ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1140

Making Sense of Item Response Theory in Machine Learning

Fernando Martínez-Plumed[1] and **Ricardo B. C. Prudêncio**[2] and **Adolfo Martínez-Usó**[3] and **José Hernández-Orallo**[4]

Abstract. Item response theory (IRT) is widely used to measure latent abilities of subjects (specially for educational testing) based on their responses to items with different levels of difficulty. The adaptation of IRT has been recently suggested as a novel perspective for a better understanding of the results of machine learning experiments and, by extension, other artificial intelligence experiments. For instance, IRT suits classification tasks perfectly, where instances correspond to items and classifiers correspond to subjects. By adopting IRT, item (i.e., instance) characteristic curves can be estimated using logistic models, for which several parameters characterise each dataset instance: difficulty, discrimination and guessing. IRT looks promising for the analysis of instance hardness, noise, classifier dominances, etc. However, some caveats have been found when trying to interpret the IRT parameters in a machine learning setting, especially when we include some artificial classifiers in the pool of classifiers to be evaluated: the optimal and pessimal classifiers, a random classifier and the majority and minority classifiers. In this paper we perform a series of experiments with a range of datasets and classification methods to fully understand how IRT works and what their parameters really mean in the context of machine learning. This better understanding will hopefully pave the way to a myriad of potential applications in machine learning and artificial intelligence.

1 INTRODUCTION

It is no news that most techniques in artificial intelligence, even with a strong theoretical background, are ultimately evaluated empirically, by comparing them and other methods against a set of problems, taken from a benchmark or repository. Aggregated performance metrics, such as an average quality measure over a set of problems, or the use of statistical pairwise comparisons are commonplace. However, having a greater value of an aggregated metric or more "win pairs" against another technique only provides summarised information. In many areas it is important to analyse the particular problems for which the best techniques usually fail, whereas other simpler techniques can succeed. Is it because the problem is pathological? Or is it because the techniques have some lacunas?

In this paper we analyse these questions using item response theory (IRT), a group of modelling and statistical tools borrowed from psychometrics that are designed to provide a precise characterisation of items and subjects, by analysing their responses [5, 13, 4]. Similarly to IRT, in our context we define the proficiency (or ability) of a model or technique as the level of hard instances this technique is able to solve. For instance, if a classifier solves all the simple instances but none of the difficult ones, the classifier may be worse (in terms of proficiency) than a classifier that solves most of the simple ones and some of the difficult ones. We focus on supervised machine learning and classification in particular, since there are many interesting questions such as instance hardness, noise handling, outliers, meta-learning, borderline areas, etc., that can find a parallel in IRT and have a particular understanding under this theory.

IRT shows a dual behaviour in the way that classifier ability and instance difficulty are estimated at the same time, both depending on the other classifiers and instances. Instance difficulty, or hardness, is an important feature of an instance. Actually, in machine learning, it has been recently demonstrated that incorporating instance hardness into the learning process can significantly increase classification performance [12, 9]. However, instance hardness gives a very limited perspective of what is happening with an instance. Apart from difficulty, a very interesting parameter in IRT is the *discrimination* parameter 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 classifiers for a certain dataset. Thus, some instances are only solved by more proficient classifiers. However, some other difficult instances are not solved by these classifiers. Is there anything special in these points? Are some of them really difficult instances while the others being just noise? Can we detect them using IRT?

Also, looking at the classifiers, we can think about what makes certain classifier proficient or whether there are more suitable classifiers for more difficult instances. We can 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, but for easy instances another classifier may be more robust. These *classifier characteristic curves* may be very interesting to analyse machine learning models.

IRT may be the right tool to analyse all these questions. However, there are some issues about the use of IRT that need more understanding. After a previous preliminary use of IRT in machine learning only using one classifier technique (random forests) [10], we have extended the analysis for more datasets and many more models, and we have found some caveats when trying to locate classifiers and understand the parameters of the items (difficulty, discrimination and guessing) and especially the abilities of the classifiers. Solving these caveats is necessary to clarify the previous questions and make full sense of IRT in machine learning.

