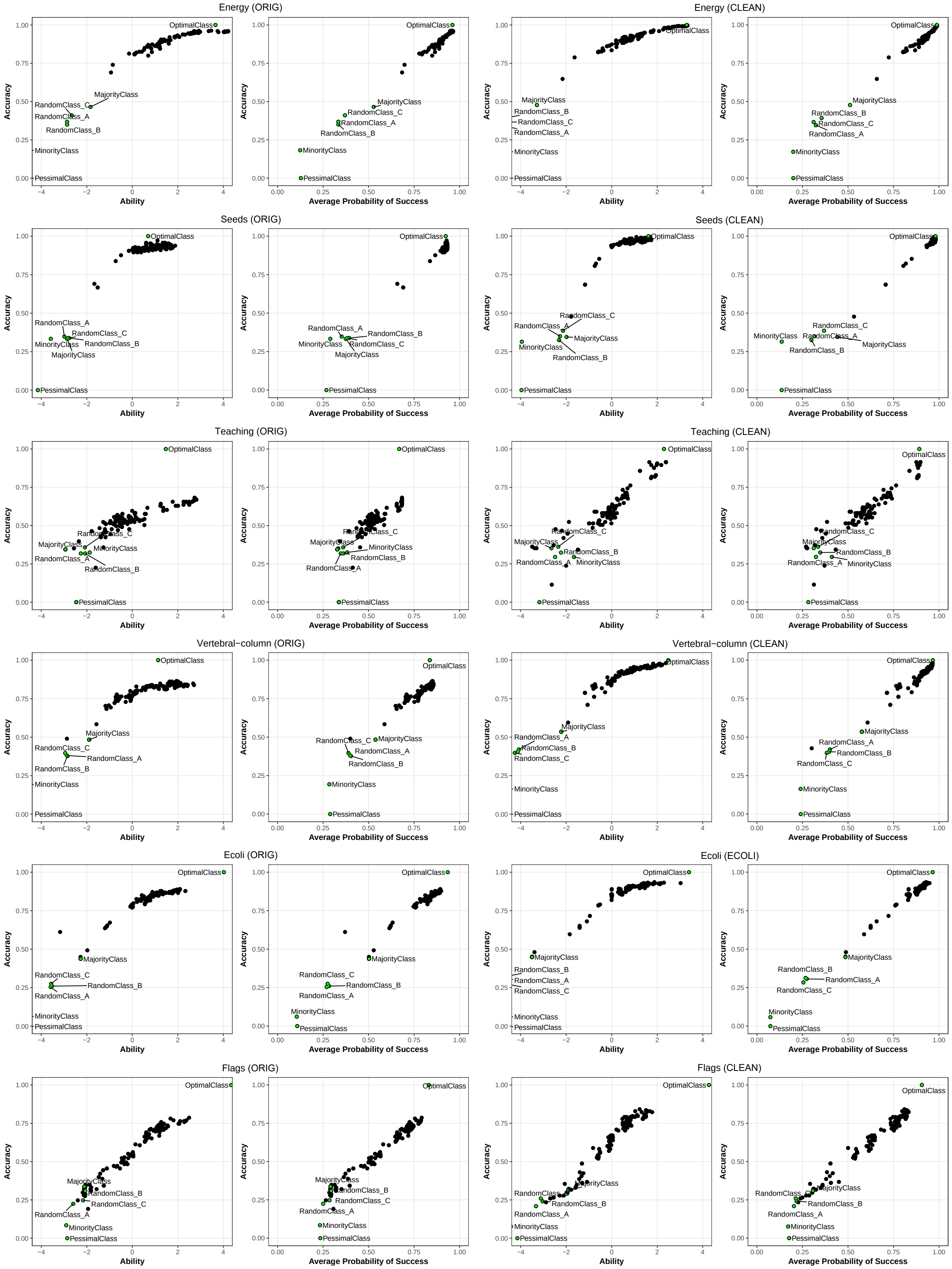

Figure 13: (Two leftmost plots) Original UCI datasets (7-12). (Two rightmost plots) UCI datasets (7-12) where those instances with a negative discrimination parameter (a) have been removed. In both scenarios we display two scatter plots: one showing the relationship between the ability parameter θ and the classifier accuracy and; and the second one, showing the relationship between the average probability of success $pSuccess(\theta_c)$ given the ability of the classifier and their accuracy.

