Adaptive Speed Tests*

Daniel Bengs and **Ulf Brefeld**†
German Institute for Educational Research
Frankfurt am Main, Germany
{bengs,brefeld}@dipf.de

Abstract

The assessment of a person's traits such as ability is a fundamental problem in human sciences. We focus on assessments of traits that can be measured by determining the shortest time limit allowing a testee to solve simple repetitive tasks, so-called speed tests. Existing approaches for adjusting the time limit are either intrinsically non-adaptive or lack theoretical foundation. By contrast, we propose a mathematically sound framework in which latent competency skills are represented by belief distributions on compact intervals. The algorithm iteratively computes a new difficulty setting, such that the amount of belief that can be updated after feedback has been received is maximized. We provide theoretical analyses and show empirically that our method performs equally well or better than state of the art baselines in a near-realistic scenario.

1 Introduction

The assessment of a person's traits such as ability is a fundamental problem in the human sciences. Perhaps the most prominent example is the Programme for International Student Assessment (PISA) launched by the Organisation for Economic Cooperation and Development (OECD) in 1997. Traditionally, assessments have been conducted with printed forms that had to be filled in by the testees (paper and pencil tests). Nowadays, computers and handhelds become more and more popular as platforms for conducting studies in social sciences; electronic devices not only facilitate data acquisition and processing, but also allow for real-time adaptivity and personalization.

Psychological testing differentiates between two types of tests, namely *power* and *speeded* tests [Furr and Bacharach, 2007]. The former uses items with a wide range of difficulty levels, so that testees will almost surely be unable to solve all items, even when they are given unlimited time. On the contrary, speeded tests deploy homogeneous items that are easy to solve. The difficulty in speeded tests is realized by narrow time intervals in which the response has to be given. In adaptive speed tests, the latent competency parameter $\hat{\theta}$ encodes for instance reaction time, concentration, or awareness of the testee. An example of such a test is the Frankfurt Adaptive Concentration Test II (FACT-II) [Goldhammer and Moosbrugger, 2007] where a simple multiplicative update of the estimate $\hat{\theta}$ is applied for the adaptation process.

In this paper, we present a novel framework for learning competency parameters in speeded tests. The formal problem setting resembles a game played in rounds. In each round, the goal is to gain as much information as possible on the difficulty setting θ corresponding to the testee's competency. The uncertainty of an estimate $\hat{\theta}$ is represented by a belief distribution over a compact interval. At round t, a new estimate $\hat{\theta}_t$ is drawn, such that $\hat{\theta}_t$ divides the belief mass in two equally sized halves. The testee solves the item which realizes a difficulty level of $\hat{\theta}_t$. The agent observes the response ρ_t. We differentiate three cases: (i) if $\hat{\theta}_t < \theta_t$, the difficulty induced by $\hat{\theta}_t$ was *too easy* for the testee and $\rho_t = 1$, (ii) in case $\hat{\theta}_t > \theta_t$, the setting as *too difficult* and $\rho_t = -1$, and (iii) $\theta_t = \hat{\theta}_t$ which corresponds to a *just right* setting and response $\rho_t = 0$. A similar scenario for discrete variables has been studied by Missura and Gärtner [2011] in the context of computer games.

Before we continue with the presentation of our method, note that the problem setting does not match traditional approaches, including standard supervised (e.g., binary classification) and unsupervised (e.g., density estimation) settings, as the feedback needs to be viewed a directional and not a point-wise one and we cannot make assumption on the testee or stationarity of the observations due to learning effects and tiredness. Thus, the directional feedback is used to update exactly half of the belief mass for maximal information gain. The rationale behind this update strategy is the following: once we observe that $\hat{\theta}$ is *too difficult*, it is highly probable that all difficulty levels $\tilde{\theta} > \hat{\theta}$ are also *too difficult*. A similar argument holds vice versa for *too easy*. The directional feedback is therefore used as a nominal reward that triggers the update process. We present results on the step size of the proposed algorithm and show that it performs equally well or better than state of the art baselines in a near-realistic scenario modelling testee behaviour.