In this paper, we present this novel IRT-based approach with potential applications in machine learning and artificial intelligence. We analyse the instance-wise performance of a great variety of classifiers

[1] DSIC, Universitat Politècnica de València, Spain, email:fmartinez@dsic.upv.es
[2] Centro de Informática, Universidade Federal de Pernambuco, Recife (PE), Brasil, email:rbcp@cin.ufpe.br
[3] Universitat Jaume I de Castelló, Spain, email:auso@uji.es
[4] DSIC, Universitat Politècnica de València, Spain, email:jorallo@dsic.upv.es

and determine those cases for which classifiers fail. In a nutshell, this paper contributes to clarify how IRT works, highlighting the role it can play in potential machine learning applications.

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 focuses on the inferred instance parameters: whether difficulty really represents instance hardness, the interpretation of the discrimination parameter (especially when close to zero or negative) and the relation between the class distribution and the guessing parameter. Section 5 focuses on the inferred classifier ability, how it relates to ability and the effect of removing the instances with negative discrimination. Section 6 discusses the findings of the previous sections and gives a global interpretation of what the IRT parameters mean and how they should be used. Section 7 closes the paper with the prospective applications, once seen that IRT and machine learning make sense together.

2 BACKGROUND

It is clear that the better we understand how a technique behaves for a range of problems the more possibilities we have for an accurate evaluation, the right selection of the optimal technique for a given problem and the improvement of the technique themselves. Let us first analyse why some practices in AI and machine learning may be benefited by a more detailed analyses, especially in terms of discrimination, and then we will see what IRT may bring.

2.1 Motivation

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, 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. 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 more 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 metalearning [3], 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 cannot be assigned to accuracy or cost. For instance, it is easy to get high accuracy for a very imbalanced binary problem while it is very difficult for a problem with ten balanced classes. Accuracy or any other common metrics [6] are not very related to the difficulty of a dataset. There have been some recent analysis of repositories [15, 8], 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 [12], 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. However, this average hardness misses 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 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.

2.2 Item response theory

Item response theory (IRT) [5, 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 response for an item is a function of the examinee's ability (or proficiency) and some item's parameters. There are models developed in IRT for different kinds of response, 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, in which $U_{ij} = 1$ for a correct response and $U_{ij} = 0$ otherwise. Let θ_j be the ability or proficiency of j. Now, assuming that the result only depends on the ability and no longer on the particular classifier, 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:

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

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

- Difficulty (b_i): it is the location parameter of the logistic function and can be seen as a measure of item difficulty. When $c_i = 0$,

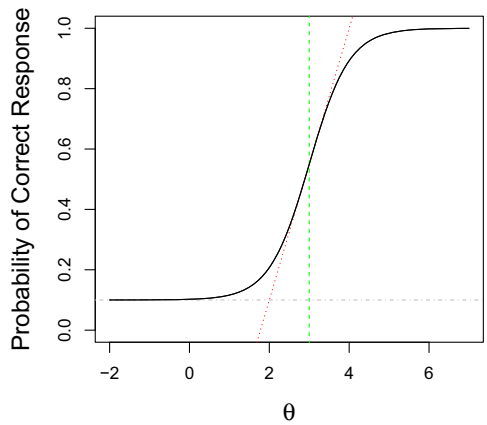

Figure 1: Example of a 3PL IRT model (in black), with slope $a = 2$ (discrimination, in red), location parameter $b = 3$ (difficulty, in green) and guessing parameter $c = 0.1$ (chance, in grey).

then $P(U_i = 1|b_i) = 0.5$. It is measured in the same scale of the ability;
- 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 instance;
- 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 difficulty 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's parameters (when examinees' abilities are known) or the abilities (when items' parameters are known). A more difficult, but common, situation is the estimation when both the items' parameters and respondents' abilities are unknown. In this situation, an iterative two-step method (Birnbaum's method [2]) can be adopted:

- Step (1) Start with initial values for abilities θ_j (e.g., random values or the number of right 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, items' parameters and abilities are simultaneously estimated only based on a set of observed responses to items, with no strong 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. In a very preliminary analysis of the application of IRT to machine learning [10], we addressed classification problems, and we identified items with instance and individuals with classifiers. Accordingly, we can talk about instance difficulty, instance discrimination, instance guessing and instance characteristic curves, but also we can talk about 'classifier abilities'.

