A three-step classification algorithm to assist criteria grid assessment

Damien Follet, Nicolas Delestre, Nicolas Malandain, Laurent Vercouter

A three-step classification algorithm to assist criteria grid assessment

Damien Follet, Nicolas Delestre, Nicolas Malandain, Laurent Vercouter
MIU, LITIS
INSA
76800 Rouen, France
first_name.last_name@insa-rouen.fr

Abstract

Nowadays, companies and the educational system are using more and more diagnostic assessments like criteria grids. But manually marking criteria grids is burdensome. We propose a modelling of criteria grid assessment and a domain-independent classification algorithm to suggest appropriate assessment to teachers. Empirical results on toy data indicate that our approach is reliable even with small amount of training data and fast enough to be used without perceivable delay.

1 Introduction

Nowadays, companies and the educational system are using more and more diagnostic assessments. Diagnosis allows companies to find the best suited person to hold a specific post. It also allows teachers to identify more precisely student gaps and to focus customized courses on these critical points [1].

Criteria grids support diagnostic assessment and give a richer insight of a student skill profile than a mere scoring mark. However marking becomes harder because teachers have to assess more than one single criteria, because a criteria grid is composed of a set of ordinal scales. With an ordinal scale, a person can express ordinal levels of agreement with a proposition. Ordinal values are ordered qualitative values [2]. Applied to pedagogical means, a teacher can match each skill or attribute with a qualitative level of mastering in relation to the student answers.

As marking becomes harder, suggesting appropriate assessments could relieve the teacher. In this paper no assumptions are made concerning the application field: Mathematics, History or Literature, or the nature of the items: Multiple Choice Question, short answers question, fill-in the gaps ... We only assume that answers are nominal, i.e. qualitative values that do not have a natural ordering [2]. Therefore the only meaningful distances between two nominal values are frequency distances.

We aim at building a supervised learning algorithm which would assign nominal answers to ordinal levels based on a small set of manually assessed productions. In order to suit the teacher requirements, this algorithm must be of low computational complexity and reliable even with few manually assessed examples. Moreover the teacher should be able to modify the items or the field of application on its own, i.e. without any intervention of a knowledge engineering expert. Therefore our goal is a domain-independent learning algorithm.

Existing classification algorithms are not suited for nominal answers. Some of them use the information contained in the distance between two answers, like k-Nearest Neighbours using an euclidean metric or a learned Mahalanobis Metric [3] (MMk-NN). But distances that are not related to frequency are not suitable for nominal answers. Algorithms like Weighted k-Nearest Neighbours (Wk-NN) use frequency distance as Modified Value Difference Metric [4]. But when an attribute

depends on nominal answers that are associated to more than one item, the distance may be distorted and lead to a prediction error. Conversely, a frequency distance between nominal answers associated to all items may take into account answers whose an attribute does not depend on, and lead to a prediction error through overfitting.

Some of them use notions closely related to invariant spaces under addition or multiplication by scalar, like Kernels: matrix algebra, Support Vector Machine: scalar product and Neural Networks: addition and multiplication of input data by scalar. But nominal answer spaces are not invariant under addition or multiplication by scalar [2].

Some of them use propositional logic, like Bayesian Networks. Each nominal answer and each nominal level is represented by one propositional variable and one boolean constraint which ensures that only one answer is affected to one item at the same time. Then the number of answer-level conditional dependencies is exponential compared to the number of answers and levels. Moreover the learning algorithm has to check all constraints for every considered conditional dependency in order to reject non-valid dependencies. Then its complexity is multiplied in vain and it will be slower than its polytomous equivalent.

Some of them use Shannon entropy to classify nominal answers into nominal levels, like decision tree algorithm Iterative Dichotomiser 3 (ID3). However these algorithms often overfit the training data and mispredict new productions. Pruning methods can reduce this tendency but do not suppress it completely. We will use this algorithm as a reference for performance tests.

Thus a specific non-overfitting learning algorithm which assign nominal answers to ordinal levels need to be developed, based on a suited modelling.

Existing nominal answer modellings are not suited either. Andrich [5], Bock [6], Rost [7], Samejima [8] and Muraki [9] can model nominal answers and ordinal levels using a parameter for each item, answer and level of attribute: they can not be used on different application domains without a knowledge engineering expert. Modellings based on Q-matrix [10][11][12] can model nominal answers thanks to a good-answer checking function and describes item-attribute dependencies as a boolean matrix. That way, Q-matrix can only express dichotomous attribute, but ordinal attributes are polytomous. Thus a specific modelling need to be developed too.