The remainder is organized as follows. Section 2 reviews related work. We present our main contributions, the learning agent and a theoretical analysis in Sections 3 and 4, respectively. Section 5 reports on simulation studies and Section 6 concludes.

2 Related Work

Motivated by applications in computer games as well as teaching systems, Missura and Gärtner [2011] considered the problem of dynamic difficulty adjustment. They formalized the problem setting as a game between a master and a player played in rounds $t = 1, 2, \cdots$, where the mas-

*This paper is a short version of [Bengs and Brefeld, 2013].

†UB is also affiliated with Technische Universität Darmstadt.

ter predicts the difficulty setting for the next round. After the player has finished his turn, the master receives feedback and updates the belief on the difficulty settings and predicts the setting for the next round. The authors introduce the Partial Ordered Set Master (POSM) algorithm that represents the set of admissible difficulty settings as a finite discrete set $\mathcal{K}$ endowed with a partial ordering $\prec$. We will show later that the POSM algorithm for the case of a totally ordered set of difficulty settings is contained as a special case within our framework.

Csáji and Weyer [2011] investigate the problem of estimation in the presence of noise using a binary sensor with adjustable threshold. Their approach estimates a constant $\theta^* \in \mathbb{R}$ that is disturbed by additive, i.i.d. noise. The threshold θ_t is assumed to be adjustable based on all previous observations and threshold values. Under mild assumptions on the distribution of the noise, they derive a strongly consistent estimator for θ^* based on stochastic approximation. In contrast to them , we do not make any assumptions on the distribution of the value to be estimated or on its stationarity.

In the field of psychometrics, only a few adaptive speed tests have been designed. For the assessment of concentration ability, Goldhammer and Moosbrugger [2007] suggested the Frankfurt Adaptive Concentration Test II (FACT-II). As FACT-II conceptualizes concentration as the ability to respond to stimuli in the presence of distractors, testees are shown a set of items comprising of target and non-target items. They are instructed to hit one button, if a target item is present, and another button, if no target item is among the items shown. After each round t, exposure time is adjusted until a liminal exposure time is reached that just allows the testee to solve the task. Starting with a fixed initial exposure time θ_1, updating is performed multiplicatively depending on whether a response is given in time or not.

3 A Learning Agent for Parameter Estimation in Speeded Tests

We cast the problem of learning competency parameters in speeded tests as a game between an agent $\mathcal{A}$ and a testee $\mathcal{T}$ played in rounds $t = 1, 2, \ldots$ on a continuous interval of difficulty settings $\Theta = [a, b]$. Θ is governed by a total order relation $>$ induced by the real numbers corresponding to the *more-difficult-than relation*. We assume that at each round, there is a *just right* setting $\theta_t \in \Theta$ for the testee $\mathcal{T}$. At round t, (i) the agent chooses a setting $\hat{\theta}_t \in \Theta$ based on the current belief, (ii) the testee responds, and (iii) the agent observes directional feedback of the form $\rho_t \in \{-1, 0, +1\}$ subject to the following rule:

$$\rho_t = \begin{cases} +1 & \text{if } \hat{\theta}_t < \theta_t \text{, too easy} \\ 0 & \text{if } \hat{\theta}_t = \theta_t \text{, just right} \\ -1 & \text{if } \hat{\theta}_t > \theta_t \text{, too difficult} \end{cases}$$

Note that the *just right* setting remains hidden to the agent at all times.

In the course of the game, the agent is choosing actions $\hat{\theta}_t$ from the space of possible actions Θ that lead to a reward signal ρ_t depending on the state of the environment θ_t. The goal of the agent is to reach the rewarding state of having selected the *just right* setting by avoiding the punishing signals associated with *too difficult* or *too easy* settings.

