Designing Intelligent Agents to Judge Intrinsic Quality of Human Decisions

Tamal T. Biswas
Department of CSE, University at Buffalo, Amherst, 14260, NY, U.S.A.

Keywords: Decision Making, Depth of Search, Item Response Theory, Chess, Data Mining, Judging of Learning Agents.

Abstract: Research on judging decisions made by fallible (human) agents is not as much advanced as research on finding optimal decisions. Human decisions are often influenced by various factors, such as risk, uncertainty, time pressure, and *depth* of cognitive capability, whereas decisions by an intelligent agent (IA) can be effectively optimal without these limitations. The concept of 'depth', a well-defined term in game theory (including chess), does not have a clear formulation in decision theory. To quantify 'depth' in decision theory, we can configure an IA of supreme competence to 'think' at depths beyond the capability of any human, and in the process collect evaluations of decisions at various depths. One research goal is to create an intrinsic measure of the depth of thinking required to answer certain test questions, toward a reliable means of assessing their difficulty apart from item-response statistics. We relate the depth of cognition by humans to depths of search, and use this information to infer the quality of decisions made, so as to judge the decision-maker from his decisions. We use large data from real chess tournaments and evaluations from chess programs (AI agents) of strength beyond all human players. We then seek to transfer the results to other decision-making fields in which effectively optimal judgments can be obtained from either hindsight, answer banks, powerful AI agents or from answers provided by judges of various competency.

1 INTRODUCTION

In most applications related to human decision making, the actors are aware of the true or expected value and cost of the actions. The available choices are deterministic and known to the actor, and the goal is to find some choice or allowed combination of choices that maximizes the expected utility value. The decisions taken can be either dependent or independent of actions taken by other entities that are part of the decision-making problem. In *bounded rationality*, however, such optimization is often not possible due to time constraints, the lack of accurate computation power by humans, the cognitive limitation of mind, and/or insufficient information possessed by the actor at the time of taking the decision. With these limited resources, the decision-maker in fact looks for a solution that seems satisfactory to him rather than optimal. Thus bounded rationality raises the issue of getting a measure of the quality of decisions made by the person.

Humans make decisions in diverse scenarios where knowledge of the best outcome is uncertain. This pertains to various fields, for example online test-taking, trading of stocks, and prediction of future events. Most of the time, the evaluation of decisions considers only a few parameters. For example, in test-taking one might consider only the final score; for a competition, the results of the game; for the stock market, profit and loss, as the only parameters used when evaluating the quality of the decision. We regard these as *extrinsic* factors.

Although bounded-rational behavior is not predicated on making optimal decisions, it is possible to re-evaluate the quality of the decision, and thus move from bounded toward strict rationality, by analyzing the decisions made with entities that have higher computing power and/or longer timespan. This approach gives a measure of the *intrinsic* quality of the decision taken. Ideally this removes all dependence on factors beyond the agent's control, such as performance by other agents (on tests or in games) or accidental circumstances (which may affect profit or loss).

Decisions taken by humans are often effectively governed by *satisficing*, a cognitive heuristic that looks for an acceptable sub-optimal solution among possible alternatives. Satisficing plays a key role in bounded rationality contexts. It has been documented

T. Biswas T..
Designing Intelligent Agents to Judge Intrinsic Quality of Human Decisions.
DOI: 10.5220/0005288406080613
In *Proceedings of the International Conference on Agents and Artificial Intelligence* (ICAART-2015), pages 608-613
ISBN: 978-989-758-074-1

in various fields including but not limited to economics, artificial intelligence and sociology (WikiBooks, 2012). We aim to measure the loss in quality and opportunity from satisficing and express the bounded-rational issues in terms of *depth* of thinking.

In multiple-choice question scenarios, there is no standard metric to evaluate answers. Any aptitude test allows multiple participants to answer the same problem, and based on their responses the difficulty of the problem is measured. The desired measure of difficulty is used when calculating the relative importance of the question on their overall scores. The first issue is how to distinguish the *intrinsic* difficulty of a question from simple poor performance by respondents? A second issue is how to judge whether a question is hard because it requires specialized knowledge, requires deep reasoning, or is "tricky"—with plausible wrong answers. Classical test theory approaches are less able to address these issues owing to design limitations such as having only a few choices with a unique correct answer.