First we introduce notions and notations used in this article. Then we define assessment function and its associated abstract function to model human assessment. We propose a domain-independent algorithm which assign nominal answers to ordinal levels. Finally we present empirical evidences on toy data that suggest that our algorithm is fast and reliable even with small amount of training data.

2 Notions and notations

In this part, we introduce notions and notations that will be used thereafter.

i_l is the item indexed by l. IS (Item Set) is the set of all items. A student associates an answer to every item. $Dom(i_l)$ is the set of all possible answers to item i_l.

a_m is the attribute indexed by m. AS (Attribute Set) is the set of all attributes. The teacher associates a level to every attribute for every student. $Dom(a_m)$ is the set of all possible levels of the attribute a_m.

A production p is a vector of answers. $p|_{i_l}$ is the answer of p which is associated to the item i_l. $PS = \underset{i_l \in IS}{\otimes} Dom(i_l)$ (Production Set) is the set of all possible productions. Given a set of items E, a reduced production $p|_E$ is a vector of answers associated to items belonging to E.

A profile pf is a vector of levels. $pf|_{a_m}$ is the level of pf associated to the attribute a_m. $PFS = \underset{a_m \in AS}{\otimes} Dom(a_m)$ (ProFile Set) is the set of all possible profiles. Given a set of levels E, a reduced profile $pf|_E$ is a vector of levels associated to attributes belonging to E.

$\#E$ is the cardinality of the set E.

3 Assessment function and search spaces

In order to give assessment suggestions to teachers, we have to mimic human assessment.

We assume that human assessment is deterministic under certain conditions: assessing a small set of anonymized and digitalized productions with criteria grids. We make this approximation because human assessment bias like central tendency effet, socio-arithmetical hypothesis, tiredness, halo effet, sequence effet, inertia effet, contamination effet, vagueness effect and automatic strictness [13] have less impact in this way: the less impact these bias have, the more determinist human assessment is.

Therefore we will collect manually-assessed examples for the automatic learning algorithm according to these conditions.

Assessment function will be used to mimic human assessment. Such function associates every possible production to a profile and satisfies one reasonable property: an assessment function is partitionable.

Definition. An assessment function $f_{eval} : PS \rightarrow PFS$ is partitionable if and only if for every attribute a_m, $f_{eval}|_{a_m}$ is surjective.

$\Leftrightarrow \forall a_m \in AS, \forall n \in Dom(a_m), \exists p \in PS \; as \; f_{eval}(p)|_{a_m} = n$

Informally, for every attribute a_m an assessment function associates at least one production of PS to each level $n \in Dom(a_m)$.

Moreover an assessment function f_{eval} is a mathematical function, so for every attribute a_m it associates every production of PS to exactly one level $n \in Dom(a_m)$. We can conclude that for every attribute a_m, $f_{eval}|_{a_m}$ is a a_m-partition of PS such as each cluster of the partition stands for a specific level $n \in Dom(a_m)$.

Therefore the assessment function space is identical to the juxtaposition of a_m-partition spaces for every $a_m \in AS$. This juxtaposition is exponential: given the number of items, attributes, answers per item and levels per attribute we can calculate its size $\#ASFS$ (ASsessment Function Space). Given $D_i^{\#IS}$ the size of PS, $D_a^{\#AS}$ the size of PFS and $\left\{ {n \atop p} \right\}$ the Stirling number of second kind[1], the size of ASFS is:

$$\#ASFS = \#AS * \left\{ {D_i^{\#IS} \atop D_a} \right\} * D_a!$$

This expression is not intuitive since the Stirling number is hard to express: $\left\{ {n \atop p} \right\} = \frac{1}{p!} \sum_{k=0}^{p} (-1)^{p-k} \binom{p}{k} k^n$. However we can easily express its lower bound: $L(n,p) \leq \left\{ {n \atop p} \right\}$ given n and p.

$$L(n,p) = \tfrac{1}{2}(p^2 + p + 2) * p^{n-p-1} - 1 = O(p^{n-p+1})$$

Therefore we can express the lower bound of the size of ASFS:

$$\#ASFS \geq O(\#AS * D_a^{D_i^{\#IS} - D_a + 1} * D_a!)$$

Example.
Given $\#IS = 5$, $\#AS = 5$, $D_i = 5$ and $D_a = 4$, there are 3125 different productions and 625 different profiles. So there are more than $\#ASFS \geq 30 * 4^{3123} \simeq 5 * 10^{1881}$ different assessment functions.

The size of assessment function space is exponential: the bigger this space is, the more errors an overfitting algorithm will do. In order to work on a smaller search space, we will use a less informative function called the *abstract function*. For every attribute a_m, this function returns the set of items whose a_m depends on.