ID	Classifier	Acc
Rnd	Random classifier	0.54
fda	Flexible discriminant analysis	0.83
rpart	Recursive partitioning	0.84
JRip	Propositional rule learner	0.87
J48	Decision tree	0.89
SVM	Support vector machine	0.96
IBK	2-nearest neighbours	0.93
RF	Random forest	0.96
NN	Neural network	0.97

Table 5: Classifiers of interest (using default parameters) and their accuracy for the Heart-statlog dataset. The selected classifiers are a representative sample of the main families of classifiers in machine learning.

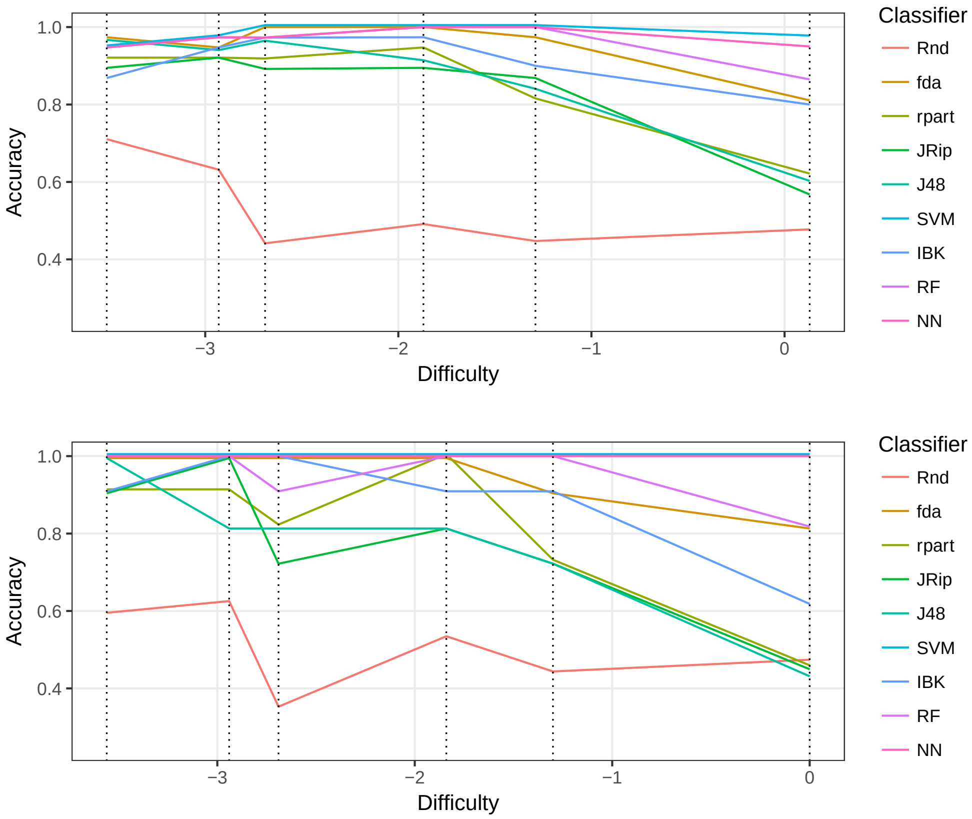

Figure 14: Empirical classifier characteristic curves (across bins on the difficulty parameter) of the classifiers in Table 5 (Heart dataset, negative discrimination instances filtered out). Dashed black vertical lines represent the average difficulty values for the instances in each bin. (Top) CCC obtained with the 70% of Heart-statlog using cross-validation. (Bottom) CCC obtained with the rest (30%) of (unseen) data used as validation set.