Accordingly, we have identified three research goals:

1. Find an intrinsic way to judge the difficulty of decision problems, such as test questions,
2. Quantify a notion of *depth of thinking*, by which to identify satisficing and measure the degree of boundedness in rational behavior.
3. Use an application context (namely, chess) in which data is large and standards are well known so as to calibrate extrinsic measures of performance reflecting difficulty and depth. Then transfer the results to validate goals 1 and 2 in applications where conditions are less regular.

Putting together all these aspects, we can develop intelligent agents (IAs) that can segregate humans by their skill level via rankings based on their decisions and the difficulty of the problems faced, rather than being based only on total test scores and/or outcomes of games. IAs can be used for automated personnel assessment which by analyzing performances of an actor can pinpoint weaknesses and strength to help him improve.

In our setting, we have chosen chess games played by thousands of players spanning a wide range of ratings. The moves played in the games are analyzed with chess programs, called *engines*, which are known to play stronger than any human player. We can assume that given considerable time, an engine can provide an effectively optimal choice at any position along with the numeric value of the position, which exceeds the quality of evaluation perceived by even the best human players. Our intelligent agents use these engines as a knowledge-base to produce the final judgment.

This approach can be extended to other fields of bounded rationality, for example stock market trading and multiple choice questions, for several reasons, one being that the model itself does not depend on any game-specific properties. The only inputs are numerical values for each option, values that have authoritative hindsight and/or depth beyond a human actor's immediate perception. Another is the simplicity and generality of the mathematical components governing its operation, which are used in other areas.

Our position is that as automated tools for judging personnel results come into prominence, we will need a uniform structure for designing and calibrating them. Our model embraces both values and preference ranks, lends itself to multiple statistical fitting techniques that act as checks on each other, and gives consistent and intelligible results in the chess domain. This paper demonstrates the richness and efficacy of our modeling paradigm. It is thus both a rich testbed for measuring interoperability between intelligent agents and a fulcrum for transferring evaluation criteria established with big data to other applications.

2 BACKGROUND

Sequential sampling/accumulation based models are the most influential type of decision models to date. *Decision field theory* (DFT) applies sequential sampling for decision making under risk and uncertainty (Busemeyer and Townsend, 1993). One important feature of DFT is 'deliberation', i.e., the time taken to reach to a decision. DFT is a dynamic model of decision making that describes the evolution of the preferences across time. It can be used as a predictor not only of the decisions but also of the response times. Deliberation time (combined with the threshold) controls the decision process. The threshold is an important parameter which controls how strong the preference needs to be to get accepted.

Although *item response theory* (IRT) models do not involve any decision making models directly, they provide tools to measure the skill of a decision-maker. IRT models are used extensively in designing questionnaires which judge the ability or knowledge of the respondent. The *item characteristic curve* (ICC) is central to the representation of IRT. The ICC plots $p(\theta)$ as a function of θ, where θ and $p(\theta)$ represent the ability of the respondent and his probability of choosing any particular choice, respectively. Morris and Branum et al. have demonstrated the application of IRT models to verify the ability of the respondents with a particular test case (Morris et al., 2005).

On the chess side, a reference chess engine $E \equiv E(d, mv)$ was postulated in (DiFatta et al., 2009). The parameter d indicates the maximum depth the engine can compute, where mv represents the number of alternative variants the engine used. In their model, fallibility of human players is associated to a likelihood function L with engine E to generate a stochastic chess engine $E(c)$, where $E(c)$ can choose any move among mv alternatives with non zero probability defined by the likelihood function L.

In relation to test-taking and related item-response theories (Baker, 2001; Thorpe and Favia, 2012; Morris et al., 2005), our work is an extension of Rasch modeling (Rasch, 1960; Andrich, 1988) for *polytomous* items (Andrich, 1978; Masters, 1982; Ostini and Nering, 2006), and has similar mathematical ingredients (cf. (Wichmann and Hill, 2001; Maas and Wagenmakers, 2005)). Rasch models have two main kinds of parameters, *person* and *item* parameters. These are often abstracted into the single parameters of actor *location* (or "ability") and item *difficulty*. It is desirable and standard to map them onto the same scale in such a way that '*location* > *difficulty*' is equivalent to the actor having a greater than even chance of getting the right answer, or of scoring a prescribed norm in an item with partial credit. For instance, the familiar F-to-A grading scale may be employed to say that a question has exactly B-level difficulty if half of the B-level students get it right. The formulas in Rasch modeling enable predicting distributions of responses to items based on differences in these parameters.