The previous analysis in [10] was restricted to just one classifier technique (random forest) varying the number of trees. As a consequence, results were very smooth and expectable, as the higher the number of trees the higher the ability. We already found some caveats, as our lack of understanding of the 'guessing' factor, which we wanted to connect to the class distribution. However, we did not know how IRT would behave for a pool of diverse classifiers and, most especially, what parameters we could get when we have special classifiers, such as a random classifier, a majority classifier or a perfect classifier. The IRT packages used then and the number of datasets were also limited, so we had a very restricted view of the application of IRT.

In this paper we widen our focus and do a more consistent experimental analysis on a range of different classifiers and datasets, showing and explaining some unexpected results.

3 METHODOLOGY

We want to design a realistic and reproducible experimental scenario[5] where we can compare a wide range of classifiers with a diversity of datasets[6].

For estimating good IRT models, a reasonably high number of individuals is needed. Since we equate individuals with instances this is not an issue if we do not use datasets with a very small number of examples N (¡100).

However, the most critical issue is obtaining a large population of techniques. In order to achieve that, we used 128 classifiers arising from 15 different families (decision trees, rule-based methods, discriminant analysis, Bayesian, neural networks, support vector machines, boosting, bagging, stacking, random forests, nearest neighbours, partial least squares, principal component regression and logistic and multinomial regression). In order to obtain 128 classifiers, but still heterogeneous, we modified their parameters[6] to obtain several different models per technique. For instance, a pool of classifiers was produced by Random Forests (RF) trained with different numbers of trees. All the classifiers are implemented in R. Some use a particular package while others use the classifier through the interface provided by the caret[7] package. We learned the models adopting 10-fold cross-validation.

Apart from the classifiers generated by machine learning techniques, we also introduced some artificial classifiers:

- Three random classifiers (*RndA*, *RndB*, *RndC*), the three equally using the prior class probabilities but included to analyse variability.
- 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: an optimal/pessimal classifier (*Opt*, *Pess*) which always

[5] For reproducibility, all the experiments can be found in https://github.com/nandomp/IRT_params/blob/master/Experiments.md.

[6] The whole list of classifiers' parameters, data, plots, configuration files and code is in https://github.com/nandomp/IRT_params.git.

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

succeeds/fails respectively.

These seven artificial classifiers are used as baselines, establishing a continuum from the pessimal to the optimal. The random ones are in between, plus the minority and majority class.

We have performed a series of experiments using the Cassini[8] toy dataset and a set of eight real datasets from the UCI repository [1]. For space reasons, in the paper we only show the results for the "Cassini" and the "Heart-Statlog" dataset. "Cassini" is a 3-class bivariate toy dataset composed by 200 instances with a 10% of random noise we put on purpose (see Figure 2), and we use it as an illustrative dataset. "Heart-Statlog" is a binary dataset which has 270 instances and 13 attributes containing heart disease data (see Figure 5) and we use it as a representative dataset.

For all datasets separately we develop the parameter tuning in order to obtain the logistic models and proficiency models. The binary results from the classifiers are always obtained by using the test "fold", therefore never using the train sets to obtain the responses. In particular, a 3-parameter IRT model (based on logistic functions) is learned for each instance, fitting the classifiers' correct response probability according to their abilities and the guess parameter. We adopt MLE to estimate all the models' parameters of all instances and the classifiers' ability simultaneously, as usual in IRT. In particular, for generating the IRT models, we used the `ltm`[11] R package, which implements the previously mentioned Birnbaum's method.

The model parameters characterise the instance difficulty, guess parameter and discrimination power, as we will analyse in the following section. The ability of a classifier is also estimated by the MLE method in the IRT package under different contexts (levels of instance difficulty). As we will see in Section 5, it is related (non-linearly) to classification accuracy.

4 ANALYSIS OF THE INSTANCE PARAMETERS

The item parameter that is easiest to understand is difficulty. Because of the MLE estimation method, the value is not equal but well correlated to the percentage of classifiers that predict it correctly. Difficulty can be estimated by other (simpler) methods, such as [12], and it has no interpretation problems, so let us focus on the discrimination and the guessing parameters next.