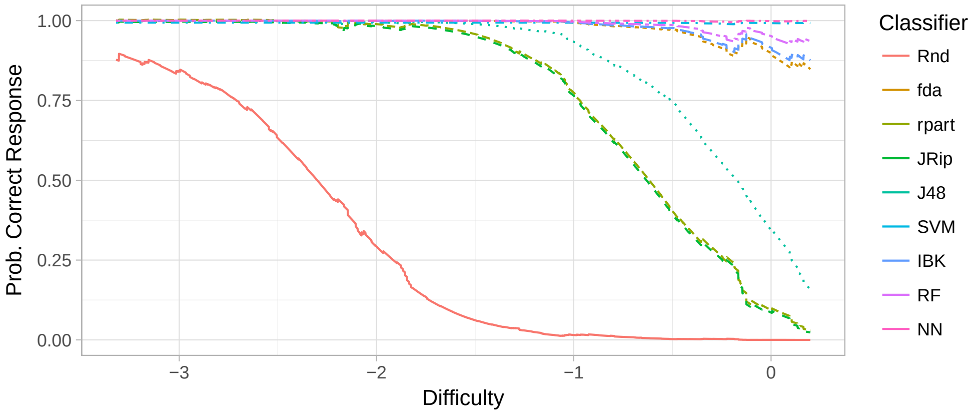

Figure 15: Theoretical classifier characteristic curves, i.e., probability of a correct response as a function of the difficulty parameter for all the classifiers in Table 5 (Heart dataset, negative discrimination instances filtered out).

see some progression in the curves (sufficient detail) but still some robustness without spurious peaks, the bins had to contain a minimum number of instances in each interval. As the dataset contains 226 instances (with positive discrimination), and after a split (70-30%), about 70 of examples in the test set, we wanted a minimum number of 10 examples per bin (with equal size this meant 6 bins approximately). Regarding the results, the classifiers are roughly constant for the first three bins (easiest instances), corresponding to 34% of the instances considered. Apart from the random classifier, all get good results for these easy instances. From the fourth bin, instances become more difficult in such a way that it is possible to start distinguishing the classifiers' abilities and some degrade sooner than others. For instance, *rpart* had a very good result for easy instances but some problems with those of medium difficulty. In the fifth bin (17% of the instances), *rpart* obtained the worst response probability (0.81), while the best classifiers are still the *NN*, *SVM* and *RF* with a response probability equal to 1. Finally, for the latest bin (17% of the instances), the really hard instances, *NN* is the best classifier, followed by *SVM* with response probabilities 0.98 and 0.94. The most striking and interesting case is *J48*. From being among the best classifiers for low and medium difficulties it becomes the second-worst for high difficulties. This suggests that the notion of difficulty that IRT infers may be related to Thornton's separability index, defined as the percentage of the closest examples that are of the same class [25, 26], which is equivalent to the instance hardness measure kDN. However, as we saw in Figure 6, the correlation was not that strong.

Finally, in Figure 14 (bottom) we use a validation dataset (30% of the examples that have not seen before) in order to show whether the previous CCC plot can be used to select the (set of) best classifier(s) according to the difficulty ranges of the instances. Since we do not know the difficulty values of these unseen examples, we estimate them in an straightforward way, by averaging the difficulty values of the most similar examples in the original set. Then, we classify these instances by using the previous set of classifiers and plot the results in a new CCC according to the estimated difficulties . The results are consistent with the top figure. This means that, if there is any way to anticipate the difficulty of an instance (e.g., using proximity as done here), we can decide which classifiers are preferable for that particular instance.

As commented before, CCCs follow the philosophy of the Person Characteristic Curves (PCC), which accommodate item difficulty in the x-axis (same as in CCCs) and the probability of correct response in the y-axis (similar to accuracy in CCCs). Actually, the empirical value of the probability of correct response is accuracy. In the same way we construct the empirical CCCs, we can also plot this theoretical probability of a correct response. This is just using the IRT model and plotting the curve varying the difficulty. Note that we need to take average values for discrimination and guessing parameters for each range of difficulty values. Taking the abilities of all the classifiers that we used to plot the CCCs in 14 (top), we obtain the "theoretical" plot in Figure 15, which is closely related to the CCCs obtained for each classifier in the previous figure.