3 CHESS ENGINES AND METRICS USED

3.1 Chess Engines and Their Evaluations

The Universal Chess Interface (UCI) protocol used by most major chess engines specifies two basic modes of search, called *single-pv* and *multi-pv*, and organizes searches in both modes to have well-defined stages of increasing depth.[1] Depth is in unit of *plies*, also called *half-moves*.[2] In single-pv mode, at any depth, only the best move is analyzed and reported fully. If a better move is found at a higher depth, the evaluation of the earlier selected move is not necessarily carried forward any further. Whereas, in multi-pv mode, we can select the number ℓ of moves to be analyzed fully. The engine reports the evaluation of each of the ℓ best moves at each depth. In our work, we run the engine in ℓ-pv mode with $\ell = 50$, which covers all legal moves in most positions and all reasonable moves in the remaining positions. Figure 1 shows output from the chess engine Stockfish 3 in multi-pv mode at depths up to 19.[3]

We aim to incorporate the idea of depth or process of deliberation by fitting the moves by the chess players to itemized skill levels based on *depth of search* and *sensitivity*. The ability of a player can be mapped to various depths of the engines. An amateur player's search depth for choosing any move may often not exceed two plies, whereas for a grandmaster it might be possible to analyze moves at ply-depths as high as 20. In this model we will attempt to generate a mapping between engine depths and player ratings, and use it to quantify depths of thinking for human players of all rating levels.

3.2 Concept of Depth of Thinking

In most decision theory literature, *deliberation time* is measured in units of seconds. In real-life decision making, when we try to judge the quality of a decision, it is very difficult to store the exact timing information for each decision. Moreover, the popular belief that quality of decisions is directly proportional to the deliberation time is not applicable in every scenario. Sometimes the correct decision looks reasonable at the beginning of deliberation, loses its 'charm' after a while, yet finally appears as the best choice to the decision-maker.

Chess tournaments place limits on the collective time for decisions, such as giving 120 minutes for a player to play 40 moves, but allow the player to budget this time freely. Meanwhile, chess offers an intrinsic concept of depth apart from how much time a player chooses to spend on a given position. In game theory, depth represents the number of plies a player thinks in advance. In chess, a turn consists of two plies, one for each player. We can visualize depth in chess as the depth of the game tree. In our model, the

[1] In all but a few engines the depths are successive integers. (The engine Junior used to produce evaluation at depths at interval of 3 i.e., 3-6-9-12....) Also 'pv' stands for "principal variation".

[2] A move by White followed by a move by Black equals two plies.

[3] The position is at White's 29^{th} move in the 5^{th} game of the 2008 world championship match between Kramnik-Anand with Forsyth Edwards Notation (FEN) code "8/1b1nkp1p/4pq2/1B6/PP1p1pQ1/2r2N2/5PPP/4R1K1 w - - 1 29". Values are from White's point of view in units of *centipawns*, figuratively hundredths of a pawn.

Table 1: Example of move evaluation by chess engines.

Moves	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Nd2	+230	+137	+002	+002	+144	+103	+123	+158	+110	+067	+064	+006	+002	+024	+013	-037	-018	000	000
Qg8	+205	+205	-023	-023	-059	-031	-058	-065	-066	-066	-053	-053	-103	-053	-053	-053	-053	-053	-053
Qh5	+101	+101	+034	+034	+034	-031	-058	-065	-066	-066	-053	-053	-103	-053	-053	-053	-053	-053	-053
Kf1	+108	+108	+108	+082	+029	+006	-087	-087	-090	-087	-048	-048	-087	-087	-077	-092	-092	-092	-092
Bxd7	+139	+139	-023	-023	-031	-039	-071	-071	-016	-020	-023	-023	-023	-017	-043	-042	-042	-083	-095
Rd1	+044	+044	+044	+016	-100	-094	-104	-124	-121	-121	-139	-143	-136	-150	-148	-122	-109	-122	-109
Nh4	+284	+161	+161	+161	+129	+116	+102	+046	+063	+028	+025	+028	-014	-078	-087	-097	-097	-127	-131
Kh1	+078	+078	+078	+051	-037	000	-019	-165	-165	-140	-140	-124	-157	-152	-185	-158	-158	-158	-172
Qg5	-107	-107	-091	-107	-113	-113	-130	-120	-202	-202	-197	-209	-200	-202	-200	-200	-189	-201	-174
Ng5	+402	+299	+299	+242	+163	+090	+008	+008	-033	-048	-041	-067	-067	-067	-115	-150	-150	-194	-177
Qh4	-107	-107	-107	-107	-113	-113	-130	-120	-202	-202	-186	-209	-203	-202	-200	-200	-189	-201	-191
Rf1	+003	+003	+003	-022	-138	-138	-138	-150	-168	-196	-183	-181	-220	-216	-205	-203	-211	-224	-205
h3	+084	+084	+084	+057	-237	-207	-230	-230	-257	-292	-279	-258	-249	-250	-253	-248	-249	-213	-236
Nxd4	-074	-074	-030	-054	-128	+243	+139	+139	+139	+091	+098	+098	+107	+093	+082	+061	-259	-250	-250
h4	+081	+081	+081	+055	-267	-267	-252	-243	-251	-255	-255	-247	-232	-246	-221	-244	-253	-253	-253
Ra1	+020	+020	+020	-007	-120	-120	-133	-145	-174	-196	-170	-211	-213	-172	-200	-217	-231	-231	-274
Rb1	+022	+022	+022	-005	-158	-158	-158	-145	-223	-196	-179	-172	-179	-209	-209	-217	-231	-231	-274
Qh3	+093	+093	+050	+050	-059	-019	-104	-104	-126	-208	-239	-210	-259	-217	-279	-310	-312	-312	-298
a5	+136	+136	+102	-191	-181	-181	-181	-288	-288	-288	-304	-327	-375	-376	-345	-428	-428	-430	-424
Be2	+097	+048	+062	+062	-051	-075	-205	-205	-278	-278	-282	-352	-379	-379	-375	-406	-447	-456	-451