Definition. An abstract function $f_{abs} : AS \times (2^{IS} - \{\emptyset\})$ is associated to an assessment function f_{eval} if and only if for every attribute a_m, the function $f = \{(p|_{f_{abs}(a_m)}, pf|_{a_m}) \; as \; f_{eval}(p) = pf\}$

[1] the Stirling number of second kind represents the number of possible partitions of a set composed of n elements, such as every partition has exactly p unlabeled non-empty parts.

is a function, i.e. there is no $f_{abs}(a_m)$-reduced production associated to two different levels of attribute a_m by f.

The size of the ABFS (ABstract Function Space) is: $\#ABFS = \#AS * (2^{IS} - 1)$.

Example.
Given $\#IS = 5$, $\#AS = 5$, $D_i = 5$ and $D_a = 4$, there are only $5 * (2^5 - 1) = 155$ different abstract functions.

In the next section, we describe our classification algorithm based on abstract functions and assessment functions.

4 Classification algorithm

We developed an algorithm which classifies nominal answers into ordinal levels. It takes a set of manual examples TS (Training Set) and returns one assessment function f_{eval}, associated to the abstract function f_{abs}.

An example e is composed of a production $e.p$ and a profile $e.pf$.

4.1 Algorithm summary

Our algorithm is divided in three steps: computing the compatible abstract function set, selecting the best compatible abstract function and deducing the associated assessment function. The next sections will describe these steps more precisely. Simple numerical application will illustrate each one of these steps using the following example set TS.

Given i_1 and i_2 two items, a_1 an attribute, $Dom(i_1) = \{a; b; c; d; e\}$, $Dom(i_2) = \{a; b; c; d; e\}$, $Dom(a_1) = \{1; 2; 3; 4\}$ and $TS = \{(ac, 1); (bc, 1); (cd, 1); (dc, 2)\}$ such as $(ab, 1)$ means "production $(i_1 \leftarrow a; i_2 \leftarrow b)$ is associated to $(a_1 \leftarrow 1)$ by the assessment function". A graphical representation of TS is shown on figure 1.

i_1	i_2	a_1
a	*c*	1
b	*c*	1
c	*d*	1
d	*c*	2

Figure 1: Graphical representation of TS

4.1.1 Compatible abstract functions

First, for every attribute a_m we compute the set of all possible a_m-compatible abstract function with TS.

Definition. An abstract function $f_{abs}(a_m) \subseteq IS$ is a_m-incompatible with TS if and only if $\exists ex_1, ex_2 \in TS \; as \; ex_1.c|_{f_{abs}(a_m)} = ex_2.c|_{f_{abs}(a_m)} \wedge ex_1.p|_{a_m} \neq ex_2.p|_{a_m}$. Otherwise it is a_m-compatible with TS.

Note that this concept is closely related to assessment functions being mathematical functions: no production can be associated to two different levels of attribute.

Example.

1. $i_1 \leftarrow \{a; b; c\}$ for $a_1 = 1$, $i_1 \leftarrow \{d\}$ for $a_1 = 2$,
   $\{a; b; c\} \cap \{d\} = \emptyset \Rightarrow i_1$ is a_1-compatible with TS
2. $i_2 \leftarrow \{c; d\}$ for $a_1 = 1$, $i_2 \leftarrow \{c\}$ for $a_1 = 2$,
   $\{c; d\} \cap \{c\} = \{c\} \Rightarrow i_2$ is a_1-incompatible with TS
3. $\{i_1; i_2\} \leftarrow \{ac; bc; cd\}$ for $a_1 = 1$, $\{i_1; i_2\} \leftarrow \{dc\}$ for $a_1 = 2$,

$\{ac; bc; cd\} \cap \{dc\} = \emptyset \Rightarrow \{i_1; i_2\}$ is a_1-compatible with TS

4.1.2 Choosing the best compatible abstract function

Secondly, for every attribute a_m and every a_m-compatible abstract function $f_{abs}(a_m)$ we compute its complexity score. The a_m-compatible abstract function with the lowest score is the best one and is selected: "the simplest necessary hypothesis is the more likely".

Definition. The complexity score $score(f_{abs}(a_m))$ of an abstract function $f_{abs}(a_m) \subseteq IS$ for the attribute a_m is the product of the size score and the total option score.

Definition. The size score $s_{\text{size}}(f_{abs}(a_m))$ of an abstract function $f_{abs}(a_m) \subseteq IS$ for the attribute a_m is the number of items belonging to $f_{abs}(a_m)$.