4.1 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 be used to indicate if the instance is useful to distinguish between strong or weak classifiers for a problem. With the aim of better understanding the meaning of this parameter, we first used the toy Cassini dataset (see Figure 2). In this case, 200 IRT models were built (one per instance) and 128 values of ability for the set of classifiers were estimated. Some examples of item characteristic curves[9] (ICCs) are presented in Figure 3. What we see is that some instances are more difficult than others: instance "b" has difficulty 0.9 while instance "c" has difficulty -2.1. The slopes for the four instances here are positive but with pretty different slope values (see Table 1).

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

[9] All ICCs for the Cassini and the UCI datasets are in https://github.com/nandomp/IRT_params/tree/master/_guessingParam_

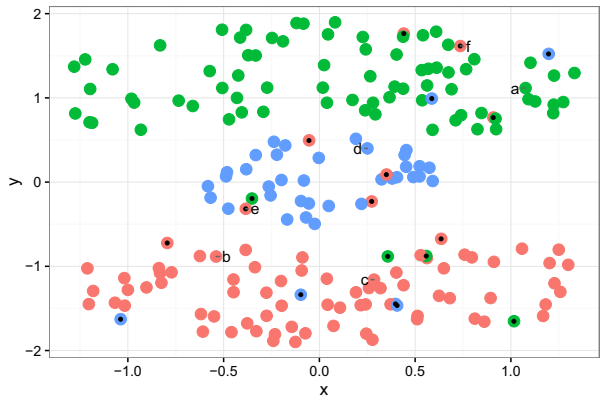

Figure 2: Visualisation of the Cassini toy dataset. Different colours represent different classes. Those instances with negative slope are represented with a black dot inside. ICCs of those instances labelled with a letter are shown in Figures 3 and 4.

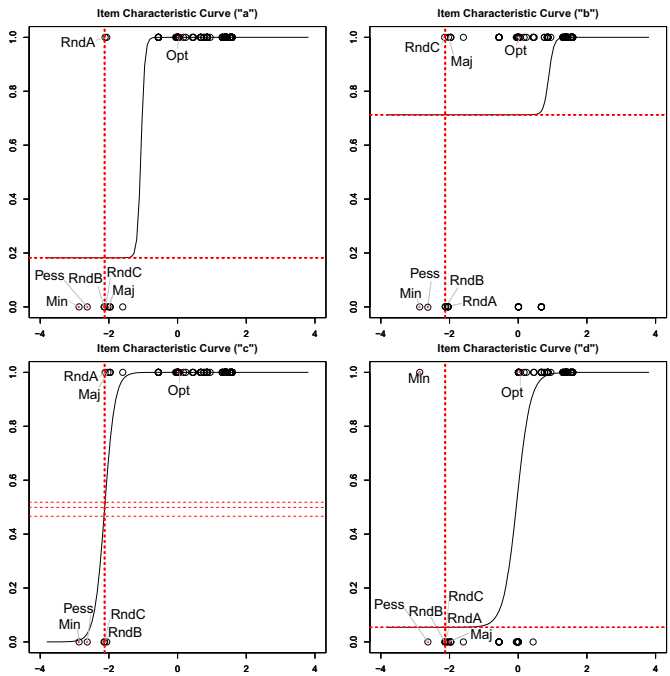

Figure 3: Examples of ICCs (with positive slope) of the points labelled in Figure 2 from "a" to "d". Classifier abilities are also included in the ICCs, plotted at $y = 1$ if they succeed on the instance, and at $y = 0$ otherwise. Artificial classifiers are named. The probability of correct response for the ability values of the three random classifiers are plotted with red dashed lines (cut points) usually appearing together or very close.

Item	Guess	Difficulty	Discrimination
a	0.18204	-1.06159	18.70137
b	0.71252	0.89022	13.67411
c	0.00016	-2.12133	6.81183
d	0.05437	-0.03429	5.10973
e	0	-0.99836	-1.57099
f	0	-2.11275	-1.70781

Table 1: ICC parameters for the plots in Figures 3 and 4

From the 200 instances, 180 had positive slopes (i.e., positive discrimination values), matching the common assumption of IRT and the nice ICCs on Figure 4. 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 a black dot inside. Figure 4 shows two ICCs examples for these cases "e" and "f". As 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, noisy instances put on purpose are *exactly* those that have negative slope. 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.