The general idea of our approach is the following: We use a function $w_t : [a, b] \to (0, \infty)$ to model the agent's belief at time t about the optimal action based on the experience gathered at time-steps $1, \ldots, t-1$. Suppose that the agent selects a setting $\hat{\theta}_t$ and receives feedback $\rho_t = +1$ (*too easy*). Because of the transitivity of the ordering of difficulty settings, the agent not only learns about $\hat{\theta}_t$ as an isolated point, but also learns that all settings $\tilde{\theta}$ which are easier than $\hat{\theta}_t$, i.e., $\tilde{\theta} < \hat{\theta}_t$, would also have been *too easy* and the agent updates the belief on the whole interval $[a, \hat{\theta}_t]$. The mass of belief that can be updated is then given by

$$A_t(\hat{\theta}_t) := \int_a^{\hat{\theta}_t} w_t(x)dx.$$

Similarly, if $\rho_t = -1$, the belief in the interval $[\hat{\theta}_t, b]$ can be updated according to

$$B_t(\hat{\theta}_t) := \int_{\hat{\theta}_t}^b w_t(x)dx.$$

If $\rho_t = 0$, there is no reason to update belief, because current knowledge has led to a correct prediction. We devise the following strategy for predicting $\hat{\theta}_t$ and updating belief: The difficulty setting $\hat{\theta}_t$ for the upcoming round is selected in order to allow to update as much belief as possible after feedback has been obtained. That is, we select $\hat{\theta}_t$ so that

$$\hat{\theta}_t = \operatorname{argmax}_{\tilde{\theta} \in [a,b]} \min \left\{ A_t(\tilde{\theta}), B_t(\tilde{\theta}) \right\}. \qquad (1)$$

It can easily be seen that this amounts to selecting $\hat{\theta}_t$ such that

$$A_t(\hat{\theta}_t) = \frac{1}{2} \int_a^b w_t(x)dx.$$

Equivalently, $\hat{\theta}_t$ can be characterized by $A_t(\hat{\theta}_t) = B_t(\hat{\theta}_t)$. Because w_t is non-negative by assumption, the mapping $\hat{\theta}_t \mapsto A_t(\hat{\theta}_t)$ strictly increasing and thus bijective, so $\hat{\theta}_t$ is uniquely determined if only $\int_a^b w_t(x)dx \neq 0$. In order to derive an algorithm from this framework, we need to specify the space of belief functions $\mathcal{W}$ and the belief updating rule

$$\mathcal{W} \times \{-1, 0, 1\} \to \mathcal{W}, \quad (w_t, \rho_t) \mapsto w_{t+1}.$$

The next section introduces strategies to learn the agent.

3.1 Interval Subdivision Agent

While there is no restriction on the space of belief functions arising from the general framework, we choose to use the space of non-negative step functions on $[a, b]$ for $\mathcal{W}$ and an exponential updating rule based on interval subdivision. That is, we divide the interval containing the actual prediction $\hat{\theta}_t$ at $\hat{\theta}_t$ and update the belief values to the left or right of $\hat{\theta}_t$ depending on the feedback ρ_t by multiplying with a parameter $\beta \in (0, 1)$. Formally, denoting by χ_M the characteristic or indicator function of a set $M \subset \mathbb{R}$, we write w_t as a sum

$$w_t = \sum_{i=1}^{N_t} y_i^{(t)} \chi_{I_i^{(t)}}$$

for some $N_t \in \mathbb{N}$, where $y_i^{(t)} \geq 0$ is the value w_t takes on the i^{th} interval given by

$$I_i^{(t)} = [x_{i-1}^{(t)}, x_i^{(t)})$$

for $i = 1 \cdots, N_t - 1$ and $I^{(t)}_{N_t} = [x_{N_t-1}, x_{N_t}]$. The interval endpoints are defined by a partition

$$a = x^{(t)}_0 < x^{(t)}_1 < x^{(t)}_2 < \cdots < x^{(t)}_{N_t} = b$$

of $[a, b]$. Denoting the index of the interval containing $\hat{\theta}_t$ by i^*_t, we update

$$w_{t+1} = \sum_{i=1}^{i^*_t-1} \beta y_i \chi_{I^{(t)}_i} + \beta y_{i^*_t} \chi_{[x_{i^*_t-1}, \hat{\theta}_t)} + y_{i^*_t} \chi_{[\hat{\theta}_t, x_{i^*_t})} + \sum_{i=i^*_t+1}^{N_t} y_i \chi_{I^{(t)}_i},$$