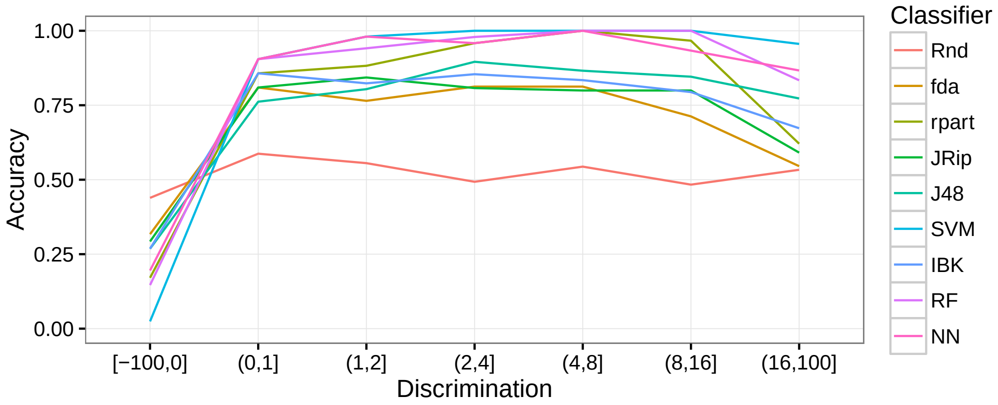

Figure 16: Empirical classifier characteristic curves plots (across bins on the discrimination parameter) of the classifiers in Table 5 for the Heart dataset.

We also propose a different CCC plot using the discrimination parameter instead of difficulty on the x-axis. Figure 14 presents this variant of CCC for the same classifiers in Table 5, also for the Heart dataset. In this case, we analyse the original non-filtered version because we are interested in the analysis of negative discriminations. The construction procedure is similar to the previous case: collect binary responses and divide the instances in bins ordered by the discrimination parameter. For each bin, we plot on the x-axis the average range of the discrimination of the instances in the bin and in the y-axis we plot the frequency of correct responses (accuracy) of the classifier.

The shape of the curves is as surprising as interesting. The results are very bad for negative discriminations, but there is also a slightly weak area for very high discriminations (very steep slopes). If we look first at the negative discriminations, we see that some methods that are very good for positive-discrimination instances (e.g., SVM) are very bad for negative-discrimination instances. Actually, for the Heart dataset, it seems as if the best techniques in terms of accuracy (SVM or even RF) are also the best predictors for discrimination (and vice versa). Actually, there seems to be a general pattern that the best models for positive-discrimination instances are the worst for negative-discrimination instances, although this requires further research. Of course, as expectable, the random model is the best for the negative-discrimination instances, as this model is flat.

The most interesting part for classifier evaluation happens at the right end. It seems that those instances with very high slopes (very high discriminations) are the ones that can better discriminate between different techniques. In other words, for medium discriminations, results are tighter and we would need several instances to tell one classifier from another, but for high discrimination, only a few examples may suffice. This links to one of the original motivations of IRT, being able to assess individuals (in our case classifiers) with as few items (in our case examples) as possible.

7. Discussion

The previous sections have analysed the item parameters and the classifier abilities in order to have a better understanding of IRT when applied to classification problems in machine learning. When looking at an instance, we see that its difficulty can be caused by several reasons: it can be borderline, it can be surrounded by examples of a different class (very low separability index), it can be an outlier, etc. With the discrimination parameter, we at least can see whether it is difficult because only good methods are able to identify it (but still solvable), having positive slope, or it is difficult because no method gets it right (having a flat slope), or even good methods fail especially (because they want to find a pattern for it). This suggests that discrimination can be very useful to analyse noise, on one hand, but also to analyse how expressive classifiers are, and whether they overfit, on the other hand. Many possible applications are suggested in [5] with instance hardness to characterise “why some instances are difficult to classify correctly”, but we can give a more comprehensive view with discrimination and difficulty.

Another thing we have clarified is the guess parameter. This has to be ruled out as having any connection with the class distribution or imbalance. The inclusion of random classifiers is useful to see how difficulty and ability can be calibrated. For instance, we could scale the difficulties and abilities values such that they are zero for random classifiers, which would help interpretability. Nevertheless, we have seen that by including the 3PL models that have the flexibility of the guess parameter, we obtain better fits, fewer negative discrimination values and neater relations of ability with accuracy. Indeed, the IRT models can be improved if the classes are balanced after training, in order to calculate the IRT parameters.