evaluation of each chess position comes from the engine with values for each move at each depth individually. We use regression measure effect on thinking by utilizing move-match statistics for various depths.

3.3 Concept of Difficulty of a Problem

While the notion of difficulty of a problem is well known in the IRT literature, the concept seems to be little studied in decision making theories. We argue that having many possible options to choose from does not make the problem hard. Rather the difficulty lies in how close in evaluation the choices are to each other, in how "turbulent" they are from one depth to the next. The perceptions of difficulty differ among decision-makers of various abilities. The difficulty parameter β is the point on the ability scale where a decision-maker has a 0.50 probability of choosing the correct response.

3.4 Concept of Discrimination

The discriminating power α is an item (or problem) parameter. An item with higher discriminating power can differentiate decision-makers around ability level β better. For 2PL logistic IRT model, α contributes to the slope of the ICC at β. In our domain, higher discrimination may come from problems in which option values change markedly between depths, as exemplified in Table 1 by the rows for moves Nd2 and Nxd4. Less competent decision-makers may be attracted to answers that look good at low depths, but lose value upon greater reflection.

4 JUDGING THE DECISIONS AND THE DECISION MAKERS

Each chess position can be compared to a question asked to a student to answer. We treat the positions as independent and identically distributed (iid). The upper bound for the number of reachable chess positions is $10^{46.25}$ (Chinchalkar, 1996) and even in top level games, players often leave the "book" of previously played positions by move 15 or so. The lack of critical positions that have been faced by *many* players makes it hard to derive the item discrimination and difficulty parameters for chess positions in the traditional IRT manners.

For typical IRT models, the expectation of the correct response for a particular examinee (of a certain given ability) for a question is determined by the ratio of the number *m* of respondents with correct answers to the total number *n* of respondents. If we know the abilities of the respondents beforehand, we can create *k* subgroups of examinees, where each subgroup has the same ability. Assuming each subgroup consists of f_j respondents where $j \in (1..k)$, and r_j in each subgroup give the correct answer, the probability of answering correctly is deemed to be $p_j = r_j / f_j$. But in our chess domain we do not have '*n*'. So instead we use the utility values (evaluations) of the engines to generate the probability. There is a clear advantage in adopting this approach. Besides mitigating the problem of having enough respondents/players, we do not need any additional estimation to evaluate the ability parameter of the examines, rather the evaluation at various depths yields this. The various depth parameters without any additional tweaks work comparably to the ability parameter of the IRT models.

For achieving this goal, we need to address the fundamental question of how to calculate the estimate of the probability of playing the correct move, which is a similar paradigm for a decision-maker's probability of finding the optimal solution. This part plays the most critical role in the whole design and requires us to introduce techniques to convert utilities into probabilities.