Definition. The total option score $s_{\text{alternatives}}(f_{abs}(a_m))$ of an abstract function $f_{abs}(a_m) \subseteq IS$ for the attribute a_m is the sum of each n-option score given the level $n \in Dom(a_m)$.

Definition. The n-option score of $f_{abs}(a_m) \subseteq IS$ for the attribute a_m given $n \in Dom(a_m)$ is the number of different $f_{abs}(a_m)$-reduced answer vector associated to n: $s_{\text{alternatives}}(f_{abs}(a_m), n) = \#A$ such as $A = \{v_r \; as \; \exists ex \in TS, ex.c|_{f_{abs}(a_m)} = v_r \wedge ex.p|_{a_m} = n\}$.

Example.

1. $s_{\text{size}}(\{i_1\}) = 1$, $s_{\text{alternatives}}(\{i_1\}, a_1) = 3 + 1 = 4$
   because $a_1 \leftarrow \{a; b; c\}$ for $a_1 = 1$ and $\{d\}$ for $a_1 = 2$.
   $\Rightarrow score(\{i_1\}, a_1) = 1 * 4 = 4$.
2. $s_{\text{size}}(\{i_1; i_2\}) = 2$, $s_{\text{alternatives}}(\{i_1; i_2\}, a_1) = 3 + 1 = 4$
   because $i_2 \leftarrow \{ac; bc; cd\}$ for $a_1 = 1$ and $\{dc\}$ for $a_1 = 2$.
   $\Rightarrow score(\{i_1; i_2\, a_1) = 2 * 4 = 8$.

We computed all possible abstract function scores for a_1 and $\{i_1\}$ is the lowest scoring of all. Then $f_{abs}(a_1) = \{i_1\}$.

4.1.3 Deducing the associated assessment function

Finally, we compute the partial assessment function f_{eval} from TS and $f_{abs}(a_m)$ for every attribute a_m. Thereafter the teacher can correct or complete this partial function through interactions that add new examples to TS.

Example.
Given $f_{abs}(a_1) = \{i_1\}$ from the previous step and $TS = \{(ac, 1); (bc, 1); (cd, 1); (dc, 2)\}$,

1. $(ac, 1) \Rightarrow (\{i_1 \leftarrow a\}, a_1 \leftarrow 1)$
2. $(bc, 1) \Rightarrow (\{i_1 \leftarrow b\}, a_1 \leftarrow 1)$
3. $(cd, 1) \Rightarrow (\{i_1 \leftarrow c\}, a_1 \leftarrow 1)$
4. $(dc, 2) \Rightarrow (\{i_1 \leftarrow d\}, a_1 \leftarrow 2)$

The assessment function f_{eval} is composed of the 4 couples given above. Figure 2a gives a graphical representation of f_{eval} .

4.2 Using the assessment function

We can use this assessment function f_{eval} during the decision stage: we can infer the level of a_1 from answers associated to i_1, as shown in figure 2b.

5 Testing on toy data set

In order to estimate our algorithm performances, we compared it to three reference algorithms: ID3 [14], Wk-NN [4] and MMk-NN (k-NN using Mahalanobis Metric Learning [3]). As recommended in [4], we used $k = 1$ for Wk-NN. Further to preliminary tests on equivalent TS, the best performances of MMk-NN correspond with $m = 5$ and $k = 10$ so we used these values for this test.

i_1	i_2	a_1
a	*	1
b	*	1
c	*	1
d	*	2

(a) learned f_{eval}

i_1	i_2	a_1
a	b	1
b	d	1
c	e	1
d	a	2

(b) 4 new productions

Figure 2: Learned assessment function f_{eval} and 4 new productions assessed using it.

Table 1: Toy data tests results

	delay (μs)				error (%)	
algorithms	learning	σ_{learning}	decision	σ_{decision}	error	σ_{error}
our algorithm	6 252,2	1 303,7	55,2	38,1	20,8	7,3
ID3	343,1	97,0	43,0	14,9	49,1	9,1
MMk-NN	364,6	129,5	595,1	108,6	65,2	7,1
Wk-NN	20 942,5	1 318,1	6 673	502,8	63,5	7,5

We arbitrarily set the following constants: $\#IS = 8$ items, $D_i = 5$ different answers per item, $\#AS = 5$ attributes, $D_a = 4$ different level per attribute. The two sets P_{TS} (Training Set) and P_{DS} (Decision Set) are composed of 10 productions each such as $P_{TS} \cap P_{DS} = \emptyset$. We randomly generated 100 couples (P_{TS}, P_{DS}) ; each one was tested with 100 randomly generated assessment functions. These assessment functions are generated as follow: for every attribute a_m, we randomly choose a subset of IS to be the abstract function for a_m. Then for every attribute a_m we affect a random level $n \in Dom(a_m)$ to each set of answers associated to $f_{abs}(a_m)$: result is the random assessment function f_{eval}.