The comparison with instance hardness measures has also shed light about the parameters. We have seen that they provide something new, which had not been captured by the previous measures. The explanation is clear here: hardness measures are population-independent while IRT parameters depend on a population. Of course, this is at the same time an advantage and a disadvantage, IRT parameters are relative to the chosen population, but if we want to analyse how items behave with a representative set of classifiers, this can give us more information. There is another populational-based measure, IH. However, this measure confounds difficulty and discrimination, and it is not sufficient for explaining the different kinds of instances.

When analysing abilities, one of the first surprising results was that the optimal classifier does not get the highest ability. This is not a mistake but a way to maintain the consistency between the *expected* responses produced by the logistic models for good classifiers and their observed responses. If an instance has a negative slope, the expected response of the optimal classifiers for that instance is close to zero. However, the observed response of the optimal classifier is always one. So, one way to produce a better fit of the observed responses for that instance in isolation would be to demote the ability of the optimal classifier. In this way, the difference between the expected response (defined by the ICC) and the observed response of the optimal classifier would not be so large.

Actually, if a classifier is predicting all test instances correctly, we do have a perfect classifier (probably because the dataset is very easy). In usual circumstances, with imperfect classifiers, noisy datasets, etc., it makes sense again to demote the optimal classifier, especially considering that other actually good classifiers also make mistakes for the noisy instances. So, in this way, ability is a very interesting measure that portrays a different information than accuracy. Actually, considering that the optimal classifier should have maximum ability was a wrong premise when we started the analysis of IRT in machine learning.

Once we have a solid understanding of IRT for classification problems, we can now suggest five main application areas. First, IRT could be useful to improve classifier methods. We have seen that the discrimination parameter could be used to identify those instances with noise or with particular characteristics, or where the classifiers overfit. This can be done with the training dataset (using cross-validation) for a pool of common classification techniques (preferably efficient). Then, several criteria for exclusion of some instances can be implemented during the learning of more computationally-demanding or less robust techniques. Also, for some incremental methods (or new methods to be developed) it might be useful to order the examples in some way, starting from those that are easier and more discriminative, and let the classifier be refined for other more complex examples afterwards. Of course, all this should be analysed in conjunction with the relevant literature in instance selection [27, 28, 29].

The second (related) area is the construction of algorithm portfolios [30, 31]. By properly understanding where the difficult instances are and how algorithms behave with them (most able algorithms being more capable or not for these instances, according to discrimination), we can design better portfolios.

Actually, the third area is classifier selection during deployment. If there is any way to anticipate the difficulty of an instance, we can decide which classifiers are preferable for that particular instance (such as [32], with no retraining, more similar to a reframing by difficulty, [33]), looking at the classifier characteristic curves. The difficulty of an instance could be explored by comparing the predictions of several classifiers (not against the true label, which is unknown, unlike we have done here with IH) or by comparing it with other instances (in the training data) for which we have previously determined its difficulty.

The fourth area goes around a better understanding of datasets. For instance, IRT parameters are latent variables that could be related to the manifest variables, the dataset features. For instance, we can find whether a particular subset of features contributes very significantly to the values of difficulty or discrimination. This can give us clues about how to avoid difficult items (e.g., in active learning, [34]) or how to ensure that we pick discriminative items (e.g., again, in active learning). Also, new feature selection methods could be developed by taking the IRT parameters into account.

The fifth main area is evaluation. Actually, IRT was introduced for that. One possible direction is the use of IRT to produce more discriminating datasets, by removing the instances with negative or flat discrimination. It is a quite common practice in machine learning that new methods are compared using 20 or 30 datasets from a repository, when it is well known that most of them are not very discriminative. If we 'clean' the datasets in order to remove the instances with negative discrimination, we can get that the abilities are better aligned with the quality for all examples. However, excluding instances in a dataset may not necessarily be a good idea, even for the estimation of IRT parameters in the test set. Every case must be addressed individually, so this particular aspect remains an open issue. Also, we can compare abilities between different datasets, which could be normalised to be commensurate and calculate averages for a set of classifiers, something that for accuracy or other metrics is not advisable, as the magnitudes can be incommensurate.

Finally, the most common application of IRT is in adaptive testing [35, 36]. Selecting the items that are most discriminating for a particular dataset may minimise the number of instances that are required to estimate the ability of a new classifier, also by adapting the difficulty of the items to the classifier as the estimation proceeds. This could be useful especially in applications where we can ask for the label of selected instances, and they have a high (expert) cost. As in IRT, a good estimation of ability using adaptive testing could be done with about a dozen instances.