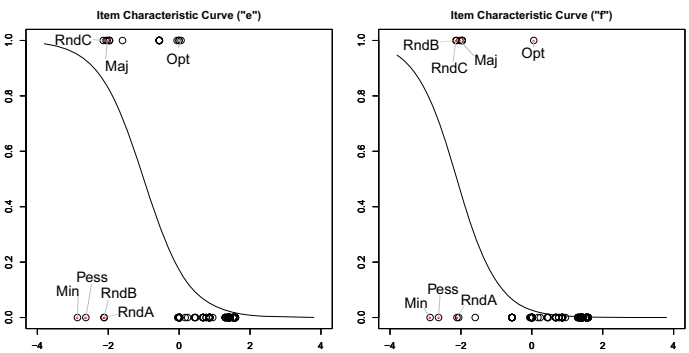

Figure 4: Examples of ICCs (negative slope) of the points labelled in Figure 2 as "e" and "f". Classifier abilities are plotted in $y = 1$ if they succeed, otherwise, in $y = 0$. Artificial classifiers are named.

By looking at the discrimination parameter for several instances, we now 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.

4.2 Understanding the guess 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 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 chances to succeed. For instance, on a multiple choice testing item with four possible answers, the guessing parameter is 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 guess 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

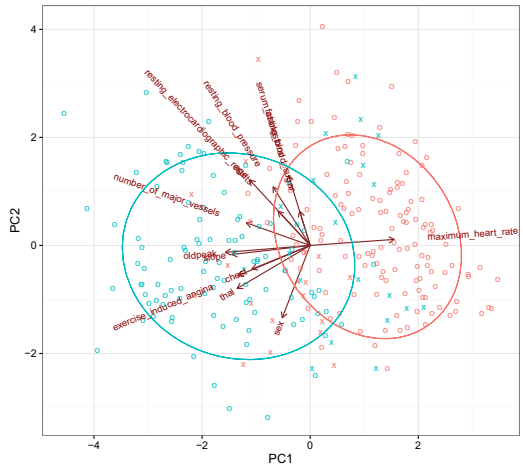

Figure 5: Visualisation of the Heart dataset using the first principal components. Different colours represent different classes. Crosses indicate the instances with negative slopes. Normal data ellipses are represented for each group (class) as well as the arrows for the original variables (dataset attributes).

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). From the rest of experiments with UCI datasets used we concluded exactly the same —initially surprising— fact.

But, interestingly, if we plot as a cut points (dashed red lines) the probability of correct response for the ability values of the three random classifiers, we get some interesting but disparate values, in this case 0.18, 0.71, 0.5 and 0.06 on Figure 3. However, we can compute the average conditional probability of success of these random classifiers (or all that have ability the same ability, denoted by θ_c), for all instances, i.e.,

$$pSuccess(\theta_c) = \frac{\sum p_i(U_i = 1|\theta_c)}{N}$$

where c is the classifier and N is the number of instances. The values of $pSuccess(\theta_c)$ for the three random classifiers are 0.35, 0.36 and 0.36. 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.

5 ANALYSIS OF ABILITY AND CLASSIFIER CHARACTERISTIC CURVES

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 subjects (classifiers). What information can we extract about the classifiers using IRT? Directly, IRT estimates a value of ability θ for each classifier. How is this indicator interpreted? This is what we see next.

5.1 Estimated abilities and actual classifier quality

Figures 6 and 7 show the estimated abilities of all classifiers for the Cassini and Heart-statlog datasets against accuracy (Left) and against the average probability of success $pSuccess(\theta_c)$ given the ability of the classifier (Right). In both cases, we see a strong correlation, 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 left and right plots 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.