in case $\rho_t = 1$ and analogously for $\rho = -1$,

$$w_{t+1} = \sum_{i=1}^{i^*_t-1} y_i \chi_{I_i} + y_{i^*_t} \chi_{[x_{i^*_t-1}, \hat{\theta}_t)} + \beta y_{i^*_t} \chi_{[\hat{\theta}_t, x_{i^*_t})} + \sum_{i=i^*_t+1}^{N_t} \beta y_i \chi_{I_i}.$$

Finally, if $\rho_t = 0$ no update is necessary and $w_{t+1} = w_t$. The belief function can be stored and updated efficiently by storing the endpoints $x^{(t)}_1, \cdots, x^{(t)}_{N_t-1}$ and function values $y^{(t)}_1, \cdots, y^{(t)}_N$. Also, our particular choice of $\mathcal{W}$ makes the computation of $\hat{\theta}$ simple and inexpensive: As w is a step function, its integral over θ is given by

$$\int_a^b w_t(x)dx = \sum_{i=1}^{N_t-1} y_i \left(x_{i+1} - x_i\right)$$

The initial belief function w_1 can be tailored to incorporate prior knowledge about where to expect θ_1. In the absence of prior knowledge on the distribution of θ, $w_1 \equiv 1$ serves as a possible initialization.

3.2 Limited-memory Interval Subdivision Agent

The memory usage of the internal subdivision agent (ISA) at time t is in $\mathcal{O}(t)$. Indeed, if w_0 is represented by N interval-value pairs, each step adds at most one node in the belief function. A limit on the amount of memory consumed by ISA can be imposed by limiting interval subdivision. Thus, the limited-memory ISA (LISA) only subdivides intervals when subdivision results in intervals of width greater than a given parameter $\epsilon > 0$.

4 Theoretical Analysis

In this section we present a theoretical analysis of the ISA algorithm. We are interested in characterizing convergence properties of ISA under different assumptions. The simplest assumption that can be made about the *just right* setting is that it remains constant at all times. That is, $\theta_t \equiv c$ for $c \in [a, b]$ and all $t \in \mathbb{N}$. We now present a bound on the step size between successive predictions by ISA. The bound follows directly from Lemma 1.[1]

Lemma 1. *Let $f : [a, b] \to (0, \infty)$ be bounded and integrable on $[a, b]$. Let $\beta \in (0, 1)$. Let $\theta_1, \theta_2 \in [a, b]$ be numbers such that $\int_a^{\theta_1} f(x)dx = \frac{1}{2}\int_a^b f(x)dx$ and $\int_a^{\theta_2} \hat{f}(x)dx = \frac{1}{2}\int_a^b \hat{f}(x)dx$, where*

$$\hat{f}(x) = \begin{cases} \beta f(x) & \text{if } a \le x \le \theta_1 \\ f(x) & \text{if } \theta_1 < x \le b \end{cases}.$$

Then $\theta_1 < \theta_2$ and

$$\frac{1-\beta}{4M} \int_a^b f(x)dx \le \theta_2 - \theta_1 \le \frac{1-\beta}{4m} \int_a^b f(x)dx. \quad (2)$$

where $M := \max_{x \in [a,b]} f(x)$ and $m := \min_{x \in [a,b]} f(x)$.

Lemma 1 says that if the difficulty level $\hat{\theta}_t$ estimated by ISA is *too easy* ($\rho_t = 1$), the new estimate will be greater than its predecessor, that is $\hat{\theta}_{t+1} > \hat{\theta}_t$ holds. Analogously the case $\rho_t = -1$ implies $\hat{\theta}_{t+1} < \hat{\theta}_t$. We use the inequality to derive a bound on the step size of ISA in the following Theorem 1.