The use of IRT comes with the extra cost of estimating the parameters, which involves some computational effort (especially for large datasets) and may require to reiterate the estimation in order to avoid local minima, depending on the IRT models and the particular library. The compensation comes with the detailed analysis of items and techniques that IRT provides. It is important to note that for many of the applications mentioned above, this analysis could be done at the time benchmarks are configured, so that the parameters for a set of datasets and algorithms can be reused by other researchers if are made available, as we have done here with our experiments.

Overall, apart from the experimentation with real classifiers and datasets, we have used artificial datasets and artificial classifiers, which have brought an excellent opportunity to analyse how IRT works and clarify their interpretation. We expect that further research can do this for other supervised machine learning tasks (e.g., regression, for which other non-dichotomous IRT models should be explored), but also for weakly supervised machine learning (e.g., reinforcement learning). Another very interesting area would be incremental learning or situations where we increase the number of examples gradually. This scenario would show an evolution of the ability of several classifiers and also the difficulty of the instances (the increase in size is assumed to be made in such a way that the increase is done, incrementally, as supersets). Also, we have limited our analysis to the mapping between instances and items (instance-wise), but we could also consider datasets as items (i.e., dataset-wise), leading to interesting connections with meta-learning [14]. Finally, we hope that this paper encourages other people to analyse where and how IRT can be useful for machine learning and other areas of artificial intelligence (e.g., planners, SAT solvers, etc.) with increasing availability of experimental results to be analysed.

Acknowledgements

We thank Peter Flach, Cèsar Ferri and the rest of the REFRAME project for some useful discussions about IRT in machine learning. We are also grateful to Aiden Loe, who helped us with some R libraries and the notion of fit.

This work has been partially supported by the EU (FEDER) and Spanish MINECO grant TIN2015-69175-C4-1-R, the Air Force Office of Scientific Research under award number FA9550-17-1-0287, and the REFRAME project, granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences Technologies ERA-Net (CHIST-ERA), and funded by MINECO in Spain (PCIN-2013-037) and by Generalitat Valenciana PROMETEOII/2015/013. Ricardo Prudêncio is financially supported by CNPq (Brazilian Agency). José Hernández-Orallo was supported by a Salvador de Madariaga grant (PRX17/00467) from the Spanish MECD for a research stay at the Leverhulme Centre for the Future of Intelligence (CFI), Cambridge, a BEST grant (BEST/2017/045) from the Valencia GVA for another research stay also at the CFI, and an FLI grant RFP2.

References

[1] F. Martínez-Plumed, R. B. Prudêncio, A. Martínez-Usó, J. Hernández-Orallo, Making sense of item response theory in machine learning, in: European Conference on Artificial Intelligence, ECAI, 2016, pp. 1140–1148.

[2] S. E. Embretson, S. P. Reise, Item response theory for psychologists, L. Erlbaum, 2000.

[3] D. Thissen, H. W. (Eds), Test Scoring, Lawrence Erlbaum Associates Publishers, 2001.

[4] R. J. De Ayala, Theory and practice of item response theory, Guilford Publications, 2009.

[5] M. R. Smith, T. Martinez, C. Giraud-Carrier, An instance level analysis of data complexity, Machine learning 95 (2) (2014) 225–256.

[6] R. B. Prudêncio, C. Castor, Cost-sensitive measures of instance hardness, in: First International Workshop on Learning over Multiple Contexts in ECML 2014. Nancy, France, 19 September 2014, 2014.

[7] F. Martínez-Plumed, J. Hernández-Orallo, Ai results for the Atari 2600 games: difficulty and discrimination using IRT, 2nd International Workshop Evaluating General-Purpose AI (EGPAI 2017), Melbourne, Australia, 2017.

[8] R. B. Prudêncio, J. Hernández-Orallo, A. Martínez-Usó, Analysis of instance hardness in machine learning using item response theory, in: Second International Workshop on Learning over Multiple Contexts in ECML 2015. Porto, Portugal, 11 September 2015, 2015.