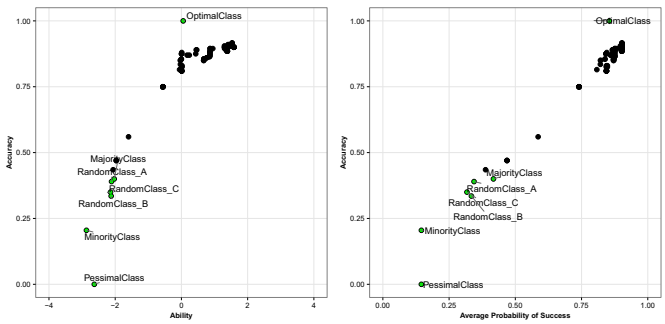

Figure 6: Original Cassini dataset. (Left) Scatter plot showing the relationship between the ability parameter θ and the classifier accuracy. (Right) Scatter plot showing the relationship between the average probability of success $pSuccess(\theta_c)$ given the ability of the classifier and their accuracy.

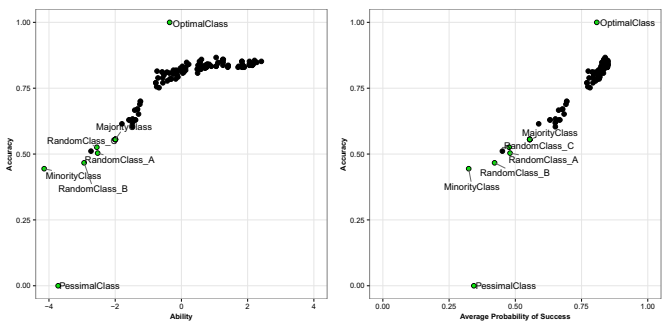

Figure 7: Original Heart dataset. (Left) Scatter plot showing the relationship between the ability parameter θ and the classifier accuracy. Negative values of the discriminant parameter instances greatly affect the estimation of the classifier ability parameter. (Right) Scatter plot showing the relationship between the average probability of success $pSuccess(\theta_c)$ given the ability of the classifier and their accuracy.

We have tried to understand this surprising result and we have observed that the instances with a negative value of the discrimination parameter greatly affect the estimation of the ability parameter of the classifiers. 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 8 and 9. Now the Optimal and Pessimal classifiers are actually the best and worst classifiers respectively. Also, we see that the accuracies are now much better. In fact, by removing the examples with negative discrimination, there are classifiers that can get almost 100% accuracy.

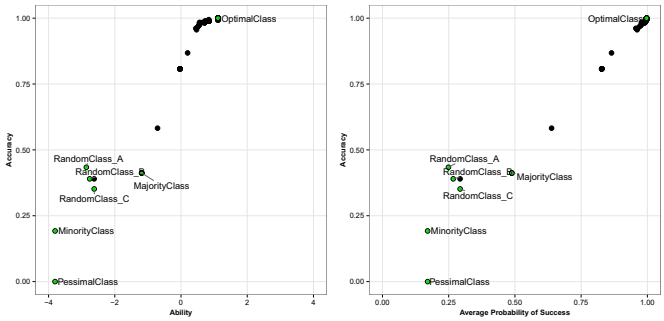

Figure 8: Cassini dataset where those instances with a negative discriminant parameter (a) have been removed. (Left) Scatter plot showing the relationship between the ability parameter θ and the classifier accuracy. (Right) Scatter plot showing the relationship between the classifier average probability of success $pSuccess(\theta_c)$ and their accuracy.

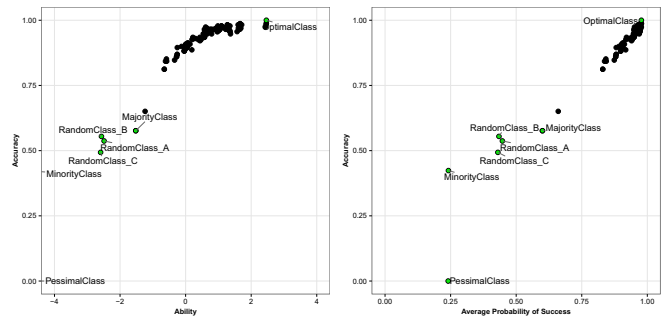

Figure 9: Heart dataset where those instances with a negative discriminant parameter (a) have been removed: (Left) Scatter plot showing the relationship between the ability parameter θ and the classifier accuracy. (Right) Scatter plot showing the relationship between the classifier average probability of success $pSuccess(\theta_c)$ 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 do worse with items with negative discrimination. This is consistent with the item parameters. This suggests the common practice in IRT of removing the instances with low or negative discrimination, leaving only the items that are useful to evaluate respondents. 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. In other words, for instances with negative discrimination parameters, IRT considers that those that succeed may be because they are less able, either because they overfit, underfit or by chance.