Theorem 1. *Let $\left(\hat{\theta}_t\right)_{t=1}^N$ be a sequence of estimations generated by ISA with parameter β. Then for $t = 1, \ldots, N-1$ it holds that*

$$\frac{1-\beta}{4M_t} \int_a^b w_t(x)dx \le \left|\hat{\theta}_{t+1} - \hat{\theta}_t\right| \le \frac{1-\beta}{4m_t} \int_a^b w_t(x)dx,$$

where

$$M_t := \max_{x \in [a,b]} w_t(x)$$

and

$$m_t := \min_{x \in [a,b]} w_t(x).$$

Theorem 1 bounds the minimal and maximal difference between successive estimates by ISA. Note that the bounds are invariant under rescaling of the belief function, but depend on the parameter β that controls learning rate: If β is small, new experience is given more weight and the lower bound on step size is greater than its analogue for $\beta \approx 1$ which gives less weight to new information.

We now investigate the relation between LISA and POSM for a completely ordered set which we denote by $\Theta' = \{1, \cdots, N\}$ for some $N \in \mathbb{N}$, endowed with the natural ordering. The following proposition holds:

Proposition 1. *Let $N \in \mathbb{N}$, $\Theta' = 1, \ldots, N$ endowed with the natural ordering be the set of difficulty levels for POSM and let $[a, b] = [0, N]$. Let $\beta \in (0, 1)$, $\epsilon < 1$. Define the initial belief function w_0 for $LISA$ by $x_i = i$ for $i = 0, \cdots, N$ and $y_i = 1$ for $j = 1, \cdots, N$. Denote by $\operatorname{ind}(x)$ the function mapping $x \in [a, b]$ to Θ' such that $x \in [x_{\operatorname{ind}(x)-1}, x_{\operatorname{ind}(x)})$. Then, given a sequence of feedback $(\rho_t)_{t\in\mathbb{N}}$, the estimates $(\tilde{k}_t)$ produced by POSM coincide with $(\operatorname{ind}(\hat{\theta}_t))_{t=1}^N$.*

The result stated in Proposition 1 explains to some extent why ISA and LISA expose a behaviour qualitatively similar to that of POSM in the setting of our experiments. As we show in the next section, the LISA and ISA algorithms are able to exploit the continuous setting, outperforming POSM by a significant margin.

5 Empirical Results

For our experiments, we simulate near-realistic scenarios to create settings that reflect behaviour observed in adaptive psychological speed tests or computer games. We compare the empirical performance of ISA and LISA to state-of-the-art baselines POSM and the algorithm used by FACT-II .

Throughout all our experiments, we use $\Theta = [0, 1]$. Note that this does not limit generality, as every compact interval can be rescaled and shifted to match Θ. To allow for a fair comparison, the set of difficulty settings for POSM consists of N equidistantly sampled points in Θ, where N is

[1] Detailed proofs are presented in [Bengs and Brefeld, 2013].

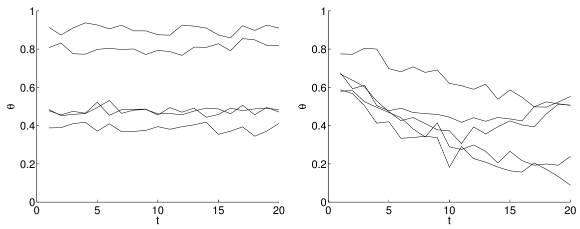

Figure 1: Randomly parametrized functions modelling θ in absence (left) and presence of drift (right). In both scenarios white noise is added.