We can assume that a player of Elo rating e on average plays or thinks up to/around depth d. For any particular depth d, for a position t, we have a number ℓ of available options $a_1, a_2, \ldots, a_\ell$ and a list of corresponding values $U_d = (u_1^d, u_2^d, \ldots, u_\ell^d)$. A player does not know the values, but by means of his power of discrimination can assign higher probability of playing a move i with higher u_i^d. For our basic dichotomous model, we are only concerned about the the probability of playing the best move where there are only two binary decisions possible, namely P_i and Q_i, which represent the probability of playing the correct and some incorrect move, respectively. It is possible to extend this to polytomous cases, along lines of the partial credit model of Muraki (1992).

4.1 Converting Utilities into Probabilities

For calculating the probability, we measure the deviation δ of all the legal moves from the best evaluation ($u_{*,d}$) at any particular depth d for any particular position. This generates the delta vector $\Delta_d = \delta_{1,d}, \delta_{2,d}, \ldots, \delta_{\ell,d}$. If the best move at depth d is m_j, where $j \in \{1, \ldots, \ell\}$, then $u_{*,d} = u_{j,d}$ and $\delta_{j,d} = 0$. We perform prior scaling based on the evaluation of the position before the played move. For generating probabilities from utility values, we used exponential transformations with fixed parameters, via $p_i = \frac{e^{-2\delta_i}}{\sum_{j=1}^{\ell} p_j}$. The choice of the constant 2 can be modified by fitting the sensitivity parameter s at a later stage, but nonetheless promises to be a good starting point.

4.2 Fitting ICC for Estimating Item Parameters

When IRT models are employed in test-taking applications, the parameters involved are called *item discrimination* $\alpha_i \in (0, +\infty)$ and *item difficulty* $\beta_i \in (-\infty, +\infty)$. They are related by:

$$P_i(\theta) = P(\alpha_i, \beta_i, \theta) = \frac{1}{1 + e^{-\alpha_i(\theta - \beta_i)}}. \qquad (1)$$

Once the probabilities of the moves are calculated, our next task is generate item parameters for the two-parameter logistic ICC model used in IRT literature. We simplify Equation (1) by setting $a = \alpha$, $b = \alpha\beta$. The resulting equation becomes:

$$P_i = P(\theta_i) = P(a, b, \theta_i) = \frac{1}{1 + e^{-a\theta_i + b}}. \qquad (2)$$

For estimating the item parameters from the probability we have already deduced, we use least squares estimation to minimize: $L(a,b) = \sum_{i=1}^{d} (p_i - P_i)^2$. Here the residual $p_i - P_i$ is the difference between the actual probability value of the dependent variable and the value predicted by the model. Estimation of a and b can be performed by Newton-Raphson based iterative procedure. The measure of a and b is used to judge the item parameters of the decision problem.

4.3 The ICC-Move Choice Correspondence

When an IRT model is deployed in the context of a theory of testing, the major goal is to procure a measure of the ability of each examinee. In item response theory, this is standardly the maximum likelihood estimate (MLE) of the examinee's unknown ability, based upon his responses to the items of the test, and the difficulty and discrimination parameters of these items. When we apply this idea for chess moves assessed by various chess engines, we follow the same procedure. We first calculate the MLE for the moves the player played. This is performed by evaluating the positions by various chess engines and then assigning the probability of playing the correct move at every depth. Finally we use maximum likelihood estimation to get the ability parameter of the player. We convert the ability parameter to the intrinsic rating by regressing on data set specific to players of known ratings.

For the completion of this estimation we make four assumptions. First, the value of the item parameters are known or derived from engine evaluation. Second, examinees are i.i.d sample or independent objects and it is possible to estimate the parameters for examinees independently. Third, the positions given to the players are independent objects too. Though the positions may come from the same game we assume those to be uncorrelated. Fourth, all the items used for MLE are modeled by the ICCs of the same family.

If a player $j \in \{1, \ldots, N\}$ faces n positions (either from a single game or any set of random positions) and the responses are dichotomously scored, we obtain $u_{i,j} \in \{0,1\}$ (1 for matching; 0 for not) where $i \in \{1, \ldots, n\}$ designates the items. This yields a vector of item responses of length n: $U_j = (u_{1j}, u_{2j}, \ldots, u_{nj})$.