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, it seems that removing these instances can have the advantage that ability may be more reliable than accuracy to select the best classifiers. Before a more extensive analysis is done, we will not

run into any conclusions, especially because a better way of comparing classifiers is by looking ability as a function of the difficulty and discrimination of the instances, as we explore next.

5.2 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 on 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 10 presents the CCC of the classifiers in Table 2 for the Heart dataset 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 6 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. In this experiment, we excluded the instances with negative slopes.

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 2: 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.

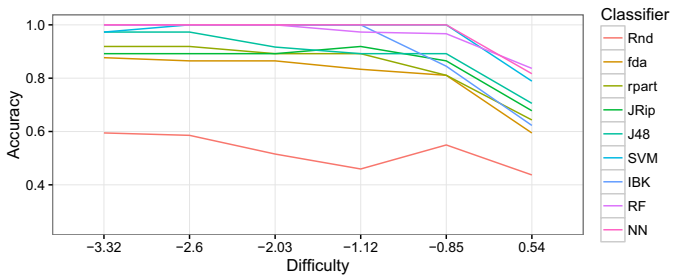

Figure 10: CCC plots (across bins on the difficulty parameter) of the classifiers in Table 2 for the Heart dataset (negative discrimination instances filtered out).

In Figure 10 the classifiers are roughly constant for the first two 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 third 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, J48 had a very good result for easy instances but has some problems with those of medium difficulty. In the fifth bin (17% of the instances), *fda* and *rpart* obtained the worst response probability (0.81), while the best classifiers are still the *NN* and *SVM* with a response probability equal to 1. Finally, for the latest bin (17% of the instances), the really hard instances, *RF* is the best classifier, followed by *NN* and *SVM*, with response probabilities 0.83, 0.81 and 0.78. The most striking and interesting case is *IBk*. From being the best classifier 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, which is defined as the percentage of the closest examples that are of the same class [7, 14]

We also propose a different CCC plot using the discrimination parameter instead of difficulty on the x-axis. Figure 10 presents this variant of CCC for the same classifiers in Table 2 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 as in 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.

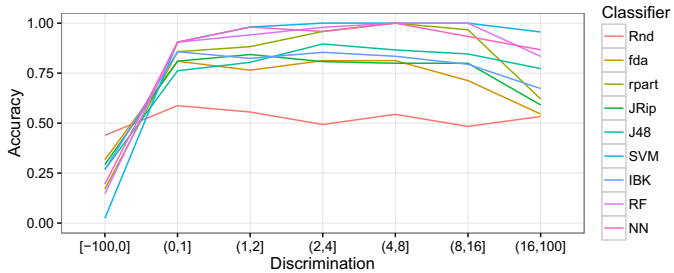

Figure 11: CCC plots (across bins on the discrimination parameter) of the classifiers in Table 2 for the Heart dataset.

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 instance. Actually, it seems as if SVM errors could be the best predictor for discrimination (and vice versa). Actually, there is a general pattern that the best models for positive-discrimination instances are the worst for negative-discrimination instances. 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.

6 DISCUSSION

The previous two sections have analysed the item parameters and the classifier abilities in order to have a better understanding of IRT when applied to machine learning classification.

When looking at instances, we see that their 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, 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.

Another thing we have clarified is the guess parameter. This has to be ruled out as having any connection with the class distribution. 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.

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 instances correctly, either the dataset is very simple, the classifier is overfitting the data or, as usual in IRT, the classifier is cheating (basically what we are doing here with the Optimal classifier, having access to the true labels). 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 made 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.

It is important to highlight that IRT evaluates classifiers in terms of the other classifiers that we include in the pool. This relativeness has also its positive side, especially if we include a range of diverse classifiers in the pool according to those found in many machine learning or data mining suites. Actually, it is for this kind of pool for which we have to select the best model.