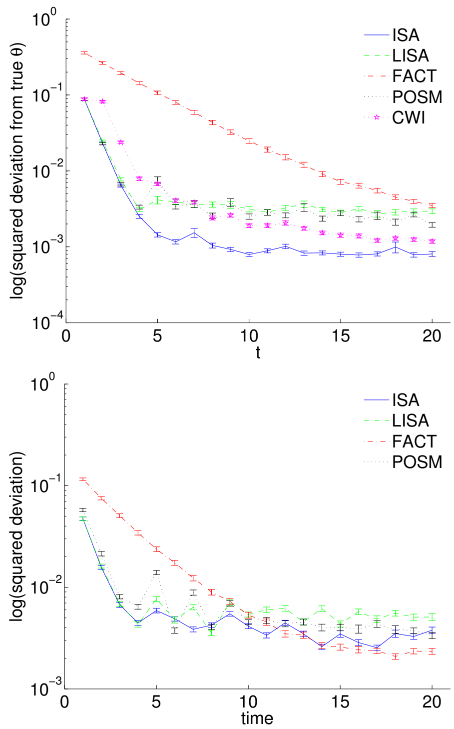

Figure 2: Squared deviations from true θ for the constant (top) and the drift (bottom) setting.

the number of time steps used. This choice guarantees that the number of subdivisions made by ISA and LISA is less than or equal to the number of settings available to POSM. Thus, all approaches have access to the same amount of resources. We use optimal parameters for ISA, LISA and POSM chosen by model selection.

We consider two distinct settings: In the first setting, the true parameter θ remains constant and is sampled from a uniform distribution. For the constant setting, we also include Csáji-Weyer-Iteration (CWI) [Csáji and Weyer, 2011] as an additional baseline. In the second setting, we simulate learning and tiredness effects. The true parameter θ thus underlies drifts and the resulting distribution is not stationary. Additionally, observations are disturbed by additive noise originating from a Gaussian distribution. Figure 1 shows sample observations for the two settings. In both settings, we conduct 500 repetitions with randomly drawn sequences θ_t and report on averaged deviations and standard errors.

Figure 2 (top) shows the results for the constant setting. All algorithms need some time to adapt to the noisy θ_t. The three learning algorithms and CWI, however, approach the true θ significantly faster than FACT. CWI and ISA approximate the true θ more closely with ISA realizing quicker convergence and smaller error. The squared error is smallest for ISA, followed by the almost equally performing LISA and POSM. FACT is outperformed by all four competitors by a large margin (see also Table 1).

Table 1: Sum of squared deviations from true θ, average over 500 runs.

	ISA	LISA	POSM	FACT	CWI
const.	3.3842	4.3905	4.2441	34.8336	5.9575
drift	3.4027	4.0825	4.4171	9.4808	–

Figure 2 (bottom) summarizes the results for the drift setting. ISA performs best, followed by LISA and POSM. Again FACT is outperformed significantly by the others. The squared errors are similar or smaller for all algorithms than they are in absence of drift (see Table 1), showing that all algorithms can deal with drift well. The performance of FACT even proves significantly better than in the setting without drift. This effect can be explained by the fact that the model of drift employed here favors evolutions of θ starting in the upper range of Θ. Note that FACT always initializes θ_0 with the highest possible value which highly affects its performance in the first iterations. The other algorithms thus benefit in the beginning from intitalizing θ with the mean of the search space. However, different choices are possible.

6 Conclusion

We have introduced a mathematically sound learning framework for parameter adaptation in speeded tests. Our approach does not make any assumptions on the distribution of the true parameter and is therefore deployable in settings characterized by parameter drift and additive noise. Empirically, we have shown that the algorithm performs equally or better than state of the art baselines in different scenarios modelling testee behaviour under different assumptions.

References

[1] D. Bengs and U. Brefeld. A Learning Agent for Parameter Adaptation in Speeded Tests. In *Proceedings of the ECML/PKDD Workshop on Reinforcement Learning from Generalized Feedback: Beyond Numeric Reward*, 2013.

[2] B. C. Csáji and Weyer. System identification with binary observations by stochastic approximation and active learning. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*:3634–3639 , 2011.

[3] R. M. Furr and V. R. Bacharach. *Psychometrics: An Introduction*, SAGE Publications, Thousand Oaks, California, 2007.

[4] O. Missura and T. Gärtner. Predicting Dynamic Difficulty. In *Advances in Neural Information Processing Systems 24*: 2007–2015, 2011.

[5] H. Moosbrugger and F. Goldhammer. *FAKT-II Frankfurter Adaptiver Konzentrationsleistungs-Test II*, Huber, Bern, 2007.