In order to test an assessment function f_{eval} with a couple (P_{TS}, P_{DS}), we used f_{eval} to label P_{TS} to constitute the training set TS. For each algorithm, TS was given during the training stage. Then during the decision stage, we asked the algorithm to predict labels of P_{DS}, then we compared these predictions to DS the real labels of P_{DS} using f_{eval} .

To summarise: $\#IS = 8, D_i = 5, \#AS = 5, D_a = 4$, $\#TS = 10, \#DS = 10$ on 10 000 simulations. Results are displayed in the table 1. They show that our algorithm is the more reliable: the mean percentage error for our algorithm is 21%, compared with 49% for ID3, 64% for Wk-NN and 65% for MMk-NN.

They also show that during the learning stage ID3 is the quickest, closely followed by MMk-NN ; our algorithm is 20 times slower than ID3 but still quite brief with less than 7 ms, and Wk-NN is 60 times slower than ID3. However during the decision stage, ID3 and our algorithm are the quickest with about 50 μs, MMk-NN is 12 times slower and Wk-NN is 130 times slower.

6 Conclusion

We have presented a new representation for criteria grid assessment: the assessment function associated to an abstract function, and a new domain-independent algorithm suited to classify nominal answers into ordinal levels. According to empirical results on toy data, our algorithm is reliable even with small amount of training data and fast enough to be used without perceivable delay.

We now have to extend our empirical comparison to real data and use optimization methods to speed up our algorithm. Moreover, it could be interesting to study the relationship between our complexity score and the well-known Kolmogorov complexity [15][16][17] and to describe the differences between our approach and Minimum Description Length theory [18] [19]. Finally, we could extend this algorithm to classify complex answers like nominal-labeled graphs.

References

[1] Chris Rust, Margaret Price, and Berry O'Donovan. Improving students' learning by developing their understanding of assessment criteria and processes. *Assessment and Evaluation in Higher Education*, 28(2):147–164, 2003.

[2] Stanley Smith Stevens. On the theory of scales of measurement, 1946.

[3] Shiming Xiang, Feiping Nie, and Changshui Zhang. Learning a mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 41(12):3600–3612, 2008.

[4] Scott Cost and Steven Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine learning*, 10(1):57–78, 1993.

[5] David Andrich. A rating formulation for ordered response categories. *Psychometrika*, 43(4):561–573, 1978.

[6] R Darrell Bock. Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika*, 37(1):29–51, 1972.

[7] Jürgen Rost. Measuring attitudes with a threshold model drawing on a traditional scaling concept. *Applied Psychological Measurement*, 12(4):397–409, 1988.

[8] Fumiko Samejima. Estimation of latent ability using a response pattern of graded scores. *Psychometrika monograph supplement*, 1969.

[9] Eiji Muraki. A generalized partial credit model: Application of an em algorithm. *Applied psychological measurement*, 16(2):159–176, 1992.

[10] Tiffany Barnes. Evaluation of the q-matrix method in understanding student logic proofs. In *FLAIRS Conference*, pages 491–496, 2006.

[11] Mark J Gierl, Jacqueline P Leighton, and Stephen M Hunka. An ncme instructional module on exploring the logic of tatsuoka's rule-space model for test development and analysis. *Educational Measurement: Issues and Practice*, 19(3):34–44, 2000.

[12] Menucha Birenbaum, Anthony E Kelly, and Kikumi K Tatsuoka. Diagnosing knowledge states in algebra using the rule-space model. *Journal for Research in Mathematics Education*, pages 442–459, 1993.

[13] Dieudonn Leclercq, J Nicaise, and Marc Demeuse. *Docimologie critique*. U.P. de Lige, 2004.

[14] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[15] Andrei N Kolmogorov. On tables of random numbers. *Sankhyā: The Indian Journal of Statistics, Series A*, 25(4):369–376, 1963.

[16] S Legg. Solomonoff induction (technical report cdmtcs-030), centre for discrete mathematics and theoretical computer science, university of auckland, 2011.

[17] Ray J Solomonoff. Algorithmic probability: Theory and applications. In *Information Theory and Statistical Learning*, pages 1–23. Springer, 2009.

[18] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465 – 471, 1978.

[19] Pieter Adriaans and Paul Vitanyi. The power and perils of mdl. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 2216–2220. IEEE, 2007.