[9] J. Lalor, H. Wu, H. Yu, Beyond majority voting: Generating evaluation scales using item response theory, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2016) 648–657.

[10] J. Lalor, H. Wu, T. Munkhdalai, H. Yu, An analysis of machine learning intelligence, CoRR abs/1702.04811.

[11] M. Brundage, Modeling progress in AI, AAAI 2016 Workshop on AI, Ethics, and Society.

[12] J. Hernández-Orallo, The Measure of All Minds: Evaluating Natural and Artificial Intelligence, Cambridge University Press, 2017.

[13] J. Hernández-Orallo, Evaluation in artificial intelligence: From task-oriented to ability-oriented measurement, Artificial Intelligence Review 48 (3) (2017) 397–447.

[14] P. Brazdil, C. Giraud-Carrier, C. Soares, R. V. (Eds), Metalearning - Applications to Data Mining, Springer, 2009.

[15] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recognition Let. 30 (1) (2009) 27–38.

[16] J. Vanschoren, J. N. van Rijn, B. Bischl, L. Torgo, OpenML: Networked science in machine learning, SIGKDD Explor. Newsl. 15 (2) (2014) 49–60.

[17] N. Macià, E. Bernadó-Mansilla, Towards UCI+: A mindful repository design, Information Sciences 261 (2014) 237 – 262.

[18] V. J. Hodge, J. Austin, A survey of outlier detection methodologies, Artificial Intelligence Review 22 (2) (2004) 85–126.

[19] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM '08, IEEE Computer Society, 2008, pp. 413–422.

[20] A. Birnbaum, Statistical Theories of Mental Test Scores, Addison-Wesley, Reading, MA., 1968, Ch. Some Latent Trait Models and Their Use in Inferring an Examinees Ability.

[21] M. Lichman, UCI machine learning repository, `http://archive.ics.uci.edu/ml`, University of California, Irvine, School of Information and Computer Sciences" (2013).

[22] D. Rizopoulos, ltm: An r package for latent variable modeling and item response theory analyses, Journal of statistical software 17 (5) (2006) 1–25.

[23] R. P. Chalmers, et al., mirt: A multidimensional item response theory package for the R environment, Journal of Statistical Software 48 (6) (2012) 1–29.

[24] Y. Zhao, R. Hambleton, Software for IRT analyses: Descriptions and features, Center for Educational Assessment Research Report (652).

[25] J. Greene, Feature subset selection using thorntons separability index and its applicability to a number of sparse proximity-based classifiers, in: Proceedings of the 12th Annual Symposium of the Pattern Recognition Association of South Africa, 2001.

[26] C. Thornton, Truth from trash: How learning makes sense, MIT Press, 2002.

[27] A. L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial intelligence 97 (1) (1997) 245–271.

[28] H. Liu, H. Motoda, Instance selection and construction for data mining, Vol. 608, Springer Science & Business Media, 2013.

[29] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. Kittler, A review of instance selection methods, Artificial Intelligence Review 34 (2) (2010) 133–143.

[30] C. P. Gomes, B. Selman, Algorithm portfolios, Artificial Intelligence 126 (1) (2001) 43 – 62.

[31] L. Xu, F. Hutter, H. H. Hoos, K. Leyton-Brown, Satzilla: portfolio-based algorithm selection for sat, Journal of artificial intelligence research 32 (2008) 565–606.

[32] S. Visweswaran, G. F. Cooper, Learning instance-specific predictive models, Journal of Machine Learning Research 11 (Dec) (2010) 3333–3369.

[33] J. Hernández-Orallo, A. Martínez-Usó, R. B. Prudêncio, M. Kull, P. Flach, C. Farhan Ahmed, N. Lachiche, Reframing in context: A systematic approach for model reuse in machine learning, AI Communications 29 (5) (2016) 551–566.

[34] B. Settles, Active learning literature survey, University of Wisconsin, Madison 52 (55-66) (2010) 11.

[35] W. J. Van der Linden, C. A. Glas, et al., Computerized adaptive testing: Theory and practice, Springer, 2000.

[36] H. Wainer, N. J. Dorans, R. Flaugher, B. F. Green, R. J. Mislevy, Computerized adaptive testing: A primer, Routledge, 2000.