From our third assumption, all the u_{ij} are i.i.d samples. Considering all the assumptions, the probability of the vector of item responses for a given player can be produced by the likelihood function

$$Prob(U_j|\theta_j) = \prod_{i=1}^{n} P_i^{u_{ij}}(\theta_j) Q_i^{1-u_{ij}}(\theta_j). \qquad (3)$$

This yields the log-likelihood function

$$\begin{aligned} L &= \log Prob(U_j|\theta_j) \\ &= \sum_{i=1}^{n} [u_{ij} \log P_{ij}(\theta_j) + (1 - u_i j) \log Q_{ij}(\theta_j)]. \end{aligned}$$

Since the item parameters for all the n items are known, only derivatives of the log-likelihood with respect to a given ability will need to be taken:

$$\begin{aligned} \frac{\partial L}{\partial \theta_j} = \sum_{i=1}^{n} u_{ij} \frac{1}{P_{ij}(\theta_j)} \frac{\partial P_{ij}(\theta_j)}{\partial \theta_j} \\ + \sum_{i=1}^{n} (1 - u_{ij}) \frac{1}{Q_{ij}(\theta_j)} \frac{\partial Q_{ij}(\theta_j)}{\partial \theta_j}. \qquad (4) \end{aligned}$$

When Newton-Raphson minimization is applied on L, an ability estimator θ_j for the player is obtained. The value of θ_j determines the ability of the decision maker.

5 CONCLUSION AND PROSPECTS

In this position paper, we have outlined several measurement procedures to quantify the quality of human decisions. Our proposed model generates the prediction of choices made by any decision-makers for any problems. It also ranks the decision-makers by the quality of the decisions made. The model is established via the evaluations generated by an AI agent of supreme strength, and uses this information as the knowledge-base for the IA to analyze the problem.

These procedures can also be employed to model an IA to mimic a decision-maker by tuning down to match the decision-maker's native characteristics. Numerous aspects like speed-accuracy trade-off, effect of procrastination and impact of time pressure also can be analyzed, and their effect on performances by the decision-makers can be tested. Other fields where this model can be applied include, but are not limited to, economics, psychology, test-taking, sports, stock market trading, and software benchmarking. Of these fields test-taking has the closest formal correspondence to our chess model.

We aim thereby to shed light on the following problems, for application domains such as test-taking for which we can establish a correspondence to our chess model: Do the intrinsic criteria for mastery transferred from the chess domain align with extrinsic criteria inferred from population and performance data in the application's own domain? How close is the agreement and what other scientific regularities, performance mileposts, and assessment criteria may be inferred from it? What does this say about distributions, outliers, and the effort needed for mastery?

REFERENCES

Andrich, D. (1978). A rating scale formulation for ordered response categories. *Psychometrika*, 43:561–573.

Andrich, D. (1988). *Rasch Models for Measurement*. Sage Publications, Beverly Hills, California.

Baker, F. B. (2001). *The Basics of Item Response Theory*. ERIC Clearinghouse on Assessment and Evaluation.

Busemeyer, J. R. and Townsend, J. T. (1993). Decision field theory: a dynamic-cognitive approach to decision making in an uncertain environment. *Psychological review*, 100(3):432.

Chinchalkar, S. (1996). An upper bound for the number of reachable positions. *ICCA JOURNAL*, 19(3):181–183.

DiFatta, G., Haworth, G., and Regan, K. (2009). Skill rating by Bayesian inference. In *Proceedings, 2009 IEEE Symposium on Computational Intelligence and Data Mining (CIDM'09), Nashville, TN, March 30–April 2, 2009*, pages 89–94.

Maas, H. v. d. and Wagenmakers, E.-J. (2005). A psychometric analysis of chess expertise. *American Journal of Psychology*, 118:29–60.

Masters, G. (1982). A Rasch model for partial credit scoring. *Psychometrika*, 47:149–174.

Morris, G. A., Branum-Martin, L., Harshman, N., Baker, S. D., Mazur, E., Dutta, S. N., Mzoughi, T., and McCauley, V. (2005). Testing the test: Item response curves and test quality. *Am. J. Phys.*, 74:449–453.

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

Ostini, R. and Nering, M. (2006). *Polytomous Item Response Theory Models*. Sage Publications, Thousand Oaks, California.

Rasch, G. (1960). *Probabilistic models for for some intelligence and attainment tests*. Danish Institute for Educational Research, Copenhagen.

Thorpe, G. L. and Favia, A. (2012). Data analysis using item response theory methodology: An introduction to selected programs and applications. *Psychology Faculty Scholarship*, page 20.

Wichmann, F. and Hill, N. J. (2001). The psychometric function: I. Fitting, sampling, and goodness of fit. *Perception and Psychophysics*, 63:1293–1313.

WikiBooks (2012). Bestiary of behavioral economics/satisficing — Wikibooks, the free textbook project. [Online; accessed 7-August-2014].