7 CONCLUSION

In this paper we have clarified the use of IRT for an instance-wise analysis of datasets and classifiers (we left dataset-wise analysis for future work). We identified several issues of confusion in the interpretation of the parameters (discrimination, guessing, ability) and we have now fully understood their meaning in the context of classification. After this, it is now the time to explore all the potential applications.

There are three main application areas. The first area is that 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 techniques, such as ensembles or deep learning techniques. Also, for some incremental methods (or new methods to be developed) it might be useful to order the examples in some way, starting for those that are easier and more discriminative, and let the classifier be refined for other more complex examples afterwards.

The second area is classifier selection during deployment. If there is any way to anticipate the difficulty of an instance, we can decide which classifier is preferable for that particular instance, looking at the classifier characteristic curves. The difficulty of an instance could be explored by comparing the predictions of several classifiers or by comparing it with other instances (in the training data) for which we have previously determined its difficulty.

The third main area is evaluation. Actually, IRT was introduced for that. One possible direction is the use of IRT to produce more discriminative datasets, by removing the instances with negative 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 can be more significative about the quality of a method. 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. Selecting the items that are most discriminative 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.

Of course, there might be criticisms too for the use of IRT in machine learning. For instance, the IRT approach can be computationally expensive to fit IRT models for millions of instances, which is very common now in real applications (even if we can always use sampling). Of course, using IRT for calculating difficulty alone is an overkill. The strength of IRT is the derivation of the discrimination parameters and the interpretation of abilities.

Overall, we hope that this paper encourages other people to analyse where and how IRT can be useful for machine learning. In this paper, 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 classification and other supervised machine learning tasks (e.g., regression, for which other IRT models exists), but also for weakly supervised machine learning (e.g., reinforcement learning) or for AI in general (e.g., planners, SAT solvers, etc).

ACKNOWLEDGEMENTS

We would like to thank Peter Flach, Cèsar Ferri and the rest of the REFRAME project for some useful discussions about IRT in machine learning.

This work has been partially supported by the EU (FEDER) and Spanish MINECO grant TIN2015-69175-C4-1-R, 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.

REFERENCES

[1] K. Bache and M. Lichman. UCI machine learning repository, 2013. 4

[2] A. Birnbaum, *Statistical Theories of Mental Test Scores*, chapter Some Latent Trait Models and Their Use in Inferring an Examinees Ability, Addison-Wesley, Reading, MA., 1968. 3

[3] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta (Eds), *Metalearning - Applications to Data Mining*, Springer, 2009. 2

[4] Rafael Jaime De Ayala, *Theory and practice of item response theory*, Guilford Publications, 2009. 1, 2

[5] S. E. Embretson and S. P. Reise, *Item response theory for psychologists*, L. Erlbaum, 2000. 1, 2

[6] C. Ferri, J. Hernández-Orallo, and R. Modroiu, 'An experimental comparison of performance measures for classification', *Pattern Recognition Let.*, **30**(1), 27–38, (2009). 2

[7] John 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). 7

[8] Núria Macià and Ester Bernadó-Mansilla, 'Towards UCI+: A mindful repository design', *Information Sciences*, **261**, 237 – 262, (2014). 2

[9] Ricardo BC Prudêncio and Carlos 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). 1

[10] Ricardo BC Prudêncio, José Hernández-Orallo, and Adolfo 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). 1, 3

[11] Dimitris Rizopoulos, 'ltm: An r package for latent variable modeling and item response theory analyses', *Journal of statistical software*, **17**(5), 1–25, (2006). 4

[12] Michael R Smith, Tony Martinez, and Christophe Giraud-Carrier, 'An instance level analysis of data complexity', *Machine learning*, **95**(2), 225–256, (2014). 1, 2, 4

[13] David Thissen and Howard Wainer (Eds), *Test Scoring*, Lawrence Erlbaum Associates Publishers, 2001. 1

[14] Chris Thornton, *Truth from trash: How learning makes sense*, Mit Press, 2002. 7

[15] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo, 'OpenML: Networked science in machine learning', *SIGKDD Explor. Newsl.*, **15**(2), 49–60, (2014). 2